METHODS IN MOLECULAR BIOLOGY™

Series Editor
John M. Walker
School of Life Sciences
University of Hertfordshire
Hatfield, Hertfordshire, AL10 9AB, UK

For further volumes:
http://www.springer.com/series/7651

Systems Metabolic Engineering

Methods and Protocols

Edited by

Hal S. Alper

Department of Chemical Engineering, Cockrell School of Engineering,
The University of Texas at Austin, Austin, Texas, USA

Humana Press

Editor
Hal S. Alper
Department of Chemical Engineering, Cockrell School of Engineering
The University of Texas at Austin
Austin, Texas, USA

Additional material to this book can be downloaded
from http://extras.springer.com

ISBN 978-1-4939-6292-1 ISBN 978-1-62703-299-5 (eBook)
DOI 10.1007/978-1-62703-299-5
Springer New York Heidelberg Dordrecht London

Printed on acid-free paper

Humana Press is a brand of Springer
Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Metabolic engineering has always been focused on using a systems-level view to analyze cellular metabolism and predict optimal rewiring of metabolic networks. Since its inception, significant advances have been made in our capacity to obtain high-resolution details about cellular state. In response to this newfound capacity, Systems Metabolic Engineering emerges as a paradigm that connects the area of systems biology with metabolic engineering goals. Specifically, this field incorporates large-scale data collection/high-throughput biology and in silico modeling efforts along with new capacities for genome-wide engineering to accomplish the goal of improving a cellular phenotype or pathway flux. These technologies and efforts continue to expand the global systems-level view of metabolic engineering by shifting focus away from individual pathways and toward the collective, interconnected nature of metabolism and regulation. The advances of systems biology enable high-throughput collection of genomic, transcriptomic, proteomic, metabolomic, and fluxomic data. However, this immense snapshot of cells brings about a large challenge for data collection, integration, interpretation, synthesis, and ultimately perturbation to the cell. Moreover, the rate of data generation is also being matched by our rate of multiplexed engineering of pathways and genomes.

The ultimate goal of a Systems Metabolic Engineering approach is to systematically and robustly define the specific perturbations necessary to alter a cellular phenotype. The tangible outcome of such an approach would be a complete cell model capable of (1) simulating cell and metabolic function and (2) predicting phenotypic response to changes in media, gene knockouts/overexpressions, or incorporation of heterologous pathways. While the field is not yet at this point, it is clearly on a trajectory toward such capacity. The field of Systems Metabolic Engineering has already proven to be a successful paradigm for improving pathway performance for small molecules in both the laboratory and industrial setting. As techniques continue to improve, the design cycle for engineering a cell will be greatly reduced.

As stated above, great strides have been made in advancing the key aspects of a Systems Metabolic Engineering approach. Thus, the aim of this book is to describe the methodologies and approaches in the area of Systems Metabolic Engineering and provide a step-by-step guide for their implementation. In particular, four major tenants of this approach will be discussed: (1) modeling and simulation, (2) multiplexed genome engineering, (3) 'omics technologies, and (4) large data-set incorporation and synthesis. Each of these four capacities plays an important role in the design cycle for strain improvement. Tools and protocols within each of these tenets will be described to enable facile implementation of a Systems Metabolic Engineering approach using model host organisms. This book is designed especially for metabolic engineers, molecular biologists, and microbiologists who are proficient in the genetic manipulation of organisms. The coverage of this material is quite broad to allow for accessibility by novices and experts alike. It is hopeful that this book will serve as a guide to implementing the most recent approaches in Systems Metabolic Engineering.

Austin, Texas, USA *Hal S. Alper*

Contents

Part III Systems-Level 'Omics Tools

Part IV Integrating Large Datasets for Modeling and Engineering Applications

Contributors

Michiel Akeroyd • *DSM Biotechnology Center, Delft, The Netherlands*

Maciek R. Antoniewicz • *Metabolic Engineering and Systems Biology Laboratory, Department of Chemical and Biomolecular Engineering, University of Delaware, Newark, DE, USA*

Kazuharu Arakawa • *Institute for Advanced Biosciences, Keio University, Fujisawa, Kanagawa, Japan*

Aaron A. Best • *Department of Biology, Hope College, Holland, MI, USA*

Edik M. Blais • *Department of Biomedical Engineering, University of Virginia, Charlottesville, VA, USA*

Viktor M. Boer • *DSM Biotechnology Center, Delft, The Netherlands*

Erin Boggess • *Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA*

Linda Broadbelt • *Department of Chemical and Biological Engineering, Northwestern University, Evanston, IL, USA*

Pablo Carbonell • *Institute of Systems & Synthetic Biology (ISSB), Evry, France*

Sriram Chandrasekaran • *Institute for Systems Biology, Seattle, WA, USA; Center for Biophysics and Computational Biology, University of Illinois at Urbana-Champaign, Urbana, IL, USA*

Arvind K. Chavali • *Department of Biomedical Engineering, University of Virginia, Charlottesville, VA, USA*

Tao Chen • *Key Laboratory of Systems Bioengineering, Ministry of Education, Tianjin University, Tianjin, China; Department of Biological Engineering, School of Chemical Engineering and Technology, Tianjin University, Tianjin, China*

William R. Cluett • *Department of Chemical Engineering and Applied Chemistry, University of Toronto, Toronto, ON, Canada*

Lydia M. Contreras • *Department of Chemical Engineering, Cockrell School of Engineering, The University of Texas at Austin, Austin, TX, USA*

Matthew DeJongh • *Department of Computer Science, Hope College, Holland, MI, USA*

Scott Devoid • *Argonne National Laboratory, MCS Division, Argonne, IL, USA*

Julie Dickerson • *Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA*

Jing Du • *Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA*

Jean-Loup Faulon • *Institute of Systems & Synthetic Biology (ISSB), Evry, France*

Stephen S. Fong • *Department of Chemical and Life Science Engineering, Virginia Commonwealth University, Richmond, VA, USA*

Philippe Gaudin • *DSM Biotechnology Center, Delft, The Netherlands*

Pratish Gawand • *Department of Chemical Engineering and Applied Chemistry, University of Toronto, Toronto, ON, Canada*

Grant Gelderman • *Department of Chemical Engineering, Cockrell School of Engineering, The University of Texas at Austin, Austin, TX, USA*

Ryan T. Gill • *Department of Chemical and Biological Engineering, Engineering Center, University of Colorado Boulder, Boulder, CO, USA*
Christopher M. Gowen • *Department of Chemical Engineering and Applied Chemistry, University of Toronto, Toronto, ON, Canada*
Wilbert H.M. Heijne • *DSM Biotechnology Center, Delft, The Netherlands*
Christopher Henry • *MCS Division, Argonne National Laboratory, Argonne, IL, USA*
Laura Jarboe • *Department of Chemical and Biological Engineering, Iowa State University, Ames, IA, USA*
Lara J. Jazmin • *Department of Chemical and Biomolecular Engineering, Vanderbilt University, Nashville, TN, USA*
Renger H. Jellema • *DSM Biotechnology Center, Delft, The Netherlands*
Tao Jin • *Department of Chemical and Biological Engineering, Iowa State University, Ames, IA, USA*
Max E. Joffe • *Department of Chemical and Biomolecular Engineering, University of Maryland, College Park, MD, USA; Interdisciplinary Graduate Program in the Biomedical and Biological Sciences, Vanderbilt University, Nashville, TN, USA*
Katy C. Kao • *Department of Chemical Engineering, Texas A&M University, College Station, TX, USA*
Radhakrishnan Mahadevan • *Department of Chemical Engineering and Applied Chemistry, University of Toronto, Toronto, ON, Canada*
Thomas J. Mansell • *Department of Chemical and Biological Engineering, Engineering Center, University of Colorado Boulder, Boulder, CO, USA*
Hilly Menke • *DSM Biotechnology Center, Delft, The Netherlands*
Sabine Metzelaar • *DSM Biotechnology Center, Delft, The Netherlands*
Matthew Moura • *Department of Chemical and Biological Engineering, Northwestern University, Evanston, IL, USA*
Shilpa Nargund • *Department of Chemical and Biomolecular Engineering, University of Maryland, College Park, MD, USA*
Hadi Nazem-Bokaee • *Department of Biological Systems Engineering, Virginia Tech, Blacksburg, VA, USA*
Maurien M.A. Olsthoorn • *DSM Biotechnology Center, Delft, The Netherlands*
Ross Overbeek • *Fellowship for Interpretation of Genomes, Burr Ridge, IL, USA*
Stephan Pabinger • *Division for Bioinformatics, Biocenter, Innsbruck Medical University, Innsbruck, Austria*
Gianni Panagiotou • *School of Biological Sciences, The University of Hong Kong, Pokfulam Road, Hong Kong; Novo Nordisk Foundation Center for Biosustainability, Technical University of Denmark, Hørsholm, Denmark*
Jason A. Papin • *Department of Biomedical Engineering, University of Virginia, Charlottesville, VA, USA*
Anne-Gaëlle Planson • *Institute of Systems & Synthetic Biology (ISSB), Evry, France*
Nathan D. Price • *Institute for Systems Biology, Seattle, WA, USA*
Simon Rasmussen • *Department of Systems Biology, Center for Biological Sequence Analysis, Technical University of Denmark, Kgs Lyngby, Denmark*
Luis H. Reyes • *Department of Chemical Engineering, Texas A&M University, College Station, TX, USA*

Carlos M.F.M. Ribeiro • *DSM Biotechnology Center, Delft, The Netherlands*
Liam Royce • *Department of Chemical and Biological Engineering, Iowa State University, Ames, IA, USA*
Cees M.J. Sagt • *DSM Biotechnology Center, Delft, The Netherlands*
Ryan S. Senger • *Department of Biological Systems Engineering, Virginia Tech, Blacksburg, VA, USA*
Shuobo Shi • *Key Laboratory of Systems Bioengineering, Ministry of Education, Tianjin University, Tianjin, China; Department of Biological Engineering, School of Chemical Engineering and Technology, Tianjin University, Tianjin, China*
Thomas Sicheritz-Pontén • *Department of Systems Biology, Center for Biological Sequence Analysis, Technical University of Denmark, Kgs Lyngby, Denmark*
Evangelos Simeonidis • *Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Esch-sur-Alzette, Luxembourg; Institute for Systems Biology, Seattle, WA, USA*
Kieran Smallbone • *Manchester Centre for Integrative Systems Biology, University of Manchester, Manchester, UK*
Ganesh Sriram • *Department of Chemical and Biomolecular Engineering, University of Maryland, College Park, MD, USA*
Hein Stam • *DSM Biotechnology Center, Delft, The Netherlands*
Natalie J. Stanford • *Manchester Centre for Integrative Systems Biology, University of Manchester, Manchester, UK*
Masaru Tomita • *Institute for Advanced Biosciences, Keio University, Fujisawa, Kanagawa, Japan*
Zlatko Trajanoski • *Division for Bioinformatics, Biocenter, Innsbruck Medical University, Innsbruck, Austria*
Dennis Tran • *Department of Chemical and Biomolecular Engineering, University of Maryland, College Park, MD, USA*
Vitali Tugarinov • *Department of Chemistry and Biochemistry, University of Maryland, College Park, MD, USA*
Keith Tyo • *Department of Chemical and Biological Engineering, Northwestern University, Evanston, IL, USA*
D.B.R.K. Gupta Udatha • *Department of Chemical and Biological Engineering, Industrial Biotechnology, Chalmers University of Technology, Gothenburg, Sweden*
Fredoen Valianpour • *DSM Biotechnology Center, Delft, The Netherlands*
Veronika Vonstein • *Fellowship for Interpretation of Genomes, Burr Ridge, IL, USA*
Joseph R. Warner • *OPX Biotechnologies, Inc., Boulder, CO, USA*
James Winkler • *Department of Chemical Engineering, Texas A&M University, College Station, TX, USA*
Laurence Yang • *Intrexon Corporation, San Diego, CA, USA*
Tae Hoon Yang • *Genomatica, Inc., San Diego, CA, USA*
Jamey D. Young • *Department of Chemical and Biomolecular Engineering, Vanderbilt University, Nashville, TN, USA*
Yongbo Yuan • *Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA*
Huimin Zhao • *Department of Chemical and Biomolecular Engineering, Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL, USA;*

Departments of Chemistry, Biochemistry, and Bioengineering, Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Xueming Zhao • *Key Laboratory of Systems Bioengineering, Ministry of Education, Tianjin University, Tianjin, China; Department of Biological Engineering, School of Chemical Engineering and Technology, Tianjin University, Tianjin, China*

Part I

Modeling and Simulation Tools

Chapter 1

Genome-Scale Model Management and Comparison

Stephan Pabinger and Zlatko Trajanoski

Abstract

Spurred by recent innovations in genome sequencing, the reconstruction of genome-scale models has increased in recent years. Genome-scale models are now available for a wide range of organisms, and models have been successfully applied to a number of research topics including metabolic engineering, genome annotation, biofuel production, and interpretation of omics data sets. The challenge is how to manage the large amount of data in genome-scale models and perform comparative analysis to gain new biological insights. In this chapter, important standards for genome-scale modeling are outlined. Furthermore, management strategies as well as existing repository and construction tools are discussed. As the comparison of models is an important aspect during the development and analysis stages, available methods are presented and existing software solutions are reviewed.

Key words: Systems biology, Metabolomics, Genome-scale modeling, Model management, Model comparison, Data standards, Software tools

1. Introduction

Recent advances in genome sequencing and the development of comprehensive high-throughput experiments and techniques have fostered the assembly of genome-scale metabolic models (1). The reconstruction efforts aim at incorporating all known metabolic reactions for a particular organism into a standardized format which allows researchers to study all processes and dynamic interactions at molecular level.

Due to the extraordinary growth in industrial production of bulk chemicals, pharmaceuticals, enzymes, and biofuels (2), the field of metabolic modeling has received increasing attention within the last few years (3, 4). In the past decade, more than 50 genome-scale metabolic models have been built for a variety of single- and multicellular organisms (5).

By converting reconstructions into a mathematical system, models can be used to reproduce assumptions and parameters of

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_1,

an observed behavior and may help to increase the knowledge of its dynamic functionality. Thus, in silico models can be employed to predict phenotypes or the outcome of different perturbations for an investigated system using a variety of emerging mathematical techniques (6). Genome-scale models have already proven to be valuable for strain engineering, which aims at improving production yield and stability (7). Moreover, they provide an additional way to analyze various omics data sets (8).

All genome-scale models have in common that they comprise a huge amount of metabolites and connecting reactions, which requires advanced tools for efficient analysis and comparison (9). Databases and repositories are needed to store and manage the wealth of information, and sophisticated software tools are required to analyze the data and develop new models.

In this chapter, important aspects for managing and comparing genome-scale models are outlined. It starts with a description of standards that should be taken into account when constructing and managing metabolic models (Subheading 2). Next, details about the management process are discussed (Subheading 3) followed by approaches and tools for comparing genome-scale models (Subheading 4).

2. Standards for Genome-Scale Models

For a long time, biologists have used various notations to describe different aspects of processes, objects, and observations. A noticeable example is the multitude of naming schemes for genes and proteins, which are often incompatible between and even within organisms (10). Public data repositories frequently contain overlapping entries which describe the same biological objects but use different names and identifiers (11).

Nowadays, the community has been recognizing the need for consistency in naming biological objects (e.g., genes and proteins) and the importance of unique identifiers (12). In order to ensure the reusability of a model and allow comparisons between models, it is essential that synonyms of one biological object can be mapped back to official names (11). Furthermore, the utility of reconstructions that apply to certain standards is greatly enhanced by the use of semantic annotations that unambiguously describe each compartment, metabolite, and reaction within the reconstruction (13). The consideration and use of established standards for both the used software and the reconstruction process itself ensures data integrity and compatibility. In addition, it is important that a structure is provided in which different kinds of data can be efficiently represented and exchanged. Despite the recent improvements of

standards, the community is currently lacking a fixed structure for linking experimental data to models, which should be addressed in the future of genome-scale modeling (11) (see Note 1).

In the following paragraphs, important data standards and conventions are described, which should be considered when working with genome-scale models.

2.1. SBML and CellML

The Systems Biology Markup Language (SBML) (14) is an XML-based exchange format for the description of structural and dynamic biological models. It represents the minimum information needed to simulate the models' dynamic behavior and provides a common intermediate format that can be used to define models in regulatory networks, signaling pathways, gene regulation networks, and metabolic pathways (10). SBML is used by a large set of software applications (see Note 2).

The CellML (15) language is another XML-based markup language designed for the description of mathematical models. The purpose of CellML is to store and exchange models in a format that enables researchers to reuse components from one model to another. CellML is similar to SBML but is not specific to descriptions of biochemistry and provides greater scope for model modularity and reuse. SBML defines compartment elements using a dedicated outside attribute, while CellML lacks information on spatial dimensions. The description of a reaction is more detailed in CellML than in SBML since both substance and reaction are represented by components that are connected by linking variables (16).

All SBML elements can be annotated with Systems Biology Ontology (SBO) (17) terms, which is useful for additional semantic information. The CellML framework lacks a dedicated tag and is not associated with any particular ontology. It uses a metadata framework for referencing components in external resources. Although SBML is widely used in the community, it currently puts little effort into describing substance types and experimental evidence (16). However, it provides several features for describing information important for simulations, such as initial amount and concentration of the metabolites.

2.2. MIRIAM

The "Minimum Information Requested In the Annotation of biochemical Models" (MIRIAM) (18) specification proposes a set of rules for curating quantitative models and defines procedures for encoding and annotating reconstructions of biological systems. MIRIAM identifiers are used to specify the biological meaning of SBML elements where each identifier is a single unique string that unambiguously references one object in an external resource. It uses an annotation scheme for external resources that requires the use of unique resource identifiers (URIs) to determine model constituents, such as the model itself, compartments, metabolites, or

reactions. In SBML, MIRIAM annotations are implemented with the help of the RDF (Resource Description Framework) format. RDF is a semantic web language, which is used to formally describe information about objects by using unique identifiers.

2.3. SBGN

The Systems Biology Graphical Notation (SBGN) (19) is a standard for visualizing biological network models. It is intended to describe a system in detail and without any ambiguities to foster the efficient storage, exchange, and reuse of information. SBGN is made up of three languages that represent different views of biological systems: Process Description Language (PDL), describing the biochemical reactions responsible for the behavior of the system; Entity Relationship Language (ERL), which focuses on the relations that are supposed to represent independent information; and Activity Flow Language (AFL), which visualizes how information is propagated through a biological system.

2.4. BioPAX

Biological Pathway Exchange (BioPAX) (20) is a standard language to represent metabolic and signaling pathways, molecular and genetic interactions, as well as gene regulation networks. It aims at facilitating the exchange of pathway data and biological models. Furthermore, the BioPAX community tries to coordinate the work on improving standards with the SBML, CellML, and SBGN modeling standards.

2.5. Annotation of Models

Although standards have been available for some time, many existing models rely on proprietary naming of metabolites and enzymes. Reconstructions are based on a multitude of different sources and may contain incomplete and contradictory information about *putative* and *probable* genes. Annotations are a valuable method to reference model components into external resources. The general suggestion is to store metadata as links to ontologies or controlled vocabulary repositories instead of using free text (16). Moreover, previously described standards should be considered when developing new models to create a comprehensive and consistent reconstruction. A strong advantage of annotations in genome-scale models is the ability to find true differences between the organism, rather than artifacts from the reconstruction process. Although annotating a model requires extra work during the construction process, the effort will pay off as models have a higher impact and remain valid in other contexts (10) (see Note 3). However, annotations should be carefully selected since they might be unspecific or simply wrong. Furthermore, they may describe the same chemical entity or process, but point to a different web resource, or entries of different web resources share subtle biochemical relationships. As the integration of annotations in existing models can be a tedious work, several tools have been developed to ease the annotation process (21, 22).

SemanticSBML (23) is a web application for viewing and editing biochemical models. It supports adding and modifying annotations within a model and allows users to check the validity of annotations. Moreover, models can be changed or automatically merged where equivalent elements from two models are matched together. The freely available Java library libAnnotationSBML (24) aims at facilitating the manipulation of SBML annotation terms and modifying already annotated models.

The Metannogen (25) software suite allows researchers to browse and annotate existing biological models using the MIRIAM standard. It provides direct access to the models stored in the BioModels (26) database and Java Web Simulation (JWS) (27) online system and supports the reconstruction of new networks.

3. Management

The management of genome-scale reconstructions includes constructing, storing, and sharing models. Figure 1 provides an overview of methods and aspects that need to be considered for managing and comparing models. The metabolic network reconstruction process is usually labor intensive and includes several iterative cycles between experiments and model updates, which leads to the generation of multiple intermediate model versions (5, 28). To assist researchers in iterative model building, a variety of applications have been published which are discussed in Subheading 3.1.

Important prerequisites when choosing a repository for storing models are the provided data security mechanisms and the offered tools for further model analysis. Moreover, the ability to easily export models for analysis into external tools should be available in the used application. To allow the exchange of metabolic reconstructions, it is necessary that data and models are described according to community-accepted standards and that sufficient annotation and metadata are available. As conforming to common standards may be time-consuming and complicated, existing data management systems try to improve this process by providing tooling, expertise, and best practice guidelines (11). Software systems should be selected which support researchers by providing extra tooling to help understand which minimum information is required. However, a difficult aspect of data management is the reluctance of researchers to release their data, and in most cases models are only shared at the point of publication. Researchers should therefore ensure that data is secured when new unpublished models are analyzed or made public, and ideally the used software should support sharing of individual data sets. Furthermore, sharing with colleagues and the research community should also be considered.

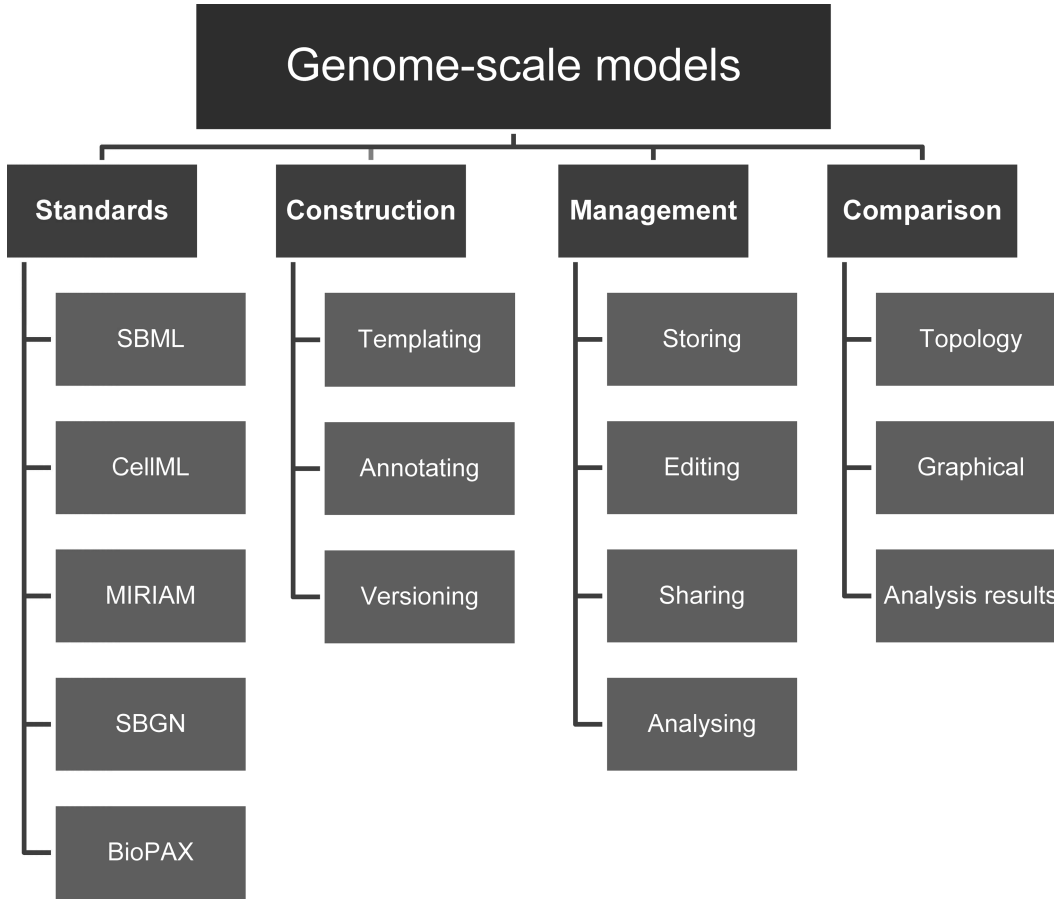

Fig. 1. Depicted are various aspects that need to be taken into account when managing and comparing genome-scale models.

The flexibility and adaptability of biological systems to varying external conditions are of high importance to the experimental context in which data was obtained. Therefore, data supplementing genome-scale models should be documented with enough description to enable reproducibility of analysis and the reconstruction results.

To this end, a number of repositories for collecting genome-scale models of biological systems have been published. They are a great way to share new reconstructions and usually provide sophisticated querying and browsing functionality. Repositories should contain all additional information about a specific reconstruction which is necessary for model interpretation without needing a reliance on the associated publication (11).

The generation of new genome-scale metabolic models has been described in a protocol, which comprises 96 distinct steps (28). Usually, reconstructions from genome annotation undergo multiple rounds of manual curation and may contain several gaps or

incorrect annotations (29). As a result, numerous intermediate versions are created which should be correctly versioned and managed. As the process is usually very labor- and time-intensive, a number of software tools and packages have been published, which try to facilitate the reconstruction of new models.

In the following section, common tools are discussed that provide repository functionality for genome-scale models as well as assist researchers in the creation of new metabolic reconstructions (see Note 4). Table 1 provides a summary of mentioned applications and compares availability as well as reconstruction, repository, comparison, and analysis functionalities.

3.1. Software Tools

BioModels Database (26) is a popular database for storing, searching, and retrieving published mathematical models. Currently, BioModels is available in its twenty-third release and contains 424 fully curated, MIRIAM-compliant models and 443 other reconstructions. Models are annotated and cross-referenced with external data resources, such as publications, databases of compounds and pathways, and controlled vocabularies. BioModels supports the download of models in *SBML format* as well as various other formats. The models in its curated section have been described in peer-reviewed publications and were edited by professional curators to ensure a high quality. The curation process ensures that a model is syntactically correctly encoded and verifies the accuracy of the biological information. Moreover, the logical model composition is checked, and the reproducibility of the described behavior is tested. New models can be submitted in SBML or CellML format and undergo a curation phase before publishing. In addition to the online system, the BioModels Database itself can be stored locally if researchers do not want to publicly share their models but still use the whole functionality of the application.

The Java Web Simulation (JWS) (27) application provides a collection of curated models of biological systems. User-generated models can be uploaded and stored in the JWS database and need to be encoded either in the SBML format or in the JWS format. Furthermore, the JWS online interface allows users to run simulations and analyses on these models in a web browser via an easy-to-use interface. In addition, it provides an automated SBGN schema generator and a tool for adding MIRIAM annotations.

The knowledge base of biochemically, genetically, and genomically structured genome-scale metabolic network reconstructions (BIGG) (29) is currently hosting 10 different genome-scale models. All reactions in the database are mass and charge balanced, and models can be browsed and exported as SBML files. Due to the underlying application infrastructure, it is not easily possible to incorporate new metabolic reconstructions. For visualization purposes, curated metabolic maps are provided for many organisms.

Table 1
Software tools for model management, comparison, and analysis

Name	Online/offline	Repository	Construction	Comparison	Analysis	Publication
BIGG	✓/–	✓	–	–	–	(29)
BioCyc	✓/✓	✓	✓	✓	✓	(30)
BioModels	✓/✓	✓	–	–	–	(26)
BridgeDb	–/✓	–	✓	–	–	(22)
CADLIVE	–/✓	–	✓	–	✓	(37)
CycSim	✓/–	✓	✓	–	✓	(48)
FAME	✓/✓	–	✓	–	✓	(9)
FMM	✓/–	–	✓	–	–	(49)
JigCell	–/✓	–	✓	✓	✓	(50)
JWS	✓/–	✓	–	–	✓	(27)
MEMOSys	✓/✓	✓	✓	✓	–	(35)
Metannogen	✓/✓	–	✓	–	–	(25)
MetExplore	✓/–	✓	–	–	✓	(33)
MetNetMaker	–/✓	–	✓	–	–	(36)
Model SEED	✓/–	–	✓	–	✓	(1)
OptFlux	–/✓	–	–	✓	✓	(40)
Pathway Tools	–/✓	–	✓	✓	✓	(31)
rBioNet	–/✓	–	✓	–	–	(39)
Saint	✓/✓	–	✓	–	–	(21)
SEEK	✓/✓	✓	–	–	✓	(11)
SemanticSBML	–/✓	–	✓	✓	–	(23)
VANTED	–/✓	–	✓	–	✓	(51)

The BioCyc collection (30) includes databases for most completely sequenced eukaryotic and prokaryotic species. Each database contains the genome and the predicted metabolic network of one organism. BioCyc provides global and comparative analysis methods for genomes and metabolic networks. The downloadable version allows researchers to create own databases and build new metabolic models.

The Pathway Tools (31) software system has been developed to support, amongst other use cases, the creation, annotation, comparison, and analysis of metabolic models. It assists researcher in the

development of new networks based on the annotated genome by integrating several bioinformatics data types. The analysis functionality for metabolic models facilitates finding dead-end metabolites and the identification of potential drug targets. Pathway Tools provides advanced querying tools and offers an automatic display functionality of metabolic pathways and full metabolic networks.

In addition to the mentioned widely used software suites, new interesting projects have been published that provide unique features for sharing and storing genome-scale models. The web-based software FAME (9) offers visualization of metabolic models by generating interactive maps with elements that can be clicked to access additional information. It allows users to upload their own models or build new models based on the Kyoto Encyclopedia of Genes and Genomes (KEGG) (32) database. The application features easy edition of the flux bounds on all internal and exchange reactions, editing existing reactions, and modifying objectives and export models as SBML. SEEK (11) is an open-source software project that uses a web-based infrastructure to enable sharing and exchanging biological models. It allows researchers to analyze their data with commonly used tools from the community. MetExplore (33) is an online accessible application which stores metabolic networks of 160 organisms using information mainly from BioCyc-like databases. It proposes several functions to perform flux balance analysis (FBA) (34) and supports mapping of metabolomics experiments onto filtered metabolic networks.

The metabolic model research and development system (MEMOSys) (35) is an online application for the management, storage, and development of genome-scale metabolic models. It has been specifically designed to support researchers in the iterative model development process by providing a built-in automatic version control system. The system stores each addition, modification, or deletion of an entry as a new revision, which allows researchers to access the complete developmental history of reactions, metabolites, and models. Moreover, for each version of a model, the complete network can be queried, compared, and exported. MEMOSys features private and public models as well as a supervision process for including new modifications into existing models. Components can be annotated with references to external databases using the MIRIAM notation, and all models can be exported in the SBML format. MEMOSys is freely available and can be locally installed.

The generation of new models poses various challenges to researchers, which are addressed by several software tools. The platform Model SEED (1) assists users in the generation, optimization, curation, and analysis of genome-scale metabolic models. It is capable of generating draft models based on the assembled genome sequence. The generation workflow includes genome annotation, creation of biomass reactions, reaction network assembly, gap

filling, and model optimization. Various completed models are maintained by scientists, and reconstructions can be analyzed using flux variability analysis (FVA) and FBA. The open-source tool MetNetMaker (36) assists researchers in the creation of FBA-ready metabolic networks. It uses the KEGG LIGAND naming convention and offers a reaction creator to design custom reactions. The application features the determination of dead-end compounds (compounds which are created but not used or used and not created) and allows users to save models in Excel or SBML files. The CADLIVE (computer-aided design of living systems) (37) application has been designed for constructing and analyzing large-scale biological networks. Networks can be graphically edited or constructed and are stored in a database. CADLIVE features the automatic conversion of biochemical network maps into mathematical models to simulate model dynamics and analyze topological features. It offers a pathway search module for virtual knockout mutants and is capable of automatically creating layouts of biochemical network maps by calculating the coordinates of all metabolites. Embedded within the COBRA toolbox (38) is the tool rBioNet (39), which supports the reconstruction of new genome-scale models. Metabolites and reactions can be imported from existing models, whereby newly created networks are checked for dead-end metabolites as well as mass and charge balance. Models can be exported in the SBML file format.

4. Comparison

An important feature of genome-scale models is their possibility to extract biologically relevant information from the wealth of existing data. The comparison of models is one valid approach to reveal interesting insights based on the relationship between the structure of a metabolic network and the resulting phenotype of an organism. In addition, the development of novel models is often based on existing metabolic reconstructions, where comparison results could help in finding white spots in the cellular networks that might deserve further modeling efforts. During the reconstruction process, it is therefore reasonable to search for relevant models and check how models differ, overlap, and complement each other (23).

In order to allow a biological meaningful comparison of models and avoid comparing reconstruction artifacts, components need to be annotated with unique identifiers (11). Without standardized annotations, it can be impossible to determine whether the same biological object is being referred to across multiple data files. Public databases and commonly used resources provide a collection of official names that should be used for biological entities.

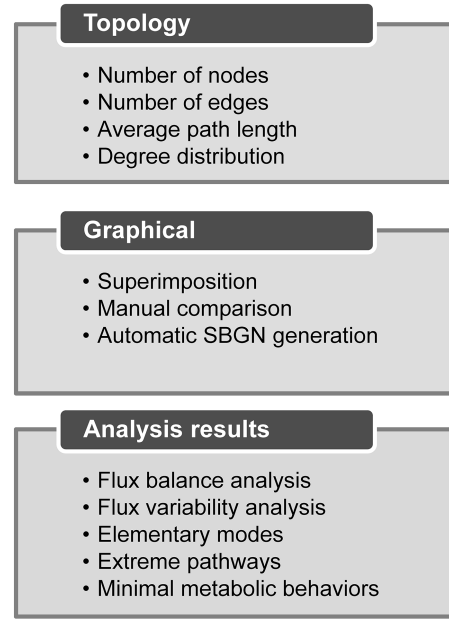

Fig. 2. Displayed are several methods to compare genome-scale models.

However, the decision to include or remove certain genes in a new model still involves a degree of subjectivity based on the used data sources (10) (see Note 5). Thus, the comparison result of models could still yield undesirable differences even with the careful use of standards and annotations.

The comparison of models can be based on various biological or mathematical aspects: (1) details of the network topology, (2) graphical differences, or (3) analysis results (see Fig. 2). Differences in the topology of a model can be calculated by comparing, among others, number of nodes and edges, the average path lengths, or degree distribution.

Recently, Oberhardt et al. (5) presented a method for aligning metabolic reconstructions of two organisms where nonbiological differences are removed from the models, while the reconstructions are kept as true as possible to the underlying biological data. Therefore, models can be compared against one another in confidence that differences reflect biological differences instead of noise. The graphical comparison of networks is supported by the OptFlux (40) application, which provides a visualization module to superimpose metabolic models. Another tool that allows the graphical comparison of models is semanticSBML (23). The software is capable of aligning several models in a tree view and can be used to detect similarities and conflicts between their elements. The Pathway Tool's suite (31) offers comparative analyses of organism-specific databases to identify differences in reactions, pathways, and genome organization. The comparison functionality

of MEMOSys (35) allows researchers to compare different models or different versions of one model. The system reports comparison results for reactions, metabolites, and genes using graphical Venn diagrams and detailed lists of equal and unique entities for each model.

Differences between models can also be characterized by comparing the result of analysis methods. Flux balance analysis, elementary modes (41), extreme pathways (42), and minimal metabolic behaviors are widely used methods to investigate the behavior of metabolic models. Over the past years, several toolboxes have been developed which can be applied to study and analyze genome-scale models (38, 43–46).

5. Notes

1. Each model should specify what data set was used for construction and validation. A possible place is the addition of an annotation directly in the model.
2. A good resource to find more about additional software packages is the SBML website: http://sbml.org.
3. It is a good practice to annotate models during the reconstruction process and not just at the time of publication.
4. Meta-information on existing web repositories and databases is listed in the PathGuide repository: http://www.pathguide.org.
5. To increase the usability of reconstructions, the Gene Ontology (GO) (47) Evidence Code may be used, which describes the confidence of model elements with terms such as "inferred by curator" or "inferred from experiment."

References

1. Henry CS, DeJongh M, Best AA et al (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. Nat Biotechnol 28:977–982
2. de Jong B, Siewers V, Nielsen J (2012) Systems biology of yeast: enabling technology for development of cell factories for production of advanced biofuels. Curr Opin Biotechnol 23(4):624–630
3. Gavrilescu M, Chisti Y (2005) Biotechnology-a sustainable alternative for chemical industry. Biotechnol Adv 23:471–499
4. Hatti-Kaul R, Törnvall U, Gustafsson L et al (2007) Industrial biotechnology for the production of bio-based chemicals–a cradle-to-grave perspective. Trends Biotechnol 25:119–124
5. Oberhardt MA, Puchałka J, Martins dos Santos VAP et al (2011) Reconciliation of genome-scale metabolic reconstructions for comparative systems analysis. PLoS Comput Biol 7: e1001116
6. Oberhardt MA, Palsson BØ, Papin JA (2009) Applications of genome-scale metabolic reconstructions. Mol Syst Biol 5:320
7. Lee SY, Lee D-Y, Kim TY (2005) Systems biotechnology for strain improvement. Trends Biotechnol 23:349–358
8. Patil KR, Nielsen J (2005) Uncovering transcriptional regulation of metabolism by using metabolic network topology. Proc Natl Acad Sci U S A 102:2685–2689

9. Boele J, Olivier BG, Teusink B (2012) FAME, the Flux Analysis and Modeling Environment. BMC Syst Biol 6:8
10. Krause F, Schulz M, Swainston N et al (2011) Sustainable model building the role of standards and biological semantics. Methods Enzymol 500:371–395
11. Wolstencroft K, Owen S, du Preez F et al (2011) The SEEK: a platform for sharing data and models in systems biology. Methods Enzymol 500:629–655
12. Howe D, Costanzo M, Fey P et al (2008) Big data: the future of biocuration. Nature 455:47–50
13. Herrgård MJ, Swainston N, Dobson P et al (2008) A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. Nat Biotechnol 26:1155–1160
14. Hucka M, Finney A, Sauro HM et al (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19:524–531
15. Miller AK, Marsh J, Reeve A et al (2010) An overview of the CellML API and its implementation. BMC Bioinformatics 11:178
16. Strömbäck L, Hall D, Lambrix P (2007) A review of standards for data exchange within systems biology. Proteomics 7:857–867
17. Courtot M, Juty N, Knüpfer C et al (2011) Controlled vocabularies and semantics in systems biology. Mol Syst Biol 7:543
18. Le Novère N, Finney A, Hucka M et al (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). Nat Biotechnol 23:1509–1515
19. Le Novère N, Hucka M, Mi H et al (2009) The systems biology graphical notation. Nat Biotechnol 27:735–741
20. Demir E, Cary MP, Paley S et al (2010) The BioPAX community standard for pathway data sharing. Nat Biotechnol 28:935–942
21. Lister AL, Pocock M, Taschuk M et al (2009) Saint: a lightweight integration environment for model annotation. Bioinformatics 25:3026–3027
22. van Iersel MP, Pico AR, Kelder T et al (2010) The BridgeDb framework: standardized access to gene, protein and metabolite identifier mapping services. BMC Bioinformatics 11:5
23. Krause F, Uhlendorf J, Lubitz T et al (2010) Annotation and merging of SBML models with semanticSBML. Bioinformatics 26:421–422
24. Swainston N, Mendes P (2009) libAnnotationSBML: a library for exploiting SBML annotations. Bioinformatics 25:2292–2293
25. Gille C, Hübner K, Hoppe A et al (2011) Metannogen: annotation of biological reaction networks. Bioinformatics 27:2763–2764
26. Li C, Donizelli M, Rodriguez N et al (2010) BioModels Database: an enhanced, curated and annotated resource for published quantitative kinetic models. BMC Syst Biol 4:92
27. Snoep JL, Olivier BG (2002) Java Web Simulation (JWS); a web based database of kinetic models. Mol Biol Rep 29:259–263
28. Thiele I, Palsson BØ (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. Nat Protoc 5:93–121
29. Schellenberger J, Park JO, Conrad TM et al (2010) BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions. BMC Bioinformatics 11:213
30. Karp PD, Ouzounis CA, Moore-Kochlacs C et al (2005) Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. Nucleic Acids Res 33:6083–6089
31. Karp PD, Paley SM, Krummenacker M et al (2010) Pathway Tools version 13.0: integrated software for pathway/genome informatics and systems biology. Brief Bioinform 11:40–79
32. Kanehisa M, Goto S, Sato Y et al (2012) KEGG for integration and interpretation of large-scale molecular data sets. Nucleic Acids Res 40: D109–D114
33. Cottret L, Wildridge D, Vinson F et al (2010) MetExplore: a web server to link metabolomic experiments and genome-scale metabolic networks. Nucleic Acids Res 38:W132–W137
34. Orth JD, Thiele I, Palsson BØ (2010) What is flux balance analysis? Nat Biotechnol 28:245–248
35. Pabinger S, Rader R, Agren R et al (2011) MEMOSys: bioinformatics platform for genome-scale metabolic models. BMC Syst Biol 5:20
36. Forth T, McConkey GA, Westhead DR (2010) MetNetMaker: a free and open-source tool for the creation of novel metabolic networks in SBML format. Bioinformatics 26:2352–2353
37. Kurata H, Inoue K, Maeda K et al (2007) Extended CADLIVE: a novel graphical notation for design of biochemical network maps and computational pathway analysis. Nucleic Acids Res 35:e134
38. Schellenberger J, Que R, Fleming RMT et al (2011) Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nat Protoc 6:1290–1307

39. Thorleifsson SG, Thiele I (2011) rBioNet: a COBRA toolbox extension for reconstructing high-quality biochemical networks. Bioinformatics 27:2009–2010
40. Rocha I, Maia P, Evangelista P et al (2010) OptFlux: an open-source software platform for in silico metabolic engineering. BMC Syst Biol 4:45
41. Schuster S, Dandekar T, Fell DA (1999) Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. Trends Biotechnol 17:53–60
42. Schilling CH, Letscher D, Palsson BO (2000) Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. J Theor Biol 203:229–248
43. Heino J, Calvetti D, Somersalo E (2010) Metabolica: a statistical research tool for analyzing metabolic networks. Comput Methods Programs Biomed 97:151–167
44. Hoops S, Sahle S, Gauges R et al (2006) COPASI–a COmplex PAthway SImulator. Bioinformatics 22:3067–3074
45. Jensen PA, Lutz KA, Papin JA (2011) TIGER: toolbox for integrating genome-scale metabolic models, expression data, and transcriptional regulatory networks. BMC Syst Biol 5:147
46. Cvijovic M, Olivares-Hernández R, Agren R et al (2010) BioMet Toolbox: genome-wide analysis of metabolism. Nucleic Acids Res 38: W144–W149
47. Ashburner M, Ball CA, Blake JA et al (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet 25:25–29
48. Le Fèvre F, Smidtas S, Combe C et al (2009) CycSim–an online tool for exploring and experimenting with genome-scale metabolic models. Bioinformatics 25:1987–1988
49. Chou C-H, Chang W-C, Chiu C-M et al (2009) FMM: a web server for metabolic pathway reconstruction and comparative analysis. Nucleic Acids Res 37:W129–W134
50. Vass MT, Shaffer CA, Ramakrishnan N et al (2006) The JigCell model builder: a spreadsheet interface for creating biochemical reaction network models. IEEE/ACM Trans Comput Biol Bioinform 3:155–164
51. Junker BH, Klukas C, Schreiber F (2006) VANTED: a system for advanced data analysis and visualization in the context of biological networks. BMC Bioinformatics 7:109

Chapter 2

Automated Genome Annotation and Metabolic Model Reconstruction in the SEED and Model SEED

Scott Devoid, Ross Overbeek, Matthew DeJongh, Veronika Vonstein, Aaron A. Best, and Christopher Henry

Abstract

Over the past decade, genome-scale metabolic models have proven to be a crucial resource for predicting organism phenotypes from genotypes. These models provide a means of rapidly translating detailed knowledge of thousands of enzymatic processes into quantitative predictions of whole-cell behavior. Until recently, the pace of new metabolic model development was eclipsed by the pace at which new genomes were being sequenced. To address this problem, the RAST and the Model SEED framework were developed as a means of automatically producing annotations and draft genome-scale metabolic models. In this chapter, we describe the automated model reconstruction process in detail, starting from a new genome sequence and finishing on a functioning genome-scale metabolic model. We break down the model reconstruction process into eight steps: submitting a genome sequence to RAST, annotating the genome, curating the annotation, submitting the annotation to Model SEED, reconstructing the core model, generating the draft biomass reaction, auto-completing the model, and curating the model. Each of these eight steps is documented in detail.

Key words: Model SEED, RAST, Automated metabolic model reconstruction, Flux balance analysis, Gap filling, Microbial metabolism, Systems metabolic engineering

1. Introduction

Over the past decade, genome-scale metabolic models have proven to be a crucial resource for predicting organism phenotypes from genotypes. These models provide a means of rapidly translating detailed knowledge of thousands of enzymatic processes into quantitative predictions of whole-cell behavior. They have been applied extensively to identify essential genes and genes sets, predict organism phenotypes and growth conditions, design metabolic engineering strategies, and simulate the effects of transcriptional regulation on organism behavior (1). Yet until recently, the pace of new metabolic model development was eclipsed by the pace at which new genomes were being sequenced. To address this problem, the Model SEED

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_2,

framework (http://www.theseed.org/models/) (2) was developed as a means of automatically producing draft genome-scale metabolic models to increase the pace of new model development and close the gap between the number of metabolic models and the number of sequenced genomes. The Model SEED integrates existing methodologies (3–9) and introduces new techniques to automate nearly every step of the metabolic reconstruction process (10), enabling generation of functioning draft models from assembled genome sequences in approximately 48 h. Today, the Model SEED has been applied to generate over 15,000 draft metabolic models, including a model of the over 3,500 complete prokaryotic genome sequences currently available in GenBank (11).

A genome-scale metabolic model consists of three primary components: (1) a list of reactions that take part in the metabolic pathways of the organism including reaction stoichiometry and reversibility, (2) a set of gene–protein-reaction (GPR) associations that capture how gene activity is related to the activity of metabolic reactions, and (3) a biomass composition reaction that indicates which small molecules must be produced for an organism to grow and divide (12). All of these components are used in a method called flux balance analysis (FBA) to simulate microbial metabolism in a specified environmental condition.

FBA involves the use of linear optimization to define the limits on the metabolic capabilities of a model organism by assuming that the interior of the cell exists in a quasi-steady-state (13–16). This quasi-steady-state assumption is enforced by a set of linear mass balance constraints written for each metabolite included in the model. These mass balance constraints and reaction flux bounds form a set of underdetermined linear equations with many possible solutions. Because these equations are underdetermined, an optimization criterion is used to capture the most physiologically relevant region of the solution space. The optimization criteria vary depending on the application, but the most common criterion is the maximization of growth yield (16, 17). Maximum growth yield is simulated by maximizing the flux through the biomass reaction in the model, while the uptake of nutrients is fixed at a specific ratio. This is a meaningful optimization criterion because organisms have been observed to grow at the maximum predicted yield when nutrients are plentiful (18).

In this chapter, we provide detailed descriptions of how to construct a new draft genome-scale metabolic model starting from a new unannotated genome sequence. We describe in detail how to use the SEED (4) and RAST (3) tools to annotate a genome and review the genome annotations. We then describe how to use the Model SEED and other tools to construct, review, and analyze a metabolic model from the RAST annotation. We also move beyond simple descriptions of how to use these tools to also include details on how these tools work "under the hood." These details are useful

for developing a complete understanding of the data and assumptions that enter into the automated model reconstruction process. Overall, we break down the automated model reconstruction process into eight sequential steps including (1) submitting a genome sequence to RAST, (2) annotation of the genome, (3) review and curation of the annotation, (4) submitting a RAST annotation to Model SEED, (5) reconstruction of a core metabolic model, (6) generation of a draft biomass composition reaction, (7) autocompletion of the metabolic model, and (8) review and curation of the metabolic model.

2. Materials

2.1. Requirements for Submitting a Genome Sequence to Automated Annotation

1. A FASTA file containing the sequence of all chromosomes and plasmids for the microbial genome to be annotated, typically obtained from the submitters' own sequencing project or GenBank (http://www.ncbi.nlm.nih.gov/genbank/).
2. Web access to the RAST home page (http://rast.nmpdr.org/) or an installation of the myRAST desktop application (http://blog.theseed.org/servers/).
3. A user account in the SEED, which can be freely obtained via the SEED registration page (http://rast.nmpdr.org/?page=Register).

2.2. Data Supporting the RAST Approach to Automated Genome Annotation

1. A curated, controlled vocabulary of functional roles that define the specific biological functions that will be mapped onto genes in the annotation process (e.g., pyruvate kinase (EC 2.7.1.40)).
2. An organization of related functional roles into a set of well-curated subsystems (e.g., glycolysis and gluconeogenesis).
3. A large number of diverse microbial genome sequences to serve as the reference genomes to which all other genome sequences will be mapped for initial annotation.
4. A database of protein families, called FIGfams (19) in the SEED, that represent isofunctional homologues.

2.3. Data Supporting the Curation of RAST Genome Annotations

1. Web access to the RAST (http://rast.nmpdr.org/) and PubSEED (http://pubseed.theseed.org) home pages.
2. At least some genome sequences of organisms that are phylogenetically close to your organism to support comparative genomics approaches.

3. Some expertise in the subsystems you are curating, including knowledge of how biological functions in the subsystem interact (e.g., neighboring steps in a metabolic pathway).

2.4. Requirements for Submitting a Genome for Automated Model Reconstruction

1. A genome sequence annotated by RAST (see Subheading 3.1) or a genome currently available in the PubSEED (http://pubseed.theseed.org).
2. Web access to the Model SEED home page (http://www.theseed.org/models/).
3. A user account in the SEED, which can be freely obtained via the SEED registration page (http://rast.nmpdr.org/?page=Register).

2.5. Data Supporting Reconstruction of Metabolic Models in Model SEED

1. A comprehensive database of the biochemical reactions that comprise the known metabolic pathways that will be included in the models.
2. An annotation ontology with a strict controlled vocabulary.
3. A curated mapping from the reactions in the biochemistry database to protein complexes and from protein complexes to functional roles in the annotation ontology.
4. A genome with genes consistently annotated with the annotation ontology.
5. A list of spontaneous and universal reactions that should be added to all models.

2.6. Data Supporting Generation of Biomass Composition Reactions in Model SEED

1. An estimation of the fraction of biomass that consists of DNA, RNA, protein, lipids, cell wall, and cofactors and an estimation of growth-associated ATP consumption.
2. An approximate estimation of the amino acid composition of protein, the nucleotide composition of RNA, and the GC content of the genome (see Notes 3–4).
3. An annotation ontology with a strict controlled vocabulary.
4. A list of potential lipid, cell wall, and cofactor with conditions on what metabolic functions and subsystems are indicative of a dependence on these metabolites (stated in terms of the annotation ontology).
5. A genome with genes consistently annotated with the annotation ontology.

2.7. Data Supporting Model Auto-completion in the Model SEED

1. A biochemistry database for which generic reactions, unbalanced reactions, nonmicrobial reactions, and lumped reactions have been removed.
2. A media condition in which the auto-completion will be performed.

3. A mapping between reactions and compounds in the model and reactions and compound in the biochemistry database.
4. Optimization software capable of solving a large-scale mixed-integer optimization problem.

2.8. Requirements for Reviewing and Curating a Model SEED Model

1. Access to the Model SEED website (http://www.theseed.org/models/).
2. Cytoscape (http://www.cytoscape.org/) and CytoSEED plugin (http://sourceforge.net/projects/cytoseed/) for viewing metabolic models.
3. Software for running flux balance analysis on metabolic models using SBML files. For example, the COBRA Toolbox (opencobra.sourceforge.net/) or OptFlux (www.optflux.org/).

3. Methods

As described above, the automated genome annotation and metabolic model reconstruction process using the SEED and Model SEED approach can be broken down into eight sequential steps, which we describe in detail here: (1) submitting a genome sequence to RAST, (2) annotation of the genome, (3) review and curation of the annotation, (4) submitting a RAST annotation to Model SEED, (5) reconstruction of a core metabolic model, (6) generation of a draft biomass composition reaction, (7) auto-completion of the metabolic model, and (8) review and curation of the metabolic model.

3.1. Submitting a Genome Sequence to RAST for Automated Annotation

One of the simplest methods for obtaining a consistent annotation for a new genome sequence is to submit the sequence to the RAST server for genome annotation. The RAST server has essentially automated the genome annotation process, requiring the user to supply only a genome sequence and a few simple parameters to get the process started. Once started, the entire annotation process is typically complete in less than 48 h. Here, we provide step-by-step instructions on how to submit a genome sequence for annotation in RAST using our genome submission website: http://rast.nmpdr.org/.

1. All users desiring to submit a genome to RAST for annotation must first register for a SEED user account. Registration is completely open and free of charge and simply provides a mechanism by which we can assign job ownership and ensure private access to submitted genomes and subsequent genome annotations. New users can register for accounts using the account registration webpage on RAST: http://rast.nmpdr.org/?page=Register (Fig. 1a).

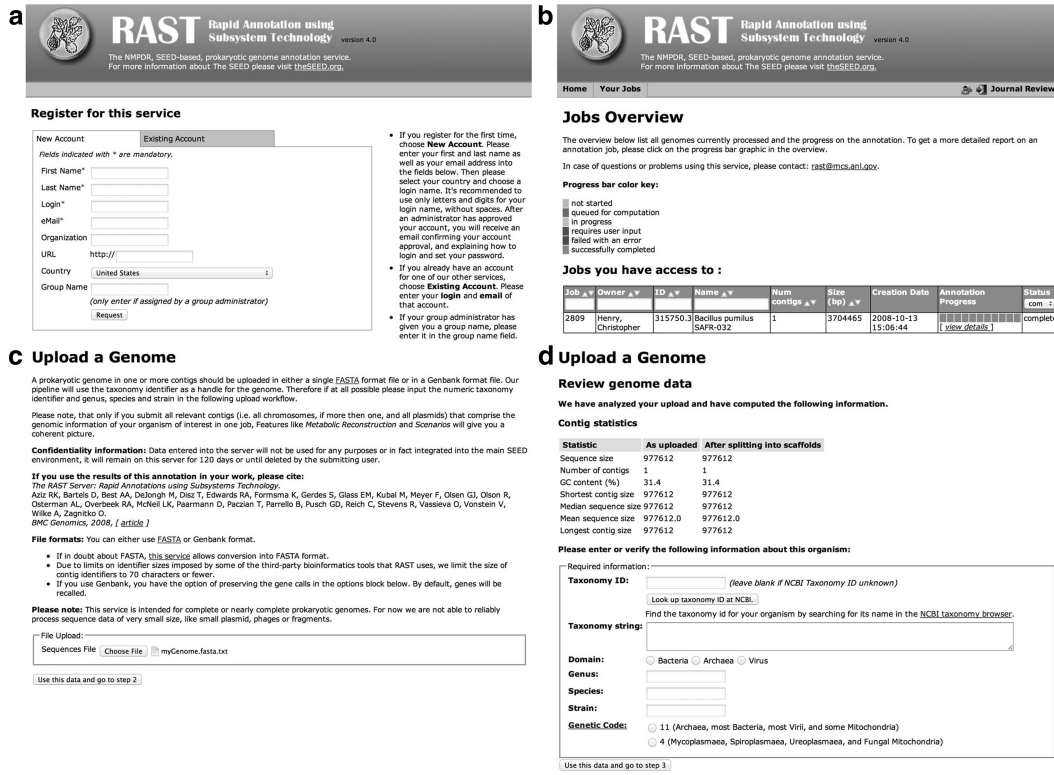

Fig. 1. Web interface for submitting genome sequences to RAST for automated annotation.

2. Once a user has registered for an account, the next step is to log in to RAST using the registered account credentials. Once logged in, the user will be taken to the "Job Overview" page (Fig. 1b), which summarizes all currently running and complete genome annotation jobs submitted by the user to RAST. To upload a new job, simply hover over the "Your Jobs" entry in the menu bar at the top of the page and select the "Upload new job" item.
3. Now you will arrive on the "Upload a genome" page of RAST (Fig. 1c). Simply click on the box labeled "Sequences file," select the FASTA file on your computer, and click the "Use this data and go to step 2" button.
4. If the upload of your FASTA file goes well, you will arrive on the "Review genome data" page (Fig. 1d). This page includes a preliminary analysis of your genome that includes the length of the chromosomes found in your FASTA file. This is useful to ensure the upload and parsing of your file went well. This page also requests an NCBI taxonomy ID (e.g., 83333), an NCBI taxonomy (e.g., Bacteria; . . .*Escherichia coli*), domain (e.g., Bacteria), genus (e.g., *Escherichia*), species (e.g., *coli*), strain (e.g., K-12), and genetic code (e.g., 11) for your genome. Simply fill in the data and click on the "Use this data and go to step 3" button.

5. This will take you to the final input screen, which requests optional information including the sequencing technique (e.g., nearly always pyrosequencing today), coverage (e.g., typically over 10×), the number of contiguous strings of DNA (e.g., the number of chromosomes for completely assembled genomes), and the average read length (e.g., 100–200 on modern pyrosequencing machines). This page also enables the user to select additional optional parameters to be used during the RAST annotation process (3). These include selecting the gene calling methodology (RAST or Glimmer3 (20, 21)), selecting the release of FIGfams (19) to be used (FIGfams are the protein families used as references against which all RAST annotations are made), and binary parameters for fixing errors, fixing frameshifts, building a metabolic model (see Subheading 3.4), backfilling gaps, and disabling replication. We recommend reviewing RAST documentation for details on these parameters (http://www.nmpdr.org/FIG/wiki/view.cgi/Main/RAST); otherwise, the default settings work well. Once the desired settings have been selected, simply click on the "Finish the upload" button to submit the genome for annotation to RAST. The job is scheduled and run on computer servers maintained at Argonne National Laboratory.
6. Users can check on the status of their genome annotation job at any time by returning to the "Job Overview" page on RAST, which will provide regular updates on job progress. RAST jobs are accessible only by the user and RAST administrators, unless a user wishes to make the annotated genome accessible by a wider audience. In that case, the user must explicitly grant access privileges to users as desired.

It is worth noting that RAST itself can run on relatively cheap equipment (under $10,000) and be used to annotate tens of genomes per day. The existing servers support much higher loads (occasionally reaching several hundred genomes per day). RAST scales, so it would be straightforward to support thousands of jobs per day. Over its lifetime thus far, over 50,000 jobs have been submitted to the RAST annotation service.

3.2. The RAST Approach to Automated Genome Annotation

While a user who has submitted a genome sequence to RAST for annotation does not need to know how the RAST annotation process works, it is still useful to know the details of the process to better understand the assumptions and caveats that go into a RAST annotation. When a newly sequenced prokaryotic genome is submitted to RAST for annotation, the job goes through roughly five steps:

1. First, there is a targeted search for a small, well-defined set of elements. Currently, these include rRNAs, tRNAs, genes relating to synthesis and use of selenocysteine, and genes relating to the synthesis and use of pyrrolysine. The set of elements sought will undoubtedly expand as new tools to recognize elements like microRNAs and CRISPRs will be added.
2. Then an iterative step to identify protein-encoding genes is initiated. The bulk of the effort normally is based on use of Glimmer (20, 21). The search usually involves an attempt to recognize common genes, use these as a training set, recall using the training set, and then attempt to remove lengthy overlaps and to fill unusually large gaps.
3. Once estimates of protein-encoding genes (PEGs) have been derived, an initial pass is made to assign functions to the subset of PEGs that can be reliably assigned functions based on the FIGfams/kmers. In genera with numerous existing well-annotated reference genomes, this step often assigns functions to over 90% of the PEGs. In diverse genomes, the percentage can be far less. A second pass is made using BLAST (22) to estimate similarities, and then these are used to assign functions. It is important to note that the reliable assignments in a *controlled vocabulary* are largely based on the first pass and that this second pass is thought of as assigning functions that are clues in an *uncontrolled vocabulary*.
4. PEGs with assigned functions in the *controlled vocabulary* can be gathered into subsystems, when the annotation tools identify *all* of the roles needed to form an active variant of this subsystem. It is precisely the cases in which most, but not all, of the needed roles are identified that require manual curation to clarify what needs to be done to achieve more accuracy and consistency. PEGs that were annotated with *uncontrolled vocabulary* or uncalled ORFs resulting from low-quality DNA sequence are usually responsible for that. For details on this process, see step 1 in Subheading 3.3.

This is an abbreviated description of the process RAST uses to annotate microbial genome sequences. It is important to note that the reference set of genomes that serve as the basis for most of the RAST annotations undergo continuous manual curation, ensuring that annotations remain up-to-date with the latest biological data, ensuring that annotations of genes and gene clusters are consistently propagated to new genomes, and supporting the construction and maintenance of the SEED subsystems and functional roles that form the foundation of the RAST annotation ontology. This manual curation is an essential ingredient contributing to the accuracy and consistency of RAST annotations. Although there is manual annotation of reference genome annotations continuously occurring in

RAST, it is still important for users to review and curate the annotations of their own genomes once the RAST automated annotation process is complete.

3.3. Reviewing and Curating a RAST Annotation

The majority of genomes submitted to RAST are "phylogenetically close" to already annotated genomes. Indeed, it is becoming common for users to submit hundreds of genomes from a single genus or even species. While close genomes do often contain quite different collections of genes, the subsystems identified in the reference genomes become good candidates for the new genomes. Thus, if we computed the "closest 30 genomes" based on rRNA comparisons or upon analysis of a collection of universal (or near universal) genes, then we could reasonably use the subsystems in these close genomes as collections of genes to be sought for in the new genome. A common error in the automated RAST annotation of a genome is the case in which all roles but one within a subsystem were recognized, but the missing role was not detected due to frameshift errors or truncations in the coding gene (i.e., to poor quality sequence data). Implementing a simple, effective way to spot these cases allows us to take advantage of the existing well-annotated close reference genomes. This is the reason why we suggest choosing the "Fix frameshifts" option when submitting the genome to RAST. Quickly identifying the subsystems present in a new genome will allow an understanding of what roles can be identified, and using these roles to characterize which complexes are present creates the needed bridge to the enzymatic ontology.

1. To compare the annotations in a new genome against those in a reference genome, RAST offers the ability to ask for a list of all subsystems present in both genomes or present in one but not the other. From the Job Overview page choose "view details" for the genome of interest. This leads to the Job Details page from which you can choose to "Browse annotated genome in SEED Viewer." The SEED Viewer is the browsing and annotation environment provided by RAST. Under "Comparative Tools," select to run the "Function based Comparison." You will be prompted to select a reference genome and run the comparison. The output is a table summarizing the similarities and differences for all genes, which were associated with subsystems. From this table, one can invoke search tools ("find" button) that will attempt to find functional role assignments that are present in the reference genome, but not in the newly annotated one. Learning how to gather and use this data is the first step towards correcting the initial RAST annotations.
2. Most manual refinements will come from comparison with reference genomes, detection of frameshifts and truncations

by walking the genome in the SEED Viewer environment, and searches for conjectured functions suggested by gap-filling algorithms. The RAST environment offers the registered user the ability to browse the genes, annotations, roles, and subsystems identified for the new genome. The capability exists to add new features, delete features, or to change the annotations associated with features. Changes made by the user can be used to recompute the subsystems ("recompute subsystems" button on the "Organism Overview" page). With experience, a user can learn to compare annotations between a set of genomes to locate potentially unidentified genes, to spot inconsistent annotations, and to locate genes unique to specific genomes (please refer to the "SEED Viewer Tutorial" under the Help menu).

3. Inevitably, there will be an ongoing need to extend the controlled vocabulary. This is achieved by adding subsystems containing the needed roles. Within the SEED Project, there is active encoding of new subsystems by experienced annotators. However, there is a growing need to make it possible for expert users to define and curate their own subsystems. To support this, we have made available within the PubSEED the possibility for users to create their own subsystems. These subsystems will get integrated into the annotation cycle. That is, they will lead to the creation of new FIGfams and then will directly impact future annotations produced by RAST.
4. Users may well have exceptional expertise but no desire to spend the effort needed to construct and maintain a subsystem. Such users can request a new subsystem be built by supplying a definition of the roles and one or more carefully annotated rows in the subsystem. To do so, choose "Request a Subsystem" under the "Navigate" menu. The request will be considered by the SEED annotation team, and if it does represent new functionality with experimental characterization, the new subsystem will get constructed and added to the collection.

Once the RAST genome annotation has been reviewed and curated, the annotated genome is ready to be submitted to the Model SEED for reconstruction of a draft genome-scale metabolic model.

3.4. Submitting a RAST Annotation for Model Construction in the Model SEED

There are two places where a RAST-annotated genome can be submitted to the Model SEED for automated reconstruction of a draft genome-scale metabolic model. First, the web interface for submitting a genome sequence for annotation in RAST includes an option to automatically submit the annotated genome for model reconstruction (see Fig. 2d and step 5 of Subheading 3.1). Here we

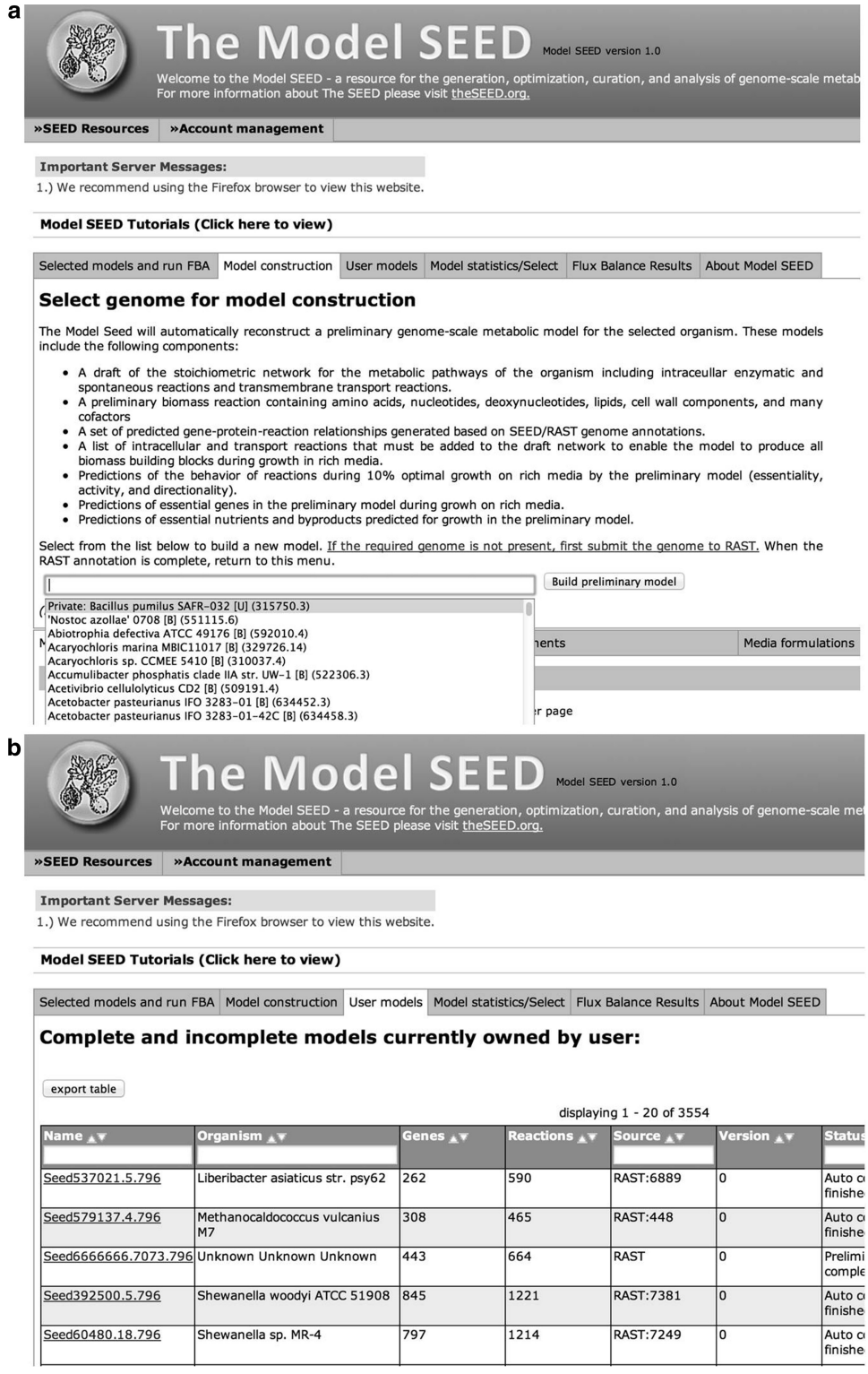

Fig. 2. Model reconstruction and user models.

describe the alternative approach of using menus available on the Model SEED website itself.

1. As with RAST, all users must first register a SEED user account before that can submit genomes for model reconstruction in Model SEED. Note that RAST, SEED, and Model SEED all share the same user registration system, meaning the username and password used to log in to RAST can also be used to log in to the Model SEED and the PubSEED. Registration is completely open and free of charge and simply provides a mechanism by which we can assign model ownership and ensure private access to private models. New users can register for accounts using the account registration webpage on Model SEED: http://www.theseed.org/seedviewer.cgi?page=Register (Fig. 2a).
2. Once an account has been registered, simply visit the Model SEED home page (http://www.theseed.org/models) and log in. Once logged in, click on the "Model Construction" tab in the upper frame of the Model SEED home page.
3. Once the "Model Construction" tab loads, a description of the model construction process will be displayed with a filter select for genomes beneath the description (Fig. 2a). Included in this filter select are all SEED genomes that are publically available for all SEED users as well as any private genome that the logged user has submitted to RAST for annotation. Note that all private genome entries in this filter select begin with the word "PRIVATE." Simply select the genome for which you want to build a model and click on the "Build preliminary model" button. Clicking this button will immediately take you to the "User models" tab of the Model SEED website, which after a moment will now display a table that includes your newly submitted model (Fig. 2b). Note that while you can select the model for viewing immediately, no data will be available until the model reconstruction process is complete.

Once a genome has been submitted for reconstruction in the Model SEED, it enters the automated model reconstruction pipeline of the Model SEED, which we describe in Subheadings 3.5–3.7. This process is entirely automated, so it is not necessary for the user to intervene at any point. This description exists to inform about what goes on under the hood of the Model SEED pipeline. The initial reconstruction of a core model (see Subheadings 3.5–3.6) requires approximately 5–10 min to complete, at which point the core model may be viewed in the Model SEED site (see Subheading 3.8). Initial reconstructions are automatically submitted for auto-completion (see Subheading 3.7), which can require up to an additional 24 h to complete.

3.5. Automated Metabolic Model Reconstruction in the Model SEED

While a user that has submitted an annotated genome for model reconstruction does not need to know how the Model SEED reconstruction works, it is still useful to know the details of the process to better understand the assumptions and caveats that go into the resulting model. To construct a preliminary model, four steps are taken:

1. A comprehensive database of biochemistry is constructed, representing all possible reactions and compounds that a model could use (see Note 1). This database consists of all of the reactions and compounds from the KEGG (23) and many published genome-scale metabolic models (24), combined into a nonredundant set. In total, this database contains over 13,000 reactions and over 16,000 reactants. All reactions that include generic reactants (e.g., alcohol) and all mass and charge imbalanced reactions are disallowed from inclusion in metabolic models (Fig. 3a).
2. All compounds in the database are adjusted to their predominant charged for at pH 7 (see Note 2), and all reactions are proton balanced using the charged forms of the reactants. Gibbs free energy change is then calculated at pH 7 for all reactions in the database using the group contribution method (6), and this data is used to predict reaction reversibility and directionality (7, 25).
3. A mapping is prepared between the SEED annotation ontology and the reactions in the biochemistry database through protein complexes as an intermediate (Fig. 3b). This mapping

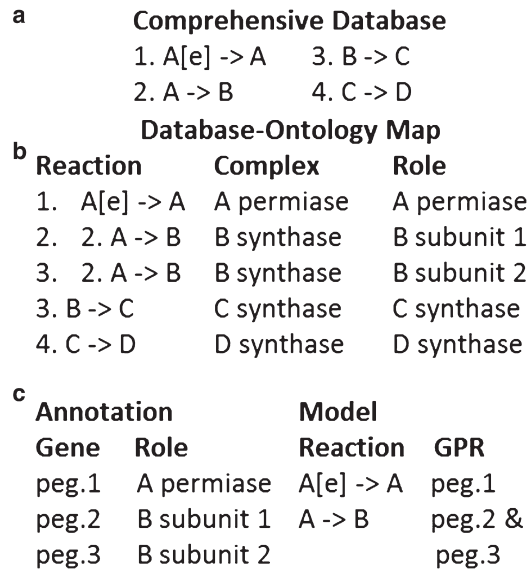

Fig. 3. Model reconstruction process.

is continuously maintained as the SEED annotation ontology and reaction database evolve over time.

4. The annotation ontology map is used to translate the gene annotations in the RAST-annotated genome into a list of metabolic reactions with associated gene–protein-reaction (GPR) rules that govern how gene activity impacts reaction activity (Fig. 3c). A list of spontaneous reactions (e.g., $CO_2 + H_2O \geq HCO_3$) is added to every core model constructed by the Model SEED, as these reactions occur regardless of the genes present in the genome. These reactions form the metabolic pathways of the core model.

This completes the reconstruction of the metabolic pathways and gene–protein-reaction associations for the metabolic model. The next step in the reconstruction process is to build a draft biomass composition reaction for the model. This step is described in detail in the next portion of this chapter (Subheading 3.6).

3.6. Construction of Draft Biomass Objective Function in the Model SEED

The biomass composition reaction (BCR) describes the relative quantity of all small molecule metabolites that must be produced in order to generate 1 g of biomass. BCRs account for proteins, lipids, DNA, RNA, cell walls, and cofactors. The small molecule building blocks of microbial biomass vary significantly depending on the metabolic pathways utilized, the electron transport chain, and the cell wall type. For this reason, BCRs are automatically assembled in the Model SEED based on the genome annotation and a set of template BCR, generated manually for four classes of cell wall: gram negative, gram positive, Mycoplasma, and Archaea. As with the core model reconstruction, it is not necessary for the user to intervene at any point during the BCR generation process in the Model SEED. We describe the process here just to improve understanding of how this process works. It is important to note that an exact BCR cannot be automatically generated from genome sequence alone, because the relative quantity of biomass components cannot be predicted exactly from genome sequence. Thus, the coefficients for metabolites in automatically generated BCRs are only approximations, and experiments must be performed to obtain exact values for these coefficients (see Note 3 for more information on the BCR coefficients).

1. The first step in the BCR generation process is to identify which BCR template to use, which requires that the input genome be classified as gram negative, gram positive, Mycoplasma, and Archaea. This classification can be done manually via experimental characterization or literature search; it can be done automatically via phylogenetic characterization of the organisms based on 16S RNA; or it can be done based on the genome annotations, as the various cell walls are associated with different

cell wall biosynthesis subsystems. The third approach is the method used by the Model SEED; a list of functional roles specific to each cell wall type has been assembled, and we predict cell wall type by assessing which list of roles has more associated genes in the input genome annotation.

2. Once the organism cell wall type has been determined, a template BCR is selected based this typing. Template BCRs typically contain approximately 100 candidate metabolites for inclusion in the model BCR. Each candidate metabolite is associated with a set of conditions that must be satisfied in order for the metabolite to be included in the model BCR. Half of the metabolites in the template BCRs are typically universal, meaning they will be included in the BCR of every model. These include all amino acids, all nucleotides, all deoxynucleotides, and many common cofactors (e.g., NAD). The remaining half of metabolites (e.g., lipids, cofactors, cell wall components) are included in the model BCR only if the genome annotation includes evidence for functional roles associated with the biosynthesis or utilization of the metabolites.
3. Once all the metabolites in the model BCR have been determined, the stoichiometric coefficients for the metabolites must be computed. The model BCR should represent the metabolites consumed to produce 1 g of biomass, so coefficients are computed such that the net mass of metabolites consumed in the BCR add up to 1 g. The mass of metabolites produced as products in the BCR (e.g., ADP, phosphate, and H^+ generated by the ATP hydrolysis) is subtracted from this net mass when adding up BCR metabolite mass (see Note 4). To support this computation, each candidate metabolite in the template BCR is associated with a category: DNA, RNA, protein, lipid, cell wall, cofactor, or energy. The template BCR includes estimations of the fraction of biomass associated of each category and the mole fraction of each small molecule to each category. These estimations are based on the values reported for the representative organisms on which these templates are based (*E. coli*, *B. subtilis*, *M. barkeri*, and *M. genitalium*). As such, it is important to note that these values are approximations for the organism being modeled, and they need to be adjusted based on experimental data before precise quantitative predictions can be produced by the model.
4. The template biomass reactions all include an "energy" category that contains ATP and water as reactants and ADP, phosphate, and H^+ as products. These metabolites represent the growth-associated ATP consumption for the organism, and the coefficient on all of these terms in the biomass reaction represents one of the most important

adjustable parameters for fitting growth–yield predictions to experimental data in metabolic models. Growth-associated ATP consumption varies widely among published metabolic models, with values typically falling between 30 and 100 mol/g cell dry weight hour (24). As with all other coefficients in the template BCR, the growth-associated ATP consumptions included in the template BCRs are approximations garnered from representative organisms. We make special mention of the growth-associated ATP consumption coefficient, because it is the largest BCR coefficient by far and as such has the greatest influence on yield computations. When building a model to compute growth yield, one of the first steps should be to adjust the growth-associated ATP consumption coefficient to fit experimentally measured growth yields.

5. DNA is the one category of BCR components for which stoichiometry can be calculated directly from the DNA sequence, as this portion of the biomass represents the replication of the chromosome itself (see Note 5). DNA coefficients are calculated by first computing the GC content of the chromosome. Then the molar fractions of the deoxynucleotides are set according to the GC content (e.g., deoxyguanine = deoxycytosine = GC and deoxyadenine and deoxythiamine = 1−GC).

At the end of this process, we will have a draft BCR, which is the final piece required to have a complete genome-scale metabolic model. The BCR is a critical element to the predictive capacity of a metabolic model, and both the coefficients and the small molecular metabolite content of the BCR require curation and adjustment. The coefficients in the BCR directly impact the ability of the model to quantitatively predict growth yield; the small molecule metabolites included in the BCR directly impact the ability of the model to qualitatively predict cell viability in a variety of environmental conditions and with a variety of genetic perturbations. Depending on the type of predictions needed, curation of the BCR should be prioritized accordingly. Although the draft metabolic model is now complete, it will almost never include all of the reactions needed to produce every component of the BCR, meaning the model will still be unable to predict growth conditions at this time.

3.7. Auto-completion of a Metabolic Model in the Model SEED

The core metabolic model and biomass objective function that are assembled by the Model SEED during the model reconstruction process contain all the data required to begin the analysis of microbial phenotypes and capabilities using approaches such as flux balance analysis (13–16). However, even when working with the most well known of microbial organisms (e.g., *E. coli*, *B. subtilis*), these core models will be of limited use initially, as they will be lacking many of

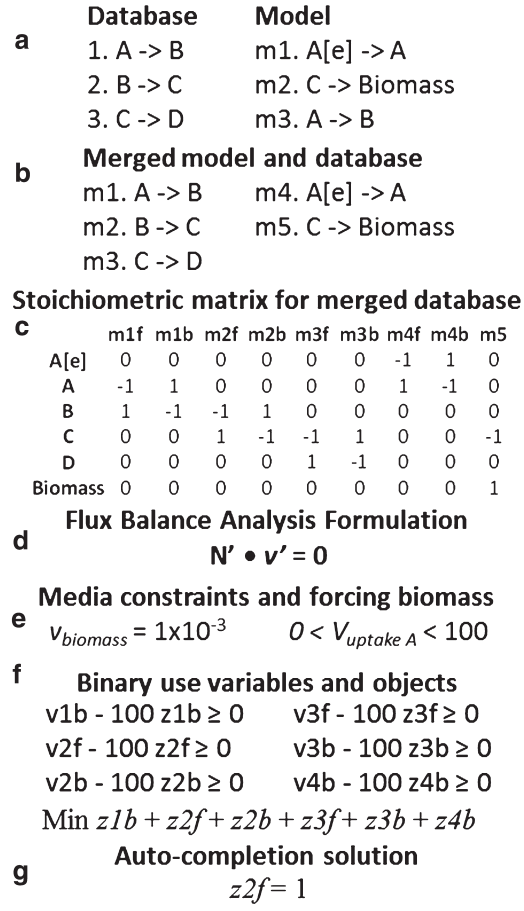

Fig. 4. Model auto-completion process.

the enzymatic steps required to produce all biomass building blocks including in the biomass objective function of the model. All core models generated by the Model SEED undergo an auto-completion process, whereby additional reactions are added to the model as needed to enable the production of all biomass components. The auto-completion process is automated by the Model SEED, but here we outline the steps of this process used by the Model SEED.

1. In the first step of the auto-completion process, a biochemistry database is prepared to serve as the source of reactions to be added to the core model to enable biomass production (Fig. 4a). Prior to use in auto-completion, all generic reactions, lumped reactions, and unbalanced reactions must be removed from the database, as these reactions can cause physiologically irrelevant pathways to be added by the auto-completion algorithm. In the Model SEED, the database used for auto-completion includes 10,516 reactions and 8,355 compounds.

2. Next, the auto-completion database is merged with the reactions and compounds of the core metabolic model while ensuring that identical compounds and reactions in the model and database are unified to produce a single nonredundant biochemical network (Fig. 4b). In the Model SEED, this process is simple as all reactions included in Model SEED models come from the Model SEED biochemistry database, meaning the models and database have a common namespace. When the model and biochemistry database namespaces are different, this merging process can be the most difficult step in the auto-completion process, due to inconsistencies in how compounds and reactions are named and represented.
3. Now the unified biochemical network and model BCR are translated into a stoichiometric matrix, where the columns are reactions, the rows are compounds, and the elements are the stoichiometric coefficients of the compounds in the reactions. As this matrix is formulated, every reaction is decomposed (regardless of reversibility) into separate forward and reverse component reactions (Fig. 4c), so that the flux through every component reaction is always greater than or equal to zero.
4. This matrix is then used to form the linear mass balance constraints of a flux balance analysis problem by setting the product of the stoichiometric matrix and the vector of fluxes through the forward and reverse component reactions to be equal to zero (Fig. 4d).
5. Next, an additional constraint is added to the linear optimization problem that forces the flux through the BCR to a positive nonzero value. Uptake and drain fluxes are also added for all metabolites that occur in the extracellular compartment. Bounds on these fluxes are adjusted as needed to represent the media conditions in which the auto-completion is being performed (Fig. 4e). Because the specific defined growth conditions for organisms are unknown, in the Model SEED auto-completion is performed in complete media, where uptake of all transportable metabolites is allowed (see Note 6).
6. To track which new reactions in the auto-completion database are to be used when flux is forced through the BCR, binary variables are associated with each component reaction that did not appear in the original model either because annotated reactions were irreversible or because no gene was annotated to perform the reaction. Each binary use variable is equal to "1" if its associated reaction is active and "0" otherwise. The objective function for the auto-completion optimization problem then becomes the minimization of the sum of the binary use

variables multiplied by a set of cost coefficients. Cost coefficients are computed for every component reaction based on thermodynamic feasibility, completion of existing pathways, the confidence in the biochemistry, and the amount of information available for the biochemistry (Fig. 4f). Costs are also commonly calculated based on blast scores for genes associated with the gap-filled reactions in other genomes (26).

7. Each solution to the auto-completion optimization problem (Fig. 4 g) represents a set of reactions that must be either added or made reversible in order to enable the metabolic model to produce biomass in the media condition selected for auto-completion (see Notes 7–8). The Model SEED will select a solution that best minimizes the auto-completion cost function, and that solution will be integrated into the core model as gap-filling reactions. However, it's important to note that many equivalent optimal solutions often exist for the auto-completion optimization, meaning the solution must be manually curated to determine if it is correct.
   Once the auto-completion process is complete, the metabolic model will be capable of producing biomass in the media condition in which the auto-completion was performed (typically complete media). At this stage, flux balance analysis may be used to generate qualitative predictions of essential genes, growth conditions, growth phenotypes, and metabolic capabilities. But this metabolic model is still a draft model, and substantial curation must be performed before the model is capable of generating accurate quantitative predictions. In the next section, the tools available for viewing and curating metabolic models will be explored, with an emphasis on the Model SEED website.

3.8. Reviewing and Curating a Model SEED Model

Once a model has been build and auto-completed in the Model SEED, the final step is to review the model and begin the process of curating the model. The Model SEED website (http://www.theseed.org/models) provides numerous interfaces for viewing metabolic model data and for comparing the model to other models in the Model SEED database. Below, we will walk through the Model Viewer interface of the Model SEED, the Cytoscape SEED interface for model viewing, and the methods for downloading model data for offline analysis.

1. The Model Viewer interface of the Model SEED is divided into two main frames: an upper frame and a lower frame. The upper frame contains tools for selecting and running analyses on models. It is split into six tabbed panes, and initially the tab labeled "Selected models and run FBA" is displayed. This tab contains a text box displaying the text "type here to see available models," which can be used to search for existing models

that are publically available. For example, if you type "k12" in this text box, you will see several publically available metabolic models for *E. coli K12*. To select your model for viewing, first make sure that you are logged into the Model SEED site. The login menu is located on the upper right-hand corner of the Model Viewer page. Next, type the name of the genome, genome ID, or name of the model you constructed in the model selection text box and find your model from among the options that appear in the filter select. Select your model and click the submit button. The Model Viewer page will reload with your model selected. see Note 9 for other methods of selecting models in the Model Viewer.

2. The upper frame will now contain some summary statistics for your model, including the number of genes, reactions, and compounds. The upper frame will also contain controls for running flux balance analysis on your model within the Model SEED environment. These will be discussed later. For now, we will review the bottom frame, where all other data related to your model will be displayed. The bottom frame is a tabbed display containing the following tabs: Map, Reactions, Compounds, Biomass Reactions, Genes, and Media formulations. Initially, the "Map" tab is selected, which enables selection of KEGG pathway maps (23) from a table showing the names of the pathway maps and the number of reactions, compounds, and EC numbers that occur in your model in each map. You can search from the list of available maps for viewing using the text box immediately below the "Name" header in the table. see Note 10 for general information about Model SEED tables. Once you've identified a metabolic map of interest, simply click on the map link. The metabolic map should immediately begin loading in the area beneath the map table. Note that you can open multiple maps at once by clicking on different map links in the map table. Maps can also be opened by clicking on map links in the metabolic maps themselves, in the reaction table, and in the compound table.

3. Scroll to the bottom of the page after a selected map has loaded. All of the reactions that are present in your model are highlighted. Hover the mouse over any reaction in the pathway map to see associated information, such as the reaction ID, the corresponding KEGG ID, and the associated gene ID. The KEGG pathway maps will only highlight reactions from the metabolic model that correspond to the KEGG reactions in the map. Metabolic models sometimes contain reactions that do not have corresponding KEGG reaction IDs; these reactions will not be displayed in the KEGG pathway maps. This may cause apparent gaps in the pathway maps (e.g., the pyruvate dehydrogenase complex in the iJR904 glycolysis/gluconeogenesis map).

To determine whether the gap is apparent or real, select the "Reactions" tab in the bottom frame of the Model SEED webpage and type the name of a compound adjacent to the gap (e.g., "pyruvate") in the text box below the "Equation" column header and press the "enter" key. The table will show all reactions containing that compound as a substrate or product (e.g., rxn00154 which represents a condensed version of the reaction catalyzed by the pyruvate dehydrogenase complex). The map view is the first location where model curation can begin. It's useful to examine the maps associated with the central metabolic pathways. Look for pathways where many reactions are present and only single steps are missing. These pathways are prime candidates for additional gap filling. When filling gaps in this manner, look at the gene page in the SEED for genes associated with reactions around the gap. These regions of the chromosome often contain the genes that may be associated with the reaction gap. Also look for reactions that are isolated from the rest of the model in the metabolic maps. These "island" reactions are prime candidates for removal.

4. Now select the "Reactions" tab on the lower frame of the Model Viewer. Once the tab finishes loading, you should see a table of all reactions currently included in your model. This table includes Model SEED reaction IDs (which you can click on to load a separate reaction page with images of all reactants), reaction names, reaction equations (you can click on compound names to load a separate compound page showing structure and listing all reactions the compound takes part in), functional roles and subsystems mapped to the reaction in the SEED ontology (these mappings were used to generate the core model as described in Subheading 3.5), KEGG maps (click on these links to load the associated map in the "Maps" tab), EC numbers, KEGG IDs, notes for the reaction in your model, and a list of the genes mapped to your reaction in your model (full GPR rules for reactions are shown on the reaction pages). This view is the best location to systematically see all the reactions that were added to the model during the auto-completion process. Simply go to text box in the model column of the reaction table, type "Gapfilling," and press enter. The table will now list all reactions that were gap filled in the model. Now critically examine each gap-filled reaction. If the reaction appears to be a correct addition to the model, go to the KEGG maps associated with the gap-filled reaction and repeat the curation steps described in step 3. If it appears that the gap-filled reaction is wrong, consider why that gap-filled reaction was added. For example, if a folate transporter was incorrectly added, this still indicates that the folate biosynthesis pathways

in the model are incomplete. This provides a hint as to which correct gap-filling reactions must be added.

5. Now select the "Compounds" tab on the lower frame of the Model Viewer. This tab contains a table of all compounds currently included in your model. This table includes Model SEED compound IDs (which you can click on to load the compound page), compound names, molecular formula, molecular weight, molecular charge, KEGG maps (click on these links to load the associated map in the "Maps" tab), KEGG IDs, and an indication of the compartments where each compound appears in the model. Note that all molecular properties displayed for compounds in this table were computed at pH 7.
6. Now select the "Biomass Components" tab on the lower frame of the Model Viewer. This tab contains a table of all compounds currently included in the biomass reaction of your model. This table includes the compound ID, names, formula, mass, charge, KEGG map, KEGG ID, and coefficient for each compound in the biomass reaction. The entries in the biomass component table should be reviewed, with special attention paid to the cofactors, lipids, and cell wall components included in the BCR. Compare the BCR of the draft model with BCR from other models of phylogenetically close organisms to identify if important components were excluded or if components were included that should not be there.
7. Now select the "Genes" tab on the lower frame of the Model Viewer. This tab contains a table of all genome features that appear in the annotated genome for which the model was reconstructed. This table includes gene IDs (click on these to go to the gene page in SEED), start, length, direction of transcription, functional annotation, predicted and experimental essentiality, and a list of the reactions mapped to each gene in your model. This is the best place to view the entire genome annotation that was used to assemble the model. Explore the annotation, check the gene calls, and look for large blocks of unannotated genes that might be indicative of a problem. These issues can then be resolved using the annotation curation tools available in RAST and PubSEED (see Subheading 3.3).
8. Finally, select the "Media formulations" tab on the lower frame of the Model Viewer. This tab contains a table of all media formulations currently loaded into the Model SEED database. The table includes media IDs and a list of media compounds in terms of names and compound IDs. This table is useful for quickly identifying the media conditions on which you want to simulate for model using flux balance analysis.

9. To run flux balance analysis on your model, return to the "Selected models and run FBA" tab of the upper frame of the Model Viewer website. Below the model information in this tab is a header entitled "Click here to run FBA on selected models" – click on this header to reveal a text box where you can select from a set a predefined media formulations. The default media condition for FBA is "Complete," meaning that all compounds for which the metabolic model has transport reactions are present in the medium. Click the "Run" button, and the top frame will switch to the "Flux Balance Results" tab. After FBA has completed, the upper frame will display a table containing the FBA results. If the model predicts growth on the selected medium, the "Growth" column will display the growth rate. Select the corresponding checkbox in the "Select" column and click on "View Selected Results." The "Reactions" tab in the bottom frame is updated to display the flux for each reaction in the rightmost column.
10. Another mechanism for viewing model content is the Cytoscape SEED plugin for Cytoscape. The CytoSEED viewer (27) for Model SEED models provides a more flexible environment for viewing metabolic models. CytoSEED is a plugin for the Cytoscape biological network viewer (28), and instructions for installing and using CytoSEED are available at http://www.cs.hope.edu/cytoseed/.
11. One of the most powerful mechanisms for model curation is the comparison of your model with the model of another more well studied by phylogenetically close organism. The Model SEED site facilitates such comparison by enabling the user to select multiple models at once. To compare models, simply use the model filter select in the "Selected models and run FBA" tab of the upper frame of the Model Viewer website to select another model. Once the desired model is selected, once again click on the "Select model" button. The Model Viewer site will now reload, but this time, both your model and the new model will be selected at the same time. You can use this mechanism to select up to five models at once for side-by-side comparison. All views described above will then contain data for all selected models.
12. Model SEED models may also be downloaded to enable their use with other available flux balance analysis platforms. In the "Selected models and run FBA" tab of the upper frame of the Model Viewer website, the model information table includes a series of download link in the far right-hand column. Three formats are available for model download: LP format, SBML format, and Excel format. The LP format defines the constraints, variables, and objective function for the linear optimization problem defined by running flux balance analysis on your model. LP files can be manipulated and simulated using command line

interfaces for most linear optimization solvers (e.g., CPLEX, GLPK, SCIP). The SBML format is a standard format for metabolic models. These files can be loaded into FBA software platforms such as the COBRA Toolbox (29) or OptFlux (30) for simulation and analysis. These FBA software packages feature numerous studies that apply the model to predict phenotypes and behavior. They also include algorithms to refine a model based on experimental data (8). The Excel format download includes three worksheets containing the reaction, compound, and gene data for the model. This format mirrors much of the data displayed on the Model Viewer website.

At this stage, the draft Model SEED model can be applied to predicting phenotypes that may be compared with phenotypic data available in the literature. Whole-genome transposon mutagenesis, gene knockout studies, and Biolog phenotyping arrays are all valuable tools for model testing and validation. Powerful computational techniques such as GrowMatch (8) now exist for reconciling metabolic models with experimental phenotype data. In addition to testing the capacity of the model to correctly predict microbial viability, the model can also be applied to predict growth yields, which can be compared to experimentally observed growth curves. Prior to attempting to predict growth yields, first ensure that the ATP biosynthesis mechanism being employed by the model is physiologically reasonable. Often the auto-completion process will produce models that generate ATP in physiologically unreasonable ways. Correct auto-completion of electron transport chains in metabolic models remains an open problem and an active area of work. Finally, take advantage of the comparative tools in the Model SEED that compare the draft models with other more well-curated models. Comparison with published models of similar organisms is the fastest way to identify and correct errors that occur in the annotation and automated model reconstruction process.

A detailed protocol (10) does exist for manually creating new genome-scale metabolic models. In this chapter, we have focused on the tools and steps taken to automatically produce a draft metabolic model, but this reconstruction protocol remains one of the best resources available for any researcher contemplating the development of a new metabolic model. We recommend reading this protocol in detail, comparing the protocol with the content of this book chapter, and using the protocol as a guide to the process of curating and revising your draft metabolic model.

4. Notes

1. The biochemistry database used by the Model SEED to construct metabolic models (see Subheading 3.5) is available for browsing at http://seed-viewer.theseed.org/models. If no model is selected, the "Reactions" and "Compounds" tabs will contain all reactions and compounds available in the database (13,000 and 16,000 entities, respectively). Where possible, we have also preserved connections to the KEGG pathway maps, which may be viewed under the "Maps" tab. Additional information about each reaction is available through the link on the reaction id, e.g., "rxn00123." This page includes thermodynamic reversibility, Gibbs free energy estimates, and a table of alternate names under the "Database Links" tab. A similar set of information is available for compounds under the compound ID, e.g., "cpd00456."
2. There is free software available, called MarvinBeans (http://www.chemaxon.com/products/calculator-plugins/), for predicting the predominant charged form of compounds at a specified pH. The software requires only a MOL file with the compound molecular structure as input. The command to determine the predominant charge at pH 7 is "cxcalc -N hi majorms -H 7 -f mol:-a example.mol > chargedExample.mol".
3. We mention that the specific metabolite coefficients in the BCR are only approximations garnered from representative organisms on which the template BCRs are based. The same is true for the specific metabolites included in the BCR as well, although we do strive to be as comprehensive as possible with that list. We also emphasize the need to refine the BCR metabolite coefficients based on experimental data before precise quantitative flux predictions can be obtained. Here, we note the substantially greater importance of obtaining correct values for the growth-related ATP consumption over obtaining correct values for specific biomass composition of proteins, RNA, cofactors, lipids, and cell walls. This is important to emphasize, because calculating growth-related ATP consumption requires only simple growth curves measured in a variety of growth conditions. Full biomass compositions can be much more challenging to obtain exact values and have less value in models.
4. Why are there products besides biomass in my Model SEED BCR? Often products other than biomass are added to the BCR of metabolic models. In some cases, this is done for modeling reasons. For example, two of the most common products in BCR are ADP and phosphate, which are present because they

are the products of hydrolysis of ATP and water. Most BCR include the reactants and products of ATP hydrolysis to represent the energy consumed by the synthesis of biomass from small molecule metabolites. For example, DNA, RNA, and protein polymerization reactions all hydrolyze ATP into ADP to provide energy for the hydrolysis process. In other cases, products are added to biomass to recycle metabolites that are consumed during the synthesis of biomass metabolites, but for which no biosynthesis pathways exist. Examples are the protein components attached to CoA and ACP (the protein components are attached in the metabolic pathways, so we can capture the essentiality of the enzymes that perform this ligation step) or the molecule dimethylbenzimidazole, for which the biosynthesis pathway is unknown.

5. Students learning about metabolic modeling and biomass objective functions often ask why amino acid and RNA nucleotide BCR coefficients cannot be calculated from the direct translation of the DNA sequence as is done for DNA deoxynucleotide coefficients. This view would be correct if all genes and proteins were expressed by the cell in equal quantities at all times. Unfortunately, this is not the case, and in fact, it is so far from reality as to be an extremely poor assumption to make.
6. In the model auto-completion process, choosing a minimal media for the auto-completion is preferred as it reduces the available solution space during the auto-completion optimization. This will result in more specific gap-filling predictions, and it will produce a model that is more likely to grow in many of the observed growth conditions.
7. The model auto-completion process involves solving a large mixed-integer linear optimization problem with potentially tens of thousands of binary variables. The most common open-source optimization package, GLPK, is not capable of solving this problem in a reasonable amount of time. Alternative MILP optimization packages must be used. Open-source alternatives include SCIP (http://scip.zib.de/), CBC (http://www.coin-or.org/projects/Cbc.xml), and SYMPHONY (http://www.coin-or.org/projects/SYMPHONY.xml). Commercial software includes CPLEX (www.ibm.com/software/integration/optimization/cplex-optimizer/). In our experience, the CPLEX solver outperforms all others by a significant margin, although all the open-source solvers we list above will solve most auto-completion problems in less than 24 h.
8. When solving the mixed-integer optimization problem during the auto-completion process, extreme caution must be taken

with the enforcement of the binary use variables. Although the problem constraints state that the flux through a component reaction must be zero if its corresponding use variable is zero, all optimization solvers have a tolerance setting that dictates the maximum amount by which a constraint may be violated. This tolerance can provide numerical "wiggle room" that allows reactions to carry a small amount of flux while keeping the binary use variable at zero. This small amount of flux can be sufficient to satisfy the minimal flux through the biomass reaction, while in fact, the current solution is not feasible. To avoid this problem, we recommend performing auto-completion with very low tolerance settings (e.g., 1×10^{-9}) and with large bounds on reaction flux and metabolite uptake (e.g., 10,000).

9. Some of the other tabs in the upper frame of the Model Viewer website provide alternative mechanisms for loading model summary statistics and selecting models for detailed viewing. The "User models" tab contains a table of all private models currently owned by the logged in user. This tab is a useful mechanism for quickly checking the status and availability of all your private models in the Model SEED. The "Model Statistics/select" tab includes a table of all models in the Model SEED that you currently have access to (including public models and private models). This tab is useful for quickly viewing, comparing, and querying metadata for all Model SEED models.
10. In general, all tables in the Model SEED website have controls for searching and sorting based on columns. When a text box is present immediately below a column header, type your search text in the text box and press the "enter" key; the table contents will be filtered to display only the entries that match your search text. The table can be restored by deleting the search text from the text box and pressing the "enter" key. Additionally, some table headers have two arrows immediately to their right; click the "up" arrow to sort in ascending order and the "down" arrow to sort in descending order.

Acknowledgements

We acknowledge the entire SEED, Model SEED, and CytoSEED teams at Argonne National Laboratory, Fellowship for Interpretation of Genomes, Hope College, and University of Chicago for efforts on the frameworks described in this chapter. This work was supported by the US Department of Energy under contract DE-

ACO2-06CH11357 (SD, CH), the National Institute of Allergy and Infectious Diseases under contract HHSN266200400042C (RO), and the National Science Foundation under grants MCB-0745100 and DBI-0850546 (MD, AB, VV, RO).

References

1. Feist AM, Palsson BO (2008) The growing scope of applications of genome-scale metabolic reconstructions using *Escherichia coli*. Nat Biotechnol 26:659–667
2. Henry CS, DeJongh M, Best AA, Frybarger PM, Linsay B, Stevens RL (2010) High-throughput generation, optimization, and analysis of genome-scale metabolic models. Nat Biotechnol 1672:1–6
3. Aziz RK, Bartels D, Best AA, DeJongh M, Disz T, Edwards RA, Formsma K, Gerdes S, Glass EM, Kubal M, Meyer F, Olsen GJ, Olson R, Osterman AL, Overbeek RA, McNeil LK, Paarmann D, Paczian T, Parrello B, Pusch GD, Reich C, Stevens R, Vassieva O, Vonstein V, Wilke A, Zagnitko O (2008) The RAST Server: rapid annotations using subsystems technology. BMC Genomics 9:75
4. Overbeek R, Disz T, Stevens R (2004) The SEED: a peer-to-peer environment for genome annotation. Commun ACM 47:46–51
5. DeJongh M, Formsma K, Boillot P, Gould J, Rycenga M, Best A (2007) Toward the automated generation of genome-scale metabolic networks in the SEED. BMC Bioinformatics 8:139
6. Jankowski MD, Henry CS, Broadbelt LJ, Hatzimanikatis V (2008) Group contribution method for thermodynamic analysis of complex metabolic networks. Biophys J 95:1487–1499
7. Henry CS, Zinner J, Cohoon M, Stevens R (2009) *i*Bsu1103: a new genome scale metabolic model of *B. subtilis* based on SEED annotations. Genome Biol 10:R69
8. Kumar VS, Maranas CD (2009) GrowMatch: an automated method for reconciling in silico/in vivo growth predictions. PLoS Comput Biol 5:e1000308
9. Suthers PF, Dasika MS, Kumar VS, Denisov G, Glass JI, Maranas CD (2009) A genome-scale metabolic reconstruction of *Mycoplasma genitalium*, *i*PS189. PLoS Comput Biol 5: e1000285
10. Thiele I, Palsson B (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. Nat Protoc 5:93–121
11. Schuler GD, Epstein JA, Ohkawa H, Kans JA (1996) Entrez: molecular biology database and retrieval system. Methods Enzymol 266:141–162
12. Edwards JS, Palsson BO (2000) The *Escherichia coli* MG1655 in silico metabolic genotype: its definition, characteristics, and capabilities. Proc Natl Acad Sci U S A 97:5528–5533
13. Papoutsakis ET, Meyer CL (1985) Equations and calculations of product yields and preferred pathways for butanediol and mixed-acid fermentations. Biotechnol Bioeng 27:50–66
14. Jin YS, Jeffries TW (2004) Stoichiometric network constraints on xylose metabolism by recombinant *Saccharomyces cerevisiae*. Metab Eng 6:229–238
15. Varma A, Palsson BO (1994) Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type *Escherichia coli* W3110. Appl Environ Microbiol 60:3724–3731
16. Varma A, Palsson BO (1993) Metabolic capabilities of *Escherichia coli*. 2. Optimal-growth patterns. J Theor Biol 165:503–522
17. Varma A, Palsson BO (1993) Metabolic capabilities of *Escherichia coli*.1. Synthesis of biosynthetic precursors and cofactors. J Theor Biol 165:477–502
18. Edwards JS, Ibarra RU, Palsson BO (2001) In silico predictions of *Escherichia coli* metabolic capabilities are consistent with experimental data. Nat Biotechnol 19:125–130
19. Meyer F, Overbeek R, Rodriguez A (2009) FIGfams: yet another set of protein families. Nucleic Acids Res 37:6643–6654
20. Delcher AL, Harmon D, Kasif S, White O, Salzberg SL (1999) Improved microbial gene identification with GLIMMER. Nucleic Acids Res 27:4636–4641
21. Delcher AL, Bratke KA, Powers EC, Salzberg SL (2007) Identifying bacterial genes and endosymbiont DNA with Glimmer. Bioinformatics 23:673–679
22. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25:3389–3402

23. Kanehisa M, Goto S (2000) KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res 28:27–30
24. Feist AM, Herrgard MJ, Thiele I, Reed JL, Palsson BO (2009) Reconstruction of biochemical networks in microorganisms. Nat Rev Microbiol 7:129–143
25. Kummel A, Panke S, Heinemann M (2006) Systematic assignment of thermodynamic constraints in metabolic network models. BMC Bioinformatics 7:512
26. Krumholz EW, Yang H, Weisenhorn P, Henry CS, Libourel IG (2012) Genome-wide metabolic network reconstruction of the picoalga Ostreococcus. J Exp Bot 63:2353–2362
27. DeJongh M, Bockstege B, Frybarger P, Hazekamp N, Kammeraad J, McGeehan T (2012) CytoSEED: a Cytoscape plugin for viewing, manipulating and analyzing metabolic models created by the Model SEED. Bioinformatics 28:891–892
28. Smoot ME, Ono K, Ruscheinski J, Wang PL, Ideker T (2011) T. Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 27:431–432
29. Becker SA, Feist AM, Mo ML, Hannum G, Palsson BO, Herrgard MJ (2007) Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox. Nat Protoc 2:727–738
30. Rocha I, Maia P, Evangelista P, Vilaca P, Soares S, Pinto JP, Nielsen J, Patil KR, Ferreira EC, Rocha M (2010) OptFlux: an open-source software platform for in silico metabolic engineering. BMC Syst Biol 4:45

Chapter 3

Metabolic Model Refinement Using Phenotypic Microarray Data

Pratish Gawand, Laurence Yang, William R. Cluett, and Radhakrishnan Mahadevan

Abstract

Phenotypic microarray (PM) is a standardized, high-throughput technology for profiling phenotypes of microorganisms, which allows for characterization on around 2,000 different media conditions. The data generated using PM can be incorporated into genome-scale metabolic models to improve their predictive capability. In addition, a comparison of phenotypic profiles of wild-type and gene knockout mutants can give essential information about gene functions of unknown genes. In this chapter, we present a protocol to refine preconstructed metabolic models using the PM data. Both manual refinement and algorithmic approaches for integrating the PM data into metabolic models have been discussed.

Key words: Phenotypic microarrays, Biolog^{TM}, Metabolic models, Model refinement, Bilevel optimization, Systems metabolic engineering

1. Introduction

Current high-throughput omics technologies generate large datasets that can be integrated into cellular models to improve their predictive capability (1). Data generated from different high-throughput techniques such as transcriptomics, proteomics, genomics, and metabolomics has been previously used to expand metabolic model constructions (2–4). Data from phenomics, a complementary high-throughput technique for characterizing microbial phenotypes, can provide a wealth of information on the physiology of the organism that can be incorporated into metabolic models (3–5). Phenomics involves characterization of cellular fitness (phenotype) in response to genetic or environmental perturbations (1). Elaborate phenotypic characterization of microorganisms has found applications in studying gene function, taxonomic classification, improving industrial bioprocess, and model improvement in systems biology (6).

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_3,

Phenotype microarrays (PMs), developed by Bochner et al. (7) and commercialized by Biolog Inc. (http://www.biolog.com/), is a standardized high-throughput phenotyping platform in 96-well plate format. The platform allows growth assay of bacterial cells under 1,920 different environmental conditions that are preformulated in 20 plates (PM1-20). Different environmental conditions include different carbon sources (plates PM1 and PM2), different nitrogen sources (PM3, PM6-8), different phosphorus and sulfur sources (PM4), different osmolarity conditions (PM9), different pH values (PM10), and different toxic chemicals at different concentrations (PM11-20) (6). Phenotype microarrays use tetrazolium dye chemistry to monitor respiration instead of directly measuring growth. Alternatively, turbidity may be measured (using the Biolog Turbidimeter) to directly quantify cell density, which can be used to estimate growth rate.

For the experimental protocol of PMs, the reader is referred to previous studies (7, 8). Briefly, bacterial cells are first grown on a universal growth medium such as R2A agar. The cultivated cells are used to form cell suspensions of known cell density (85% transmittance). These cell suspensions are diluted, mixed with tetrazolium dye, and are inoculated (typically 100–150 μL volume) into 96-well plates containing preformulated, dried nutrients. The 96-well plates are then incubated into a plate reader, and the colorimetric signal intensity is recorded (generally every 15 min).

The data thus obtained from PM analysis can be used for a variety of applications including refining of metabolic models. This chapter elaborates a protocol for refining metabolic models using PM data.

2. Materials

A desktop computer with the following software: MATLAB (MathWorks®), COBRA Toolbox v2.0 (can be downloaded from http://www.cobratoolbox.org), OmniLog PM software (Biolog Inc.), a MATLAB compatible LP and MILP solver such as CPLEX (IBM Inc., Armonk, NY, USA), and MATLAB compatible NLP solver such as CONOPT (ARKI Consulting & Development, A/S).

3. Methods

The process of incorporating PM data for model improvement can be divided into four major steps as shown in Fig. 1.

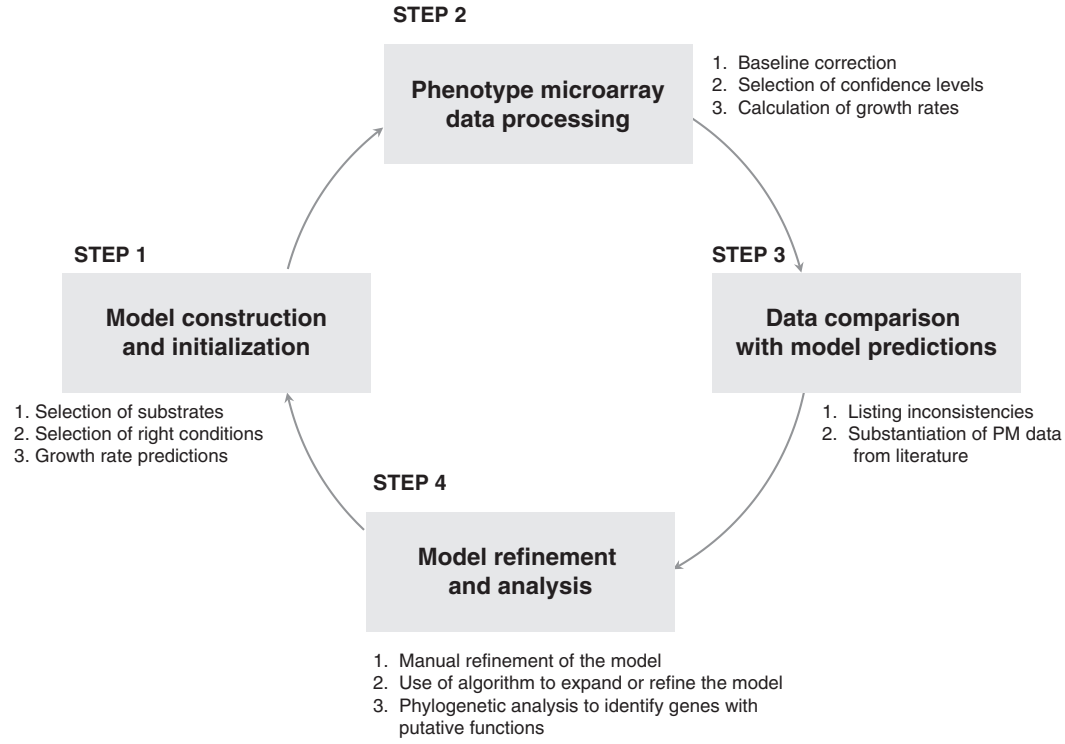

Fig. 1. Important steps involved in model enhancement using PM data.

The following text describes important steps that would typically be used to improve model predictions using PM data.

3.1. Model Initialization

1. The metabolic model intended to be refined should be present in an SBML file, which can be read in MATLAB environment using COBRA Toolbox (COBRA Toolbox v2.0 is recommended) (9). COBRA Toolbox contains commonly used scripts for analysis, manipulation, and visualization of metabolic models.
2. Set free exchange for CO_2, H_2O, and metal ions (including H^+) by setting the lower and upper exchange flux limits to -1×10^3 mmol/g dcw/h and 1×10^3 mmol/g dcw/h, respectively.
3. Choose appropriate conditions, either aerobic or anaerobic, by setting O_2 lower exchange flux limits to -20 mmol/g dcw/h and 0 mmol/g dcw/h, respectively (see Note 1).
4. Set lower limits of all the exchange reactions for all substrates to 0 mmol/g dcw/h, except the substrate under consideration. Set the upper limit of all the substrates to 1×10^3 mmol/g cell/h to allow free secretion.

5. Iteratively calculate growth rates on all the different substrates that are included in the model. To calculate the growth rate on a test substrate, use the exchange flux of 5 mmol/g dcw/h (see Note 2).
6. Mark no growth if the growth rate prediction is less than or equal to 10^{-8} h^{-1}.
7. If the PM analysis has been carried out to compare the wild type with mutants, the growth rates for the gene knockout mutants should be calculated by the setting both the upper and lower bounds of the reaction(s) corresponding to the deleted gene to 0 mmol/g dcw/h (see Note 3).

3.2. Processing of PM Data

1. OmniLog PM software is used to visualize the kinetic data from the PM experiments. Alternatively, web-based PMViewer software can be used for viewing and analyzing PM data (8). Using the parametric module in the OmniLog software, organisms can be compared for the basic differences such as lag time and growth rate (see Note 4).
2. For baseline correction and assignment of a confidence level to the data, find the standard deviation (SD) in the negative control wells. A well absorbance reading greater than 1.2–2 times the SD of the negative wells can be considered as a physiological response.
3. Alternatively, make a baseline correction to all the readings by subtracting the average of negative control well readings.
4. Confidence level can be assigned to the test well measurements to ensure right interpretation of the data. Assignment of confidence is especially important if the absorbance data is not collected continuously (every 15 min) but is obtained at intermittent time points (every 24 h). Based on the consistency of the absorbance signals of a well, the measurement can be high confidence, medium confidence, or low confidence. For example, if 10 readings are available for a well and if 9–10 readings show positive response, the data is high-confidence data; if only 5–8 out of 10 readings show positive response, the data is medium-confidence data; and if only 1–5 wells show positive readings, the data is low-confidence data (see Note 5).
5. Similar confidence levels can be assigned to kinetic plots, if biological replicates have been used for the experiment.
6. *Normalization.* Normalization of PM data is important when two different strains of an organism, or a wild type and a mutant, are being compared to each other. Normalization of data accounts for the inherent differences in the growth rates of the compared organisms. For normalization of an individual well signal, divide the signal with the average signal across the 96-well plate. For normalization of the kinetic data, repeat the above step for each time point (10) (see Note 6).

7. *Calculation of growth rate.* Growth rate can be calculated from the PM data if 750 or 600 nm wavelength was used for monitoring the optical densities of the cells. Measurements at 570 nm can be used as growth analogue, to measure relative growth rates (see Note 7). To find the growth rate from PM data, fit the exponential growth phase to an exponential equation ($Ae^{-\mu t}$, where μ is the growth rate). Alternatively, the PM data can also be fitted to a logistic equation (11) (see Note 8).

3.3. Data Comparison

1. *Direct qualitative comparison of in silico and PM data.* High-confidence data obtained from PM assays can be directly compared to the *in silico* predictions obtained in Subheading 3.1. Lower- and medium-confidence data needs further substantiation from the literature resources to infer the physiology with confidence. Check the agreement between the *in silico* predicted growth phenotype (growth or no growth) to the PM results. Note all the inconsistencies between the *in silico* predictions and the PM data and classify them in the following two categories:

   NG/G—In silico no growth/*in vivo* growth.

   G/NG—In silico growth/*in vivo* no growth.

   Infer the inconsistencies based on the decision chart provided in Fig. 2.

2. *Data comparison for gene knockout mutants.* PM data for mutants can be compared to either wild-type PM data or *in silico* growth predictions for the gene deletion mutant. Comparison of wild-type and mutant PM data can give useful information about the function of the deleted gene. Such data can be used for gene annotation, which can then be used for model expansion. Similarly, inconsistencies arising from the comparison between the mutants' PM data and *in silico* predictions can be used for direct model refinement. Identify the inconsistencies between observed and predicted growth phenotype for the mutant and classify the inconsistencies as NG/G and G/NG. Interpret the results based on the decision chart provided in Fig. 3.

3. *Comparison of growth rates with the model predicted growth rates.* Comparison between the predicted and experimental growth rate can be used for model refinement mainly using algorithmic methods. The data comparison can be used for obtaining the right objective function for the model (ObjFind) (12), finding enzyme capacity constraints (OCCI) (13), and adding or removing reactions to the model (GrowMatch) (14). These algorithmic methods have been explained further in this chapter.

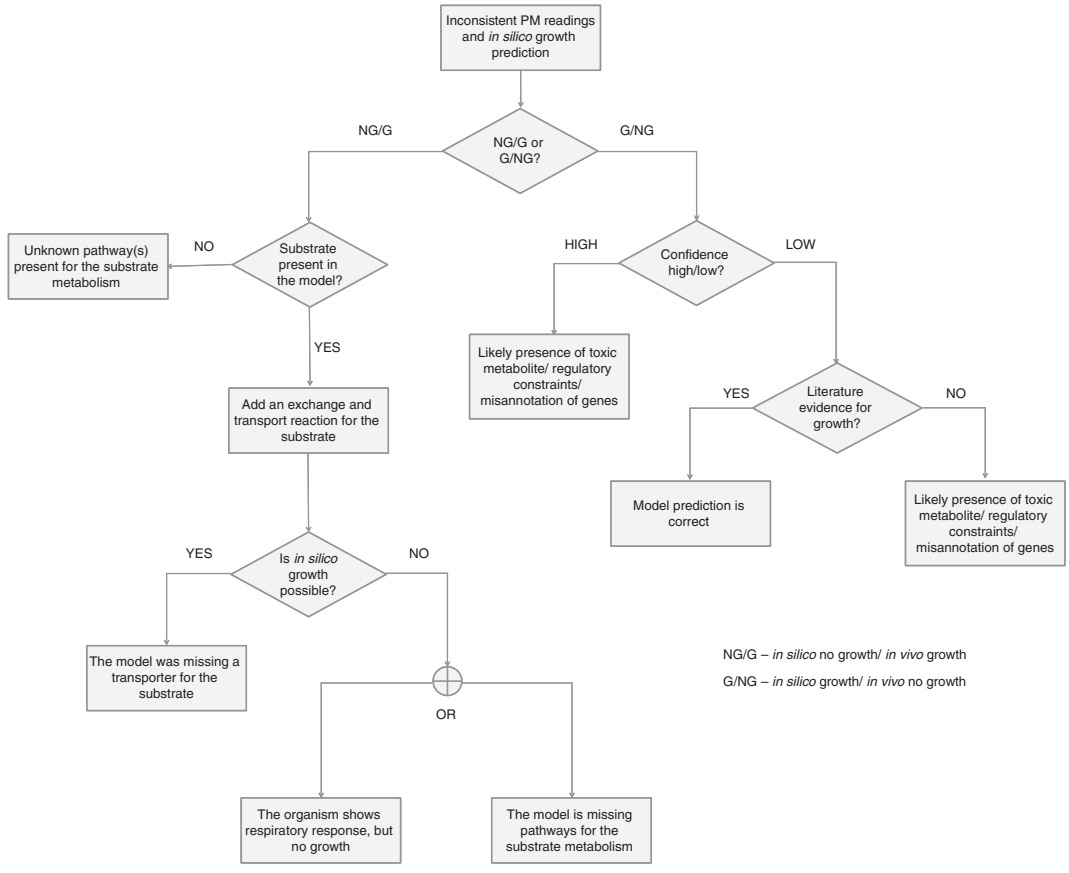

Fig. 2. Decision tree for inferring inconsistencies between *in silico* model predictions and PM data.

3.4. Model Refinement

3.4.1. Manual Model Expansion

1. *Addition of a transporter*. In cases where NG/G inconsistencies are due to a missing transporter for a particular substrate, addition of an exchange and a transport reaction to the model can reconcile the *in silico* and *in vivo* data. To establish a genetic evidence for existence of such a transporter, putative genes should be identified using similarity searches (BLAST) and from transporter databases such as TCDB (http://www.tcdb.org/) and TransportDB (http://www.membranetransport.org/) (see Note 9).
2. *Addition of a pathway*. Additional pathways may be needed in the following cases.

   Presence of diverged isozymes that have been incorrectly annotated or not annotated is anticipated: To identify the diverged isozymes, carry out sequence similarity searches using BLAST with relaxed stringency. In case evidence for such an isozyme is established, add the appropriate reaction (s) and update the corresponding gene protein relationships

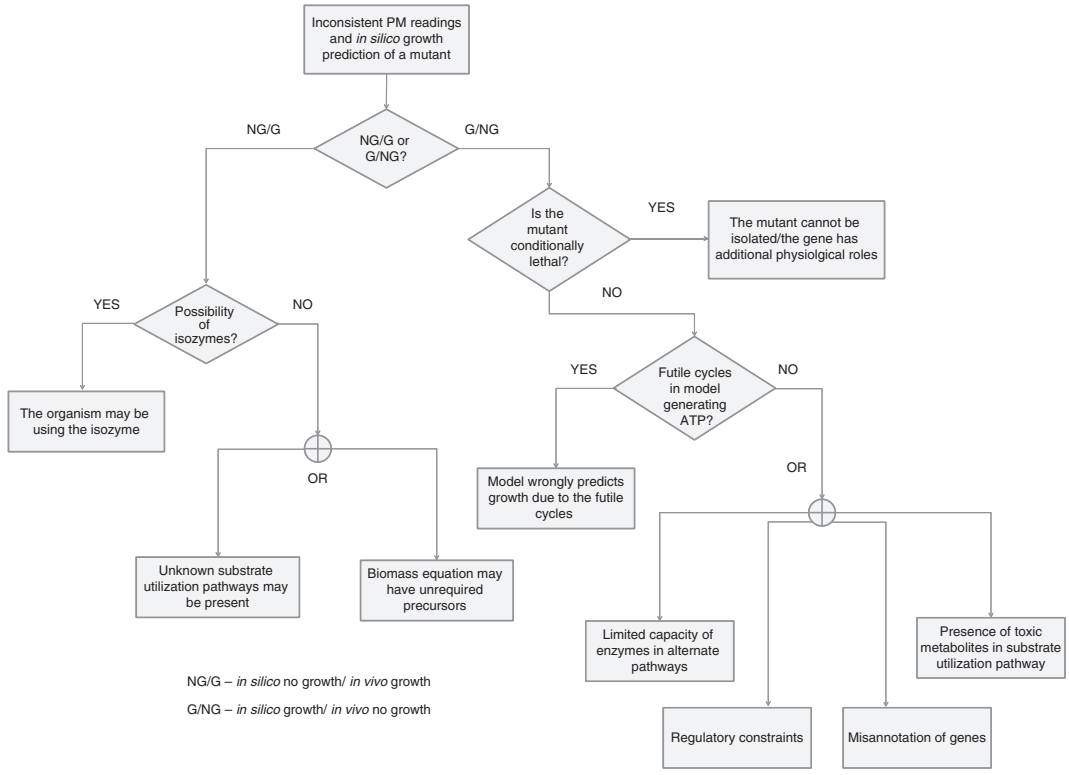

Fig. 3. Decision tree for inferring inconsistencies between *in silico* model predictions and PM data for gene knockout mutant.

(GPR). Ensure that the reaction added is charge and mass balanced (see Note 10).

Presence of an alternative pathway for a metabolite existing in the model: A starting point for addition of such pathways is to identify the network gaps, if any (COBRA Toolbox v2.0 can be used for gap analysis). Based on the gaps present in the network, predict the reactions that would be required to enable growth. After identifying the required reactions, carry out sequence similarity search to identify the putative genes for the predicted reactions. Update the model by adding the putative pathway after mass and charge balancing.

Presence of a pathway for a metabolite not present in the model: If PM shows growth on a substrate that is not present in the model, there is a possibility of existence of unknown transporters and pathways for catabolizing the substrate. The first step to identify these pathways is to identify the reactions that other organisms may use to metabolize same or similar substrate. Identify all the reactions that the substrate can undergo using databases such as KEGG (http://www.genome.jp/kegg/) and MetaCyc (http://metacyc.org/). Search (using sequence similarity) for the putative enzymes

responsible for these reactions in the genome of the target organism. The putative pathways and transporters can be added to the model after ensuring the charge and mass balance (see Note 11).

3. *Removal of a reaction*. Removal of a reaction from the metabolic model is required if a gene is misannotated. Remove the pathway from the metabolic model by deleting the corresponding reaction from the stoichiometric matrix and removing the corresponding bounds, objective coefficient, and GPR associations.
4. *Updating the biomass equation*. Biomass equation is based on the experimental determination of cell components and may, in some cases, contain components that are not essential for growth under all conditions. Presence of components that are not required during all growth conditions may be identified from the NG/G type inconsistencies. For example, in *Bacillus subtilis* some lipids and cell wall components are not required under normal growth conditions; however, the biomass composition defined *in silico* needs these components to be synthesized (4). Using the PM data, update the biomass equation by removing the nonessential components such that *in silico* predictions conform to the experimental observations.
5. *Addition of the regulatory constraints*. Some proteins may not be available under all environmental conditions due to regulatory repression. Though most metabolic models do not account for regulatory constraints, first approximation of regulation can be achieved by blocking the repressed reactions under certain conditions. To reconcile the *in silico* and PM and data, block the appropriate reactions by setting the upper and lower bounds to 0 mmol/g dcw/h.

3.4.2. Algorithmic Refinement of Constraint-Based Models Using PM

1. Computational algorithms are available for accelerating model refinement (14–16). Here, we describe the steps required to use such algorithms. We will consider two alternative formulations: a mixed-integer linear program (MILP) and a mathematical program with complementarity constraints (MPCC) (see Notes 12 and 13). As with manual model refinement, we use the notation G/NG to denote the case of *in silico* (predicted) growth when no growth is observed and NG/G to denote the case of no growth *in silico* but growth is observed.
2. Construct a reaction database that includes reactions not present in the original, organism-specific model (see Note 10). This expanded database is typically referred to as the universal database (14, 15).

3. Convert the reaction database into a constraint-based model, namely, by expanding the original model's stoichiometric matrix, reaction names, compound names, GPR associations, subsystem definitions, flux bounds, objective function, and the mass balance constraint vector ($S.v = b$). In addition, the expanded model must now include labels for each reaction that indicates whether the reaction was originally present in the current model version or not.
4. *MILP for resolving G/NG inconsistencies.* Formulate a bilevel optimization problem where the inner problem maximizes growth rate, while the outer problem forces growth rate to zero by determining (using the integer variables) which reactions to delete from the model or which reversible reactions should be revised to be irreversible. To correct reversibility of a reaction, the reversible fluxes will first need to be split into forward and reverse fluxes. see ref. 14 for the actual formulation (see Notes 14–17).
5. *MILP for resolving NG/G inconsistencies.* Formulate a single-level MILP using the expanded model, where a minimal number of external reactions are added to the model in order to meet a user-defined minimum growth rate.

   Here, a binary variable is assigned to each external reaction that is in the universal database but not in the original model. The binary variables are equal to 1 if the external reaction is added to the model or equal to 0 if it is not added. The minimum number of external reactions is found by minimizing the sum of these binary variables.
6. For both G/NG and NG/G, identify alternate optimal solutions by iteratively adding integer cuts and resolving the MILP.
7. Formulate a mathematical program with complementarity constraints (MPCC) to identify reactions to remove.
8. *MPCC for resolving G/NG inconsistencies.* Combine all or part of the tested environments or genetic backgrounds into a single model by stacking the flux bounds (reflecting the available substrate and/or genetic background) and the stoichiometric matrix. Only the original model (not the expanded model) is used here.
9. Formulate a bilevel optimization problem in which the inner problem maximizes growth rate, while the outer problem forces growth rate to zero by quantitatively adjusting flux bounds (see OCCI algorithm described in ref. 13 for the formulation). The optimization problem must also include constraints ensuring that the same revised flux bounds are applied across all conditions tested so that a single model that accurately describes all conditions emerges.
10. Reformulate the bilevel optimization problem into a single-level MPCC and use nonlinear programming (NLP) solvers

to obtain a solution (e.g., BARON, Ipopt, CONOPT) (see Notes 18 and 19).

11. *Assess the optimal solution*. If only a lower or upper bound is fixed to zero, while the other bound is nonzero, then reaction reversibility should be revised. If both the lower and upper bounds are fixed to zero, then the reaction should be deleted.

4. Notes

1. There can be additional growth requirements under anaerobic conditions. For example, anaerobic growth of *Saccharomyces cerevisiae* requires ergosterol, which is not required under aerobic growth conditions. If exchange of such additional compounds is not allowed, the model may predict zero growth under anaerobic conditions.
2. Substrate utilization rate may affect the model predictions of growth. For example, model predictions for growth of *B. subtilis* on glycine depends on the exchange flux. At 1 mmol/g dcw/h, the model predicts no growth; however, at 5 mmol/g dcw/h, the model predicts positive growth (4).
3. Make sure that all the reactions corresponding to the deleted gene are deleted in the model. Refer to the gene protein relationship (GPR) matrix to find out all the reactions corresponding to the gene of interest.
4. Two wavelengths are generally used to measure the response of the organism: 570 and 750 nm. Absorbance at 570 nm measures the color change of tetrazolium dye in response to the respiration of the organism, whereas absorbance at 750 nm indicates a change in the cell density. Another option to measure cell density is to skip the addition of the dye and measure absorbance at the standard 600 nm.
5. Obtaining high confidence levels in N and S plates can be sometimes difficult due to the high background of the negative wells.
6. Caution should be exercised in interpreting the normalized data as sometimes normalization can give very dissimilar patterns compared to the raw data, without any biological underpinning (11).
7. Note that 570 nm measures the respiratory response of the cells and cells can respire without growing. For example, *B. subtilis* shows respiratory response in presence of acetate but cannot use it as a carbon source (4).

8. The growth rates calculated from PM data may not agree with the experimental growth rates found using more elaborate cultivation techniques such as flasks or fermenters. This is due to the difference in the culture conditions such as shaking and settling of cells in the wells.
9. There is a possibility of diffusive transport of a substrate without a transporter in cell. There could also be preannotated nonspecific transporter/facilitator that allows the substrate uptake.
10. While various online databases exist, the task of ensuring that the reactions are balanced in terms of mass and charge; that the reactions, compounds, and subsystem names are consistent; that GPR relations are properly defined (important when tracing revised reactions back to genetic evidence); that compartments are consistent between external and the original reactions; etc., is often laborious and nontrivial. Therefore, the reader may wish to obtain access to databases that are already curated for use with constraint-based models of metabolism (17).
11. Note that promiscuous enzymes can act on multiple substrates and the enzymes involved in metabolizing the substrate may be preannotated promiscuous.
12. Alternative hypotheses will often explain a given phenotype measurement. The algorithms described here are capable of generating alternate hypotheses in an algorithmic manner (i.e., integer cuts for MILP, multiple starting solutions or more advanced global optimization methods for NLP).
13. The algorithms can be used with larger datasets where knockout mutants are grown under different environments (16, 18, 19). The same principle applies when using automated curation methods, where the fluxes corresponding to deleted reactions are constrained to zero.
14. Only the original metabolic model (not the expanded model) is required here since reactions will not be added. To solve the bilevel MILP problem, reformulation to a single-level MILP is typically performed (14).
15. More often than not, best results are obtained from GrowMatch and related MILP-based algorithms after tuning the MILP solver settings. For CPLEX, we use 1e-8 or smaller for integer tolerance. The reason is that the integer variables are being multiplied by a large number (e.g., 1,000); therefore, multiplying even 1e-8 by 1,000 would yield 1e-5. Thus, the solver would tolerate a flux of 1e-5 when in fact we intended to force the flux closer to zero using the integer variable.

16. A G/NG inconsistency may be due to inactivity of a reaction due to transcriptional regulation. This hypothesis can be rigorously tested by including models of transcriptional regulation, as reported in Barua et al. (16).
17. While commercial MILP solvers can typically guarantee global optimality of the solution (i.e., no better solution exists), in some cases, the solver may simply take too long to practically find a globally optimal solution. Therefore, a time limit is specified on the MILP solver (e.g., 1 week). The resulting solution is not guaranteed to be globally optimal, but useful solutions are often returned.
18. Certain global or local NLP solvers can be accessed through the NEOS web server (20).
19. If a nonconvex optimization problem like OCCI is used, the solution must be assessed for global optimality. Since a rigorous guarantee of global optimality is not always accessible, starting the solver from many (e.g., 100 for a model having <100 reactions) starting solutions may suffice. Also consider validating results using global optimization solvers like BARON (21), which is accessible online through the NEOS server (20).

References

1. Joyce AR, Palsson BØ (2006) The model organism as a system: integrating 'omics' data sets. Nat Rev Mol Cell Biol 7:198–210
2. Covert MW, Knight EM, Reed JL et al (2004) Integrating high-throughput and computational data elucidates bacterial networks. Nature 429:92–96
3. Feist AM, Henry CS, Reed JL et al (2007) A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol Syst Biol 3:121
4. Oh Y, Palsson BØ, Park SM et al (2007) Genome-scale reconstruction of metabolic network in *Bacillus subtilis* based on high-throughput phenotyping and gene essentiality data. J Biol Chem 282:28791–28799
5. Oberhardt MA, Puchałka J, Fryer KE et al (2008) Genome-scale metabolic network analysis of the opportunistic pathogen *Pseudomonas aeruginosa* PAO1. J Bacteriol 190:2790–2803
6. Bochner BR (2009) Global phenotypic characterization of bacteria. FEMS Microbiol Rev 33:191–205
7. Bochner BR, Gadzinski P, Panomitros E (2001) Phenotype microarrays for high-throughput phenotypic testing and assay of gene function. Genome Res 11:1246–1255
8. Borglin S, Joyner D, Jacobsen J et al (2009) Overcoming the anaerobic hurdle in phenotypic microarrays: generation and visualization of growth curve data for *Desulfovibrio vulgaris* Hildenborough. J Microbiol Methods 76:159–168
9. Schellenberger J et al (2011) Quantitative prediction of cellular metabolism with constraint-based models: the COBRA toolbox v2.0. Nat Protoc 6:1290–1307
10. Weber KP, Grove JA, Gehder M et al (2007) Data transformations in the analysis of community-level substrate utilization data from microplates. J Microbiol Methods 69:461–469
11. Sturino J et al (2010) Statistical methods for comparative phenomics using high-throughput phenotype microarrays. Int J Biostat 6:29
12. Burgard AP, Maranas CD (2003) Optimization-based framework for inferring and testing hypothesized metabolic objective functions. Biotechnol Bioeng 82:670–677
13. Yang L, Mahadevan R, Cluett WR (2008) A bilevel optimization algorithm to identify

enzymatic capacity constraints in metabolic networks. Comput Chem Eng 32:2072–2085

14. Kumar VS, Maranas CD (2009) GrowMatch: an automated method for reconciling in silico/in vivo growth predictions. PLoS Comput Biol 5:e1000308
15. Reed JL, Patel TR, Chen KH et al (2006) Systems approach to refining genome annotation. Proc Natl Acad Sci U S A 103:17480–17484
16. Barua D, Kim J, Reed JL (2010) An automated phenotype-driven approach (GeneForce) for refining metabolic and regulatory models. PLoS Comput Biol 6:e1000970
17. Kumar A, Suthers PF, Maranas CD (2012) MetRxn: a knowledgebase of metabolites and reactions spanning metabolic models and databases. BMC Bioinformatics 13:6
18. Ito M, Baba T, Mori H et al (2005) Functional analysis of 1440 *Escherichia coli* genes using the combination of knock-out library and phenotype microarrays. Metab Eng 7:318–327
19. Tohsato Y, Mori H (2008) Phenotype profiling of single gene deletion mutants of *E. coli* using Biolog technology. Genome Inform International Conference on Genome Informatics 21:42–52
20. Czyzyk J, Mesnier MP, More JJ (1998) NEOS server. IEEE Comput Sci Eng 5:68–75
21. Sahinidis N (1996) BARON: a general purpose global optimization software package. J Global Optimiz 8:201–205

Chapter 4

Linking Genome-Scale Metabolic Modeling and Genome Annotation

Edik M. Blais, Arvind K. Chavali, and Jason A. Papin

Abstract

Genome-scale metabolic network reconstructions, assembled from annotated genomes, serve as a platform for integrating data from heterogeneous sources and generating hypotheses for further experimental validation. Implementing constraint-based modeling techniques such as flux balance analysis (FBA) on network reconstructions allows for interrogating metabolism at a systems level, which aids in identifying and rectifying gaps in knowledge. With genome sequences for various organisms from prokaryotes to eukaryotes becoming increasingly available, a significant bottleneck lies in the structural and functional annotation of these sequences. Using topologically based and biologically inspired metabolic network refinement, we can better characterize enzymatic functions present in an organism and link annotation of these functions to candidate transcripts; both steps can be experimentally validated.

Key words: Metabolic network, Gap filling, Orphan reactions, Flux balance analysis

1. Introduction

Of the 2,000+ genomes that have been sequenced, around 40% of the protein products that have been identified have no described function (1). Over 5,000 enzymatic functions have been described across all species, but more than a third have no known corresponding genes or proteins (2, 3). Bridging these gaps of knowledge between gene and function is important to fully utilize data available in the postgenomic era. Here, we describe computational methods available in genome-scale metabolic modeling that aim to provide hypotheses that fill in these knowledge gaps as well as methods to experimentally validate these computational predictions.

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_4,

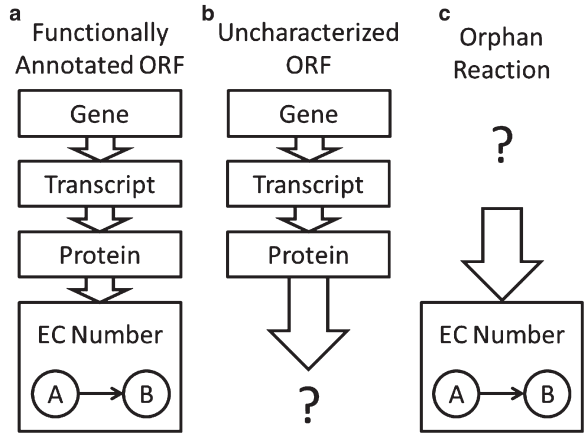

Fig. 1. Description of functional annotation of metabolic enzymes and current deficiencies. (**a**) Functionally annotated open reading frame (ORF) which represents a gene, a transcript, a protein, and its enzymatic function (represented by an EC number). (**b**) Structurally annotated ORF that has no characterized functional annotation. (**c**) Evidence exists for an enzymatic activity in an organism, but the orphan reaction has no assigned gene, transcript, or protein.

1.1. Genome Annotation

There are two major forms of genome annotation: functional and structural annotation. Functional annotation involves an understanding of the biological functions inherent in the genome, while structural annotation includes identifying coding regions of DNA that encode for a protein product, known as open reading frames (ORFs). Several computational techniques are available to structurally annotate ORFs directly for a newly sequenced organism (4, 5) (see Note 1), but functional annotation can be more challenging to assign as the sequence of an ORF alone does not necessarily describe its biological function (6). In addition, only around 1% of protein sequences have experimentally derived annotations (7); thus, computational techniques are necessary for feasibly assigning functional annotations.

Protein structure and function is often well conserved between organisms, so we can infer function between homologous genes or proteins across organisms (8). This is useful for identifying potential functions of uncharacterized ORFs (see Fig. 1), such as those in a newly sequenced genome. However, this approach can be limited because there may be multiple homologous sequences when comparing a query sequence against full sequences of all other organisms (termed the metagenome), and in reality only one of the many homologous sequences may actually demonstrate the appropriate enzymatic activity. The error rate for function assigned by sequence similarity may be as high as 49% (9); thus, a more guided approach is necessary for assigning function to ORFs.

1.2. Genome-Scale Metabolic Network Reconstruction

Genome-scale metabolic reconstructions serve as a platform for integrating data from heterogeneous sources and generating hypotheses for further experimental validation. Metabolic networks are constructed from existing genome annotations and manually expanded from literature-based sources and biochemical information contained in publicly available databases (10). Resulting models contain a comprehensive set of known biochemical reactions and their associated ORFs. Implementing a systems-level approach allows for the identification of potential gaps in knowledge based on discrepancies between model predictions and experimental data (e.g., gene essentiality screens) as well as topological features of the network (e.g., pathways resulting in dead ends). With the assistance of semiautomated algorithms and manual inspection, we can fill in these knowledge gaps by modifying the network to include additional biochemical reactions that were previously missing or by removing functions that were improperly added by previous annotators. By finding ORFs encoding for enzymes orthologous to those that catalyze the same functions in other organisms, we can improve both the structural and functional annotation of the genome for an organism of interest while also creating a higher-quality metabolic model.

In this chapter, we describe methods to

1. Predict missing and misannotated biochemical reactions for a given organism using metabolic network reconstructions. These include biologically inspired refinements, which bridge the gap between model predictions and experimental data, as well as topologically based algorithms that find and fill blocked pathways in a given network. These methods help improve the functional annotation of the genome for the organism of interest as well as improve the predictive ability of the metabolic model.
2. Assign candidate ORFs to novel functions as well as to existing functions that lack ORFs (orphan reactions; see Fig. 1). These relationships provide the link between functional and structural annotation, which are both important to a higher-quality annotation and metabolic model.
3. Use a systems approach to decide on which network modifications to include (and further validate) when posed with multiple gap-filling solutions.
4. Perform experiments to verify existence of candidate ORFs. This will help strengthen our confidence in both the structural and functional annotation of the genome in the organism of interest.

2. Resources

2.1. Genomic Information, Bioinformatics Tools, and Biochemical Databases

2.1.1. Genome Sequence for an Organism of Interest

To improve the genome annotation of an organism using a metabolic network approach, we need a whole genome DNA sequence. This information is important to identifying ORFs that may catalyze newly added reactions.

Availability: National Center for Biotechnology Information (NCBI) GenBank (11) (http://www.ncbi.nlm.nih.gov/genbank/)

Other bioinformatics resources outside the scope of metabolic modeling are available through the NCBI at: http://www.ncbi.nlm.nih.gov/guide/.

2.1.2. Information on Enzyme Commission (EC) Classifications

The EC classification system is used to define enzymatic activities that can occur within different organisms, and an EC number characterizes in part the functional annotation for an enzyme (and correspondingly for the catalyzed reaction(s)). ECs are classified according to the following hierarchical scheme: EC-1 (oxidoreductases), EC-2 (transferases), EC-3 (hydrolases), EC-4 (lyases), EC-5 (isomerases), and EC-6 (ligases). There can be several subclasses under these six categories. For example, the enzyme hexokinase, which is associated with an EC number of 2.7.1.1, belongs to the class on "transferases" (enzymes that aid in the transfer of a functional moiety from one metabolite to another) and the subclass on "transferring phosphorous-containing groups." Other enzymes in the same subclass include glucokinase (2.7.1.2) and galactokinase (2.7.1.6). The database ENZYME (part of ExPASy's suite) (2.1.4) contains information on EC numbers.

Availability: http://enzyme.expasy.org/

2.1.3. BLAST (Basic Local Alignment Search Tool)

BLAST computes the sequence similarity between sequences of amino acids or nucleic acids (12). This bioinformatics tool allows for quantitative, high-throughput comparisons in sequences between organisms in order to identify homologous ORFs that may share functional annotations (8, 13).

Availability: http://blast.ncbi.nlm.nih.gov

2.1.4. Biochemical Databases

Reconstructing the metabolism of a particular organism involves integrating biochemical information from various publicly available databases and experimental literature sources. Below, we provide a list of some important publicly available biochemical databases that contain information on genome, enzymes, reactions, and/or

pathways. The list below is not intended to be comprehensive; rather, it provides a flavor for the kinds of publicly available resources that can be used in the genome-scale metabolic reconstruction and modeling process.

KEGG (Kyoto Encyclopedia of Genes and Genomes) database contains comprehensive data on known enzymatic reactions that occur across various organisms (14–16).

Availability: http://www.genome.jp/kegg

ExPASy (Expert Protein Analysis System) contains comprehensive information on EC numbers and protein structure (17, 18).

Availability: http://expasy.org/

SEED allows for quickly generating automated draft metabolic networks for prokaryotic organisms of interest (19).

Availability: http://www.theseed.org

MetaCyc contains comprehensive information on pathways and enzymes across many organisms (20, 21).

Availability: http://metacyc.org/

GeneDB is a pathogen genome database maintained by the Wellcome Trust Sanger Institute (22).

Availability: http://www.genedb.org/Homepage

MetRxn allows queries of a comprehensive metabolite/reaction database and comparisons of metabolites/reactions between KEGG, MetaCyc, several metabolic reconstructions, and more (23).

Availability: http://metrxn.che.psu.edu/

UniProt is a comprehensive knowledge base of annotated protein sequences across many organisms (24).

Availability: http://www.uniprot.org/

MetaBase is a wiki database of biological databases (25).

Availability: http://metadatabase.org

2.2. High-Throughput Experimental Data

Substrate utilization assays: experimental observations of cellular growth under different substrate conditions

Availability: Experimental literature

Gene essentiality assays: experimental observations of cellular growth when a gene is knocked out or knocked down (e.g., the Keio collection of mutants for *Escherichia coli* (26, 27))

Availability: DEG (Database of Essential Genes) http://tubic.tju.edu.cn/deg/ (28, 29), OGEE (Online GEne Essentiality database) http://ogeedb.embl.de/ (30), and experimental literature

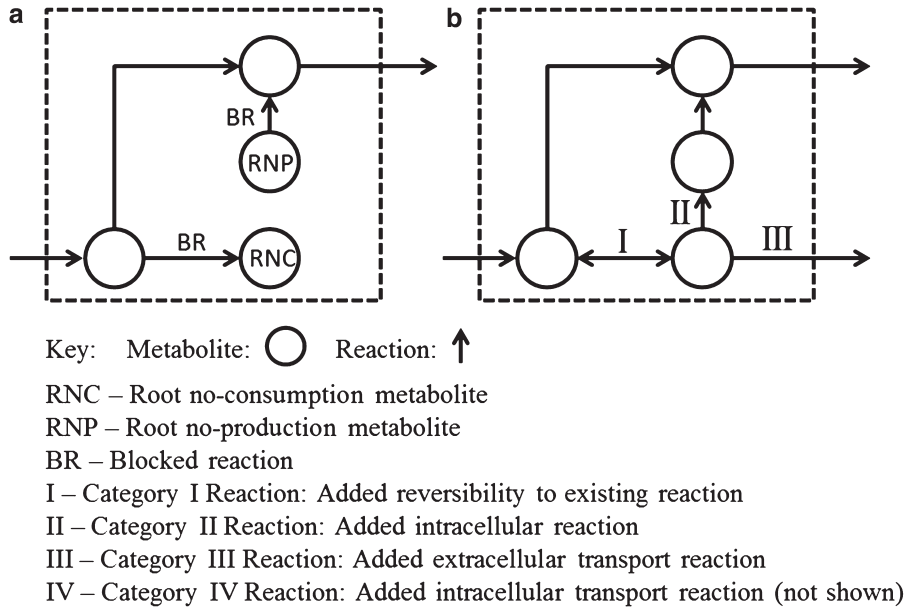

Fig. 2. Toy network depicting two blocked reactions (BR) caused by root no-consumption (RNC) and root no-production (RNP) metabolites. Categories of reactions that may be added to the network (suggestions aimed at restoring flux through the RNC metabolite): (category I) added reversibility to an existing reaction, (category II) added intracellular reaction, (category III) added extracellular transport reaction, and (category IV, not shown) added intracellular transport reaction. Note that in this case, the category II solution restores flux through both dead-end metabolites.

2.3. Metabolic Modeling Setup

2.3.1. Metabolic Model for an Organism of Interest

Methods in this chapter are geared towards improving the annotation of organisms for which a genome-scale reconstruction is available (see Note 2). A metabolic network consists of two major components: the stoichiometric matrix (S-matrix) and a set of rules for gene–protein-reaction (GPR) relationships. The S-matrix is comprised of biochemical reactions that occur in an organism while GPR relationships represent conditional statements in Boolean logic between ORFs and their enzymatic functions in the S-matrix.

Availability (typically in the SBML format (31)): BiGG database (32), MEMOSys (33), SEED (19), and literature

2.3.2. Types of Modifications That Can Be Made to the Metabolic Network

In the process of curating a metabolic reconstruction, various types of network modifications can be made (see Fig. 2):

Category I. Adding an extracellular transport reaction (exchange reaction) where the metabolite can either be taken up or secreted by the cell.

Category II. Adding a new intracellular enzymatic or spontaneous reaction.

Category III. Adding an extracellular transport reaction(exchange reaction) where the metabolite can either be taken up or secreted by the cell. These reactions define nutrient conditions.

Category IV. Adding a new transport reaction within the cell. These reactions are often lumped with category III reactions and are limited to compartmentalized metabolic models (see Note 3).

2.4. Metabolic Modeling Tools: COBRA Software

The Constraint-Based Reconstruction and Analysis (COBRA) Toolbox is a platform implemented in MATLAB (MathWorks, Natick, MA) for interrogating metabolic reconstructions. Many functions within COBRA require mathematical programming solvers. These functions often utilize data from KEGG as a reaction database.

Availability: http://opencobra.sourceforge.net/openCOBRA (34)

SBML Toolbox imports SBML-formatted metabolic models (31) into the COBRA Toolbox.

Availability: SBML Toolbox (http://sbml.org/Software/SBMLToolbox) (35)

Flux balance analysis (*FBA*) identifies a flux distribution through the reaction network that produces an optimal flux through the objective function.

Availability: COBRA Toolbox 2.0 under optimizeCbModel()

Flux variability analysis (*FVA*) computes the ranges of possible fluxes for all reactions in a network while still maintaining a primary objective flux value such as optimal biomass production (36, 37).

Availability: COBRA Toolbox 2.0 under fluxVariability()

GapFind finds all dead-end metabolites in a network including root no-production and root no-consumption metabolites (see Fig. 2a) (38).

Availability: COBRA Toolbox 2.0 under gapFind()

DetectDeadEnds finds some dead-end metabolites, all of which participate in only one reaction as determined by the S-matrix.

Availability: COBRA Toolbox 2.0 under detectDeadEnds()

Flux sampling samples feasible flux distributions without a user-defined objective function, identifying reactions that can only carry zero flux (blocked reactions).

Availability: COBRA Toolbox 2.0 under gpSampler()

SMILEY predicts a minimum set of enzymatic or transport reactions to add in order to sustain growth in one condition (39).

Availability: COBRA Toolbox 2.0 under growthExpMatch()

2.5. Metabolic Modeling Tools: Pathway Tools Software

The Pathway Tools software environment integrates genome, pathway, and regulatory data for analysis and visualization (40, 41). These functions often utilize data from MetaCyc and UniProt as a reaction database.

Availability: http://biocyc.org/download.shtml

MetaFlux utilizes multiple gap-filling methods to aid in developing metabolic models and defining a feasible biomass reaction (42).

PHFiller (Pathway Hole Filler) finds genes for orphan reactions using BLAST and protein databases (43).

PHFiller-GC (Pathway Hole Filler – Genome Context) extends on the PHFiller algorithm to use a context-specific prediction of genes for orphan reactions based on shared pathways, shared operons between proteins, shared proteins in a complex, and regulatory interactions (44).

2.6. Metabolic Modeling Tools: Stand-Alone Algorithms

GapFill suggests adding reactions to restore flux through dead-end reactions (38).

Availability: http://maranas.che.psu.edu/software.htm

GrowMatch suggests adding or removing reactions to reconcile differences between model predictions and gene essentiality screens and nutrient utilization assays (45). Requires a reaction database.

Availability: http://maranas.che.psu.edu/software.htm and modified version implemented in SEED (19)

OMNI (optimal metabolic network identification) compares metabolic flux analysis data and *in silico* predictions of flux distributions and suggests reactions to add/remove to better correlate predictions and experimental data (46).

Availability: see Supporting Information in Ref. (46)

BNICE suggests reactions that can consume or produce metabolites based on reaction rules from the EC classification system (47).

Availability: see Methods in Ref. (47)

2.7. Experimental Validation of Candidate ORFs

In recent literature, experimental validation of candidate transcripts has been performed for the alga *Chlamydomonas reinhardtii* (48–50). See Subheading 3.6 for a brief description of some key methods. Additionally, see Refs. (48–50) for a detailed description of the materials, reagents, and protocols used to perform the experimental validations.

3. Methods

Here, we describe iterative steps to improving the genome annotation for an organism of interest using genome-scale metabolic network modeling. This is done using a systems approach to

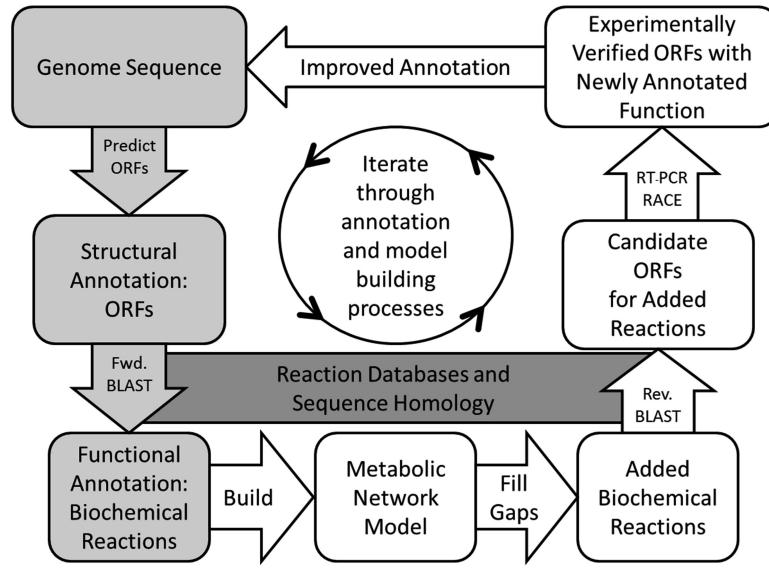

Fig. 3. Workflow for improving the annotation of a genome sequence using bioinformatics techniques (*light gray boxes*) and metabolic modeling (*white boxes*). *Dark gray box* emphasizes the importance of existing knowledge databases for biochemical reactions in assigning function through sequence homology: Forward (Fwd.) BLAST compares the sequence of an uncharacterized ORF against the genomes of other organisms in order to infer function. Reverse (Rev.) BLAST compares your organism's genome to sequences of proteins from other organisms that catalyze the same biochemical reaction to identify candidate ORFs for a reaction of interest.

identify novel reactions that take place within an organism and functionally annotate ORFs that catalyze previously uncharacterized and newly identified enzymatic functions (51). These methods are focused into four major steps, which are iteratively compatible with each other (see Fig. 3 for overview):

1. Suggest modifications to the S-matrix by adding or removing biochemical reactions based on:
   a. Manual inspection of literature evidence (Subheading 3.1)
      - Early-stage examination of central metabolism
      - Midstage multipathway refinement
      - Late-stage network validations
   b. Semiautomated analysis of network topology (Subheading 3.2)
      - Gap filling based on dead-end metabolites
      - Gap filling based on blocked reactions
   c. Semiautomated analysis of experimental data (Subheading 3.3)
      - Adding or removing reactions based on prediction/experimental discrepanciesRemoving reactions based on prediction/experimental discrepancies.

Table 1
Problem-driven methods to identify and reconcile knowledge gaps in metabolic models

Type of knowledge gaps in metabolic network	Methods to identify knowledge gaps	Methods to reconcile knowledge gaps	Resulting network modifications
Dead-end metabolites: Metabolites cannot be consumed or produced in steady-state simulations	GapFind DetectDeadEnds	GapFill BNICE	Add new reactions: Category II preferred
Blocked reactions: Reactions cannot carry flux due to dead-end metabolites	FVA Flux sampling	SMILEY	Add new reactions: Category II preferred
Metabolic model is inconsistent with biological data (e.g., fluxomics data, gene essentiality, or nutrient utilization assays)	FBA predicts no growth when growth is experimentally observed	SMILEY GrowMatch OMNI	Add new reactions: Category II preferred
	FBA predicts growth when growth is not experimentally observed	GrowMatch OMNI	Remove existing reactions: Category I or III preferred
Orphan reactions: Metabolic reactions that have no associated ORFs in the GPR	findOrphanRxns	Reverse BLAST PHFiller PHFiller-GC	Assign ORF to orphan reaction: (Pending experimental verification)

2. Assign candidate transcripts/ORFs that catalyze candidate reactions and existing orphan reactions (Subheading 3.4).
3. Manually choose network modifications at the systems level (Subheading 3.5).
4. Experimentally verify the presence and structure of candidate ORFs (Subheading 3.6).

Throughout and after the process of reconstructing a metabolic network, use these methods to identify and address knowledge gaps (see Table 1). Iterate through these semiautomated algorithms combined with manual inspection and experimental validation in order to generate a consistently high-quality metabolic model and contribute to the genome annotation of an organism.

3.1. Biologically Inspired Metabolic Network Refinement

At all stages in the reconstruction process, it is important to evaluate the functionality of the network model so that any subsequent modifications consistently lead to a higher-quality model.

Any deficiencies in model functionality should be manually examined so as to identify enzymes/reactions to fill in these gaps in knowledge.

3.1.1 *Early-stage examination of central metabolism.* While drafting a reconstruction, examine central metabolic pathways (e.g., glycolysis, TCA cycle, and pentose phosphate pathway) with literature support for completeness. Visualize these pathways in biochemical reaction databases (see Subheading 2.2) such as KEGG. For example, to evaluate functionality of glycolysis in a newly reconstructed metabolic network, use FBA to optimize for pyruvate production via the pyruvate kinase reaction under glucose-only nutrient conditions. If zero flux is obtained for the objective, manually check for gaps or deficiencies in the pathway.

3.1.2 *Midstage multipathway refinement.* At intermediate stages of model building, expand this process from individual pathways to include multiple pathways (such as amino acid and nucleotide metabolism). For example, use FBA to simulate the production of individual amino acids and nucleotides given a particular carbon source (e.g., glucose). It is important to utilize experimental literature evidence in this process (e.g., not all organisms can synthesize all 20 amino acids de novo and may need to scavenge necessary amino acids from the environment).

3.1.3 *Late-stage network validations.* In the later stages of the reconstruction process, ensure basic functionality of the model in the context of a biologically inspired objective function such as biomass or ATP production. A semiautomated algorithm for establishing a feasible biomass is *MetaFlux*. This algorithm suggests a maximum subset of biomass metabolites that can be produced given a minimum set of network modifications in the form of added/removed reactions. This algorithm accommodates an initial biomass reaction that may include some metabolites that are unable to be produced. Network changes can be manually inspected for feasibility using visualization software that is integrated with the Pathway Tools platform.

3.2. Using Network Topology to Address Dead-End Metabolites and Blocked Reactions

At any stage in the reconstruction process, there may be blocked reactions, which are reactions that cannot carry flux in a metabolic model, usually as a result of containing, being upstream (root no-consumption), or being downstream (root no-production) of a dead-end metabolite (see Fig. 2). Unblocking these reactions can be performed at the metabolite level, aimed at restoring fluxes that utilize the dead-end metabolite, or at the reaction level, aimed at

restoring flux through the blocked reaction. These two general methods may provide different gap-filling solutions, though they share the same root causes.

For network-topology-based methods in the COBRA Toolbox, set nutrient uptake (lower bound of exchange reactions) to −1 and all other reaction bounds to a large number such as 100,000 (see Note 4).

3.2.1. At the Metabolite Level: Restoring Flux Through Dead-End Metabolites

3.2.1.1 (optional) Decompartmentalize the model into only intracellular and extracellular compartments (i.e., remove subcellular compartments such as mitochondria, endoplasmic reticulum, nucleus). Dead-end metabolites may block additional reactions based on compartmentalization, and in one instance, this was addressed by decompartmentalizing the human metabolic model (52). In this case, substantially fewer reactions were blocked making this approach more manageable while forgoing the addition of category IV reactions.

3.2.1.2 Identify dead-end metabolites using one or both of the following methods:

GapFind. Use the function gapFind() with a COBRA model, which will return all root no-production (and optionally all root no-consumption) dead-end metabolites. Although gaps shown in Fig. 2 are fairly obvious, some nonobvious gaps are possible within the scope of a metabolic network (51).

DetectDeadEnds. Supply detectDeadEnds() with a COBRA model, which will return a list of all metabolites that participate in only one reaction (optionally excluding extracellular metabolites) (34). This method does not return all possible dead-end gaps, but it detects metabolites that participate in only one reaction, including reversible reactions (category I solution of a dead end), which GapFind will not detect because these are not technically dead ends. It may be preferred to identify category II solutions to incorporate the metabolite into metabolic pathways.

3.2.1.3 Suggest reactions that include dead-end metabolites as products for root no-production cases or as substrates for root no-consumption cases. Semiautomated algorithms for suggesting reactions include:

GapFill. Supply GapFill with a reaction database, a list of all reactions from the network reconstruction (see Note 5), and root no-production/consumption

metabolites. Only category I, II, and III suggestions will be returned.

BNICE. Supply BNICE with each pair of dead-end metabolites, which will suggest feasible biochemical reactions that link the two (see Note 6).

3.2.2. At the Reaction Level: Restoring Flux to Blocked Reactions

3.2.2.1 Identify blocked reactions using one or both of the following methods:

FVA. Using a COBRA model with any feasible objective function (carries positive flux using FBA), perform FVA() on all reactions (see Note 7). This may be useful for identifying blocked reactions in other contexts, but not necessarily for adding new reactions to the model. Consider all reactions that have lower and upper flux ranges of approximately [0, 0] as blocked reactions (see Note 7). Perform FVA using both the on and off setting for allowing loops (set allowLoops to 1 or 0, respectively); allowing loops will hide reactions that cannot carry flux unless a potentially thermodynamically infeasible loop is carrying flux, which may or not be relevant to the biology of the model.

Flux sampling. Execute gpSampler() on a COBRA model to sample possible fluxes throughout all reactions. This algorithm does not require an objective function. Consider reactions without nonzero sampled flux values as blocked reactions.

3.2.2.2 Suggest additional reactions that restore flux through each blocked reaction.

SMILEY. For each blocked reaction, execute growthExpMatch() with the objective function set to the blocked reaction (see Note 8) and with a relevant. with a relevant reaction database from KEGG (see Subheading 2.1). Running multiple iterations will suggest minimum subsets of additional reactions that restore flux to blocked reaction. This algorithm will suggest category I, II, III, and, if not decompartmentalized, IV reactions. For an example of applying SMILEY to restore flux through blocked reactions, see Ref. (52).

3.2.2.3 Proceed to identifying candidate ORFs for these reactions (Subheading 3.4) and narrow down choices of network modifications (Subheading 3.5) to experimentally validate (Subheading 3.6).

3.3. Gene Essentiality Screens, Nutrient Utilization Assays, and Fluxomics Data Suggest Experimentally Inspired Model Refinements

Implementing FBA on metabolic network reconstructions provides the ability to predict growth yields for organisms under different substrate conditions and genetic perturbations (53). Implementing FBA on the iAF1260 metabolic reconstruction of *E. coli* yielded an accuracy of 91% for gene essentiality predictions as compared to experimental observations (54).

Adding (or removing) reactions to reconcile predictions with experimental data:

3.3.1 Define a biologically relevant objective function for the metabolic model (55, 56). Consider a biomass function of nucleic acids, amino acids, lipids, and energy maintenance when comparing predictions to growth assays. Some experimental screens measure secretion of metabolites or other phenotypic properties, so adjust the objective of the metabolic model accordingly.

3.3.2 Define nutrient conditions relevant to biological setting. Consider carbon, nitrogen, phosphorus, and sulfur sources as well as presence of oxygen (see Ref. (57) for an example of establishing a minimal media relevant to biological conditions).

3.3.3 Manipulate the model to emulate any experimental perturbations. For example, remove relevant reactions from a COBRA model using removeRxns() or deleteModelGenes() for essentiality screens.

3.3.4 (optional) Ensure feasibility of the biomass function for at least one condition manually by FBA. Using algorithms in this section with an objective that requires a large number of additional reactions is usually neither computationally feasible nor biologically relevant.

3.3.5 Identify knowledge gaps by comparing experimental data to model data. Perform FBA for each condition and compare output to observed result (see Table 1). Possible discrepancies occur when:

(a) Model predicts growth (or other relevant output) when no growth is experimentally observed; as a result, remove reactions from network.

(b) Model predicts no growth (or other relevant output) when growth is experimentally observed; as a result, add reactions to network.

3.3.6 Suggest reactions to add (or remove) using one or more of the following semiautomated algorithms (see Note 9):

SMILEY. Execute growthExpMatch() with a relevant reaction database from KEGG (see Subheading 2.1) for each condition that needs rectification. Results include

suggestions of only additional reactions. For an example of applying SMILEY to gene essentiality data in *E. coli*, see Ref. (57).

GrowMatch. GrowMatch suggests sets of reactions to add/remove that reconcile predictions for at least one growth condition. This algorithm identifies which network changes resolve model predictions and experimental observations for one growth condition while avoiding the creation of new inconsistencies in other conditions.

OMNI. Supply OMNI with a library of reactions from a reaction database such as KEGG, a list of existing reactions that are allowed to be removed, and fluxomics data. Suggestions include both added and removed reactions that cause fluxomics data and metabolic predictions to match.

3.3.7 Proceed to identifying candidate ORFs for these reactions (Subheading 3.4) and narrow down choices of network modifications (Subheading 3.5) to experimentally validate (Subheading 3.6).

3.4. Predicting Candidate ORFs for Orphan Reactions

Orphan reactions are enzymatic reactions that have no assigned ORFs or proteins that catalyze this reaction (see Fig. 1c). The disconnect between structural annotation and functional annotation occurs either in the scope of an individual organism (local orphan) or across all organisms (global orphan). For local orphan reactions, utilizing BLAST and BLAST-related algorithms are useful for identifying potential ORFs, but global orphan reactions have no known enzymes which we can compare sequence similarity. In a metabolic model, an orphan reaction has no relationship defined in the GPR, but the stoichiometry is defined in the S-matrix. This section details methods to annotate newly added orphan reactions from Subheadings 3.1 to 3.3 as well as orphan reactions already included in the model.

3.4.1 In a COBRA model, add candidate reactions and perform findOrphanRxns() to locate all orphan reactions.

3.4.2 Identify whether each reaction is a local orphan reaction or a global orphan reaction. Query a reaction database such as KEGG or MetaCyc to see a list of annotated protein sequences across all organisms. Those that lack annotation are orphan reactions.

3.4.3 Use the following algorithms to identify candidate ORFs that may encode for enzymes that catalyze an orphan reaction of interest.

Reverse BLAST. Manually identify proteins in other organisms that share the same enzymatic function

(EC number) from MetaCyc or similar database. Choose proteins from phylogenetically similar organisms first. Perform BLAST for each candidate protein against the whole genome of the modeled organism to identify similar sequences. See Ref. (8) for details on different BLAST techniques. Results are limited to local orphans as they rely on enzymatic functions linked with ORFs in other organisms.

PHFiller. Execute PHFiller to identify proteins that catalyze the orphan reaction of interest. PHFiller returns lists of candidate ORFs from the organism's sequenced genome that may catalyze each orphan reaction. This algorithm semiautomates performing Reverse BLAST using existing protein databases, and results are limited to local orphans as they rely on enzymatic functions linked with ORFs in other organisms.

PHFiller-GC. Improvements over PHFiller in this algorithm consider not only BLAST sequence similarities but also similarity based on other associations such as shared protein complexes, shared operons, and regulatory and transcription factors. This allows the algorithm to explore beyond the structural level into pathways in identifying global orphan reactions.

3.5. Choosing Reactions to Experimentally Validate

Before proceeding to experimental methods for verifying candidate ORFs and validating their function, efforts should be made to manually curate the metabolic model. All algorithms described in Subheadings 3.1–3.3 are defined as semiautomated because manual inspection is essential to ensuring that only biologically relevant and plausible reactions are added or removed from the metabolic network. Using all gap-filling methods described in this chapter would yield an exorbitant number of suggested network modifications, so this section outlines general considerations for selecting reactions to subsequently validate.

Use a systems approach to add or remove reactions: consider simpler solutions that resolve the most knowledge gaps with the fewest network changes before addressing each problem on its own. Instead of adding all reactions from these methods to the model at once, select only a few solutions and iterate through the model and rerun gap-filling techniques. This iterative systems approach to gap filling will ensure higher-quality models and better genome annotations.

Considerations for adding reactions based on suggestions from semiautomated algorithms:

- When multiple solutions are available for the various gap-filling conditions, choose small subsets of reactions that resolve model predictions with the most experimental conditions.

- Perform several iterations of each algorithm to produce a comprehensive set of possible reactions. When solutions are limited, try other databases where available.
- Choose reactions that have candidate ORFs when possible over those that will remain as local orphan reactions.
- Examine all solutions for biological feasibility. Take into consideration literature suggestions.
- Prioritize adding category II reactions when possible (see Table 1).
- Category I and III reactions are simplified solutions (e.g., exporting a root no-consumption metabolite out of the cell to unblock a reaction). Add category I reactions only when literature supports the reversibility of an enzymatic reaction or a separate distinct enzyme can catalyze the opposite reaction. Add category III reactions only when literature supports the concept that a metabolite is excreted and/or a transporter or relevant channel has been characterized.
- Calculate feasible ranges of ΔG (free energy) for each reaction in the organism of interest and exclude thermodynamically infeasible reactions.
- Be cautious of creating thermodynamically infeasible loops when adding reactions. For example, the H+ gradient across the mitochondrial membrane fuels ATP regeneration from ADP through ATP synthase. The electron transport chain maintains this gradient by utilizing energy generated from other metabolic processes in order to pump H+ out of the mitochondrion, but additional reactions in a metabolic model may have contribute to this process. In a published reconstruction of human metabolism (58), a reversible mitochondrial transporter allows symport of lactate and H+ ions, while a separate reaction allows lactate to freely diffuse between mitchondria and the cytosol. These two reactions effectively serve as a thermodynamically infeasible alternative to the electron transport chain, fueling infinite ATP production through ATP synthase. While this loop may have been created to unblock a reaction involving lactate, it inadvertently reduces the quality of the model with respect to processes that involve ATP regeneration.

Considerations for removing reactions based on suggestions from semiautomated algorithms:

- Some metabolic models have confidence scores associated with annotated GPR relationships, such as those in the BiGG database (32). Choose to remove reactions with low confidence (annotated based on inferred function made only by sequence homology) over those with high confidence (supported with biochemical and literature evidence).

- Be cautious in removing reactions permanently. Enzymatic function may be possible at the genome scale under the right conditions, but enzyme expression or function may be dependent on specific environmental, signaling, regulatory, or time-dependent factors that prevent the organism from adapting to a particular growth condition.
- Not all reactions need to be unblocked. Evolutionary trends may have caused a loss of a key metabolic enzyme in a pathway, leaving the other members structurally and functionally intact at the molecular level, but rendered loss in functionally in the larger scope of the metabolic network. These are called biological gaps instead of knowledge gaps (51). Thus, be cautious and consider adding reactions only when justified in the scope of biology (see Subheading 3.6).
- Prioritize removing category I and III reactions over category II reactions unless literature evidence strongly supports lack of a transport or reversible reaction.

3.6. Experimentally Verifying and Sequencing Candidate ORFs

The process of adding new reactions and adopting orphan reactions (and consequently incorporating new ORFs into the model) yields an opportunity to structurally annotate candidate transcripts through experimental validation. For each candidate ORF identified in this workflow, we can experimentally verify the presence and sequence of the transcript. We can then use these results to guide the selection process involved in the refinement of our model's annotation and metabolic model. Note that the following methods were used in part in recent literature to perform model-driven experimental validation of candidate transcripts in the alga *C. reinhardtii* (48–50).

As a first step, the presence of candidate ORFs can be verified using *RT-PCR* (reverse transcription polymerase chain reaction). In this method, an RNA strand is reverse transcribed into complementary DNA (cDNA) with the aid of the enzyme reverse transcriptase. The resulting cDNA strands are used as templates in PCR reactions. RT-PCR is performed with forward and reverse primers corresponding to the putative ORFs of the enzymes of interest. This step can be used for amplification of ORFs.

If only a partial sequence of an ORF is available or if the RT-PCR step failed due to errors in ORF termini, *RACE* (rapid amplification of cDNA ends) can be performed to define the transcript boundaries (59). RACE is designed to identify the 5' and 3' ends of a transcript if the ORFs could either not be cloned or could be verified only at one end. In this method, cDNAs are generated by using PCR to amplify copies of sequence between a point within the transcript and the end (either the 3' or 5' end). The minimum information that is required is essentially a short stretch of sequence in the ORF to be cloned (60). The ORF-specific primers

are tailed with Gateway-compatible sequences (see *Gateway cloning technology* from Invitrogen). The amplicons generated from RT-PCR are cloned using a Gateway donor vector and are transformed into *E. coli*. These transformed bacteria are used as a source of template in PCR reactions.

The amplicons can be sequenced using different methods. For instance, the 5′ and 3′ ends can be verified by high-throughput *Sanger sequencing*, or ORF amplicons can be sequenced using the *Roche 454FLX Titanium sequencing* system. In the latter method, a given ORF could be considered experimentally validated if there was more than 98% coverage of the entire length of the reference sequence (i.e., predicted gene model) by the assembled contigs from the 454 reads. Successful cloning and matched sequence of a given ORF to its predicted gene model would experimentally validate the presence of the hypothesized transcript (48–50).

3.7. Outlook

Metabolic modeling can be used to improve genome annotation by filling in network gaps and linking biological functions to ORFs. Throughout this process, we want to focus on using modeling as a semiautomated tool for generating hypotheses and using manual inspection, biological reasoning, and experimental validation for revising both functional and structural annotations. Iterate through these computational and experimental steps in order to create both higher-quality models and annotations. The capabilities of these techniques will improve as annotations for other organisms become more accurate and complete, giving the opportunity to guide and accelerate annotation efforts in existing and new sequences.

4. Notes

1. Even once a genome has been completely sequenced, determining the structure of ORFs can be difficult with complex initiation, termination and splicing rules, and imperfect gene-calling algorithms (61).
2. Supplemental methods in (49, 56) describe how to draft an initial reconstruction from existing annotations.
3. see Ref. (52) for an example of distinctions between category I and III reactions. Category IV reactions are newly defined in this chapter as a decompartmentalized model was used previously.
4. Setting intracellular reactions to extremely large values, and opening all exchange reactions, allows for correct functionality of network-topology-based methods (34). More specifically, set the lower bounds and upper bounds of:

a. Forward-only reactions to [0, 10,000]
b. Reverse-only reactions to [−10,000, 0]
c. Reversible reactions to [−10,000, 10,000]
d. Exchange reactions to [−**1**, 10,000] (nutrient uptake is typically a negative flux)

If an algorithm is still incorrectly identifying gaps, then increase the magnitude of nonuptake bounds by 10, 100, or 1,000 to fix this issue.

5. Include only forward reactions. For any reversible reactions, add a new forward reaction that consumes/produces the same metabolites in the opposite direction.
6. Suggestions may include novel reactions that may not be characterized in other organisms. Preferably, only add novel reactions when a high-quality network has a few gaps that cannot be filled by other methods.
7. FVA calculates the full possible ranges of flux values for all reactions while maintaining a set percentage (default: 100%) of maximal flux through the objective. For identifying blocked reactions in the scope of the entire network, set the parameter, optPercentage, to a lower value such as 10%. In this case, maintaining 100% flux through an objective such as biomass can limit flux through alternative pathways. If all flux from a rate-limiting carbon source is allocated to biomass, then alternative reactions in nonoptimal pathways that utilize carbon will result in flux ranges of [0,0] flux, yielding unnecessarily blocked reactions.
8. For reversible reactions, solutions that restore flux through a blocked reaction in one direction will not necessarily restore flux in the opposite direction. To generate solutions that unblock the reaction in both directions, separately set the objective function to minimize and maximize flux through the blocked reaction. For unblocking multiple reactions with one suggested set of reactions, set the objective to maximize flux through a set of reactions that are blocked.
9. Run multiple iterations of each algorithm as they rely on sampling of added or removed reactions.

References

1. Blaby-Haas CE, de Crecy-Lagard V (2011) Mining high-throughput experimental data to link gene and function. Trends Biotechnol 29 (4):174–182. doi:10.1016/j.tibtech.2011.01.001
2. Hanson AD, Pribat A, Waller JC, de Crecy-Lagard V (2010) 'Unknown' proteins and 'orphan' enzymes: the missing half of the engineering parts list–and how to find it. Biochem J 425(1):1–11. doi:10.1042/BJ20091328
3. Pouliot Y, Karp PD (2007) A survey of orphan enzyme activities. BMC Bioinformatics 8:244. doi:10.1186/1471-2105-8-244

4. Rombel IT, Sykes KF, Rayner S, Johnston SA (2002) ORF-FINDER: a vector for high-throughput gene identification. Gene 282 (1–2):33–41
5. Lamesch P, Li N, Milstein S, Fan C, Hao T, Szabo G, Hu Z, Venkatesan K, Bethel G, Martin P, Rogers J, Lawlor S, McLaren S, Dricot A, Borick H, Cusick ME, Vandenhaute J, Dunham I, Hill DE, Vidal M (2007) hORFeome v3.1: a resource of human open reading frames representing over 10,000 human genes. Genomics 89(3):307–315. doi:10.1016/j.ygeno.2006.11.012
6. Frishman D (2007) Protein annotation at genomic scale: the current status. Chem Rev 107(8):3448–3466. doi:10.1021/cr068303k
7. Erdin S, Lisewski AM, Lichtarge O (2011) Protein function prediction: towards integration of similarity metrics. Curr Opin Struct Biol 21(2):180–188. doi:10.1016/j.sbi.2011.02.001
8. Emes RD (2008) Inferring function from homology. Methods Mol Biol 453:149–168. doi:10.1007/978-1-60327-429-6_6
9. Jones CE, Brown AL, Baumann U (2007) Estimating the annotation error rate of curated GO database sequence annotations. BMC Bioinformatics 8:170. doi:10.1186/1471-2105-8-170
10. Thiele I, Palsson BO (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. Nat Protoc 5(1):93–121. doi:10.1038/nprot.2009.203
11. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL (2005) GenBank. Nucleic Acids Res 33(Database issue):D34–D38. doi:10.1093/nar/gki063
12. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215(3):403–410. doi:10.1016/S0022-2836(05)80360-2
13. Johnson M, Zaretskaya I, Raytselis Y, Merezhuk Y, McGinnis S, Madden TL (2008) NCBI BLAST: a better web interface. Nucleic Acids Res 36(Web Server issue):W5–W9. doi:10.1093/nar/gkn201
14. Kanehisa M, Goto S (2000) KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res 28(1):27–30
15. Kanehisa M, Goto S, Kawashima S, Okuno Y, Hattori M (2004) The KEGG resource for deciphering the genome. Nucleic Acids Res 32(Database issue):D277–D280. doi:10.1093/nar/gkh063
16. Kanehisa M, Araki M, Goto S, Hattori M, Hirakawa M, Itoh M, Katayama T, Kawashima S, Okuda S, Tokimatsu T, Yamanishi Y (2008) KEGG for linking genomes to life and the environment. Nucleic Acids Res 36(Database issue):D480–D484. doi:10.1093/nar/gkm882
17. Gasteiger E, Gattiker A, Hoogland C, Ivanyi I, Appel RD, Bairoch A (2003) ExPASy: the proteomics server for in-depth protein knowledge and analysis. Nucleic Acids Res 31 (13):3784–3788
18. Schneider M, Tognolli M, Bairoch A (2004) The Swiss-Prot protein knowledgebase and ExPASy: providing the plant community with high quality proteomic data and tools. Plant Physiol Biochem 42(12):1013–1021. doi:10.1016/j.plaphy.2004.10.009
19. Henry CS, DeJongh M, Best AA, Frybarger PM, Linsay B, Stevens RL (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. Nat Biotechnol 28(9):977–982. doi:10.1038/nbt.1672
20. Caspi R, Altman T, Dreher K, Fulcher CA, Subhraveti P, Keseler IM, Kothari A, Krummenacker M, Latendresse M, Mueller LA, Ong Q, Paley S, Pujar A, Shearer AG, Travers M, Weerasinghe D, Zhang P, Karp PD (2012) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. Nucleic Acids Res 40 (Database issue):D742–D753. doi:10.1093/nar/gkr1014
21. Karp PD, Caspi R (2011) A survey of metabolic databases emphasizing the MetaCyc family. Arch Toxicol 85(9):1015–1033. doi:10.1007/s00204-011-0705-2
22. Hertz-Fowler C, Peacock CS, Wood V, Aslett M, Kerhornou A, Mooney P, Tivey A, Berriman M, Hall N, Rutherford K, Parkhill J, Ivens AC, Rajandream MA, Barrell B (2004) GeneDB: a resource for prokaryotic and eukaryotic organisms. Nucleic Acids Res 32(Database issue):D339–D343. doi:10.1093/nar/gkh007
23. Kumar A, Suthers PF, Maranas CD (2012) MetRxn: a knowledgebase of metabolites and reactions spanning metabolic models and databases. BMC Bioinformatics 13(1):6. doi:10.1186/1471-2105-13-6
24. Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, Yeh LS (2004) UniProt: the Universal Protein knowledgebase. Nucleic Acids Res 32(Database issue):D115–D119. doi:10.1093/nar/gkh131
25. Bolser DM, Chibon PY, Palopoli N, Gong S, Jacob D, Del Angel VD, Swan D, Bassi S, Gonzalez V, Suravajhala P, Hwang S, Romano P, Edwards R, Bishop B, Eargle J, Shtatland T,

Provart NJ, Clements D, Renfro DP, Bhak D, Bhak J (2012) MetaBase—the wiki-database of biological databases. Nucleic Acids Res 40 (Database issue):D1250–D1254. doi:10.1093/nar/gkr109

26. Baba T, Ara T, Hasegawa M, Takai Y, Okumura Y, Baba M, Datsenko KA, Tomita M, Wanner BL, Mori H (2006) Construction of *Escherichia coli* K-12 in-frame, single-gene knockout mutants: the Keio collection. Mol Syst Biol 2006:0008. doi:10.1038/msb4100050
27. Yamamoto N, Nakahigashi K, Nakamichi T, Yoshino M, Takai Y, Touda Y, Furubayashi A, Kinjyo S, Dose H, Hasegawa M, Datsenko KA, Nakayashiki T, Tomita M, Wanner BL, Mori H (2009) Update on the Keio collection of *Escherichia coli* single-gene deletion mutants. Mol Syst Biol 5:335. doi:10.1038/msb.2009.92
28. Zhang R, Ou HY, Zhang CT (2004) DEG: a database of essential genes. Nucleic Acids Res 32(Database issue):D271–D272. doi:10.1093/nar/gkh024
29. Zhang R, Lin Y (2009) DEG 5.0, a database of essential genes in both prokaryotes and eukaryotes. Nucleic Acids Res 37(Database issue): D455–D458. doi:10.1093/nar/gkn858
30. Chen WH, Minguez P, Lercher MJ, Bork P (2012) OGEE: an online gene essentiality database. Nucleic Acids Res 40(Database issue):D901–D906. doi:10.1093/nar/gkr986
31. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novere N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J, Forum S (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19(4):524–531
32. Schellenberger J, Park JO, Conrad TM, Palsson BO (2010) BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions. BMC Bioinformatics 11:213. doi:10.1186/1471-2105-11-213
33. Pabinger S, Rader R, Agren R, Nielsen J, Trajanoski Z (2011) MEMOSys: bioinformatics platform for genome-scale metabolic models. BMC Syst Biol 5:20. doi:10.1186/1752-0509-5-20
34. Schellenberger J, Que R, Fleming RM, Thiele I, Orth JD, Feist AM, Zielinski DC, Bordbar A, Lewis NE, Rahmanian S, Kang J, Hyduke DR, Palsson BO (2011) Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nat Protoc 6 (9):1290–1307. doi:10.1038/nprot.2011.308
35. Keating SM, Bornstein BJ, Finney A, Hucka M (2006) SBMLToolbox: an SBML toolbox for MATLAB users. Bioinformatics 22 (10):1275–1277. doi:10.1093/bioinformatics/btl111
36. Mahadevan R, Schilling CH (2003) The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. Metab Eng 5(4):264–276
37. Chavali AK, D'Auria KM, Hewlett EL, Pearson RD, Papin JA (2012) A metabolic network approach for the identification and prioritization of antimicrobial drug targets. Trends Microbiol 20(3):113–123. doi:10.1016/j.tim.2011.12.004
38. Satish Kumar V, Dasika MS, Maranas CD (2007) Optimization based automated curation of metabolic reconstructions. BMC Bioinformatics 8:212. doi:10.1186/1471-2105-8-212
39. Reed JL, Patel TR, Chen KH, Joyce AR, Applebee MK, Herring CD, Bui OT, Knight EM, Fong SS, Palsson BO (2006) Systems approach to refining genome annotation. Proc Natl Acad Sci U S A 103(46):17480–17484. doi:10.1073/pnas.0603364103
40. Karp PD, Paley S, Romero P (2002) The Pathway Tools software. Bioinformatics 18(Suppl 1):S225–S232
41. Karp PD, Paley SM, Krummenacker M, Latendresse M, Dale JM, Lee TJ, Kaipa P, Gilham F, Spaulding A, Popescu L, Altman T, Paulsen I, Keseler IM, Caspi R (2010) Pathway Tools version 13.0: integrated software for pathway/genome informatics and systems biology. Brief Bioinform 11(1):40–79. doi:10.1093/bib/bbp043
42. Latendresse M, Krummenacker M, Trupp M, Karp PD (2012) Construction and completion of flux balance models from pathway databases. Bioinformatics 28(3):388–396. doi:10.1093/bioinformatics/btr681
43. Green ML, Karp PD (2004) A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. BMC Bioinformatics 5:76. doi:10.1186/1471-2105-5-76
44. Green ML, Karp PD (2007) Using genome-context data to identify specific types of functional associations in pathway/genome databases. Bioinformatics 23(13):i205–i211. doi:10.1093/bioinformatics/btm213

45. Kumar VS, Maranas CD (2009) GrowMatch: an automated method for reconciling in silico/in vivo growth predictions. PLoS Comput Biol 5(3):e1000308. doi:10.1371/journal.pcbi.1000308
46. Herrgard MJ, Fong SS, Palsson BO (2006) Identification of genome-scale metabolic network models using experimentally measured flux profiles. PLoS Comput Biol 2(7):e72. doi:10.1371/journal.pcbi.0020072
47. Hatzimanikatis V, Li C, Ionita JA, Henry CS, Jankowski MD, Broadbelt LJ (2005) Exploring the diversity of complex metabolic networks. Bioinformatics 21(8):1603–1609. doi:10.1093/bioinformatics/bti213
48. Ghamsari L, Balaji S, Shen Y, Yang X, Balcha D, Fan C, Hao T, Yu H, Papin JA, Salehi-Ashtiani K (2011) Genome-wide functional annotation and structural verification of metabolic ORFeome of *Chlamydomonas reinhardtii*. BMC Genomics 12(Suppl 1):S4. doi:10.1186/1471-2164-12-S1-S4
49. Manichaikul A, Ghamsari L, Hom EF, Lin C, Murray RR, Chang RL, Balaji S, Hao T, Shen Y, Chavali AK, Thiele I, Yang X, Fan C, Mello E, Hill DE, Vidal M, Salehi-Ashtiani K, Papin JA (2009) Metabolic network analysis integrated with transcript verification for sequenced genomes. Nat Methods 6(8):589–592. doi:10.1038/nmeth.1348
50. Chang RL, Ghamsari L, Manichaikul A, Hom EF, Balaji S, Fu W, Shen Y, Hao T, Palsson BO, Salehi-Ashtiani K, Papin JA (2011) Metabolic network reconstruction of *Chlamydomonas* offers insight into light-driven algal metabolism. Mol Syst Biol 7:518. doi:10.1038/msb.2011.52
51. Orth JD, Palsson BO (2010) Systematizing the generation of missing metabolic knowledge. Biotechnol Bioeng 107(3):403–412. doi:10.1002/bit.22844
52. Rolfsson O, Palsson BO, Thiele I (2011) The human metabolic reconstruction Recon 1 directs hypotheses of novel human metabolic functions. BMC Syst Biol 5:155. doi:10.1186/1752-0509-5-155
53. Oberhardt MA, Chavali AK, Papin JA (2009) Flux balance analysis: interrogating genome-scale metabolic networks. Methods Mol Biol 500:61–80. doi:10.1007/978-1-59745-525-1_3
54. Joyce AR, Reed JL, White A, Edwards R, Osterman A, Baba T, Mori H, Lesely SA, Palsson BO, Agarwalla S (2006) Experimental and computational assessment of conditionally essential genes in *Escherichia coli*. J Bacteriol 188(23):8259–8271. doi:10.1128/JB.00740-06
55. Feist AM, Palsson BO (2010) The biomass objective function. Curr Opin Microbiol 13(3):344–349. doi:10.1016/j.mib.2010.03.003
56. Chavali AK, Whittemore JD, Eddy JA, Williams KT, Papin JA (2008) Systems analysis of metabolism in the pathogenic trypanosomatid Leishmania major. Mol Syst Biol 4:177. doi:10.1038/msb.2008.15
57. Orth JD, Palsson BO (2012) Gap-filling analysis of the iJO1366 *Escherichia coli* metabolic network reconstruction for discovery of metabolic functions. BMC Syst Biol 6(1):30. doi:10.1186/1752-0509-6-30
58. Duarte NC, Becker SA, Jamshidi N, Thiele I, Mo ML, Vo TD, Srivas R, Palsson BO (2007) Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proc Natl Acad Sci U S A 104(6):1777–1782. doi:10.1073/pnas.0610772104
59. Yeku O, Frohman MA (2011) Rapid amplification of cDNA ends (RACE). Methods Mol Biol 703:107–122. doi:10.1007/978-1-59745-248-9_8
60. Frohman MA, Dush MK, Martin GR (1988) Rapid production of full-length cDNAs from rare transcripts: amplification using a single gene-specific oligonucleotide primer. Proc Natl Acad Sci U S A 85(23):8998–9002
61. Jones SJ (2006) Prediction of genomic functional elements. Annu Rev Genomics Hum Genet 7:315–338. doi:10.1146/annurev.genom.7.080505.115745

Chapter 5

Resolving Cell Composition Through Simple Measurements, Genome-Scale Modeling, and a Genetic Algorithm

Ryan S. Senger and Hadi Nazem-Bokaee

Abstract

The biochemical composition of a cell is very complex and dynamic. It varies greatly among different organisms and environmental conditions. Inclusion of proper cell composition data is critical for accurate genome-scale metabolic flux modeling using flux balance analysis (FBA). However, determining cell composition experimentally is currently time-consuming and resource intensive. In this chapter, a method for predicting cell composition using a genome-scale model and "easy to measure" culture data (e.g., glucose uptake rate, and specific growth rate) is presented. The method makes use of a genetic algorithm for nonlinear optimization of a biomass equation (a mathematical description of cell composition). As a case study, the method was used to optimize a biomass equation for *Escherichia coli* MG1655 under multiple growth environments. The availability of experimentally determined ^{13}C flux data allowed a direct comparison with FBA predicted fluxes through the TCA cycle. Results showed dramatic improvement upon optimization of the biomass equation. In a second case study, biomass equation optimization was also applied to *Clostridium acetobutylicum*, an organism with less available biochemical cell composition data in the literature. The method produced a biomass equation highly similar to one determined experimentally for the closely related Gram-positive *Bacillus subtilis*.

Key words: Cell composition, Biomass equation, Genome-scale modeling, Genetic algorithm, Cell dynamics, Systems metabolic engineering

1. Introduction

For more than a decade now, genome-scale metabolic flux modeling has been developed to provide the critical link between the genotype of an organism and its expressed phenotype (1–4). These models serve numerous purposes from biological discovery (5–7) to designing productive microbial strains for biotechnology (8–10). A genome-scale model consists of a metabolic network that was reconstructed from its genome annotation and a mathematical description of the biochemical cell composition (called a "biomass equation"). Flux balance analysis (FBA) is one of several methods commonly used to determine how metabolic flux (bulk metabolite

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_5,

flow between enzymes) is partitioned throughout a metabolic network under different conditions. With FBA, a steady-state assumption is invoked, making the metabolic network solvable by linear programming (11, 12). In this case, an objective function (e.g., the specific growth rate) is specified, and experimental measurements (e.g., the glucose uptake rate) are used to constrain (artificially set) metabolite fluxes where possible.

The specific growth rate of the organism is equal to the flux through the "biomass equation" included in the genome-scale model. The biomass equation includes all macromolecules and solutes that comprise 1 g of dry cell weight (gDCW). The stoichiometry of the biomass equation has been found to significantly influence FBA results (13). The biomass equation often contains stoichiometric amounts of (1) amino acids used to form protein, (2) DNA, (3) RNA, (4) lipids, (5) carbon storage compounds, (6) cell wall and membrane components, and (among others) (7) free solute pools of the cytoplasm (14–16). The biomass equation also accounts for ATP demands of cellular processes not specifically included in the model. This is referred to as maintenance ATP (mATP). Early metabolic models recognized a need for a changing biomass equation that is dependent on the growth state of the *Escherichia coli* cell (17, 18), and this need was further demonstrated upon modeling different growth states of the butanol-producer *Clostridium acetobutylicum* ATCC 824 (13, 16, 19).

The method presented in this chapter enables prediction of cell composition through the use of a genetic algorithm and returns an optimized biomass equation for a given environment. This method is valuable to researchers because obtaining a full biochemical characterization of cell composition is difficult, time consuming, and expensive. We refer to these as "difficult measurements." The genetic algorithm method requires inputs of measurements obtainable by standard chromatography methods or enzymatic assays, such as the glucose uptake rate and major byproduct secretion rates by the culture. These are referred to as "easy measurements." Using the methods presented in this chapter, researchers will be able to predict (1) changes in biomass composition resulting from changes in the culturing environment and (2) cell composition of "lesser-studied" organisms that do not have a wealth of biochemical data available in the literature. Case studies are presented for *E. coli* MG1655 and *C. acetobutylicum* ATCC 824.

2. Materials

2.1. Models and a Platform for FBA

Numerous genome-scale models have been published since the initial models of *Haemophilus influenzae* (20) and *E. coli* MG1655 (21). Often these models are available as supplementary

appendices to their journal publication in either tab-delimited text format or in Systems Biology Markup Language (SBML) (22) format. As genome-scale modeling has increased in popularity, central repositories for models have been developed (23–25). In this research, we have chosen to download and work with the iJR904 model for *E. coli* MG1655 (26). Several methods exist for performing FBA calculations on a genome-scale model of choice. The constraint-based reconstruction and analysis (COBRA) Toolbox (v2.0) (27) was used with the open-source GLPK linear programming solver (28) inside the MATLAB computing environment (see Note 1). FBA was performed with multiple objective functions of (1) maximizing the specific growth rate of the cell and (2) minimizing the total metabolic flux (see Note 2). The iJR904 model included a complete set of reaction constraints and metabolite exchange reactions. The reaction constraints define the upper and lower allowable bounds (or limits) of flux through each reaction. The exchange reactions define the rate at which metabolites can enter and leave the system. Exchange reactions can serve multiple important functions. First, metabolite uptake and secretion can easily be constrained by manipulating the bounds of an exchange reaction. For example, if multiple mechanisms of glucose uptake are present, constraining the bounds of the glucose exchange reaction will ensure the total rate of glucose uptake (through multiple mechanisms) is set to a defined value. Second, exchange reactions can import/export metabolites required of a metabolic network to complete the global mass balance. The biomass equation included with the iJR904 model consists of stoichiometric amounts of 53 metabolites. These were grouped according to the following categories for optimization: (1) maintenance ATP (mATP), (2) glycogen, (3) amino acids, (4) lipids, (5) RNA, (6) DNA, and (7) peptidoglycan. Solute pools in the iJR904 model (e.g., NAD, K^+, and phosphate) were held constant.

2.2. The Fitness Function

The purpose of the method described in this chapter is predict "difficult to measure" cell composition data using "easy to measure" metabolite data (e.g., glucose uptake rate). This requires the use of a genetic algorithm (for nonlinear optimization) to adjust the stoichiometry of the biomass equation according to an objective (or fitness) function (see Note 3). In this approach, experimental data was used to determine the rates of all metabolites being transported in/out of the cell. Thus, secretion of additional metabolites (e.g., pyruvate) through exchange reactions should be negligible. The genetic algorithm adjusts the stoichiometry of the biomass equation to allow observed (1) influx of substrates (e.g., glucose, O_2), (2) efflux of metabolic byproducts (e.g., acetate, CO_2), and (3) growth given minimized flux of additional metabolites through exchange reactions. The necessity of the model to secrete additional metabolites through exchange reactions is

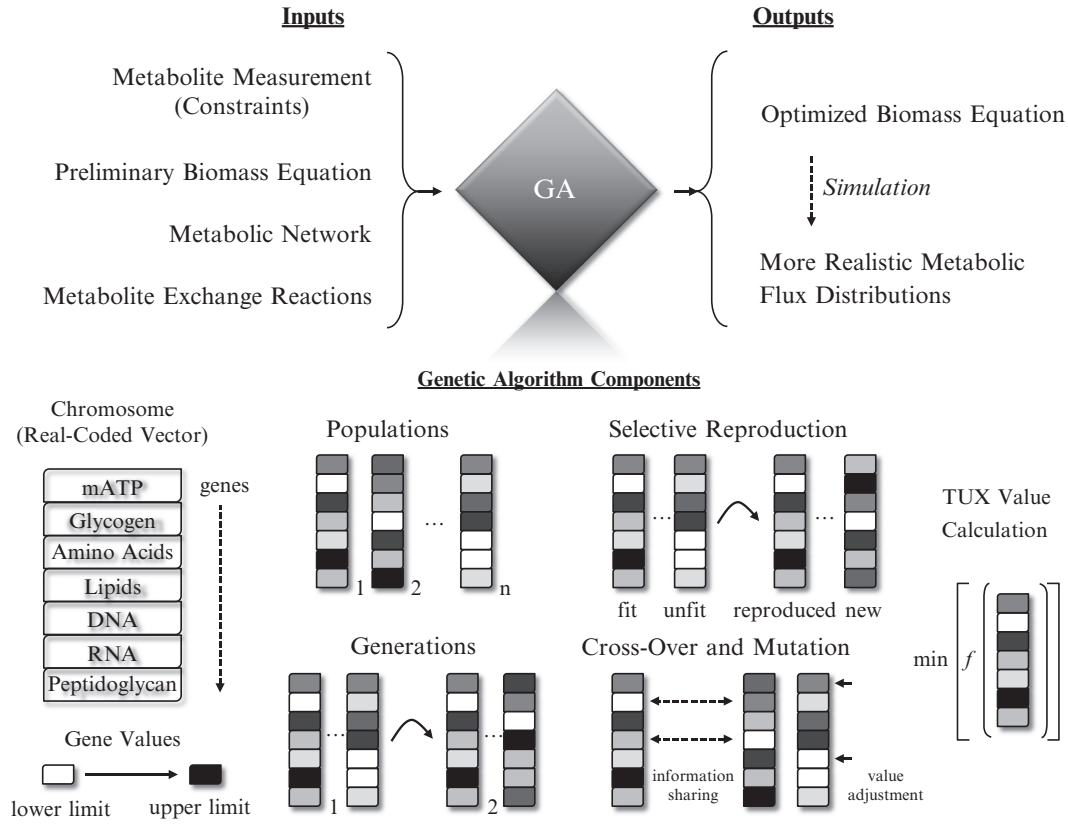

Fig. 1. Biomass equation optimization using a genetic algorithm (GA).

referred to as the *t*otal *u*nconstrained e*x*change (TUX) fluxes (see Note 4). A more "fit" biomass equation is associated with a lower TUX value.

2.3. The Genetic Algorithm

A real-coded genetic algorithm (29) was developed to perform the nonlinear optimization of updating the biomass equation stoichiometry. All calculations were performed in the MATLAB environment. Numerous combinations of "population sampling" algorithms with "mutation and crossover" operators are possible. Based on our preliminary calculations (not shown), we have developed a genetic algorithm that leads to rapid and reproducible convergence when optimizing the stoichiometry of the biomass equation. The overall setup for the genetic algorithm is shown in Fig. 1. The overall goal is to provide the algorithm with (1) metabolite flux measurements that can be used as constraints (e.g., glucose uptake rate), (2) a metabolic network with a biomass equation (e.g., iJR904), and (3) metabolite exchange reactions (included in iJR904). The output is an optimized biomass equation that will lead to more realistic flux distributions through the metabolic network when performing FBA. The genetic algorithm consists

of a real-coded chromosome (vector) that contains the biomass equation stoichiometric coefficients: (1) mATP, (2) glycogen, (3) amino acids, (4) lipids, (5) DNA, (6) RNA, and (7) peptidoglycan. Each stoichiometric coefficient is adjusted between set lower and upper limits. Initially 30 chromosomes are created using random numbers. The set of 30 chromosomes is referred to as a "population." The first population is also the first "generation." In the genetic algorithm, all chromosomes are used to evaluate the objective function (i.e., the flux through additional exchange reactions needed to complete the mass balances). Chromosomes minimizing the TUX value are deemed more "fit" and have a greater chance of being reproduced than less-fit chromosomes. The next generation of chromosomes is produced by (1) reproducing fit chromosomes of the previous generation (10%), (2) subjecting those chromosomes to BLX-α crossover (30) to generate new chromosomes (30%), (3) applying nonuniform random mutations to fit chromosomes to produce new ones (30%), and (4) randomly generating new chromosomes (30%). This study found that about 50 generations are necessary for convergence of a new optimized biomass equation. The parameter α was set equal to 0.35 in the BLX-α crossover operator. This operator works by choosing a random number of the interval $[c_{\min} - I \times \alpha,\ c_{\max} + I \times \alpha]$, where $c_{\min}$ is the minimal gene value of the two selected for crossover and $c_{\max}$ is the maximal value. The parameter I is equal to $c_{\max} - c_{\min}$. Crossovers were allowed to occur between 1 and 3 genes in the chromosome. The nonuniform mutation operator (31) requires the generation of a binary variable τ. The following shows how to calculate the new value of a gene, $c_i^{'}$, from its original value, c_i (29):

$$c_i^{'} = \begin{cases} c_i + \Delta(t, u_i - c_i) \ \textit{if} \ \tau = 0 \\ c_i - \Delta(t, c_i - l_i) \ \textit{if} \ \tau = 1 \end{cases} \tag{1}$$

Here, u_i is the upper limit of gene c_i and l_i is its lower limit. The value of y is defined as $u_i - c_i$ or $c_i - l_i$ depending on whether τ is 1 or 0. This enables calculation of the following:

$$\Delta(t, y) = y\left(1 - r^{\left(1 - \frac{t}{gmax}\right)^{b}}\right) \tag{2}$$

where r is a random number from the interval $[1, 0]$, t is the generation, $gmax$ is the total number of generations, and b is a parameter chosen by the user. We used a value of 1 for b. The nonuniform mutation operator enables large mutations in earlier generations for "searching" the solution space and smaller mutations in later generations for "fine-tuning" solutions (see Note 5).

2.4. Literature Data

The metabolic flux distributions obtained with FBA and an optimized biomass equation were compared to experimentally measured ^{13}C flux data. We located two significant datasets for this purpose (32, 33). The former dataset (33) has been modeled previously using FBA (17), so we modeled this dataset again with the iJR904 model and the biomass equation optimization algorithm to determine how closely FBA results match ^{13}C flux results through the TCA cycle. Three experimental conditions were examined: (1) aerobic growth on glucose and acetate, (2) aerobic growth on acetate only, and (3) anaerobic growth on glucose only.

3. Methods

3.1. Simulation of the iJR904 Model

The Set (I) constraints of Table 1 were applied to the iJR904 model (see Note 6) and FBA was applied (see Note 2). The data was visualized in a spreadsheet (see Note 7). Results showed metabolite uptake/secretion rates and a specific growth rate identical to those

Table 1
List of all constrained metabolite uptake/secretion fluxes and growth rates used in simulations

Constrained metabolite flux[a] (uptake/secretion) (mmol/gDCW/h)	Glucose and acetate uptake (aerobic) Constraints set (I)	Acetate uptake (aerobic) Constraints set (II)	Glucose uptake (anaerobic) Constraints set (III)
Glucose	7	0	21
CO_2	−14	−45	−9.67[b]
O_2	12.6[c]	42[b]	0
Acetate[d]	0.8	33.42	−7.67
Lactate	0	0	−16.69
Formate	0	0	−0.5
Succinate	0	0	−2.25
Ethanol	0	0	−10.46
Specific growth rate (h^{-1})	0.59	0.29	0.29

[a]Positive flux values correspond to metabolite uptake. Negative values represent secretion. All other constraints of the iJR904 model were held constant at their published values
[b]These values were updated from published values based on simulation results
[c]The value from a previous simulation result (17) was used here
[d]The acetate flux listed is the net of uptake and secretion

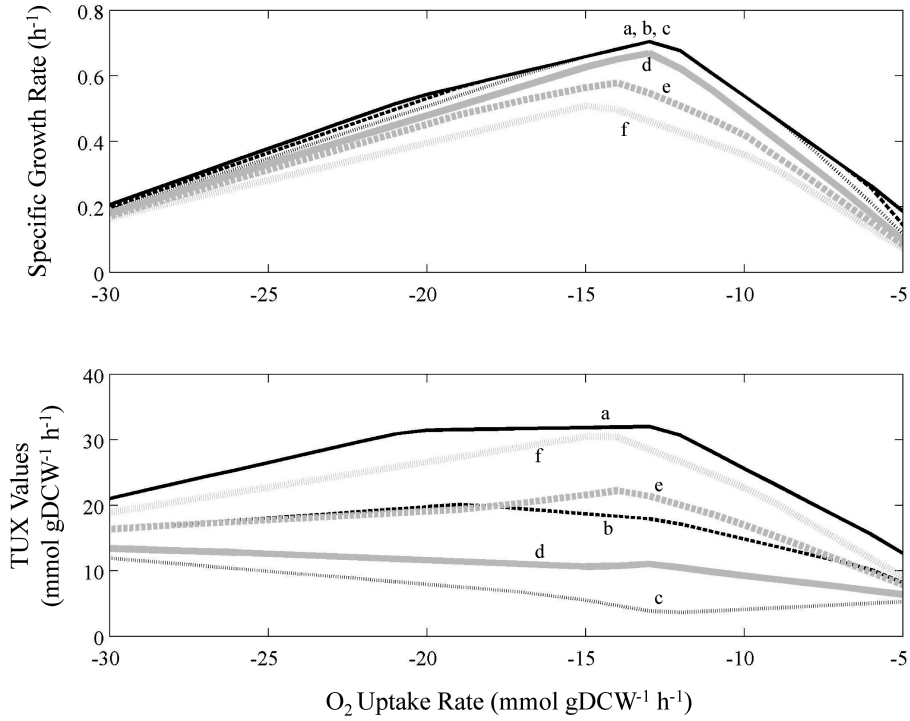

Fig. 2. The influence of maintenance ATP (mATP) with Set (I) constraints. *Top*: the relationship between the specific growth rate and oxygen uptake rate. *Bottom*: the relationship between the TUX values and the oxygen uptake rate. The following stoichiometric coefficients of mATP in the biomass equation were simulated: (a) 0, (b) 20, (c) 40, (d) 60, (e) 80, and (f) 100.

listed in Table 1. However, significant secretion of fumarate and D-gluconate were required by the model to achieve these constrained fluxes. This resulted in a significant TUX value.

3.2. Influence of mATP

The sensitivity of the biomass equation was probed by first varying the stoichiometry of mATP (see Note 8) from the published value of 45.561. The following values were installed and simulated: (1) 0, (2) 20, (3) 40, (4) 60, (5) 80, and (6) 100. All other constraints were set to the Set (I) constraints in Table 1. Results of the relationship between the specific growth rate and the O_2 uptake rate are shown in Fig. 2. Changing the mATP stoichiometry had little influence on the relationship between the specific growth and O_2 uptake rates. However, significant differences were observed for the TUX values. With an mATP value of 40 (close to the published value in iJR904), the TUX value was minimized. This method demonstrates that altering the biomass equation stoichiometry can have profound impacts on the use of exchange equations necessary to satisfy the global mass balance of the cell given Set (I) constraints.

Table 2
Reproducibility of genetic algorithm convergence using Set (I) constraints when allowing 50% variability from the iJR904 biomass equation

Iteration[a] (variation)	mATP[b]	Glycogen	Amino acids	Lipid	DNA	RNA	Peptidoglycan	TUX[c]
1 (50%)	44.463	0.231[d]	5.874	0.0124	0.0507	0.727	0.0373	0.672
2 (50%)	44.013	0.207	6.142	0.0076	0.0811	0.856	0.0412	0.539
3 (50%)	44.589	0.231	5.953	0.0112	0.0501[d]	0.694	0.0414[d]	0.718
4 (50%)	45.067	0.230	5.934	0.0137[d]	0.0501[d]	0.529	0.0414[d]	0.807
5 (50%)	43.621	0.231[d]	5.295	0.0137[d]	0.127	0.927	0.0414[d]	0.594
Average	44.350	0.226	5.840	0.0117	0.0718	0.746	0.0405	0.667
St. Dev.	0.554	0.0109	0.321	0.00250	0.03336	0.154	0.00181	0.105
iJR904	45.561	0.154	5.081	0.0091	0.1002	0.636	0.0276	2.064

[a]The genetic algorithm was run for 50 generations with 30 chromosomes per generation
[b]The maintenance ATP required of the biomass equation is represented as mATP
[c]The TUX was minimized by the genetic algorithm
[d]These values were bound by the 50% limit during the optimization

3.3. Biomass Equation Optimization Using the Genetic Algorithm

The stoichiometric coefficients of the biomass equation were allowed to vary by 50% of the values provided in iJR904. The genetic algorithm was applied to adjust these stoichiometric coefficients in order to minimize the TUX value. The genetic algorithm consisted of a 30 chromosome population and was run for 50 generations. The TUX value of the iJR904 model with the published biomass equation was 2.064. When the biomass equation was optimized, the TUX value dropped to 0.667. Results are shown in Table 2 for five independent iterations of the optimization. When compared to the iJR904 biomass equation, the stoichiometric amounts of glycogen, amino acids, and peptidoglycan increased significantly. The stoichiometry of mATP, lipid, and RNA remained largely unchanged. Several values were bound by the 50% variance limitation, suggesting that additional loosening of constraints could provide even better results. The percent variation was altered between 10 and 100% and multiple optimizations were performed. The results are shown in Fig. 3. As the constraints on the biomass equation stoichiometry were relaxed, the averaged TUX value decreased, suggesting more fit solutions were found. However, the standard deviation among replicates also increased, suggesting that more diverse solutions were found.

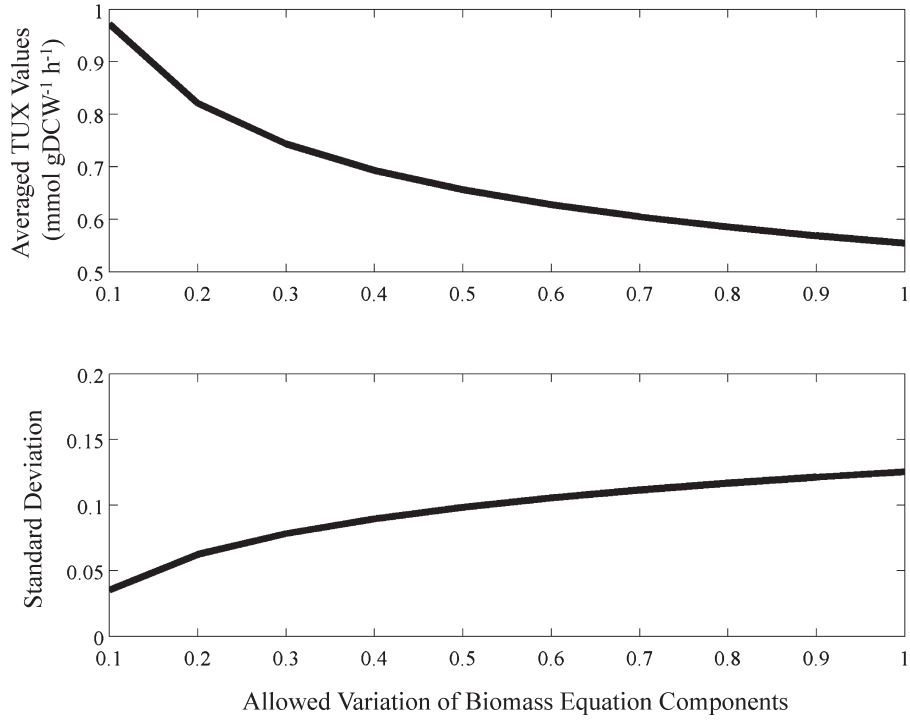

Fig. 3. The influence of limits on biomass equation convergence. The fractional variation allowed is shown (i.e., 0.1 is 10%). *Top*: the averaged TUX values of five independent optimization runs. *Bottom*: the standard deviation of the five independent optimization runs.

3.4. Multiple Environmental Conditions

A significant advantage of the algorithmic approach presented in this chapter is the ability to obtain biomass equations for multiple environments without the need to obtain "difficult measurements." The following example is presented to illustrate the method and show what results can be obtained. The iJR904 biomass equation was optimized for the three sets of constraints listed in Table 1. Results are shown in Table 3. In each case, biomass equation optimization using the genetic algorithm resulted in a greatly improved TUX value. Several important differences were observed for the resulting biomass equation in the three different environmental conditions. As the culture was restricted to growing on acetate aerobically, the cell composition was enriched with amino acids, suggesting greater protein production. This was also the case for anaerobic culture growth. In all optimized cases, the accumulation of glycogen was greater than the initial value published in the iJR904 model. This is also true for lipids and peptidoglycan. The TUX value was minimized more effectively for Set (I) constraints (i.e., aerobic growth on glucose and acetate). This is consistent with the fact that most literature data used to compile the initial draft of the biomass equation was obtained from aerobically growing cultures with a glucose substrate. This also means that additional variations of biomass equations optimized given Set (II)

Table 3
Optimized biomass equations for iJR904

Biomass equation[a]	mATP[b]	Glycogen	Amino acids	Lipid	DNA	RNA	Peptidoglycan	TUX[c]
iJR904 Set (I)[d]	45.561	0.154	5.081	0.0091	0.100	0.636	0.0276	2.064
Optimized Set (I)	44.350	0.226	5.840	0.0117	0.0718	0.746	0.0405	0.666
iJR904 Set (II)[d]	45.561	0.154	5.081	0.0091	0.100	0.636	0.0276	2.057
Optimized Set (II)	45.091	0.219	7.622[e]	0.0137	0.0996	0.470	0.0414[e]	1.284
iJR904 Set (III)[d]	45.561	0.154	5.081	0.0091	0.100	0.636	0.0276	3.321
Optimized Set (III)	43.782	0.231	7.622[e]	0.0132	0.138	0.852	0.0414[e]	1.437

[a]The genetic algorithm was run for 50 generations with 30 chromosomes per generation
[b]The maintenance ATP required of the biomass equation is represented as mATP
[c]The TUX was minimized by the genetic algorithm
[d]The originally published biomass equation with the set of constraints specified
[e]These values were bound by the 50% limit during the optimization

and Set (III) constraints are likely. Further improved biomass equations for these environmental conditions may be obtained by further loosening the limits during genetic algorithm optimization.

3.5. Comparisons Between FBA and ^{13}C Fluxes

This section describes the methods used to compare FBA and ^{13}C fluxes and provides an account of the case study performed with *E. coli* MG1655. Multiple attempts have been made to reconcile fluxes obtained from FBA with experimental flux measurements obtained using ^{13}C tracers. Some agreement has been noted (32). Our hypothesis is that an optimized biomass equation will lead to flux distributions that are more consistent with experimental measurements. To test this hypothesis, we obtained previously published (17, 33) experimentally determined flux data from the TCA cycle that correspond to the Set (I) constraints listed in Table 4. Experimentally measured fluxes and those determined by FBA are shown in Table 3. The FBA fluxes were determined for the iJR904 model with the published biomass equation first. To measure the overall correlation between ^{13}C tracer and FBA determined fluxes, the sum-squared error was calculated for all enzymes in the TCA cycle. The iJR904 published biomass equation yielded a sum-squared error value of 58.470. The FBA results made use of the glyoxylate shunt (i.e., the malate synthase and isocitrate lyase); whereas, this was not observed experimentally. Other notable

Table 4
TCA cycle modeling results using Set (I) constraints

Enzyme	Measured flux[a]	iJR904 Set (I)[b]	Optimized Set (I)[c]	Optimized Set (I*)[c,d]
Pyruvate dehydrogenase	5	8.150	6.823	6.122
Citrate synthase	5	4.532	4.128	4.329
Aconitase	5	4.532	4.128	4.329
Isocitrate dehydrogenase	5	1.993	2.630	3.318
α-Ketoglutarate dehydrogenase	3.9	1.488	2.026	2.978
Succinyl-CoA synthase	3.9	1.193	1.712	2.786
Succinate dehydrogenase	3.9	4.028	3.524	3.989
Fumarase	3.9	3.377	4.499	4.430
Malate dehydrogenase	3.9	5.945	5.947	5.470
Phosphoenolpyruvate carboxylase	2.9	0	0	0
Phosphoenolpyruvate carboxykinase	0	0	0	0
Malate synthase	0	2.569	1.473	1.041
Isocitrate lyase	0	2.539	1.444	1.011
Sum-squared error		58.470	36.155	20.352

[a]Previously published experimental data (33) was used and republished and modeled later (17)
[b]Flux results obtained using the previously published biomass equation
[c]The TUX was minimized by the genetic algorithm
[d]In this simulation, the oxygen uptake rate was constrained to 20 mmol/gDCW/h

differences include a high flux through the pyruvate dehydrogenase predicted by FBA and low flux values through the isocitrate dehydrogenase, α-ketoglutarate dehydrogenase, and succinyl-CoA synthase. When the optimized biomass equation was used, the sum-squared error was reduced to 36.155 (a 38% improvement). The use of the glyoxylate pathway was still present, but fluxes were reduced by 42%. Flux through the pyruvate dehydrogenase was reduced, and fluxes through the isocitrate dehydrogenase, α-ketoglutarate dehydrogenase, and succinyl-CoA synthase increased (but not to experimentally measured levels). To illustrate the impact of gas phase measurements, the oxygen uptake rate in the Set (I) constraints was increased to 20 mmol /gDCW /h (while leaving

everything else constant). The sum-squared error was further reduced to 20.352 (a 65% improvement), and flux through the glyoxylate shunt was reduced by 60% relative to the iJR904 published biomass equation. The same TCA cycle analysis was carried out for the Set (II) constraints (results not shown). In this case, the optimized biomass equation resulted in a 27% improvement in the sum-squared error value. Flux through the glyoxylate shunt was observed experimentally and predicted with good accuracy by FBA (within 3% for the optimized biomass equation). The major discrepancies were marked by low FBA flux predictions through the citrate synthase, aconitase, isocitrate dehydrogenase, and α-ketoglutarate dehydrogenase. The Set (III) constraints of Table 1 were also simulated and compared to experimental measurements (results not shown). The FBA results of both the iJR904 biomass equation and the optimized biomass equation correctly predicted the bifurcated TCA cycle resulting in succinate secretion and extensive use of the phosphoenolpyruvate carboxykinase. While the optimized biomass equation did not result in FBA fluxes that exactly matched experimental results from ^{13}C tracers, significant improvements were observed. These results were obtained by allowing 50% variation of biomass equation stoichiometry from the original iJR904 published biomass equation. As shown in the next case study, sometimes much larger variations are required.

3.6. Obtaining a Biomass Equation for a "Lesser-Studied" Organism

A wealth of literature data on cell composition exists for *E. coli*. This enabled the construction of such a comprehensive and highly regarded biomass equation published with the iJR904 model (26). However, what happens for organisms that are "lesser-studied" in the literature? Cell composition data is not readily available for these organisms. This happens to be the case for >99% of biodiversity. The following case study illustrates how to derive an optimized biomass equation for a "lesser-studied" organism. A genome-scale model and biomass equation were prepared for the lesser-studied Gram-positive butanol-producer *C. acetobutylicum* (13, 16, 19). Since adequate cell composition data was not available, the biomass equation for the Gram-positive *Staphylococcus aureus* N315 (15) was adopted and updated with data specific for *C. acetobutylicum* where possible (16). The resulting biomass equation was "fit" to experimental data (e.g., glucose uptake, and butanol secretion) by adjusting the mATP stoichiometry (see Note 9). This led to a value of 410 (16). The experimentally determined value is ~20 (34). The biomass equation was subjected to optimization using the genetic algorithm approach, and results are shown in Fig. 4. The optimization massively altered the biomass equation, leading to (1) a mATP stoichiometry of 35, (2) a significantly higher protein (i.e., amino acids) fraction, and (3) a much lower cell wall fraction. A genome-scale model for the "well-studied" and closely related Gram-positive *Bacillus subtilis* was published and contained an experimentally

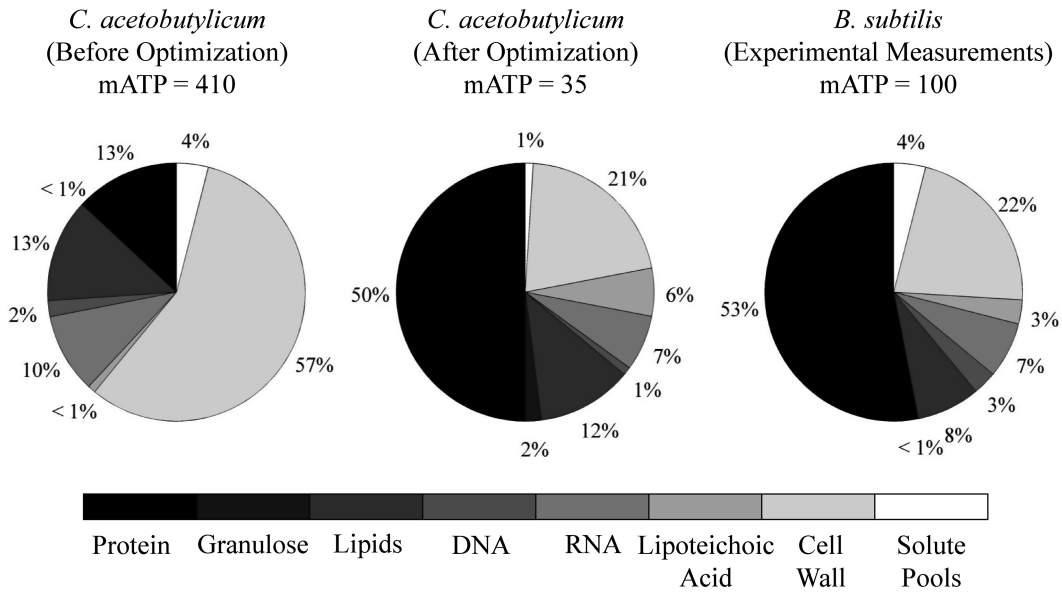

Fig. 4. Biomass equation optimization for *Clostridium acetobutylicum*. *Left*: the published biomass equation (16) derived from limited literature data (before optimization). *Middle*: the optimized biomass equation for *C. acetobutylicum*. *Right*: the biomass equation for the closely related Gram-positive *Bacillus subtilis* (35) derived from extensive experimental measurements. All percentages are given in terms of weight.

determined biomass equation (35). This is also shown in Fig. 4. The optimized biomass equation for *C. acetobutylicum* highly resembled that of published *B. subtilis* model. This technique shows that the genetic algorithm optimization approach can lead to effective biomass equations for "lesser-studied" organisms, even when the initial guess for the biomass equation is not a good fit for the genome-scale model (see Note 10).

4. Notes

1. Installation of COBRA, GLPK, and SBML.

   The COBRA Toolbox (v2.0) (27) has enabled a simplified installation for use with MATLAB. All required downloads can now be accessed from The openCOBRA Project (36). The COBRA Toolbox (v2.0) download now includes the GLPK open-source linear programming package embedded. Users simply add the COBRA Toolbox to the "MATLAB/toolbox" folder and add to the MATLAB path. In MATLAB, the command "initCobraToolbox" must be executed each time MATLAB is started. It is recommended to add this command to the MATLAB "startup.m" file. SBML functionality is now

embedded in the COBRA Toolbox (v2.0) with the installation of LibSBML (37). SBML files are easily imported into MATLAB through the use of the "translateSBML()" function.

2. How to run FBA.
   Upon installing the COBRA Toolbox (v2.0) in MATLAB and loading an SBML model, FBA is carried out by the following. The objective function is specified in the column vector called "c." All rows of "c" correspond to reactions listed in the "rxns" cell column vector. A reaction to be maximized (e.g., the biomass equation to be maximized) is set equal to 1. In the work presented in this chapter, the total flux of the system was also minimized. To do this, the command "FBAsolution = optimizeCbModel(model,(),'one')" was executed. Here, the variable "model" is the genome-scale model data structure imported from the SBML file. The resulting variable "FBAsolution" contains a variable "f" that holds the value of the optimized objective function and a variable "x" that contains flux values for every reaction in the metabolic network.
3. The objective function and the fitness function.
   For clarity, the "objective function" refers to maximizing cell growth using linear programming by GLPK, and the "fitness function" refers to optimizing the biomass equation using the genetic algorithm.
4. The total unconstrained exchange (TUX) fluxes.
   A difficult mass balance is associated with metabolism. Coming into the cell are substrates and reducing agents. Leaving the cell are oxidized products and new biomass. Experimental results may show a limited number of inputs (i.e., substrates) yield a limited number of outputs (e.g., acetate and CO_2) and new cells (of a defined cell composition). If the wrong biomass equation is used in FBA, additional products will be secreted (or imported) to satisfy the mass balance given the fluxes of inputs/outputs observed experimentally. Of course, these additional products were not observed experimentally and should be absent from the modeling results. We refer to the total flux of all additional products (or substrates) required as the TUX. This value can be calculated directly from FBA results by specifying which exchange reactions are unbound. In the iJR904 model, the bounds of the exchange reactions allow export but not import of necessary metabolites. A total of 143 exchange reactions are present in the iJR904 model and can be recognized by their "subSystem" field, which is classified as "Exchange."
5. Making sure operators do not cross limits.
   The limits of variables are imposed early in the genetic algorithm. Every time operators are applied, care must be taken to

make sure the limits are not surpassed. If an operator adjusts the value of a gene past its limit, the value was artificially set to the limit value.

6. How to apply the constraints in Table 1.
   The constraints of metabolite flux were applied to the exchange reactions of that component. Constraints are applied in the "lb" (lower bound) and "ub" (upper bound) column vector variables in the model data structure. Each row of "lb" and "ub" corresponds to the reaction listed in "rxns." For exchange reactions, fluxes leaving the cell (or system) are positive, and fluxes entering are negative. For example, to specify a glucose uptake rate of 7 mmol $gDCW^{-1}$ h^{-1}, the values of both "lb" and "ub" were amended to −7 in the row corresponding to the glucose exchange reaction (row 344 in the iJR904 model). The specific growth rate was set by adjusting the constraints of the biomass equation. This left minimization of total flux as the sole objective function for linear programming optimization. It is noted that the nitrogen source was left unconstrained in simulations due to a lack of experimental data.
7. Visualizing model outputs.
   Viewing FBA flux results in MATLAB is difficult. Often, we want to view flux results globally. To do this, we export the reactions, reaction names, bounds, and subsystems to an Excel spreadsheet. When FBA is performed, the solution vector "x" can easily be copied/pasted into the spreadsheet so that fluxes can be viewed with reactions and subsystems easily. The "vlookup" command in Excel is especially handy for retrieving fluxes for specific reactions of interest. Several other visualization tools exist as well.
8. How to vary the mATP stoichiometry.
   The stoichiometry of the biomass equation was varied by altering the stoichiometric matrix "S" directly. This is done by first finding the row of the biomass equation (i.e., 150 in iJR904) in the variable "rxns" and the metabolite to be altered in the variable "mets" (i.e., 200 for ATP in iJR904). Then the stoichiometric matrix is altered at the row of the metabolite and column of the reaction (i.e., S(200,150)) to the new value. It is important to note that reactants are given negative values in the stoichiometric matrix and products are given positive values.
9. Adjusting mATP in *C. acetobutylicum*.
   The method of fitting the biomass equation using mATP was applied in 2008 (19). Today, it is recommended to use the TUX value approach instead.
10. Setting effective biomass equation stoichiometry limits.
    Defining the limits for adjusting the stoichiometry of the biomass equation is still a topic of research. In this study, a

value of 50% variance was chosen because the iJR904 model is so highly regarded and considerable effort was devoted to compiling the biomass equation from abundant literature data. However, for the *C. acetobutylicum* biomass equation, wide limits were necessary. This produced several results that diverged but still returned good optimization scores. At this point, the user must be aware of which results are reasonable and which are not when reviewing potentially optimized biomass equations. Comparison to a similar organism with published cell composition data is currently the best way to gauge "reasonable" from "unreasonable" solutions.

References

1. Edwards JS, Ibarra RU, Palsson BO (2001) In silico predictions of *Escherichia coli* metabolic capabilities are consistent with experimental data. Nat Biotechnol 19(2):125–130. doi:10.1038/84379
2. Feist AM, Palsson BO (2008) The growing scope of applications of genome-scale metabolic reconstructions using *Escherichia coli*. Nat Biotechnol 26(6):659–667. doi:10.1038/nbt1401
3. Milne CB, Kim PJ, Eddy JA, Price ND (2009) Accomplishments in genome-scale in silico modeling for industrial and medical biotechnology. Biotechnol J 4(12):1653–1670. doi:10.1002/biot.200900234
4. Price ND, Reed JL, Palsson BO (2004) Genome-scale models of microbial cells: evaluating the consequences of constraints. Nat Rev Microbiol 2(11):886–897. doi:10.1038/nrmicro1023
5. Archer CT, Kim JF, Jeong H, Park JH, Vickers CE, Lee SY, Nielsen LK (2011) The genome sequence of *E. coli* W (ATCC 9637): comparative genome analysis and an improved genome-scale reconstruction of *E. coli*. BMC Genomics 12:9. doi:10.1186/1471-2164-12-9
6. Chandrasekaran S, Price ND (2010) Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in *Escherichia coli* and *Mycobacterium tuberculosis*. Proc Natl Acad Sci U S A 107(41):17845–17850. doi:10.1073/pnas.1005139107
7. Kim TY, Kim HU, Lee SY (2010) Metabolite-centric approaches for the discovery of antibacterials using genome-scale metabolic networks. Metab Eng 12(2):105–111. doi:10.1016/j.ymben.2009.05.004
8. Burgard AP, Pharkya P, Maranas CD (2003) Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. Biotechnol Bioeng 84(6):647–657. doi:10.1002/bit.10803
9. Fong SS, Burgard AP, Herring CD, Knight EM, Blattner FR, Maranas CD, Palsson BO (2005) In silico design and adaptive evolution of *Escherichia coli* for production of lactic acid. Biotechnol Bioeng 91(5):643–648. doi:10.1002/bit.20542
10. Ranganathan S, Suthers PF, Maranas CD (2010) OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. PLoS Comput Biol 6(4):e1000744. doi:10.1371/journal.pcbi.1000744
11. Orth JD, Thiele I, Palsson BO (2010) What is flux balance analysis? Nat Biotechnol 28(3):245–248. doi:10.1038/nbt.1614
12. Reed JL, Palsson BO (2004) Genome-scale in silico models of *E. coli* have multiple equivalent phenotypic states: assessment of correlated reaction subsets that comprise network states. Genome Res 14(9):1797–1805. doi:10.1101/gr.2546004
13. Senger RS (2010) Biofuel production improvement with genome-scale models: the role of cell composition. Biotechnol J 5(7):671–685. doi:10.1002/biot.201000007
14. Feist AM, Herrgard MJ, Thiele I, Reed JL, Palsson BO (2009) Reconstruction of biochemical networks in microorganisms. Nat Rev Microbiol 7(2):129–143. doi:10.1038/nrmicro1949
15. Heinemann M, Kummel A, Ruinatscha R, Panke S (2005) In silico genome-scale reconstruction and validation of the *Staphylococcus aureus* metabolic network. Biotechnol Bioeng 92(7):850–864. doi:10.1002/bit.20663
16. Senger RS, Papoutsakis ET (2008) Genome-scale model for *Clostridium acetobutylicum*:

Part I. Metabolic network resolution and analysis. Biotechnol Bioeng 101(5):1036–1052. doi:10.1002/bit.22010

17. Pramanik J, Keasling JD (1997) Stoichiometric model of *Escherichia coli* metabolism: incorporation of growth-rate dependent biomass composition and mechanistic energy requirements. Biotechnol Bioeng 56(4):398–421. doi:10.1002/(SICI)1097-0290(19971120)56:4<398::AID-BIT6>3.0.CO;2-J
18. Pramanik J, Keasling JD (1998) Effect of *Escherichia coli* biomass composition on central metabolic fluxes predicted by a stoichiometric model. Biotechnol Bioeng 60 (2):230–238
19. Senger RS, Papoutsakis ET (2008) Genome-scale model for *Clostridium acetobutylicum*: Part II. Development of specific proton flux states and numerically determined sub-systems. Biotechnol Bioeng 101(5):1053–1071. doi:10.1002/bit.22009
20. Edwards JS, Palsson BO (1999) Systems properties of the *Haemophilus influenzae* Rd metabolic genotype. J Biol Chem 274 (25):17410–17416
21. Edwards JS, Palsson BO (2000) The *Escherichia coli* MG1655 in silico metabolic genotype: its definition, characteristics, and capabilities. Proc Natl Acad Sci U S A 97 (10):5528–5533
22. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novere N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19 (4):524–531
23. Henry CS, DeJongh M, Best AA, Frybarger PM, Linsay B, Stevens RL (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. Nat Biotechnol 28(9):977–982. doi:10.1038/nbt.1672
24. Kumar A, Suthers PF, Maranas CD (2012) MetRxn: a knowledgebase of metabolites and reactions spanning metabolic models and databases. BMC Bioinformatics 13(1):6. doi:10.1186/1471-2105-13-6
25. Systems Biology Research Group http://gcrg.ucsd.edu/InSilicoOrganisms/OtherOrganisms. Accessed on 4/1/2012
26. Reed JL, Vo TD, Schilling CH, Palsson BO (2003) An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR). Genome Biol 4(9):R54. doi:10.1186/gb-2003-4-9-r54
27. Schellenberger J, Que R, Fleming RM, Thiele I, Orth JD, Feist AM, Zielinski DC, Bordbar A, Lewis NE, Rahmanian S, Kang J, Hyduke DR, Palsson BO (2011) Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nat Protoc 6 (9):1290–1307. doi:10.1038/nprot.2011.308
28. GLPK http://www.gnu.org/software/glpk/
29. Herrera F, Lozano M, Verdegay JL (1998) Tackling real-coded genetic algorithms: operators and tools for behavioural analysis. Artif Intell Rev 12(4):265–319
30. Eshelmann LJ, Schaffer JD (1993) Real-coded genetic algorithms and interval-schemata. In: Whitely LD (ed) Foundations of genetic algorithms 2. Morgan Kaufmann, San Mateo, pp 187–202
31. Michalewicz Z (1992) Genetic algorithms + data structures = evolution programs. Springer-Verlag, New York
32. Chen X, Alonso AP, Allen DK, Reed JL, Shachar-Hill Y (2011) Synergy between (13)C-metabolic flux analysis and flux balance analysis for understanding metabolic adaptation to anaerobiosis in *E. coli*. Metab Eng 13 (1):38–48. doi:10.1016/j.ymben.2010.11.004
33. Walsh K, Koshland DE Jr (1985) Branch point control by the phosphorylation state of isocitrate dehydrogenase. A quantitative examination of fluxes during a regulatory transition. J Biol Chem 260(14):8430–8437
34. Meyer CL, Papoutsakis ET (1989) Continuous and biomass recycle fermentations of *Clostridium acetobutylicum*.2. Novel patterns in energetics and product-formation kinetics. Bioprocess Eng 4(2):49–55
35. Oh YK, Palsson BO, Park SM, Schilling CH, Mahadevan R (2007) Genome-scale reconstruction of metabolic network in *Bacillus subtilis* based on high-throughput phenotyping and gene essentiality data. J Biol Chem 282 (39):28791–28799. doi:10.1074/jbc.M703759200
36. The openCOBRA Project http://opencobra.sourceforge.net/openCOBRA/Welcome.html. Accessed on 4/1/2012
37. LibSBML http://sourceforge.net/projects/sbml/files/libsbml/5.1.0-b0/. Accessed on 4/1/2012

Chapter 6

A Guide to Integrating Transcriptional Regulatory and Metabolic Networks Using PROM (Probabilistic Regulation of Metabolism)

Evangelos Simeonidis, Sriram Chandrasekaran, and Nathan D. Price

Abstract

The integration of transcriptional regulatory and metabolic networks is a crucial step in the process of predicting metabolic behaviors that emerge from either genetic or environmental changes. Here, we present a guide to PROM (*probabilistic regulation of metabolism*), an automated method for the construction and simulation of integrated metabolic and transcriptional regulatory networks that enables large-scale phenotypic predictions for a wide range of model organisms.

Key words: Systems biology, Metabolic networks, Transcriptional regulatory networks, Constraint-based modeling, Probabilistic regulation of metabolism, Microarray data

1. Introduction

In systems biology, we study the complexity of biological systems and try to comprehend and predict the way that the components of these systems interact. In the last two decades, advances in high-throughput experimental techniques and bioinformatics methodologies have produced an abundance of biological information. The successful management, integration, and utilization of these data are critical to enable systems biology approaches. One of the most fundamental processes necessary for life is metabolism, from which the cell harnesses energy from its food and builds the components necessary for growth and reproduction. Metabolism plays a central role in the functioning of an organism and is arguably the best understood cellular process. Therefore, systems biologists have taken an early interest in metabolic networks, their behavior, and their regulation.

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_6,

Metabolic networks display complicated structures and interactions, leading to nonlinear dynamic behaviors (1–3). The size and complexity of metabolic networks often limit our ability to test and analyze metabolism using more traditional simulation methods such as reaction kinetics, where the mechanisms or reactions and their regulation are modeled individually and in detail. Constraint-based modeling (4, 5) allows us to overcome such problems, because the only requirement is knowledge of the stoichiometry of the system in order to be able to accurately simulate the potential metabolic behavior of an organism. Over the years, a number of stoichiometry-based methodologies have been developed, with the most commonly used being flux balance analysis (FBA) (6). FBA identifies the optimal flux pattern of a network that would allow the system to achieve a particular objective, typically the maximization of biomass production.

FBA is a powerful method for predicting system behavior, but one of its drawbacks is that it ignores the often important effect of regulation. Metabolic networks are tightly controlled, in part, by intricate transcriptional networks – further increasing the complexity of the system. Being able to model this transcriptional regulation allows us to interpret the effect of mutations and environmental perturbations on functional metabolism, which in turn opens up the possibility of diagnosing metabolic disorders and identifying new drug targets.

In this chapter, we give an overview of PROM (probabilistic regulation of metabolism) (7), a method that utilizes probabilities to denote gene states and interactions between genes and transcription factors in order to enable straightforward integration of transcriptional and metabolic networks for modeling purposes. In the past, there have been relatively few efforts focused on the integration of metabolic and transcriptional regulatory networks (8, 9). PROM has shown improved results compared to previous approaches to integrate metabolism and transcriptional regulation such as regulatory FBA (RFBA) (8, 10) in studies published so far. Another benefit of PROM is that it estimates regulatory strengths automatically from high-throughput data, as opposed to the laborious manual process RFBA models are based on. Because PROM networks can be learned from high-throughput data, these models can be comprehensive, in contrast to the manually curated approaches that require extensive literature surveys. In addition, RFBA relies on Boolean logic, which has the drawback of only allowing two states for the regulated reactions: either fully active or completely inactive. PROM introduces probabilistic, soft constraints that can be automatically quantified from microarray data, thereby overcoming the limitations of RFBA (7).

2. Analysis Tools

2.1. Flux Balance Analysis

Mathematically, FBA is framed as a linear programming problem:

FBA Formulation
Maximize

$$Z = c_j v_j \tag{1}$$

subject to

$$\sum_j S_{ij} \cdot v_j = 0 \quad \forall i \tag{2}$$

$$v_j^L \leq v_j \leq v_j^U \quad \forall j, \tag{3}$$

where i indexes the set of metabolites, j indexes the set of reactions in the network, S_{ij} is the stoichiometric matrix, c_j is a vector that specifies which flux is being optimized (typically this is used for the maximization of growth), and v_j is the flux of reaction j. The objective function (1) is maximized over all possible steady-state fluxes satisfying certain stoichiometric constraints (2). In genome-scale metabolic models of microbial systems, a biomass-producing reaction is usually defined and used as the objective function. Upper and lower bounds are placed on the individual fluxes (v^U and v^L, respectively) (3). For irreversible reactions, $v^L = 0$. Specific bounds, based on enzyme capacity measurements or thermodynamic considerations, can be imposed on reactions; in the absence of any information, these rates are generally left unconstrained, i.e., $v^U = \infty$ and $v^L = -\infty$ for reversible reactions. To avoid unbounded solutions, i.e., Z reaching infinity, one rate (the input flux; typically the influx of glucose) needs to be fixed to a specific value, and all fluxes should be viewed as relative to the input flux.

2.2. Flux Variability Analysis

Flux variability analysis (FVA) (11) is used to determine the range of allowable fluxes in the optimal solutions of a constraint-based analysis problem. Using FVA, we can determine the minimum and maximum possible flux through a reaction for a given optimal growth rate. After solving the FBA formulation above and identifying the optimal growth rate v_g^*, the following algorithm is used to determine the variability of fluxes in the network:

FVA Algorithm

For $r = 1$ to R
Minimize

$$Z = v_r \tag{4}$$

subject to

$$v_{growth} \geq v_g^* \tag{5}$$

$$\sum_j S_{ij} \cdot v_j = 0 \quad \forall i \tag{6}$$

$$v_j^L \leq v_j \leq v_j^U \quad \forall j \tag{7}$$

then
maximize

$$Z = v_r \tag{8}$$

subject to (5), (6), (7)
end,

where R is the total number of reactions j in the reconstructed network.

2.3. Kolmogorov–Smirnov Test

The Kolmogorov–Smirnov test (12) is a nonparametric test for the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution or to compare two separate samples. The Kolmogorov–Smirnov test is used to check how much two of our expression profiles (described in the methods section) differ when compared to each other. The null hypothesis is that the two datasets are from the same distribution, whereas the alternative hypothesis is that they are from different continuous distributions. The Kolmogorov–Smirnov test has the advantage of making no assumption about the distribution of data. The method is used to select only those pairs of transcription factors and targets for which the target's expression changes significantly with respect to the transcription factor state.

3. Methods

In order to build an integrated model of a metabolic and transcriptional regulatory network for an organism (see Note 1 and Fig. 1), the following components are needed:

1. *The genome-scale reconstruction of the metabolic network of the organism* (13). The creation of metabolic reconstructions is often a laborious, painstaking process. Researchers either manually collect the necessary stoichiometric information from the literature, or the network is downloaded from organism-specific databases when available, with subsequent annotation and improvement of the data to make the model functional and in agreement with experimental data. Over the last 10 years, the metabolic network reconstructions of several organisms have been published and are publicly accessible. The simulation of the metabolic network within the PROM method is performed

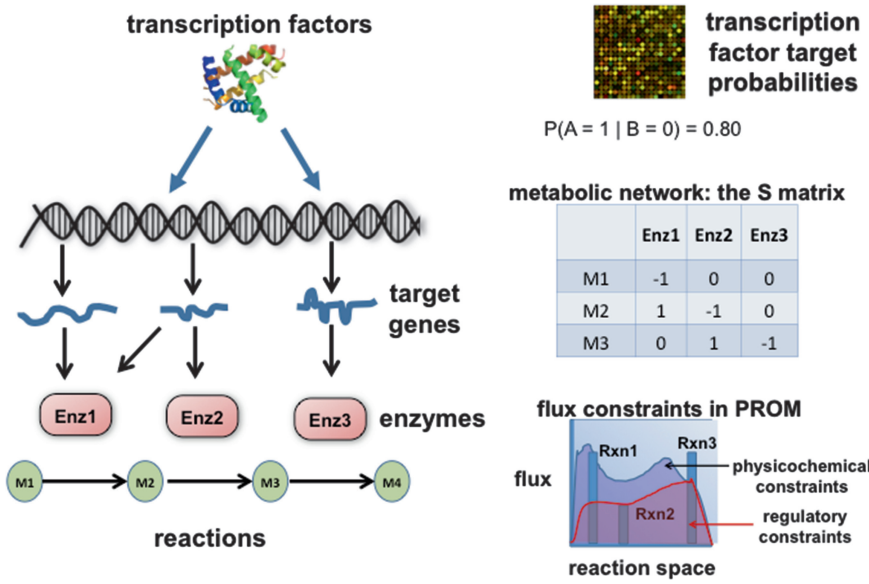

Fig. 1. The PROM method.

using FBA subject to additional constraints and a penalty function (described below) (Subheading 2.1).

2. *A regulatory network structure, which consists of a list of transcription factors, the targets of these transcription factors, and their interactions* (14). These transcriptional regulatory networks have generally been constructed based on high-throughput protein–DNA interaction data and/or statistical inference of functional relationships from genomic and transcriptomic data (15–19).
3. *A collection of gene expression data measured under different conditions, which will allow the observation of various phenotypes for the organism under study.* Ideally, the microarray data that are chosen represent a diverse number of conditions under which gene expression has been measured (see Note 2).

For the purpose of constructing the integrated metabolic– regulatory network, it is important that the PROM method takes advantage of the abundance of high-throughput data that is currently available for most organisms. From these data, the transcriptional regulatory network of the organism can be quantified statistically, similar to the probabilistic Boolean networks of Shmulevich et al. (20). From the gene expression data that are available for the organism in question, only those that involve the expression of metabolic genes are retained (see Note 3). The data are then normalized and screened for false positives using the Kolmogorov–Smirnov statistic (Subheading 2.3), and only significant interactions, defined by $P < 0.05$, are kept in the model.

PROM introduces probabilities to represent gene states and interactions between a gene and a transcription factor. For example, the probability of gene A being active when the regulating transcription factor B is not active is represented by $P(A = 1|B = 0)$, whereas the probability of both gene and transcription factor being active by $P(A = 1|B = 1)$ (see Note 3). Information from the microarray data is then used to assign values to the relationship between transcription factor and target gene. To determine the relationship between a transcription factor and its target, we first binarize the data to represent either an ON or OFF state for all the genes. We can then model the relationship between the transcription factor and the target based on the following formula:

$$P(A = 1|B = 0) = \frac{N(A = 1|B = 0)}{N(B = 0)} \tag{9}$$

where N is the number of times the event is observed. For example, if in 80% of the samples we find the gene to be on when the transcription factor is off, then the probability $P(A = 1|B = 0) = 0.8$. For transcription factors that affect more than one gene, we need to calculate this relationship for all its target genes. The fluxes of the reactions controlled by these genes can then be constrained using this information. In our example, the flux through the reaction regulated by gene A when its corresponding regulator B is turned off would be

$$P \cdot v_A^L \leq v_A \leq P \cdot v_A^U \tag{10}$$

Generally, the bounds on the fluxes of reactions in our network are redefined from having an upper bound of v_j^U to an upper bound of $P \cdot v_j^U$, where P is the probability of the gene being active under the specific phenotype. For reversible reactions, the same applies to lower bounds v_j^L (see Note 5).

Estimates for reaction bounds v_j^L, v_j^U are obtained by running the FVA algorithm (Subheading 2.2) on the unregulated metabolic model or by utilizing literature or other kinds of prior knowledge.

Unlike thermodynamic or environmental constraints that cannot be violated, we want regulatory constraints to be soft constraints to compensate for the lower confidence level and inherent uncertainty that comes from the experimentation techniques, our (lack of) understanding of the gene–transcription factor interactions and noise in the measurements. The algorithm needs to be able to exceed regulatory constraints to maximize growth if necessary but with a penalty to avoid this happening regularly. Following this procedure, we arrive at the following final formulation for the PROM model, which satisfies most or all of the regulatory constraints:

PROM Formulation
Maximize

$$Z = c_j v_j + \sum_j (\kappa_j \cdot \alpha_j + \kappa_j \cdot \beta_j) \tag{11}$$

subject to

$$\sum_j S_{ij} \cdot v_j = 0 \quad \forall i \tag{12}$$

$$P \cdot v_j^L - \alpha_j \leq v_j \leq P \cdot v_j^U + \beta_j \quad \forall j \tag{13}$$

$$\alpha_j, \beta_j \geq 0 \quad \forall j, \tag{14}$$

where $P \cdot v_j^L$ and $P \cdot v_j^U$ are the transcriptional regulation bounds, α_j and β_j are positive variables that allow deviation from those bounds, and κ_j is the cost for such deviations. The term $(\kappa_j \cdot \alpha_j + \kappa_j \cdot \beta_j)$ represents the penalty for exceeding an upper or a lower bound. The higher the value of κ, the greater the constraint on the system based on transcriptional regulation. For values of κ significantly greater than 1, the regulatory constraints become hard, and for values less than 0.1, they become insignificant. Typically, a κ value of 1 is chosen for all simulations as it represents a trade-off between the two extremes.

It is clear from the above that with PROM, gene states can take values other than just 0 and 1 due to the use of probabilities, which allows us to distinguish between strong and weak regulators. Another benefit of PROM is that we can incorporate interactions for which we have strong evidence from the literature or experimentation. If we have an example of an interaction with high-confidence proof from the literature, then the user can assign a probability of 0 or 1 for that particular interaction, setting the corresponding gene to either fully active or completely inactive. The probabilities for the rest of the interactions can then be determined based on the microarray data, following the method described here. Additional interactions involving enzyme regulation by metabolites and proteins can also be modeled and included in the PROM algorithm. If no probability can be inferred, or no data are available for a specific interaction, we set the corresponding probability to $P = 1$.

By applying the algorithm as presented above, we can then run the resulting model in order to predict the effect of knockouts of transcription factors on the metabolic fluxes. The lethality of transcription factor knockouts is predicted in a similar way to that of Shlomi et al. (21); if the respective prediction of the mutated organism's maximal growth rate is less than 5% of the wild-type growth rate, it is considered lethal, whereas knockouts that display

a growth rate lower than the wild-type growth rate are considered suboptimal.

The PROM algorithm is available for download at the following address (see Note 6): http://price.systemsbiology.net/downloads_tmp.php

4. Notes

1. Integrated metabolic–regulatory methods can be built with the PROM method for any organism for which reconstructed metabolic network models; regulatory interaction data and a sufficient amount of microarray experiment data are available.
2. The purpose of using microarrays in PROM is to quantify the relationship between transcription factors and target genes. This can be done only if we study their relationship under as many conditions as possible. If we were to use microarrays from a single condition only, we would not be able to see any change in the expression of transcription factors and target genes, and therefore, we could not learn or quantify their relationship.
3. PROM predicts phenotypes based on a gene's effect on metabolism; it cannot determine correctly the phenotypes of genes with major nonmetabolic functions.
4. When using microarray data to estimate the necessary probabilities, gene expression values under a predefined low threshold are considered inactive, and the remaining values are considered active. PROM uses the 33rd percentile as a default threshold to determine gene activity or inactivity. Generally, thresholds from 0.2 to 0.4 have provided comparably accurate predictions for the systems we have tested. The PROM algorithm, as implemented in the downloadable code, produces a warning to the user if the threshold used is not sufficient to estimate the probabilities.
5. For cases in which the probability of interaction cannot be estimated by using microarray data because of unavailability of expression data for that specific target gene or transcription factor, or if the gene was active or inactive under all conditions, we usually set the probability to a default value of 1. A value of 1 implies that the flux bounds for the reaction are not adjusted, but remain the same as in an unregulated model, whereas if the probability is set to 0, the reaction is considered inactive.
6. Pointers for running the code:
   (a) While running the PROM code, ensure that you have expression data for all the genes and regulators in the interaction data. If a relatively small fraction of the data

has missing values, PROM can impute these missing values using the k-nearest neighbors algorithm; however, it cannot handle data for genes with no expression data.

(b) As PROM predicts phenotypes based on a gene's effect on metabolism, the regulatory interactions must be between regulators and target genes that are part of the metabolic model.

(c) The names or identifiers used in the regulatory interaction data must match the names or IDs used for gene expression data and the names of the genes in the metabolic model.

(d) By default, PROM computes the predicted knockout growth rates for all the transcription factors in the network, and the growth rate is outputted in alphabetical order of the transcription factors.

Acknowledgments

We acknowledge funding from the Grand Duchy of Luxembourg for ES and NDP, a NIH Howard Temin Pathway to Independence Award in Cancer Research, an NSF CAREER grant, and the Camille Dreyfus Teacher-Scholar Program for NDP, and a Howard Hughes Medical Institute Predoctoral Fellowship for SC.

References

1. Lazebnik Y (2002) Can a biologist fix a radio? Or, what I learned while studying apoptosis. Cancer Cell 2(3):179–182
2. Mendes P, Kell D (1998) Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics 14(10):869–883. doi:10.1093/bioinformatics/14.10.869
3. Szallasi Z, Stelling J, Periwal V (2006) System modeling in cellular biology: from concepts to nuts and bolts, 1st edn. The MIT Press, Boston
4. Covert MW, Famili I, Palsson BO (2003) Identifying constraints that govern cell behavior: a key to converting conceptual to computational models in biology? Biotechnol Bioeng 84 (7):763–772. doi:Doi 10.1002/Bit.10849
5. Price ND, Reed JL, Palsson BO (2004) Genome-scale models of microbial cells: evaluating the consequences of constraints. Nat Rev Microbiol 2(11):886–897. doi:Doi 10.1038/Nrmicro1023
6. Kauffman KJ, Prakash P, Edwards JS (2003) Advances in flux balance analysis. Curr Opin Biotechnol 14(5):491–496. doi:DOI 10.1016/j.copbio.2003.08.001
7. Chandrasekaran S, Price ND (2010) Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in *Escherichia coli* and *Mycobacterium tuberculosis*. Proc Natl Acad Sci U S A 107(41):17845–17850. doi:DOI 10.1073/pnas.1005139107
8. Covert MW, Knight EM, Reed JL, Herrgard MJ, Palsson BO (2004) Integrating high-throughput and computational data elucidates bacterial networks. Nature 429(6987):92–96
9. Herrgard MJ, Lee BS, Portnoy V, Palsson BO (2006) Integrated analysis of regulatory and metabolic networks reveals novel regulatory mechanisms in *Saccharomyces cerevisiae*. Genome Res 16(5):627–635. doi:Doi 10.1101/Gr.4083206

10. Covert MW, Schilling CH, Palsson B (2001) Regulation of gene expression in flux balance models of metabolism. J Theor Biol 213 (1):73–88. doi:DOI 10.1006/jtbi.2001.2405
11. Mahadevan R, Schilling CH (2003) The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. Metab Eng 5(4):264–276. doi:DOI 10.1016/j.ymben.2003.09.002
12. Young IT (1977) Proof without prejudice—Use of Kolmogorov-Smirnov test for analysis of histograms from flow systems and other sources. J Histochem Cytochem 25 (7):935–941
13. Feist AM, Herrgard MJ, Thiele I, Reed JL, Palsson BO (2009) Reconstruction of biochemical networks in microorganisms. Nat Rev Microbiol 7(2):129–143. doi:Doi 10.1038/Nrmicro1949
14. Bansal M, Belcastro V, Ambesi-Impiombato A, di Bernardo D (2007) How to infer gene networks from expression profiles. Mol Syst Biol 3:122. doi:Artn 78. doi: Doi 10.1038/Msb4100120
15. Davidson EH, Rast JP, Oliveri P, Ransick A, Calestani C, Yuh CH, Minokawa T, Amore G, Hinman V, Arenas-Mena C, Otim O, Brown CT, Livi CB, Lee PY, Revilla R, Rust AG, Pan ZJ, Schilstra MJ, Clarke PJC, Arnone MI, Rowen L, Cameron RA, McClay DR, Hood L, Bolouri H (2002) A genomic regulatory network for development. Science 295 (5560):1669–1678
16. Schlitt T, Brazma A (2007) Current approaches to gene regulatory network modelling. BMC Bioinformatics 8. doi:Artn S9. doi: Doi 10.1186/1471-2105-8-S6-S9
17. Shen-Orr SS, Milo R, Mangan S, Alon U (2002) Network motifs in the transcriptional regulation network of *Escherichia coli*. Nat Genet 31(1):64–68. doi:Doi 10.1038/Ng881
18. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK, Hannett NM, Harbison CT, Thompson CM, Simon I, Zeitlinger J, Jennings EG, Murray HL, Gordon DB, Ren B, Wyrick JJ, Tagne JB, Volkert TL, Fraenkel E, Gifford DK, Young RA (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. Science 298(5594):799–804
19. Chandrasekaran S, Ament SA, Eddy JA, Rodriguez-Zas SL, Schatz BR, Price ND, Robinson GE (2011) Behavior-specific changes in transcriptional modules lead to distinct and predictable neurogenomic states. Proc Natl Acad Sci U S A 108(44):18020–18025. doi: DOI 10.1073/pnas.1114093108
20. Shmulevich I, Dougherty ER, Kim S, Zhang W (2002) Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics 18(2):261–274
21. Shlomi T, Cabili MN, Herrgard MJ, Palsson BO, Ruppin E (2008) Network-based prediction of human tissue-specific metabolism. Nat Biotechnol 26(9):1003–1010. doi:Doi 10.1038/Nbt.1487

Chapter 7

Kinetic Modeling of Metabolic Pathways: Application to Serine Biosynthesis

Kieran Smallbone and Natalie J. Stanford

Abstract

In this chapter, we describe the steps needed to create a kinetic model of a metabolic pathway using kinetic data from both experimental measurements and literature review. Our methodology is presented by using the example of serine biosynthesis in *E. coli*.

Key words: Mathematical modeling, Metabolism, Systems metabolic engineering

1. Introduction

Metabolic pathways are collections of enzyme-mediated reactions. Their role in living organisms is to transform cellular inputs, such as glucose, into biomass and energy, so they can grow and function. Nonlinear processes dominate the interactions between enzymes and metabolites, and hence, intuitive verbal reasoning approaches are insufficient to describe the complex dynamic behavior of such biological networks (1–3), nor can such approaches keep pace with the large increases in omics data (such as metabolomics and proteomics) and the accompanying advances in high-throughput experiments and bioinformatics. Rather, kinetic models, continuously revised to incorporate new information, must be used to guide experimental design and interpretation. Moreover, they allow the user to explore a wider variety of biological questions than can be studied using laboratory techniques alone.

In this chapter, we describe the steps needed to create a kinetic model of a metabolic pathway. We use serine biosynthesis in *Escherichia coli* (4) as an example to apply the methodology required for this purpose.

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_7,

2. Materials

2.1. Data

A mathematical description of a kinetic metabolic model may be given in differential equation form as

$$x' = Nv(x, y, p)$$
$$x(0) = x0,$$

and this may be used as a guide as to the data that must be collected to create and parameterize a kinetic model.

1. First, N is the stoichiometric matrix, which may be derived from the network structure (Table 1). Pathway stoichiometries may be found in generic databases, such as Kegg (5), or organism-specific databases such as EcoCyc (6).
2. x denotes metabolite concentrations; y denotes boundary metabolites, whose concentrations are not allowed to vary, but do affect the reaction rates. Initial concentrations for both x and y must be defined (Table 2), though note that only concentrations x will change over time. For human models, data is available at HMDB (7).

Table 1
Network stoichiometry

Reaction	Stoichiometry	E.C. code
PDH	P3G ↔ PHP	1.1.1.95
PSA	PHP ↔ PSER	2.6.1.52
PSP	PSER ↔ SER	3.1.3.3

Note that, for clarity, we have ignored the role of cofactors (such as NAD) in this model
Legend: *PDH* phosphoglycerate dehydrogenase, *PSA* phosphoserine aminotransferase, *PSP* phosphoserine phosphatase, *P3G* 3-phosphoglycerate, *PHP* phosphohydroxypyruvate, *PSER* phosphoserine, *SER* serine

Table 2
Initial metabolite concentrations

Metabolite	Concentration (mM)	ChEBI	Reference
P3G	2.36	58272	(21)
PHP	0.60	18110	(21)
PSER	0.09	57524	(21)
SER	4.90	17115	(21)

Table 3
Kinetic rate laws

Reaction	Formula
PDH	`serA*kcatA*(P3G/KAp3g)/(1 + P3G/KAp3g + PHP/KAphp)/(1 + SER/KiAser)`
PSA	`serC*kcatC*(PHP/KCphp)/(1 + PHP/KCphp + PSER/KCpser)`
PSP	`serB*kcatB*(PSER/KBpser)/(1 + PSER/KBpser + SER/KBser)`

More complex forms are available (22), but are simplified here, for clarity

Table 4
Kinetic parameters

Parameter	Value	Units	Reference
serA	1.15	mM	(22)
kcatA	0.55	1/s	(23)
KAp3g	1.2	mM	(23)
KAphp	0.0032	mM	(23)
KiAser	0.0038	mM	(23)
serC	0.1	mM	(22)
kcatC	1.75	1/s	(24)
KCphp	0.0015	mM	(24)
KCpser	0.0017	mM	(24)
serB	0.25	mM	(22)
kcatB	1.43	1/s	(22)
KBpser	0.0015	mM	(22)
KBser	0.15	mM	(22)

3. Finally, v denotes reaction rates; these are dependent on kinetic mechanisms (Table 3), parameters p (Table 4), and concentrations x and y. Databases of kinetic parameters include Brenda (8) and Sabio-RK (9).

2.2. Software

In this chapter, we will demonstrate how to build a model using Copasi (10), but the underlying principles apply to any software. The *SBML Software Guide* (11) lists alternatives that support the systems biology community standards.

3. Methods

3.1. Generating the Stoichiometric Matrix

1. Identify the important reactions that make up the pathway you wish to study. This information can be obtained from a network database. In this instance, three reactions are of interest: phosphoglycerate dehydrogenase, phosphoserine aminotransferase, and phosphoserine phosphatase.
2. Define the metabolites involved in each reaction, including any known allosteric regulation (see Note 1) or catalysis. This information can be taken from the network database, but should be checked to ensure that the reactions are stoichiometrically consistent. In this instance the reactions are described as follows:

   3-phosphoglycerate ↔ phosphohydroxypyruvate (modifiers: serA, serine)

   phosphohydroxypyruvate ↔ phosphoserine (modifiers: serC)

   phosphoserine ↔ serine (modifiers: serB)

3. Each of the reactions outlined above added to the model. Taking the first reaction as an example, in Copasi we navigate to `Model > Biochemical > Reactions`; enter the following in the chemical equation box:

   `3-phosphoglycerate = phosphohydroxypyruvate; serA serine`

   (see Note 2); and add the reaction name

   `phosphoglycerate dehydrogenase`

   (see Note 3) Return to `Model > Biochemical > Reactions` (see Note 4). All other reactions should be added in the same way.
4. At `Model > Biochemical > Species`, you will notice that seven metabolites are listed. These are the metabolites that participate in the above reactions (see Note 5). Initially, all metabolites are assumed to be variables that change over time as the system is simulated. In order for a steady state condition to be reached, system inputs and outputs have to be defined through setting some metabolites as boundary conditions. To do this in Copasi, for each metabolite corresponding to the metabolic input or output, change `Simulation Type` from `reactions` to `Fixed`. This indicates that the concentrations of these metabolites should be constant over time. For serine biosynthesis, the following need to be `Fixed`: 3-phosphoglycerate, serine, serA, serC, and serB.

Table 5
Stoichiometric matrix

	PDH	PSA	PSP
PHP	1	−1	0
PSER	0	1	−1

Legend: *PDH* phosphoglycerate dehydrogenase, *PSA* phosphoserine aminotransferase, *PSP* phosphoserine phosphatase, *P3G* 3-phosphoglycerate, *PHP* phosphohydroxypyruvate, *PSER* phosphoserine, *SER* serine

5. It is important to ensure that the model is defined in the correct units. Navigate to Model and ensure that `Time`, `Quantity Unit`, and `Volume Unit` are set accordingly to, in this instance, `s`, `mmol`, and `l`, respectively. In addition ensure that `Rate Law Interpretation` is set to `deterministic` (see Note 6).
6. The basic reaction network has now been constructed, and the stoichiometric matrix (N) is now available by navigating to `Model > Mathematical > Matrices`. It should match Table 5 (see Note 7).

3.2. Adding in Kinetic Data

To turn the stoichiometric model into a kinetic model, rate laws have to be defined for each reaction. The rate laws define the kinetic rate (v) at which the substrate metabolite/s are converted into product/s.

1. A rate law that mathematically captures reaction behavior is added to each reaction (see Note 8). Here we use irreversible Michaelis–Menten kinetics, as can be seen in Table 3.
2. The derived rate laws must be added to corresponding reactions. As an example we take phosphoglycerate dehydrogenase. In Copasi, we navigate to the corresponding reaction in the `Reactions` menu and then add a `New Rate Law` with formula

   ```
   serA*kcatA*(P3G/KAp3g)/(1 + P3G/KAp3g + PHP/
   KAphp)/(1 + SER/KiAser)
   ```

   Once the rate law has been entered, each separate variable within the math string must be defined as `Substrate` (`P3G`); `Product` (`PHP`); `Modifier` (`serA`, `SER`); or `Parameter` (`KcatA`, `KAp3g`, `KAphp`, `KiAser`).
3. The new rate law needs to be selected for the corresponding reaction back in the `Reaction` menu. At this point the associated parameter values can be added to the reaction, available in Table 4 (see Note 9).

Table 6
Steady state concentrations

Metabolite	Concentration (mM)
PHP	2.86×10^{-06}
PSER	4.59×10^{-05}

3.3. Community Standards

Describing mathematical models as above is unwieldy and error-prone and naturally leads to difficulties in reproduction of results. Thus, standards have been developed to represent models.

1. SBML (12) is an XML-based markup language to unambiguously describe models. When structured in an SBML format, over 200 software tools can be used to study the model; this includes data integration, dynamic simulation, and visualization (11). To turn the Copasi model into SBML, use `File` > `Export SBML...`.
2. SBML may be combined with Miriam (13) to annotate model entities. For example, pointing P3G to the database entry http://identifiers.org/obo.chebi/CHEBI:58272 allows its unambiguous identification and automatically links to many additional sources of information (see Note 10). The ChEBI ID is available in Table 2 for each metabolite. In Copasi, these may be added to the corresponding metabolite by navigating to the `Species` submenu `Annotation`.
3. BioModels.net (14) is a database that specializes in publishing mathematical models. There are more than 800 available for download in Miriam-annotated SBML format. The model described in this chapter is available at http://identifiers.org/biomodels.db/MODEL1203210000.

3.4. Analyzing Model Behavior

The purpose of a mathematical model is to allow us to study complex cellular interactions more easily. The next important step after model construction is to begin to analyze its behavior.

1. Living organisms can be exposed to varying external conditions, but need to maintain core cellular functions in order to remain viable. On the basis of this, it is believed that internal cellular behavior will attain a specific resting state that allows these functions to be carried out optimally. To calculate the steady state of the model in Copasi, navigate to `Tasks` > `Steady state` and click `Tasks` > `Steady state`. The result should match Tables 6 and 7 (see Note 11).
2. Under different external conditions, the steady state required to maintain core cellular functions might change rapidly.

Table 7
Steady state fluxes

Reaction	Flux (mM/s)
PDH	3.24×10^{-04}
PSA	3.24×10^{-04}
PSP	3.24×10^{-04}

The change in fluxes and metabolite concentrations towards a new steady state is known as transient behavior. The transient behavior can be assessed using `Tasks` > `Time Course` (see Note 12).

4. Notes

1. Allosteric regulators/modifiers are metabolites that positively or negatively affect the rate of an enzymatic reaction without being used or consumed within the reaction. They can form feedback mechanisms that regulate the flux through the pathway.
2. The metabolite names must be written with no gaps between the letters. For reversible reactions "=" is used, while if the reaction is irreversible, then "->" should be used. The end of the reaction metabolites and the beginning of the modifiers are signified using ";". For reactions that contain no modifiers, this should be omitted.
3. The reaction names correspond to the short form available in the below table (insert table names). You can use either form to build the model. It is important to maintain consistency though.
4. At this point that Copasi automatically assigns a rate law to mathematically describe the reaction. We will address inserting unique rate laws in the next section.
5. This is a good time to check the metabolite list. If there are more or less than the expected number of metabolites present, it is likely that there is an error in one of the equations. This should be rectified before proceeding.
6. Deterministic modeling is appropriate when the number of molecules in the system is large. When the number of molecules is small, stochastic modeling may prove more appropriate.

7. Only metabolites that can change over time are included in the stoichiometric matrix, so all `Fixed` metabolites will not appear. The names present in the matrix will also depend on the naming conventions used in step 3.
8. These can be correctly measured in laboratory experiments. Where this is not possible, there are a range of equations that can be used in order to make an approximation of the reaction rate (v) (15–19). In addition, Copasi also has a collection of equations that can be applied to reactions when the actual kinetic mechanism is not known.
9. When there are multiple substrates, products of modifiers for a given reaction ensure that the parameters correspond to the correct metabolite. This can be altered using the drop-down menu next to the parameter.
10. ChEBI (20) is a dictionary of small chemical compounds that unambiguously states the molecular structure and properties of a metabolite. By adding the ChEBI identifier to the compound, it allows the compound to always be identified regardless of the name used within the model.
11. Steady states may not closely match the experimental values and thus suggest discrepancies in the model. Refining or fitting the model, or adding new experimental data, can improve the model's utility.
12. You can change the duration of the time course and its granularity, allowing more complex behavior to be accurately tracked.

Acknowledgements

KS is grateful for the financial support of the EU FP7 (KBBE) grant 289434 "BioPreDyn: New Bioinformatics Methods and Tools for Data-Driven Predictive Dynamic Modelling in Biotechnological Applications." We thank Daniel Jameson for taking the time to comment on the manuscript. Natalie Stanford acknowledges the support of DirectFuel, a European Union Seventh Framework Programme (FP7-ENERGY-2010-1) under grant agreement n. [256808].

References

1. Lazebnik Y (2002) Can a biologist fix a radio? Or, what I learned while studying apoptosis. Cancer Cell 2:179–182
2. Mendes P, Kell D (1998) Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics 14:869–883
3. Szallasi Z, Stelling J, Periwal V (2006) System modeling in cellular biology: from concepts to nuts and bolts. MIT Press, Boston

4. Pizer LI (1963) The pathway and control of serine biosynthesis in *Escherichia coli*. J Biol Chem 238:3934–3944
5. Kanehisa M, Goto S, Sato Y et al (2012) KEGG for integration and interpretation of large-scale molecular datasets. Nucleic Acids Res 40: D109–D114, http://www.genome.jp/kegg-bin/show_pathway?org_name=eco&mapno=00260
6. Keseler IM, Collado-Vides J, Santos-Zavaleta A et al (2011) EcoCyc: a comprehensive database of *Escherichia coli* biology. Nucleic Acids Res 39:D583–D590, http://ecocyc.org/ECOLI/NEW-IMAGE?object=SERSYN-PWY
7. Wishart DS, Tzur D, Knox C et al (2007) HMDB: the Human Metabolome Database. Nucleic Acids Res 35:D521–D526
8. Scheer M, Grote A, Chang I et al (2010) BRENDA, the enzyme information system in 2011. Nucleic Acids Res 39:D670–D676
9. Wittig U, Golebiewski M, Kania R et al (2006) SABIO-RK: integration and curation of reaction kinetics data. Lect Notes Bioinformatics 4075:94–103
10. Hoops S, Sahle S, Gauges R et al (2006) COPASI—a COmplex PAthway SImulator. Bioinformatics 22:3067–3074
11. SBML software guide. http://sbml.org/SBML_Software_Guide 17 June 2011
12. Hucka M, Finney A, Sauro HM et al (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19:524–531
13. Le Novère N, Finney A, Hucka M et al (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). Nat Biotechnol 23:1509–1515
14. Li C, Donizelli M, Rodriguez N et al (2010) BioModels Database: an enhanced, curated and annotated resource for published quantitative kinetic models. BMC Syst Biol 4:92
15. Heijnen JJ (2005) Approximative kinetic formats used in metabolic network modeling. Biotechnol Bioeng 91:534–545
16. Liebermeister W, Klipp E (2006) Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. Theor Biol Med Model 3:41
17. Savageau MA (1976) Biochemical systems analysis: a study of function and design in molecular biology. Addison-Wesley, Boston
18. Smallbone K, Simeonidis E, Broomhead DS et al (2007) Something from nothing: bridging the gap between constraint-based and kinetic modelling. FEBS J 274:5576–5585
19. Visser D, Heijnen JJ (2003) Dynamic simulation and metabolic re-design of a branched pathway using linlog kinetics. Metab Eng 5:164–176
20. Degtyarenko K, de Matos P, Ennis M et al (2008) ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Res 36:D344–D350
21. Ao P, Lee LW, Lidstrom ME et al (2008) Towards kinetic modeling of global metabolic networks: Methylobacterium extorquens AM1 growth as validation. Chin J Biotechnol 24:980–994
22. Turnaev II, Ibragimova SS, Usuda Y et al (2006) Mathematical modeling of serine and glycine synthesis regulation in *Escherichia coli*. Proceedings of the fifth international conference on bioinformatics of genome regulation and structure 2:78–83
23. Zhao G, Winkler ME (1996) A novel alpha-ketoglutarate reductase activity of the serA-encoded 3-phosphoglycerate dehydrogenase of *Escherichia coli* K-12 and its possible implications for human 2-hydroxyglutaric aciduria. J Bacteriol 178:232–239
24. Drewke C, Klein M, Clade D et al (1996) 4-O-phosphoryl-L-threonine, a substrate of the pdxC(serC) gene product involved in vitamin B6 biosynthesis. FEBS Lett 390:179–182

Chapter 8

Computational Tools for Guided Discovery and Engineering of Metabolic Pathways

Matthew Moura, Linda Broadbelt, and Keith Tyo

Abstract

With a high demand for increasingly diverse chemicals, as well as sustainable synthesis for many existing chemicals, the chemical industry is increasingly looking to biosynthesis. The majority of biosynthesis examples of useful chemicals are either native metabolites made by an organism or the heterologous expression of known metabolic pathways into a more amenable host. For chemicals that no known biosynthetic route exists, engineers are increasingly relying on automated computational algorithms, as described here, to identify potential metabolic pathways. In this chapter, we review a broad range of approaches to predict novel metabolic pathways. Broadly, these can rely on biochemical databases to assemble known reactions into a new pathway or rely on generalized biochemical rules to predict unobserved enzymatic reactions that are likely feasible. Many programs are freely available and immediately useable by non-computationally experienced scientists.

Key words: Metabolic network, Metabolic pathway design, Heterologous pathways, Enzyme database searching

1. Introduction

Our twenty-first century society has an increasing demand for a range of chemicals, from fuels (1) to polymeric precursors (2, 3) to drug and drug precursors (4–6), to name a few. Not only do we need these compounds (some of which are increasingly complex) in increasing quantities, but we also need to produce these compounds sustainably, minimizing the emission of toxic contaminates, suspected climate-change agents, and reduce the energy associated with production and purification. (7).

Biosynthesis of these compounds is well positioned to meet this need, as biological systems can synthesize complex molecules in high yield in moderate (i.e., aqueous, ambient temperature, and pressure) reaction conditions (8). The capture and redirection of *existing* metabolic pathways toward the production of industrially

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_8,

useful compounds constitutes one of the many options available to us for the development of sustainable biofuel energy and for the reduction in environmental wastes from chemical processing.

However, many of the needed chemicals are not made by any organism. How then do we harness the exquisite capabilities of biology but make compounds that have not been made in nature before? The answer most likely lies in the massive amount of available biochemical information (9, 10). This information is far too complex for manual design of new pathways, as has been the major strategy to date. Rather, computational approaches to design new metabolic pathways have risen (11, 12). Here, we review several recent computational tools, all focusing on this very problem—taking large-scale biochemical data and using it to better inform the design of synthetic metabolic pathways in unicellular organisms. We present ten software programs, describing a wide range of functionalities and approaches to the question of engineering reaction pathways (Table 1). We note that this collection is not a definitive list and other flavors of metabolic pathway design can be found.

1.1. Biochemical Data Sources: The Kyoto Encyclopedia of Genes and Genomes

To design a metabolic pathway, one first needs a source of biochemical data. The Kyoto Encyclopedia of Genes and Genomes (KEGG), an online database of biochemical reactions and their corresponding enzymes and genes, is one of the largest repositories of continuously updated, verified metabolic data available (13, 14). Because it is such a large database, it is a critical resource for scientists and engineers interested in exploiting biochemistry, and from the perspective of computational tools in this chapter, KEGG very often serves as *the* source for all available metabolism to incorporate into organism models or to use in potentially novel pathways. Though most of the programs are capable of linking up to any metabolic database, KEGG is almost always the one used.

1.2. Strategies for Finding Pathways

The main obstacle to pathway discovery is that of complexity, both from the large amount of metabolic reaction data (e.g., KEGG) and from the complex state of the organism. Online databases contain intractable amounts of enzyme/reaction information for a human to determine a pathway, and the search is complicated by the inherent interconnectedness of cellular metabolism.

The two main classes overcome the issue of complexity in a different way. Graph theory-based approaches perform analysis on the reaction databases directly. Biochemical data is broken down into edges and nodes. Most commonly compounds are nodes and reactions are edges, but this can vary depending on the approach. Finding paths is then a matter of following the different routes of the graph and trying to get from the starting compound to the product. Instead of using the data of online databases directly, rule-based methods generalize those reactions into reaction rules and use those rules to map out reaction paths to and from different

Table 1
Summary of computational metabolic pathway design software

Program name	Type of program	Latest publication[a]	Run-time scales	Summarized points	
BNICE	Rule based	Wu et al. (2011) (20)	Minutes to hours	• Biochemical reactions pruned by hand from online databases, reactions generalized into "operator files" based on EC system	• Can predict novel chemistries, suite of "modules" to prune generated networks, incorporates thermodynamic calculations into reactions and MFA
PathPred	Rule based	Tokimatsu et al. (2011) (26)	Minutes to hours	• Online service offered by KEGG, KEGG reactions described by RDM patterns, RDM patterns cluster according to metabolism	• Classes of RDMs/RPAIRs used to simulate reactions, can predict novel chemistry, predictions based on structural similarity rankings, regularly updated
UM-PPS	Rule based	Ellis et al. (2012) (28)	Seconds	• Online service offered by UM-BBD, focused on xenobiotic metabolism, reactions described by generalized database of btrules, fast	• Regularly updated btrules applied to substrates, predict chemistry, can predict novel chemistry, metabolic "logic entries" prune networks, user can set parameters
PPC	Graph theory—probabilistic	Yousofshahi et al. (2011) (35)	Seconds to minutes	• Operates on/necessitates large reaction database, probabilistic selection of edges based on connectivity, incorporates FBA	• Equivalent functionality/results as extensive search, terminates at compounds in organismal model
DESHARKY	Graph theory—probabilistic	Rodrigo et al. (2008) (38)	Seconds	• Operates on/necessitates large reaction database, unweighted probabilistic selection of edges, back-step probability increasing with path length	• Calculates "loads" on cellular system to represent impact of novel pathway, metabolic load = FBA, transcriptional/translational load = kinetic/chassis model, fast

(continued)

Table 1 (continued)

Program name	Type of program	Latest publication[a]	Run-time scales	Summarized points	
ReTrace	Graph theory—atom mapping	Pitkanen et al. (2009) (43)	Minutes to hours	• Creates atom-graph instance of a reaction database, incorporates RPAIR patterns to describe reactions, maximizes atom conservation	• Conservation measured in Z_O metric, optimized *k*-shortest path search approach for fastest computation, successfully finds branching and linear pathways
PathMiner	Graph theory—atom mapping	McShan et al. (2005) (44)	Seconds	• Create "state space" of a reaction database; compounds = states, reactions = state transformations; vectorized description of the database	• Heuristics-based pathway search, use A^* metric of chemical distance as heuristical tool, additional metrics incorporated in edge cost *e*, fast
MetaRoute	Graph theory—atom mapping	Blum et al. (2008) (42)	Seconds to minutes	• Semiautomated creation of reaction rules/sets from digital databases, enzyme clustering score used to inform rule creation	• *k*-lightest path search of weighted atom graph, graph and pathways are weighted toward atom conservation, fast
OptStrain	Linear programming based	Pharkya et al. (2004) (48)	Unknown	• Universal database provides reactions for pathway discovery, multiple databases in one, linear programming techniques find pathways through yield maximization	• Minimizes necessary heterologous expression, uses accessory program (OptKnock) to tie production to biomass growth, pathway optimization based on yield

[a]Most recent publication is the most recent article found from the original authors as of writing.

compounds. In a way, rule-based methods distill the online database information down and use it to create their own, more focused reaction databases focused on the starting/target compounds. By capturing the observed chemistry in these rules, the algorithm can predict new compounds and pathways that are not found in KEGG.

2. Computational Pathway Discovery Tools

2.1. Rule-Based Tools

A significant difference between rule-based methods and the other types discussed in this chapter lies in the ability to predict novel chemistry. While other methods can only reorganize known reactions and compounds, rule-based methods use those same databases to generalize biochemistry in terms of independent reaction rules. These rules are used to generate their own databases of reactions. The rule-based systems are an approach that bridges the often overlapping fields of metabolic engineering and synthetic biology through the inclusion of novel biochemistry into pathway discovery. These chemistries may be indicative of unidentified enzymatic activities or may provide potential targets for protein engineering to alter substrate specificity. An illustrative example of rule creation, as well as the rule's application to generate novel chemistry, is shown in Fig. 1.

Here, we will review the Biochemical Network Integrated Computational Explorer (BNICE), KEGG PathPred system, and the University of Minnesota Biodegradation and Biocatalysis Database's Pathway Prediction System (UM-PPS).

2.1.1. Biochemical Network Integrated Computational Explorer

The Biochemical Network Integrated Computational Explorer (BNICE) is a rule-based system which can carry out both analysis and synthesis: Analytical tools focus on finding all paths among known metabolites, while synthesis tools allow for the identification of novel intermediate compounds and reactions (15).

Inputs and Operation

BNICE consists of four modules: NetGen, thermodynamics, pathway, and thermodynamics-based metabolic flux analysis (TMFA). NetGen predicts enzymatic reactions and products based on generalized reaction rules, and its output serves as the input data for the rest of the program's features. Thermodynamics, pathway, and TMFA are all pruning, or analytical, modules written to take the initial network of NetGen and further analyze it to identify desired reactions and pathways from the initial pool.

Chemical reactions are reproduced by files called operators, which are used to predict enzymatic reactions. These operators have been hand distilled from enzymatic databases, like KEGG and the University of Minnesota Biodegradation/Biocatalysis Database (UM-BBD). Operators are named and generalized

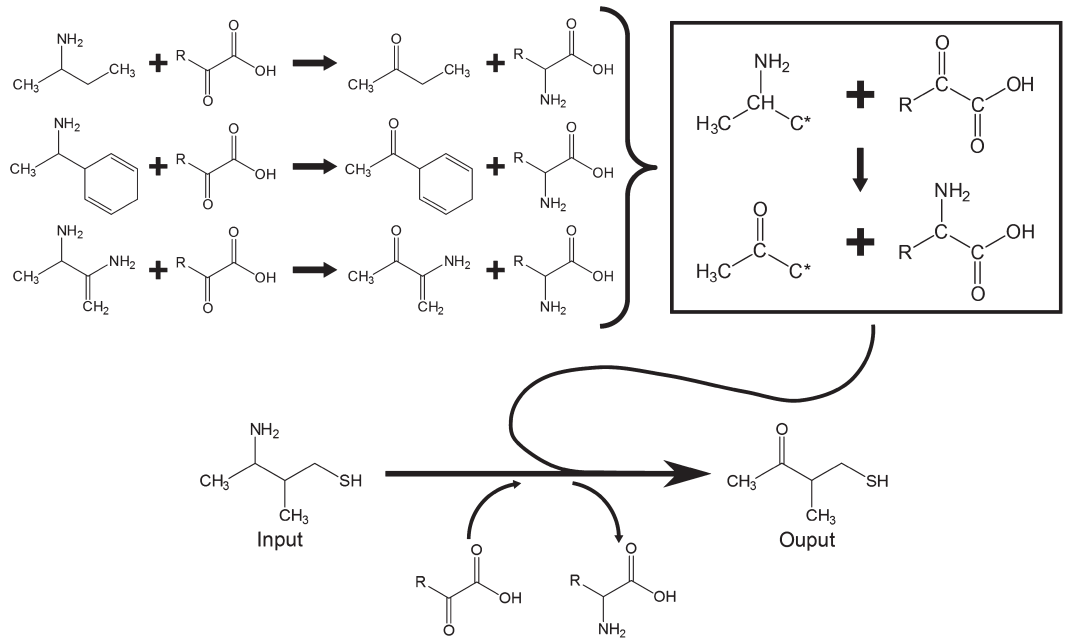

Fig. 1. An example of the rule-distillation process used in rule-based methods. Three reactions with very similar chemistries are compared, and the representative structures are used to describe the chemistry. This can then be applied to a novel substrate to potentially predict biochemistry.

according to their enzyme commission (EC) classification numbers. Rather than operators describing the *chemistry* and the *specific substrate*, the operator creation focuses on generalizing enzymatic reactions that contain the same first three EC digits and thus involve very similar chemistry, but could use different substrates.

Reactions in NetGen are simulated with a bond–electron matrix (BEM). Required reaction sites are defined in the individual operator files in a symmetric $N \times N$ matrix, where N is the number of atoms required in the reaction site. The elements of the matrix represent the bond order between overlapping rows and columns (e.g., a number 2 in an element of row O and column C would describe a double bond between a specific carbon and specific oxygen atom in the molecule). Having used the BEM to describe the required reaction site, the operator files then use an identically sized matrix with positive and negative integers to describe the making/breaking of bonds. The operators are able to cover a large spread of potential chemistries, all based on known biochemical transformations.

To generate its networks, BNICE requires an input of starting compounds, a list of operators to use in the network creation, and the number of generations to run. The pruning modules explained in the next section use output from NetGen.

Search Algorithm

Running BNICE for many generations can easily lead to a reaction network of hundreds of thousands of reactions. For this reason, we have developed the pathway, thermodynamics, and TMFA modules to take the large networks, remove undesired reactions, and provide additional useful information about the simulated biology to provide direction for potential experimental implementation.

Beginning from a starting compound, NetGen scans possible operators that can act on the start compound and generates a new pool of molecules based on the operators' chemistries. This can run for several generations to create many possible reactions. Work can also be done in retrosynthetic analysis, working backward from a product using retrosynthesis operators, which are operators with reversed directions of the initially defined chemistry (reactions that are considered physiologically reversible are handled similarly, but with both directions included in the full operator pool).

Given this initial biochemical network, pathway or thermodynamics can be used to uncover connections between pairs of desired compounds and approximate the thermodynamic changes of the reactions of interest. Pathway performs a basic depth-first search for linear pathways between the user-defined start and end points of the pathway given a maximum path length. Thermodynamics uses a group contribution method (GCM) to approximate ΔG_r across a reaction from changes in substructures (16), which have been assigned individual ΔG_f values. Using these two modules, thermodynamically favorable pathways of reasonable length are culled from the network. Lastly, to help choose from those proposed paths, TMFA can be used with each set of reactions to find those with the highest product yield, highest biomass, or other bioprocessing benchmarks. TMFA performs a flux balance analysis (FBA) of the effects from pathway integration into an organismal model, but with thermodynamic constraints on fluxes to better inform metabolism-scale effects of the pathway (17). FBA analyzes an organismal model and calculates maximum product yields as well as altered biomass rates that result from the introduction of heterologous reactions (see the OptStrain section for more details of FBA). In TMFA, metabolite activities are found, and optimal starting metabolite concentrations are suggested based on the thermodynamics.

Pathway Evaluation and System Validation

BNICE has been successfully applied to specialty chemical production (18), biodegradation of xenobiotics and environmental toxins (19), amino acid synthesis pathways (15), and biofuel production (20). Successes in these projects have independently reproduced known biological pathways and predicted novel biosynthesis routes that were already implemented in industrial settings. BNICE can be applied to a wide range of applications—novel chemistry prediction, native pathway discovery, and alternative pathway discovery. The program is written in C++ and can be run on Windows or Unix systems.

2.1.2. Cho Systems Framework

Cho et al. have implemented the BEM strategy of the BNICE algorithm and have extended it in several useful ways (21). Cho's algorithm is focused on retrosynthesis and has included modules for using chemical similarity, both for entire molecules and for substructures of a molecule, a group contribution thermodynamic analysis, pathway distance, and organism reaction specificity to help improve the selection of potentially useful pathways. Cho's algorithm has successfully predicted pathways for the synthesis of isobutanol, butyryl-CoA, and, like BNICE, 3-hydroxypropanoate.

2.1.3. PathPred

PathPred (22) is a system that utilizes the KEGG RPAIR and RDM databases, an atom-mapping rule-like system that uses KEGG's own data to break down reactions into reaction pairs with smaller rule descriptors for proposing novel metabolic pathways.

Inputs and Operation

The RPAIR database simplifies the reactions of the KEGG database and classifies reactions by reaction rules similar to BNICE, which are termed RDM patterns (for (R)eaction center atoms, (D)ifferent atoms, or (M)atched atoms, described below). The collection of these RDM patterns, and the reactant–product pairs described by the rules, is the KEGG RPAIR database (23). Reactants and products are compared and matched into reaction pairs based on a chemical similarity approach before a manual curation ensures proper pairing.

To create the RDM patterns, paired structures are compared and the overlapping substructures identified. The R atoms are those in the overlap region but on the border, hence where the reaction occurs. The D atoms are those bound to the R atoms but not in the overlap region. The M atoms are those bound to the R atoms in the overlap region. The RDM patterns describe how these three atom types change across a reaction pair and are meant to fully describe the chemistry performed in that pair (24). There are several RPAIR types that distinguish between different reaction pairings: *main*, *cofac*, *trans*, *ligase*, and *leave* pairs. PathPred utilizes only the RDM patterns from *main* pairs, which describe the pairings meant to be the focus of a particular reaction.

When the RPAIR database was first published (22), there were 7,091 reactant pairs described by 2,205 RDM patterns, with the bulk of those RDM patterns (64 %) each describing a single reaction pair.

Search Algorithm

PathPred predicts pathways from an observed clustering of RDM patterns to certain classes of metabolism (25). When running PathPred, the user must choose a type of metabolism—xenobiotic degradation or biosynthesis of secondary plant metabolites. By choosing a specific class, the program will use the associated RDM patterns. This allows for more reasonable computation times and more accurate predictions.

After selecting the type of desired prediction, the user inputs several starting parameters and is able to start a run of PathPred. Depending on the approach, either a starting compound or final compound is required (catabolism and anabolism, respectively), and additional data about chemical similarity thresholds and the number of prediction cycles can be set as well. After loading the compound, the PathPred algorithm analyzes the compound. First, it performs a similarity comparison between the input compound and the full database of KEGG compounds, looking for potential matches within a user-set threshold similarity value. Next, PathPred searches through all of the RDMs of those matched compounds and finds all RDMs that are applicable to the initial input compound. Third, the starting compound is subjected to the RDM transformations for those that matched the structural requirements. These last two steps are repeated until all transformations have been exhausted, at which point the compounds generated will be used in the first step, and the whole process is repeated for however many prediction cycles the user has specified.

Predicted pathways and reactions are ranked according to two different scoring schemes: reaction and pathway scores. The reaction score is a similarity index measure of how structurally close the compound input into the first step is to the compound that the RDM pattern is designed to act on. The pathway score is an average of the reaction scores contained within it. Compounds at the end of pathways with high pathway scores are targeted for the repeated prediction cycles.

Pathway Evaluation and System Validation

In their paper introducing the program, the authors of PathPred used their system to predict one biodegradation (1,2,3,4-tetrachlorobenzene to glycolate) and one biosynthesis (delphinidin to gentiodelphin) process, one each for the two different available sets of RDM patterns (22). The biodegradation exercise matched a documented path from the UM-BBD and found several other paths. The biosynthesis found several paths but not the known biological route, as a necessary RDM pattern was not a *main* pair and thus neglected from the process. More recently, PathPred was also successfully used to predict plant biosynthesis of fraxidin from umbelliferon (26) with several intermediary compounds known to be present in the *Saposhnikovia* root, a known biological source.

PathPred is freely available online through the KEGG database at the following address: http://www.genome.jp/tools/pathpred/. At the time of publication, PathPred was available in version 1.13.

2.1.4. University of Minnesota Pathway Prediction System

The University of Minnesota has developed one of the premier biodegradation databases, the UM-BBD. With this plethora of information, they have also taken steps to create a predictive biodegradation software program, which they have called the

University of Minnesota Pathway Prediction System (UM-PPS) (27). The program has been publicly available since late 2002/early 2003 and has seen continual development since its release, which we detail here.

Inputs and Operation

The actual use of the program is very straightforward and consists of only a few starting steps. First, the user inputs a starting compound through either a MarvinDraw applet or a SMILES string. The user is initially also given the option to limit the search to aerobic reactions. PPS then will generate and display the network in a short directed acyclic graph.

Search Algorithm

One of the original issues with the UM-PPS was that it required informed user intervention for each step. This required some knowledge of microbial biodegradation preferences in order to choose proper steps in generating a pathway. To overcome this, the software developers have implemented five network control features which capture much of the expert knowledge that was previously required, which they call "metabolic logic entries": absolute aerobic likelihood, immediate feature, relative reasoning, super rules, and variable aerobic likelihood. Full details about these features can be found elsewhere (28–30).

The core of UM-PPS is based on the distillation of the chemistries contained within the UM-BBD. The program uses generalized reaction rules, which they have termed biotransformation rules (btrules), that represent a large portion of the chemistries found on the database.

Using the chemistry of the UM-BBD, the authors of UM-PPS currently have 250 btrules for pathway prediction. These btrules are all designed to recognize and react with 50 predefined functional groups that have been distilled from the available reactions and are common across many xenobiotic metabolic reactions. When searching for potential reaction candidates, UM-PPS first performs a selection step, where btrules are matched to potential reactive sites, and then the reactions are carried out in the biotransformation step. Reaction rules are designed to be as generalized as possible, as long as the actual chemistry or known metabolism does not prevent this.

The UM-PPS and btrules are not written to describe any individual bacterium. The reactions contained within the UM-BBD (and the subsequent generalized btrules) are described based on known environmental degradation. The reactions in the database come from a wide variety of organisms and environmental observations. This is justified because of "increasing evidence that [xenobiotic degradation] is often consortial" (27). The UM-PPS is written with the intent of predicting whole-scale breakdown of xenobiotics, not necessarily the breakdowns occurring within a specific microbe. It is important to keep this in mind if using this

tool for metabolic engineering purposes, as btrule steps from different organisms may necessitate engineering beyond the introduction of proteins to a host cell.

Pathway Evaluation and System Validation

UM-PPS was validated in three ways upon publication (27). The program was able to recreate 72 % of the documented reactions of the UM-BBD at the time and gave at least one known pathway for 98 % of all UM-BBD compounds. It also reproduced five out of six biodegradation pathways as predicted by biodegradation experts for non-UM-BBD compounds. As an *in vivo* verification, three compounds predicted to release ammonia were successfully used as nitrogen sources in three cultures of soil-sampled bacteria.

UM-PPS has also been benchmarked against KEGG PathPred (discussed previously in this chapter) for biodegrative prediction (28). Not only did the UM-BBD perform equally or better for all tested compounds when compared to PathPred, but it also correctly predicted 81 % of the biodegradation routes.

The UM-PPS is freely available online for all users at the following link: http://umbbd.msi.umn.edu/predict/. The UM-BBD parent site offers all of the information about the btrules, use of PPS, all of the documented reactions on the site, and which btrules correspond to those reactions.

2.2. Graph-Based Tools: Probabilistic Approaches

Graph-based approaches find optimal or nonnative paths between substrates and products from a preexisting reaction network. There are many potential pools of reactions for these tools. In addition to the already discussed KEGG database, other options are MetaCyc (31), the UM-BBD (32), and organismal models (e.g., the *Escherichia coli iAF1260* model, (33)). Graph theory approaches take these large databases of metabolites and reactions and break them up into nodes and edges, where edges are directed arrows connecting individual nodes as illustrated in Fig. 2. Due to limitations in computational power and available time, the complexity of these massive reaction databases makes it intractable to search through every possible connection. For a breadth or depth-first search, the worst-case computational time is $O(|V| + |E|)$ (34) which, in the case of our biological networks, is a function of the total number of available nodes. For a pathway of length d in a network with an average node connectivity of b edges, the time will then be $O(b^d)$ which can rapidly become overly complex in the large, interconnected databases (e.g., KEGG).

Probabilistic analysis looks at a network and makes a decision about which edge to follow in a network, based on a probability weighting heuristic, which can vary with the individual program's approach. The two presented methods here also utilize biological models for the ranking of pathways. The two programs that we have chosen to describe are Probabilistic Pathway Construction (35) and DESHARKY (36).

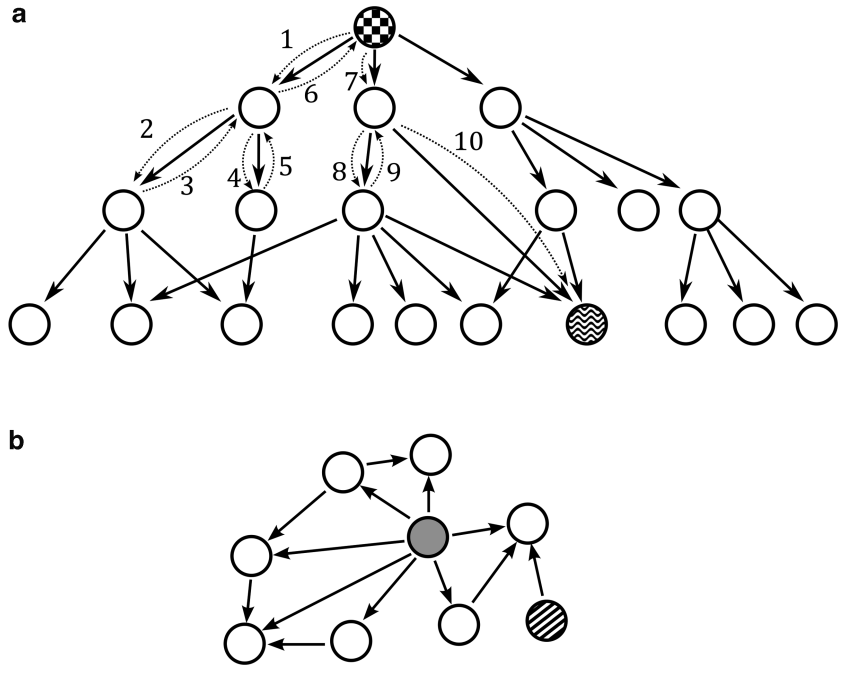

Fig. 2. Illustration of generalized graph-based analysis. (**a**) A depth-first analysis of a graphical tree. The *checkerboard and wave nodes* are the starting and target compounds, respectively. The *arrow numbers* illustrate the search with a maximum depth of 2. (**b**) Another graphical illustration demonstrating connectivity of nodes. The *gray circle* has a high connectivity, while the *striped one* a low connectivity. *Gray* could likely be a currency compound like NADH.

2.2.1. Probabilistic Pathway Construction

One option for pathway analysis of large reaction database networks is Probabilistic Pathway Construction (PPC). PPC utilizes a probabilistic graph-based search method to take large metabolic networks and find the most relevant pathways between two targeted compounds, searching for nonnative biosynthesis pathways. After discovering potential novel metabolic pathways, PPC then uses FBA to calculate the maximum yields of the desired product through all proposed pathways, subject to a biomass production constraint of 80% of that of the wild-type model.

Inputs and Operation

PPC requires three inputs: the biosynthesis product, a multi-organism reaction database to search (e.g., KEGG), and an organismal model for the expression host for FBA. The multi-organism database provides PPC potential production routes, while the organismal model gives PPC a list of native metabolites that can serve as potential pathway starting points.

Search Algorithm

PPC uses a modified depth-first graph search method to find pathway connections. A depth-first network search proceeds in steps going to nodes of deeper generations. When the search hits a stop signal (a node with no further edges or a search depth limit), it backtracks a step and proceeds to the next unexplored node.

When all edge searches from a given node have been exhausted, the program will retreat additional steps until it encounters novel edges to explore (Fig. 2a).

In PPC, molecules are treated as nodes and reactions leading to those nodes as edges. The program first looks to the designated biosynthesis product for all reactions resulting in its synthesis. It then uses a probabilistic selection to choose a reaction step. Having selected a reaction that produces the product of interest, PPC then looks to the starting substrates of that reaction and sets those as the new products, searching for reactions to those molecules with the same probabilistic scoring.

The research group tested probabilistic preferences for high connected nodes, low connected nodes, and for no inherent bias for connectivity (uniform) (Fig. 2b). In their work, the authors found that the best results were obtained with uniform connectivity.

PPC will continue to search in this manner unless it hits a predefined maximum pathway length, it encounters a compound that has already been included into the proposed pathway, or if the pathway encounters a metabolite that is native to the predefined host organism. It treats the first two cases as if it were a node with no further edges. For the third, it records the valid pathway and continues the search.

It is important to note that, being a system based on probabilities, the program must be run through many iterations in order to obtain an accurate representation of potential pathways. In the group's analysis, reliable results were returned between 500 and 1,500 iterations—the maximum yields stabilized at 500, but the average yields increased until 1,500. These results were found across several different types of synthesis products and the multiple scoring mechanisms.

Pathway Evaluation and System Validation

Having found several pathways, PPC is then able to place the paths into the organismal model and approximate theoretical maximum yields in an FBA analysis, which necessitates the selection of an organism model (37).

As validation of the pathways discovered by their program, the group looked to prior literature. While many predicted paths either had no experimental work to be found or reported production values not directly relatable to the maximum yield, there were several successful pathways previously implemented by others. These paths had been predicted independently by PPC and had comparable yield predictions.

The PPC system has a significant advantage over exhaustive searches in that it can save an enormous amount of time on longer path analyses. PPC's run-time scales linearly with respect to the number of reactions within the multi-organism database. An exhaustive search, however, becomes exponentially large with respect to the maximum pathway length. PPC's developers found

that to find a pathway of 23 steps through an exhaustive search would take approx. 400 years, while an identical search with PPC took a mere 6 min. Likewise, the results of a ten-step probabilistic search found pathways with essentially the same maximum theoretical yields as the equivalent exhaustive search. However, in this analysis, the authors have constrained themselves to a single metric of efficacy. Future studies and experimental validation will prove if additional metrics for successful pathway identification are necessary. As will be seen throughout this chapter, there currently exist many ideas about which metrics will accurately predict a pathway's success.

2.2.2. DESHARKY

DESHARKY was developed to find routes of either biosynthesis or biodegradation of compounds that are available in the KEGG database (36). The program relies heavily on the KEGG listings for all potential reactions to use in the pathways and uses a relatively simple pathway search. But where DESHARKY is unique is that it takes this a step further and uses the predicted growth rate of an organism as an indicator of how successful a given pathway will be in a physiological setting. This growth rate is determined under two independent systems, which the authors have classified as the transcriptional–translational load model and the metabolic model.

Inputs and Operation

This program contains a starter list of reactions, compounds, and enzymes from the KEGG database. The user can customize the simulation by updating this list, changing growth media components, or adding metabolites into an organism. If a compound of interest is already present within the host, then the user should input instead a set of desired termination host compounds.

To run, the program requires host organism compound data and the target compound. Biosynthesis will treat the target compound as a product, and biodegradation will use it as a substrate. Other adjustable parameters include number of iterations and maximum path length, among others.

Search Algorithm

DESHARKY takes the target compound and finds potential pathways to/from a host organism's metabolites to the input molecule. The pathway discovery is done in an unweighted probabilistic manner based on graph theory, similar to the approach taken by PPC. However, to avoid long pathways and improve convergence, each step after the first has an additional probability to retreat one step. This reversal probability increases with the number of forward steps that have been taken. Because it is a probabilistic pathway determination method, DESHARKY must be run over many iterations (the default is one million) in order to try and find all potential pathways. The program assumes all reactions of KEGG are reversible to allow both

biosynthesis and biodegradation (36). Metabolic steps with many nonnative compounds to the host organism are also disregarded.

Pathway Evaluation and System Validation

A significant challenge for novel pathway creation is not just finding the pathway but also evaluating the effects of pathway implementation on the physiological state of the organism. The biological consequences of pathways—consuming native metabolites, cytotoxic effects, and others—are often not evident within the pathways themselves. DESHARKY solves this with two independent measures of the altered cellular load resulting from a nonnative pathway: transcription–translation load and metabolic load.

Transcription–translation load estimates the negative cellular effects resulting from the resource drain for nucleic acid and enzyme polymerization, particularly the effects on RNA polymerase and nucleotide availabilities. DESHARKY uses a set of equations to tie growth rate to transcription and translation loads based on experimental measurements and first-order kinetics.

The model uses the available amino acid sequences on KEGG and an empirical mathematical cellular-chassis (organism) model they have developed to estimate reductions in growth rate arising from the heterologous pathway enzymes (38). This type of estimation of physiological effects is unique among pathway prediction systems. Metabolic load is estimated through a standard FBA approach, as used in several other programs. The two load calculators each give an independent assessment of the physiological impact from the novel enzymes introduced into the organism and provide two perspectives on cellular pathway integration, whereas most approaches would analyze only metabolic burden.

DESHARKY takes only a few seconds for each full run. The code is easily amenable to both distributed computing and modified weightings for the probabilistic searching. The program is open source, written in C/C++, and runs in UNIX environments (http://soft.synth-bio.org/desharky.html).

2.3. Graph-Based Tools: Atomistic Mapping

When searching for a pathway, one obstacle faced by graph-based tools is the effect of compounds with high degrees of connectivities (Fig. 2b) which can waste significant computational resources exploring all available edges. These compounds, like ATP or NADH, are often referred to as pool or currency metabolites and are involved in many biological reactions serving canonical roles: providing either oxidoreductive potentials or functional groups. One common solution to the high connectivity metabolite problem is to remove those nodes from the networks, though this prohibits the analysis of any pathways for the synthesis of those currency metabolites or that of any reactions where the compounds perform noncanonical roles.

The atom-tracing approach seeks to sidestep the high connectivity metabolite problem by following how bonds are made and

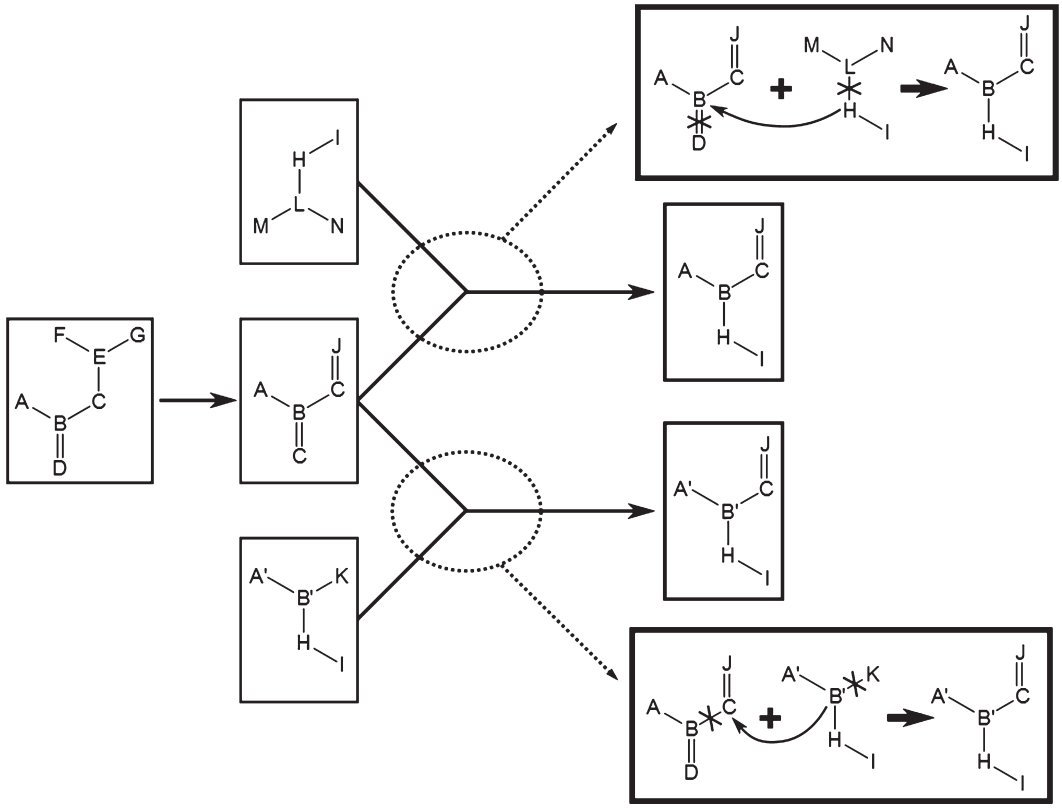

Fig. 3. A simplified graph theoretical analysis with atom mapping incorporated. Note how, although both routes yield the same product, in the *bottom path* the only atom maintained from the starting compound is C, whereas in the *top one* A, B, and C are all conserved.

broken across reactions, similar to that done by the rule-based approaches but in a much more reaction-specific manner. For most cases, currency metabolites donate relatively few atoms across a reaction (e.g., NADH is only involved in the transfer of protons and electrons). Atom-mapping rule sets can detect which substrate/product pairs have the most atoms in common, and pathway searches can focus on edges containing the highest degree of atomistic conservation, as is illustrated in Fig. 3. This strategy was pioneered by Arita (39).

These programs are able to make more informed pathway discoveries than the probabilistic tools we have discussed, with the added information from atom conservation driving pathway discovery. However, this focus may come at the cost of exploring more energetically favorable paths or pathways of significantly shorter length. The programs we have chosen to detail here are ReTrace (40), PathMiner (41), and MetaRoute (42).

2.3.1. ReTrace

ReTrace merges graph theory tracing with the specificity of atom tracing. The goal is for more accurate and concise pathway predictions—overcoming the difficulties faced by many with respect to "irrelevant connections" that often result from the searches focused on whole molecules. ReTrace also identifies branching pathways, a feature other pathway discovery tools often overlook (40).

Inputs and Operation

The program relies on KEGG for potential reaction steps and also utilizes the KEGG RPAIR database (discussed in the PathPred section) for information about where individual atoms move across reactions. This requires that a local version of the KEGG LIGAND data be downloaded from the parent website. The user inputs the source and target metabolites, and there are also several options with preset default values that the user can change for additional control.

ReTrace creates a graph of reactions connecting a starting and ending compound, using atom conservation to prioritize the most viable paths.

Search Algorithm

The overall ReTrace algorithm is made up of three procedures that are run sequentially: ReTrace, FindPath, and FindPathStart. ReTrace creates the atom and reaction graph that is used to find potential pathways. Again, KEGG is the biochemical database of choice. The graph constructed here is different from many others as both the reactions and the individual atoms of the compounds are represented by nodes, with edges describing how the individual atoms change compound locations across reactions. ReTrace constructs this graph from the database, also incorporating the stated target and source compound atoms into the map. The RPAIR reaction pairings are used to describe the core atoms for reactions listed in KEGG and how their bonds change. The program constructs a graph of source and target atom nodes and all reactions/edges of those atoms within a given number of reaction steps.

FindPath uses this graph for pathway construction and optimization, using the heuristic of atomistic conservation to rank them. Optimal predicted pathways should minimize the number of "dangling substrates," or substrates that are used in reactions of the path but enter into the pathway as foreign compounds. This helps minimize unknowns in metabolic reactions (uncertainty on the availability of a given compound), as well as yield much higher metabolic efficiencies, as energy is not wasted on removing and adding back atoms that are necessary.

Following the completion of this atom-traced graph, FindPath first finds the initial pathways within the reaction network. It runs a *k*-shortest path analysis between the source and target compounds, with the procedure FindPathStart being used to match final compound atoms to their starting compound atoms. FindPath looks through the top *k*-shortest paths to find those in which atoms are

and are not fully traced, and pathways where most atoms can be traced to source compounds are stored for further analysis. For those pathways with "unresolved atoms," or atoms untraced to one of the preset starting compounds, the program performs repeated FindPath analyses with those unresolved atoms to add more edges into the network. This is done until the program either runs out of edges to add or the atom conservation score increases beyond a preset minimum value.

Pathway Evaluation and System Validation

The authors have benchmarked their program using 13 metabolites as both source and target molecules (i.e., for each target, there were 12 sources). Computation times ranged from 1,000 s (CO_2) to 16,000 s (CMP-*N*-acetylneuraminate), the number of average discovered pathways to targets ranged from <30 (CO_2) to about 1,700 (CMP-*N*-acetylneuraminate), and the average path lengths ranged from five steps (CO_2) to 33 (*p*-Coumaroyl-CoA). The authors also showed they could find novel pathways from glucose to inosine-5′-monophosphate. Many pathways were found, including a known prokaryotic synthesis route, and routes using several alternative starting compounds were also identified. ReTrace was also used to reconstruct previously unknown pathways for the construction of amino acid carbon backbones in *T. reesei*, with success for many predicted pathways (43).

ReTrace has several unique aspects as compared to other graph-based approaches. By utilizing graph searching (which is fast, but prone to connectivity artifacts) to atomistic mapping (which is slower, but detailed), ReTrace capitalizes on each strategy's strength while minimizing its weakness. The other advantage to ReTrace is its ability to analyze branching pathways. Other graph-based tools make simplifying assumptions to pathway construction in a purely linear manner. By defining reactions and compounds as nodes, ReTrace can analyze all types of reactions—linear or branching—equally. Python code for ReTrace is freely available from http://www.cs.helsinki.fi/group/sysfys/software/retrace/ and requires the KEGG LIGAND database.

2.3.2. PathMiner

PathMiner tweaks the graph-based approach by using vectors of atoms to create "biochemical state spaces" and uses heuristic searching. Though the current implementation relies on KEGG, this approach could easily incorporate novel chemistries or other databases to predict new biochemical reactions (44).

Inputs and Operation

In PathMiner (41), the KEGG reactions and compounds are put into a biochemical state space, where compounds are the different states and the reactions constitute state transitions. This is analogous to a graph-based approach, but using vector notation. Each compound is described by a state vector **x**. Based on biochemistry, the authors have created a set of 145 unique atoms and atom bond

structures (e.g., C, O, S atoms with possible bond structures of C=O, O–S, S–S, among many others). The vector **x** contains 145 elements, and the different numbers within it detail the compound and its chemical structure. An example of this is given as $\mathbf{x}^{CO2} = [(C1)(O2)(C{=}O2)\ldots]$. Each reaction is a vector **t**, which is determined by the vector difference between states of reactant and product. Because individual compounds can perform many different chemistries, any individual **x** will likely have several **t**'s coming from it.

Combining the **t**'s with the **x**'s gives virtually the same result as would be found from a graph theory approach, but with the added benefit of the atomistic state descriptors. These 145 state elements could be easily be expanded to potentially include nonmolecular information about the compounds such as thermodynamic values (41).

Search Algorithm

With a state space defined, the authors take a computer science-based approach and view the discovery of new pathways as a classical state-search problem. An uninformed search of this space would be intractable, opening the door for heuristics to provide a quicker, informed analysis.

Chemical similarity of a molecule's state to the target molecule is used to determine which transitions/reactions are likely to lead toward the product of interest. For chemical similarity, this involves an additive combination of the distance traveled G (i.e., chemical similarity between start compound and present molecule) and the distance not yet traveled H (i.e., chemical similarity between target compound and present molecule) such that $F = G + H$. This is computed from the 145 atomic descriptors in what is essentially a state-space-based similarity index search. As these distances represent the costs of the system, the best paths are those with minimal F values. By expanding the fitness function F to include additional "edge cost" descriptors ($F = G + H + e$) for availability of precursors, heterologous vs. homologous enzymes, and others, much more sophisticated heuristics were incorporated into PathMiner (44).

After the database is fully characterized and assembled, PathMiner only requires a starting and ending compound. Searches are done by exploring all of the state transitions from each step that yields the best fitness values. The program terminates when there are no more states to explore and outputs all of the paths that were found between the two specified compounds.

Pathway Evaluation and System Validation

PathMiner's heuristic approach generally outperformed both uninformed breadth-first and depth-first searches for several different metabolic "themes"(41). In predicting vanillin synthesis (44), PathMiner found the native pathway and was able to suggest alternative host organisms for synthesis: *Brucella melitensis* or *Streptomyces coelicolor* over *E. coli*.

PathMiner uses state-space approaches and chemical distance to efficiently search known metabolic databases. Using the state-space approach, new parameters can be added to bias the network search and could be easily extended to other metabolic databases and even to potentially novel/predictive chemistries.

2.3.3. MetaRoute

MetaRoute (42) combines atom mapping and graph theory analysis with *k*-lightest path analysis to discover metabolic routes from large enzymatic databases. Here, a lightest path search will look for the pathways with the lowest connectivity values. The atom-mapping approach analyzes reactions and tracks where individual atoms go throughout a stated pathway.

Inputs and Operation

MetaRoute uses atom tracing analogous to ReTrace, where pathways with the highest number of atoms that are conserved from starting to final compound are given high scores. However, while ReTrace used RPAIR/RDM for its rules, MetaRoute developed its own rules in an automated fashion by looking for molecular substructures that are the same in reactant and product and then deducing the atoms that take part in the reaction. The KEGG and EcoCyc (45) databases were used to construct the rules (46). However, using this substructure approach can lead to redundancy because of repeated functional groups or substructures across different parts of a molecule. This redundancy in rules is solved by comparing atom-tracing reactions for enzymes clustered together based on the EC numbers. Presumably, similarly clustered enzymes should perform similar chemistries and thus have similar atom-tracing reactions. After multiple atom-tracing reactions are compared within a cluster, the atom-tracing reaction that predicts the most reactions in the cluster is chosen.

Search Algorithm

MetaRoute uses a modified *k*-lightest path search to discover novel pathways between two predefined compounds. The graph used is the reverse of the typical reaction graph, as MetaRoute uses nodes as reactions and edges as compounds. The novelty of the MetaRoute approach lies in the integration of the atom mappings into the *k*-lightest path search in what the authors have termed a "weighted atom-mapping graph." This involves two parts—the weighting of certain nodes and a structural moiety constraint. Each reaction (node) in the network has been pre-analyzed and is weighted by the atom transfer across the reaction. The *k*-lightest search maximizes the atom transfer during the search while subject to a minimum structural moiety constraint. This strategy can fail in the case of a reaction node with no atom mappings. As only 63% of the KEGG database has available reaction rules, the pathway search could potentially stall if no atoms can be traced to any products. When this happens, the program makes a choice based on the

k-lightest path criteria and then restarts the structural moiety constraint with the next compound.

Pathway Evaluation and System Validation

MetaRoute was used in a glycolysis search from D-glucose-6-phosphate to pyruvate. MetaRoute found three pathways: one documented glycolysis route and two novel paths. One of the novel pathways used an enzyme not yet classified into any documented pathways at the time of the paper's publication and had one fewer reaction than typical glycolysis. This is a strong case to use these types of computational tools for not just pathway creation but also for prediction of as yet-unknown biochemical pathways. MetaRoute has an intuitive interface, is easy to use, and is available online for free at http://abi.inf.uni-tuebingen.de/Services/MetaRoute/.

2.4. Linear Programming Tool

One approach that shares some similar approaches to graph-based methods but optimizes paths in a very different manner is the OptStrain program which uses linear programming (LP)/FBA-based tools (47).

2.4.1. OptStrain

OptStrain relies on online reaction databases for novel chemical steps and reactions to construct a pathway. Going beyond just pathway design, it will subsequently make suggestions about which genes from the native organism should be knocked out or added to enhance production.

Inputs and Operation

OptStrain uses a universal database, constructed from KEGG, UM-BBD, MetaCyc, and more, that is updated automatically. Only stoichiometrically balanced reactions from the databases are used in the search. The program also uses an organism model, which serves as the metabolic environment that novel reactions are introduced to. The program will attempt to maximize the production of a target compound given a large set of potential starting substrates.

Search Algorithm

OptStrain approaches the pathway search as an LP problem rather than a node/edge search problem. As with FBA, the universal reaction database is organized as a stoichiometric matrix **S**, with rows representing metabolites and columns representing reactions, where the element (n,m) has the stoichiometric coefficient of the nth metabolite within the mth reaction (48). Given a target product, OptStrain solves the universal matrix as an LP problem to maximize yields from a given substrate set by identifying a set of reactions that will achieve the desired biotransformation.

After identifying a set of possible reactions, OptStrain minimizes the number of new enzymes that would need to be added to the host organism. A mixed-integer linear programming problem, where heterologous enzymes to the host of interest are differentiated from native enzymes, is carried out to maximize the number of heterologous reactions that are removed while maintaining the

maximum product yield. In the last step, OptStrain utilizes another tool developed by the same authors, OptKnock (49), which does not design new pathways but attempts to optimize the host metabolism to maximize yield of the target compound through gene/reaction knockouts.

Pathway Evaluation and System Validation

In initial work, the authors optimized amino acid synthesis and found that seven amino acids could be synthesized by alternative pathways that were more energy efficient than native pathways (50). Subsequently, the authors analyzed hydrogen and vanillin synthesis (49). The first case exemplified OptStrain's ability to look to many different organisms and potential starting substrates, while the second succeeded most in minimizing heterologous genes for successful pathways. The hydrogen work predicted that *E. coli* could produce hydrogen but not in a manner coupled with growth rate. *Clostridium acetobutylicum* was identified as an additional candidate for glucose→hydrogen production that was tightly coupled with growth rate. Vanillin production in *E. coli* was predicted *de novo* with alterations similar to prior work by other researchers; however, OptStrain indicated much higher yields could be achieved with several knockouts not employed by the experimentalists.

3. Concluding Remarks

We have presented here a wide range of different computer programs of potential use in metabolic engineering. As society has a greater demand for sustainable chemicals, designing *de novo* metabolic pathways will become increasingly important. The different approaches and programs have their advantages and disadvantages. Graph-based methods can analyze known biochemistry and bring to light potentially unimaginable combinations of reactions to yield new ways to think about metabolism. This can be used to further pathway creation, as well as to inform scientists on bridging current gaps of metabolism. Several strategies, such as probabilistic searching, atom tracing, and formulating the search as an LP problem, have been successful in identifying pathways out of very large biochemical databases. However, these graph-based approaches rely completely on documented biochemical reactions and cannot predict unobserved but feasible biosynthesis pathways. To design metabolic pathways for compounds not observed to be produced by enzymes, rule-based approaches are essential. By sampling known biochemistry and distilling it into common reaction types, rule-based approaches can predict the potential reactions that could be performed on a given substrate. However, rule-based results are inherently high risk, as predicted chemistry may not be possible.

Regardless of the approach, computational tools for pathway design in metabolic engineering provide powerful methods for realizing novel biochemical processes. The tools described here, and others, demonstrate a diversity of approaches to pathway design. All of the programs, upon their release, presented some form of validation or verification of the program's efficacy for pathway prediction, typically by showing the program's ability to predict already known pathways. While few examples to date have taken a designed pathway and demonstrated that it could work (2), we expect to see a strategy of computer-aided design of metabolic pathways, with implementation, to have an increasing prominence in the design of new synthesis processes for new chemicals.

References

1. Shen CR, Liao JC (2008) Metabolic engineering of *Escherichia coli* for 1-butanol and 1-propanol production via the keto-acid pathways. Metab Eng 10:312–320
2. Yim H, Haselbeck R, Niu W, Pujol-Baxley C, Burgard A, Boldt J, Khandurina J, Trawick JD, Osterhout RE, Stephen R, Estadilla J, Teisan S, Schreyer HB, Andrae S, Yang TH, Lee SY, Burk MJ, Van Dien S (2011) Metabolic engineering of *Escherichia coli* for direct production of 1,4-butanediol. Nat Chem Biol 7:445–452
3. Kind S, Jeong WK, Schroder H, Wittmann C (2010) Systems-wide metabolic pathway engineering in Corynebacterium glutamicum for bio-based production of diaminopentane. Metab Eng 12:341–351
4. Trantas E, Panopoulos N, Ververidis F (2009) Metabolic engineering of the complete pathway leading to heterologous biosynthesis of various flavonoids and stilbenoids in *Saccharomyces cerevisiae*. Metab Eng 11:355–366
5. Ajikumar PK, Xiao WH, Tyo KE, Wang Y, Simeon F, Leonard E, Mucha O, Phon TH, Pfeifer B, Stephanopoulos G (2010) Isoprenoid pathway optimization for Taxol precursor overproduction in *Escherichia coli*. Science 330:70–74
6. Ro DK, Paradise EM, Ouellet M, Fisher KJ, Newman KL, Ndungu JM, Ho KA, Eachus RA, Ham TS, Kirby J, Chang MC, Withers ST, Shiba Y, Sarpong R, Keasling JD (2006) Production of the antimalarial drug precursor artemisinic acid in engineered yeast. Nature 440:940–943
7. Keasling JD (2012) Synthetic biology and the development of tools for metabolic engineering. Metab Eng 14:189–195
8. Stephanopoulos G, Stafford DE (2002) Metabolic engineering: a new frontier of chemical reaction engineering. Chem Eng Sci 57:2595–2602
9. Pennisi E (2005) How will big pictures emerge from a sea of biological data. Science 309:94
10. Philippi S, Kohler J (2006) Addressing the problems with life-science databases for traditional uses and systems biology. Nat Rev Genet 7:482–488
11. Copeland WB, Bartley BA, Chandran D, Galdzicki M, Kim KH, Sleight SC, Maranas CD, Sauro HM (2012) Computational tools for metabolic engineering. Metab Eng 14:270–280
12. Medema MH, van Raaphorst R, Takano E, Breitling R (2012) Computational tools for the synthetic design of biochemical pathways. Nat Rev Microbiol 10:191–202
13. Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M (2012) KEGG for integration and interpretation of large-scale molecular data sets. Nucleic Acids Res 40:D109–D114
14. Kanehisa M, Goto S (2000) KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res 28:27–30
15. Hatzimanikatis V, Li C, Ionita JA, Henry CS, Jankowski MD, Broadbelt LJ (2005) Exploring the diversity of complex metabolic networks. Bioinformatics 21:1603–1609
16. Jankowski MD, Henry CS, Broadbelt LJ, Hatzimanikatis V (2008) Group contribution method for thermodynamic analysis of complex metabolic networks. Biophys J 95:1487–1499
17. Henry CS, Broadbelt LJ, Hatzimanikatis V (2007) Thermodynamics-based metabolic flux analysis. Biophys J 92:1792–1805
18. Henry CS, Broadbelt LJ, Hatzimanikatis V (2010) Discovery and analysis of novel metabolic pathways for the biosynthesis of industrial

chemicals: 3-hydroxypropanoate. Biotechnol Bioeng 106:462–473

19. Finley SD, Broadbelt LJ, Hatzimanikatis V (2010) In silico feasibility of novel biodegradation pathways for 1,2,4-trichlorobenzene. BMC Syst Biol 4:7
20. Wu D, Wang Q, Assary RS, Broadbelt LJ, Krilov G (2011) A computational approach to design and evaluate enzymatic reaction pathways: application to 1-butanol production from pyruvate. J Chem Inf Model 51:1634–1647
21. Cho A, Yun H, Park JH, Lee SY, Park S (2010) Prediction of novel synthetic pathways for the production of desired chemicals. BMC Syst Biol 4:1–16
22. Moriya Y, Shigemizu D, Hattori M, Tokimatsu T, Kotera M, Goto S, Kanehisa M (2010) PathPred: an enzyme-catalyzed metabolic pathway prediction server. Nucleic Acids Res 38:W138–W143
23. Kotera M, Okuno Y, Hattori M, Goto S, Kanehisa M (2004) Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions. J Am Chem Soc 126:16487–16498
24. Oh M, Yamada T, Hattori M, Goto S, Kanehisa M (2007) Systematic analysis of enzyme-catalyzed reaction patterns and prediction of microbial biodegradation pathways. J Chem Inf Model 47:1702–1712
25. Oh M, Yamada T, Hattori M, Goto S, Kanehisa M (2007) Systematic analysis of enzyme-catalyzed reaction patterns and prediction of microbial biodegradation pathways. J Chem Inf Model 47:1702–1712
26. Tokimatsu T, Kotera M, Goto S, Kanehisa M (2011) KEGG and GenomeNet resources for predicting protein function from omics data including KEGG PLANT resource. Protein Function Prediction for Omics Era, 271–288
27. Hou BK, Ellis LBM, Wackett LP (2004) Encoding microbial metabolic logic: predicting biodegradation. J Ind Microbiol Biotechnol 31:261–272
28. Ellis L, Wackett L (2012) Use of the University of Minnesota biocatalysis/biodegradation database for study of microbial degradation. Microb Inform Exp 2:1
29. Fenner K, Gao J, Kramer S, Ellis L, Wackett L (2008) Data-driven extraction of relative reasoning rules to limit combinatorial explosion in biodegradation pathway prediction. Bioinformatics 24:2079–2085
30. Gao JF, Ellis LBM, Wackett LP (2010) The University of Minnesota biocatalysis/biodegradation database: improving public access. Nucleic Acids Res 38:D488–D491
31. Caspi R, Foerster H, Fulcher CA, Hopkinson R, Ingraham J, Kaipa P, Krummenacker M, Paley S, Pick J, Rhee SY, Tissier C, Zhang PF, Karp PD (2006) MetaCyc: a multiorganism database of metabolic pathways and enzymes. Nucleic Acids Res 34:D511–D516
32. Ellis LBM, Hou BK, Kang WJ, Wackett LP (2003) The University of Minnesota biocatalysis/biodegradation database: post-genomic data mining. Nucleic Acids Res 31:262–265
33. Feist AM, Henry CS, Reed JL, Krummenacker M, Joyce AR, Karp PD, Broadbelt LJ, Hatzimanikatis V, Palsson BO (2007) A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol Syst Biol 3:1–18
34. Reif JH (1985) Depth-1st search is inherently sequential. Inform Process Lett 20:229–234
35. Yousofshahi M, Lee K, Hassoun S (2011) Probabilistic pathway construction. Metab Eng 13:435–444
36. Rodrigo G, Carrera J, Prather KJ, Jaramillo A (2008) DESHARKY: automatic design of metabolic pathways for optimal cell growth. Bioinformatics 24:2554–2556
37. Papoutsakis ET (1984) Equations and calculations for fermentations of butyric-acid bacteria. Biotechnol Bioeng 26:174–187
38. Carrera J, Rodrigo G, Singh V, Kirov B, Jaramillo A (2011) Empirical model and in vivo characterization of the bacterial response to synthetic gene expression show that ribosome allocation limits growth rate. Biotechnol J 6:773–783
39. Arita M (2000) Metabolic reconstruction using shortest paths. Simulat Pract Theor 8:109–125
40. Pitkanen E, Jouhten P, Rousu J (2009) Inferring branching pathways in genome-scale metabolic networks. BMC Syst Biol 3:103
41. McShan DC, Rao S, Shah I (2003) PathMiner: predicting metabolic pathways by heuristic search. Bioinformatics 19:1692–1698
42. Blum T, Kohlbacher O (2008) MetaRoute: fast search for relevant metabolic routes for interactive network navigation and visualization. Bioinformatics 24:2108–2109
43. Jouhten P, Pitkanen E, Pakula T, Saloheimo M, Penttila M, Maaheimo H (2009) (13)C-metabolic flux ratio and novel carbon path analyses confirmed that Trichoderma reesei uses primarily the respirative pathway also on the preferred carbon source glucose. BMC Syst Biol 3:1–16

44. McShan D, Shah I (2005) Heuristic search for metabolic engineering: de novo synthesis of vanillin. Comput Chem Eng 29:499–507
45. Keseler IM, Collado-Vides J, Santos-Zavaleta A, Peralta-Gil M, Gama-Castro S, Muniz-Rascado L, Bonavides-Martinez C, Paley S, Krummenacker M, Altman T, Kaipa P, Spaulding A, Pacheco J, Latendresse M, Fulcher C, Sarker M, Shearer AG, Mackie A, Paulsen I, Gunsalus RP, Karp PD (2011) EcoCyc: a comprehensive database of *Escherichia coli* biology. Nucleic Acids Res 39:D583–D590
46. Blum T, Kohlbacher O (2008) Using atom mapping rules for an improved detection of relevant routes in weighted metabolic networks. J Comput Biol 15:565–576
47. Pharkya P, Burgard AP, Maranas CD (2004) OptStrain: a computational framework for redesign of microbial production systems. Genome Res 14:2367–2376
48. Fell DA, Small JR (1986) Fat synthesis in adipose-tissue—an examination of stoichiometric constraints. Biochem J 238:781–786
49. Burgard AP, Pharkya P, Maranas CD (2003) OptKnock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. Biotechnol Bioeng 84:647–657
50. Burgard AP, Maranas CD (2001) Probing the performance limits of the *Escherichia coli* metabolic network subject to gene additions or deletions. Biotechnol Bioeng 74:364–375

Chapter 9

Retrosynthetic Design of Heterologous Pathways

Pablo Carbonell, Anne-Gaëlle Planson, and Jean-Loup Faulon

Abstract

Tools from metabolic engineering and synthetic biology are synergistically used in order to develop high-performance cell factories. However, the number of successful applications has been limited due to the complexity of exploring efficiently the metabolic space for the discovery of candidate heterologous pathways. To address this challenge, retrosynthetic biology provides an integrated framework to formalize and rationalize the problem of importing biosynthetic pathways into a chassis organism using methods at the interface from bottom-up and top-down strategies. Here, we describe step by step the process of implementing a retrosynthetic framework for the design of heterologous biosynthetic pathways in a chassis organism. The method consists of the following steps: choosing the chassis and the target, selection of an *in silico* model for the chassis, definition of the metabolic space, pathway enumeration, gene selection, estimation of yields, toxicity prediction of pathway metabolites, definition of an objective function to select the best pathway candidates, and pathway implementation and verification.

Key words: Synthetic biology, Metabolic engineering, Retrosynthesis, Metabolic pathway

1. Introduction

Production of value-added compounds such as drugs or biofuels in chassis organisms often requires importing heterologous genes to build efficient biosynthetic pathways. Computational techniques have provided useful methods for the development of such cell factories through a streamlined process for modeling and design of a complete pathway at each step of its construction. Recent advances in systems and synthetic biology are further enabling a systematic practice through an integrated computational framework for rational biosynthetic pathway design. Nowadays, genome-scale metabolic network reconstructions providing an accurate *in silico* model of the metabolism are available for many industrial chassis organisms (1). In such models, the effects on steady-state fluxes of metabolic interventions like enhancement of substrate uptake, supplements addition, reduction of undesirable by-products fluxes, introduction of heterologous pathways, or product

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_9,

export to the extracellular medium (2) can be predicted with a remarkable degree of agreement with experimental observations (3). Furthermore, an increasingly formalization of the space of biochemical transformations in metabolic networks is allowing the designer to explore creative ways to implement alternative biosynthetic pathways (4).

To that end, metabolic modeling for heterologous pathway design can be done from two complementary approaches: a topological approach using hypergraphs, where catalytic reactions are hyperedges connecting node substrates to products, and a steady-state approach, where stoichiometry of reactions is used in order to study the properties of all feasible equilibrium states. Knowledge-based comparative analysis, graph search algorithms, and constraint-based models are alternative approaches used in order to infer meaningful pathways in metabolic networks, even if there is missing information on the enzymes.

Several metabolic databases with rich information are available: one of the most comprehensive is MetaCyc and its associated BioCyc collection of pathway/genome databases (5); similarly, KEGG is a database resource that integrates genomics, chemical, and systemic functional information (6); BRENDA is another database that contains one of the most complete collections of enzyme functional data (7). Gaps or incomplete knowledge, however, are still present in many cases, especially when looking for novel ways to synthesize compounds. In this regard, computational approaches can provide new alternatives by predicting putative heterologous pathways producing the target compound. In order to successfully handle this challenging task, the design process needs to be rationalized by following the principles of synthetic biology: modeling of the biological system of interest, modular design through standardization, goal-oriented optimization, and experimental validation. To contribute to this endeavor, we present here a retrosynthetic design approach that aims to provide a streamlined methodology for addressing the general problem of obtaining successful high-yield production of target compounds in cell factories.

1.1. The Retrosynthetic Framework for Heterologous Pathway Design

Retrosynthesis algorithms are applied to metabolic networks in order to perform a backward search from the target compound to the host metabolites through the iterative application of a defined set of biochemical transformation rules. Depending on the level of atomic resolution of those rules, recruited enzymes in the biosynthetic pathways may involve novel compound intermediates and putative reactions with unknown efficiency. A successful expression of those genes in the chassis organism needs to be addressed. Therefore, subsequent optimization of the engineered strain through genetic, metabolic, and enzyme design approaches would be usually necessary in order to maximize production yields of the target.

We present here a unified framework that combines several techniques involved in the design of heterologous biosynthetic pathways through a retrosynthetic biology approach, enabling by these means the flexible design of industrial microorganisms for the efficient on-demand production of chemical compounds of interest. The method for retrosynthetic design of heterologous pathways consists of the following steps:

1. First, the problem is defined by choosing the chassis organism and the target compound.
2. Second, an *in silico* reconstructed model of the organism containing at least the stoichiometric reactions involved in its metabolism is defined from biological databases and literature.
3. Third, the metabolic space is constructed from all known metabolic reactions and expanded to putative promiscuous reactions.
4. Fourth, heterologous pathways producing the target compound from endogenous metabolites in the chassis are enumerated using a retrosynthetic algorithm.
5. Fifth, gene sequences encoding heterologous enzymes are chosen in order to maximize gene expression and enzyme performance in the chassis organism.
6. Sixth, steady-state fluxes for each pathway are estimated through flux balance analysis.
7. Seventh, toxicity of intermediate metabolites is estimated by using a QSAR model.
8. Eighth, a cost function is defined for the pathway and the best pathways are chosen.
9. Ninth, selected pathways are implemented and their efficiency is verified.

2. Materials

The following list provides a review of the main metabolic engineering tools used at each step of the heterologous pathway design, including some specific tools for retrosynthesis design:

1. Choosing the chassis: a host organism optimized for metabolic engineering, such as strains from *Escherichia coli*, *Bacillus subtilis,* or *Saccharomyces cerevisiae*.
2. Selecting an *in silico* model of the chassis organism: from repositories of *in silico* model organisms like the databases BIGG (8) or BioModels (9).

3. Construction of the metabolic space: metabolic databases such as MetaCyc (5) or KEGG (6) and enzymatic activity databases such as BRENDA (7).
4. Pathway enumeration: software MetaHype (10).
5. Gene selection: genomics databases (e.g., UniProt, NCBI Entrez) focused on protein (enzyme) families.
6. Flux balance analysis: metabolic analysis software (e.g., COBRA (11), OptFlux (12), COPASI (13)).
7. Metabolite toxicity data for the chassis, either experimental or predicted (e.g., EcoliTox (14)).
8. Definition of a final cost function (e.g., RetroPath (15)).
9. Experimental implementation:
   (a) Molecular biology reagents for PCR and cloning
   (b) Bacterial strains for cloning and expression
   (c) Expression vectors
   (d) Growth media
   (e) Analytical techniques for protein identification (electrophoresis gel)
   (f) Chromatography system
   (g) Analytical system for metabolite identification

3. Methods

The design methodology for any metabolic engineering application starts with the selection of a chassis organism along with an associated genome-scale *in silico* model of its metabolic network (see Note 1). The retrosynthetic approach offers as well the possibility of performing an additional preliminary modeling step to expand the starting metabolic reaction space and, thus, increasing the possibility of discovering novel biosynthetic routes. These prior steps will provide the designer with a detailed knowledge base about the metabolic system that can be used advantageously at later stages of the design. In the same fashion, target compounds need to be defined at this stage.

A basic methodology for retrosynthetic design of heterologous pathways will consist of the following steps (Fig. 1): (1) choosing the chassis, (2) selecting an *in silico* model for the chassis, (3) definition of the metabolic space, (4) pathway enumeration, (5) gene selection, (6) estimation of yields, (7) toxicity prediction of pathway metabolites, (8) definition of an objective function to select the best pathway candidates, and (9) pathway implementation and verification.

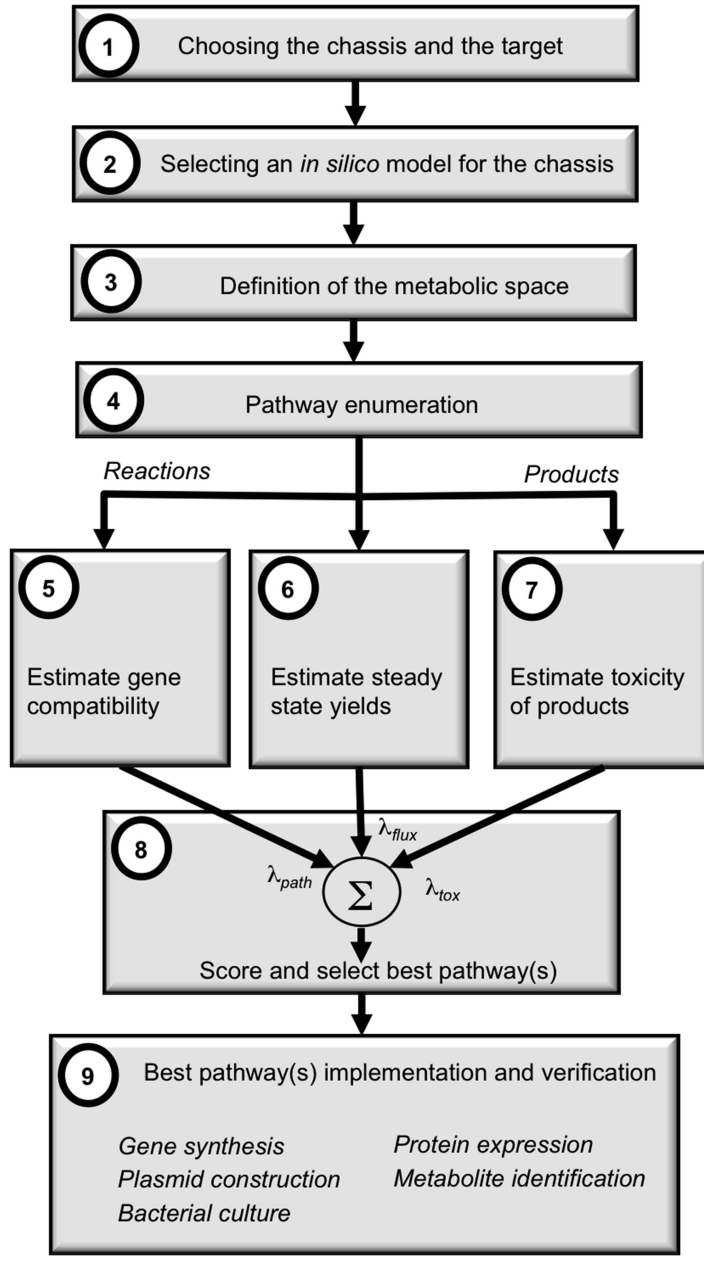

Fig. 1. Flowchart of the retrosynthetic methodology for heterologous pathway design. The process consists of nine steps: It starts by choosing the chassis organism and the target compound, followed by a selection of an *in silico* model of the chassis, and the definition of the metabolic space. The process continues with pathway enumeration; for each reaction in the pathway, genes are selected based on their compatibility to the host, and a penalty is added if the reaction appears as a putative promiscuous reaction. Next, steady-state yields are estimated, as well as toxicity of reaction products. These values are combined in order to score and select the best pathway(s), which are finally implemented for verification.

3.1. Choosing the Chassis

An early decision that necessarily influences the rest of the retrosynthetic design process is the choice of the chassis organism where the desired compound will be produced. For example, in order to increase the production of a compound naturally produced in plants, its biosynthetic pathway, if known, is imported into an industrial chassis organism. Factors that need to be considered when choosing the chassis include the following:

1. The extent and level of curation of the organism's metabolic pathways in databases.
2. The availability of a genome-wide reconstructed *in silico* model that has been experimentally verified (16) and that is ready to be used in constraint-based modeling to quantitatively estimate steady-state fluxes (see Note 2).
3. The availability of information about toxicity effects of heterologous metabolite intermediates in the organism.
4. The fact that biosynthetic pathways may involve large enzymatic complexes (such as polyketide synthases or non-ribosomal synthases for secondary metabolite synthesis) (17).
5. Similarly, specific redox reactions catalyzed by the CYP450, which is often needed in the last steps of metabolite synthesis, add another layer of complexity because of the difficulty to model these reactions and often a need to optimize further its catalytic activity through protein engineering (18).

3.2. Selecting an In Silico Model for the Chassis

In silico organism models are currently available for many industrial strains, including strains evolved for efficient production in *E. coli*, *S. cerevisiae*, or *B. subtilis* (19–21). Most of them have been deposited in open databases such as BIGG (8) or BioModels (9), and numerous tools exist for their analysis and simulation (3).

Example of Chassis Selection for Production of Resveratrol in E. coli. Resveratrol (3,5,4′-trihydroxy-trans-stilbene) is a plant phenolic compound with important associated health benefits like prevention of cardiovascular diseases, cancer, and promotion of longevity in several animal systems (22). Resveratrol, however, is only found in a limited number of plant species, including grape (*Vitis spp.*) and peanut (*Arachis hypogaea*). Because of its beneficial properties, there is an increasing interest in the optimization of the production of resveratrol in microorganisms (23, 24). Interestingly, *E. coli* provides an industrial chassis organism with one of the best characterized *in silico* models (19). As shown in Fig. 2, production of resveratrol is derived from phenylalanine that is transformed into cinnamic acid by phenylalanine ammonia lyase (PAL, EC 4.3.1.24). Next, cinnamic acid is transformed into 4-coumaric acid by cinnamate-4-hydroxylase (C4H, EC 1.14.13.11), which is further transformed into coumaroyl-CoA by the 4-coumarate:coenzyme A (CoA) ligase (4CL, EC 6.2.1.12). Then, the stilbene synthase (STS, EC

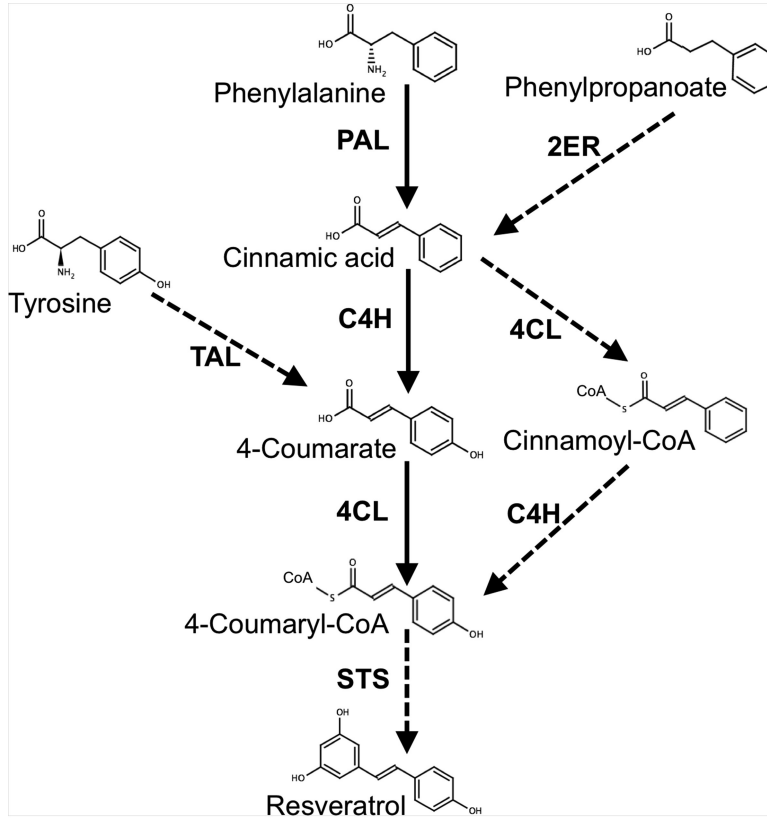

Fig. 2. Outline of phenylpropanoid alternative pathways producing resveratrol. In the center, the backbone of the native pathway is shown, consisting of phenylalanine ammonia lyase (PAL), cinnamate-4 hydroxylase (C4H), 4-coumarate:CoA ligase (4CL), and stilbene synthase (STS), the last step. 4-coumarate can be produced by an alternative pathway with the tyrosine ammonia lyase (TAL), a promiscuous reaction of PAL. Other predicted alternative routes are 2-enoate reductase (2ER) producing cinnamic acid from phenylpropanoic acid and 4CL and C4H using alternative substrates cinnamic acid and cinnamoyl-CoA, respectively.

2.3.1.95) condenses the coumaroyl-CoA and three units of malonyl-CoA to form resveratrol (23). In the rest of this chapter, we will present the different steps of a retrosynthetic design methodology for metabolic engineering of resveratrol production in *E. coli*.

3.3. Definition of the Retrosynthetic Metabolic Space

The power of retrosynthesis for heterologous pathway design resides in the way representations of chemical biotransformations can provide a generalization of important chemical features. The most valuable information (but also the most challenging) obtained from retrosynthesis analysis is the identification of putative metabolic pathways involving promiscuous biochemical transformations that often had not yet been well annotated. Several techniques have been proposed in order to expand the metabolic reaction space to contain such putative reactions. The basic idea is

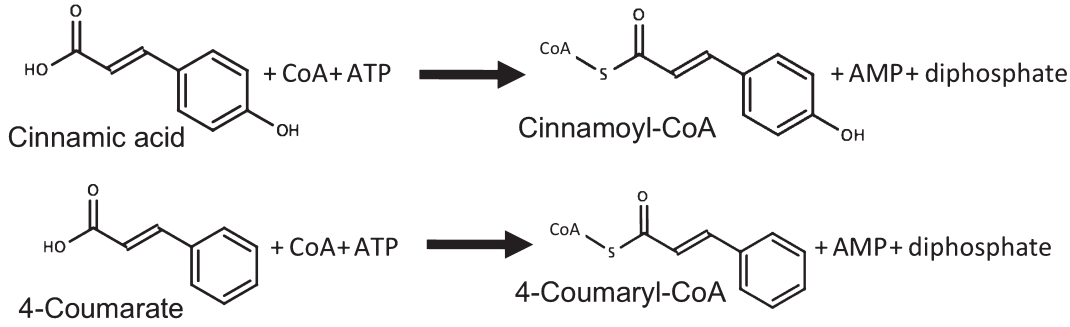

Fig. 3. Example of two promiscuous reactions generated by the same reaction rule defined by the 4CL enzyme (EC 6.2.1.12). In the first case, substrate 4-coumaric acid is converted into 4-coumaroyl-CoA, while in the second case, cinnamic is converted into cinnamoyl-CoA. Both, substrates and products, differ in the 4-hydroxy group, but the net balance of bonds that are formed or broken is identical for both reactions.

to use a set of reaction rules from which not only known reactions can be generated but also novel reactions. To define reaction rules, several representations have been used, such as those derived from bond–electron matrices (25) (see Note 3) or on the smallest molecular substructure that can be modified through the transformation (26). The combinatorial complexity associated with such representations is a major issue that we have recently addressed by proposing a trade-off solution based on molecular signatures (15). The main advantage of the molecular signature method relies on the control of the complexity of the pathway search through the selection of the level of specificity in the reaction representation.

For any reaction found through reaction rules, there are two essential questions to consider: (a) Is it a putative reaction or has it been reported previously in the databases? (b) Is there any known enzyme sequence annotated for such biochemical transformation? In addition, as much information as possible about the selected reaction needs to be collected, including if it has been observed for some enzyme as a promiscuous or side reaction, or what are their kinetics constants, cell localization, or phylogenetic diversity. Main sources for such information are enzymatic databases like BRENDA (7) and metabolic databases like MetaCyc (5) or KEGG (6).

Example of Metabolic Space Expansion. 4-coumarate:CoA ligase (4CL), an enzyme involved in the phenylpropanoid biosynthesis, attaches 4-coumaric acid to the pantetheine group of coenzyme A (CoA) to produce 4-coumaroyl-CoA, the precursor of resveratrol. Besides this native reaction, however, 4CL is reportedly able to catalyze promiscuously, among others, reactions producing caffeoyl-CoA from caffeate, feruloyl-CoA from ferulate, sinapoyl-CoA from sinapate, and cinnamoyl-CoA from cinnamic acid (27). Interestingly, substrate cinnamic acid is transformed into cinnamoyl-CoA, which may also serve as a precursor for the production of resveratrol. As shown in Fig. 3, both reactions can be derived from a single reaction rule, since the net balance of bonds that are formed and broken is identical for both reactions.

3.4. Enumerating Heterologous Pathways

The metabolic space that has been defined in the previous step consists of both endogenous and heterogeneous reactions. In order to produce exogenous compounds, the corresponding metabolic routes containing heterologous enzymes that start from endogenous metabolites must be found (see Note 4). Two methods can be applied in order to list all possible pathways leading to the target compound (10) (see Note 5): steady-state and topological methods.

Pathway Enumeration Through the Steady-State Approach. The problem of enumerating heterologous pathways can be approached by using the well-known metabolic engineering technique of computing elementary modes in a metabolic network (28). By definition, any pathway producing a target compound can be formed by some positive linear combination of the elementary modes. Here, we are focusing on a specialized version of elementary modes studies. Namely, we are interested in finding all pathways connecting endogenous metabolites to the target compound through heterologous enzymes. In particular, it is essential to correctly define what are the inputs, outputs, and the stoichiometric matrix of the metabolic system as follows:

1. Input: any metabolite that can be produced in the chassis organism.
2. Output: any metabolite that is produced and is not further consumed by the heterologous network.
3. Stoichiometric matrix: it is given by the reactions that are heterologous to the chassis organism.

Special care needs to be taken with those endogenous metabolites that are also produced in the heterologous network (usually cofactors and currency metabolites such as ATP, NAD, and protons, which participate in a large number of reactions (29), but also byproducts of the biosynthesis), since they will appear in the system defined above as both inputs and outputs, generating thus elementary modes containing loops. An easy way to prevent the enumeration of these return loops is by replacing endogenous products by a generic end node sink (see Note 6).

Under this setup, each elementary mode will correspond to a pathway that produces heterologous compounds. Because the number of pathways needs to be kept minimal, all pathways of interest producing a target compound should be contained in the elementary modes. In addition, elementary modes containing loops should not be considered as pathway candidates. Several software packages are available that compute the elementary modes of a given metabolic network. A popular implementation is Metatool (28).

Pathway Enumeration Through the Topological Approach. In the topological approach, each reaction is represented by a node of a hypergraph that is connected through hyperedges to the substrates

(see Note 7). The strategy used in the hypergraph approach for pathway enumeration is the application of a recursive backward algorithm that traverses the network starting from the target in order to search for all possible pathways connecting the target to the source (see Note 8).

The main advantage of this approach is computational efficiency. Another remarkable feature of the topological approach is that it allows for supplements and bootstraps molecules identification. Supplements are compounds that provide new biosynthetic pathways if added to the medium because they act as precursors of the target compounds. Bootstraps are compounds that are needed to be present in the medium at least in small amounts in order to allow the reactions in the pathway to start producing the target compound. A software tool for enumerating pathways using the topological approach is MetaHype (10).

Example of Pathway Enumeration of Resveratrol Producing Pathways in E. coli. Starting from precursors in *E. coli*, five alternative viable pathways producing resveratrol are identified by the retrosynthetic approach (shown in Fig. 4). One of the pathways consists of three enzymatic steps, while the rest contain four enzymes. The question that is investigated about these pathways in the next sections is how to prioritize them depending on their expected performance, as a preliminary step before selecting the ones that would eventually be implemented.

3.5. Gene Compatibility for Expression in the Host

Heterologous enzymes of the pathway need to be successfully expressed. Gene compatibility with the expression host is crucial, although it remains still a challenging task. Facilities proposing gene synthesis with codon optimization (30) have been developed in the past few years and are often used for metabolic engineering. In order to select the gene, several strategies are possible:

1. Rare codons and GC content are known parameters that can influence gene expression, as well as RNA secondary structure (30). In addition, other parameters such as sequence length or hydrophobicity might also influence a successful gene expression.
2. Homology search of heterologous genes: A blast search of the National Center of Biotechnology Information (NCBI) nucleotide data bank can identify sequences predicted to encode the enzyme having the desired activity. Phylogenetic trees can be built to identify groups of the different enzymes identified (31). Minimizing the phylogenetic distance between the chassis organism and the organism where the gene is endogenous can also help in order to choose the homologue enzyme.
3. Scoring gene compatibility: An adequate strategy for gene sequence selection is to associate a score to each sequence so that only sequences with top score are further considered.

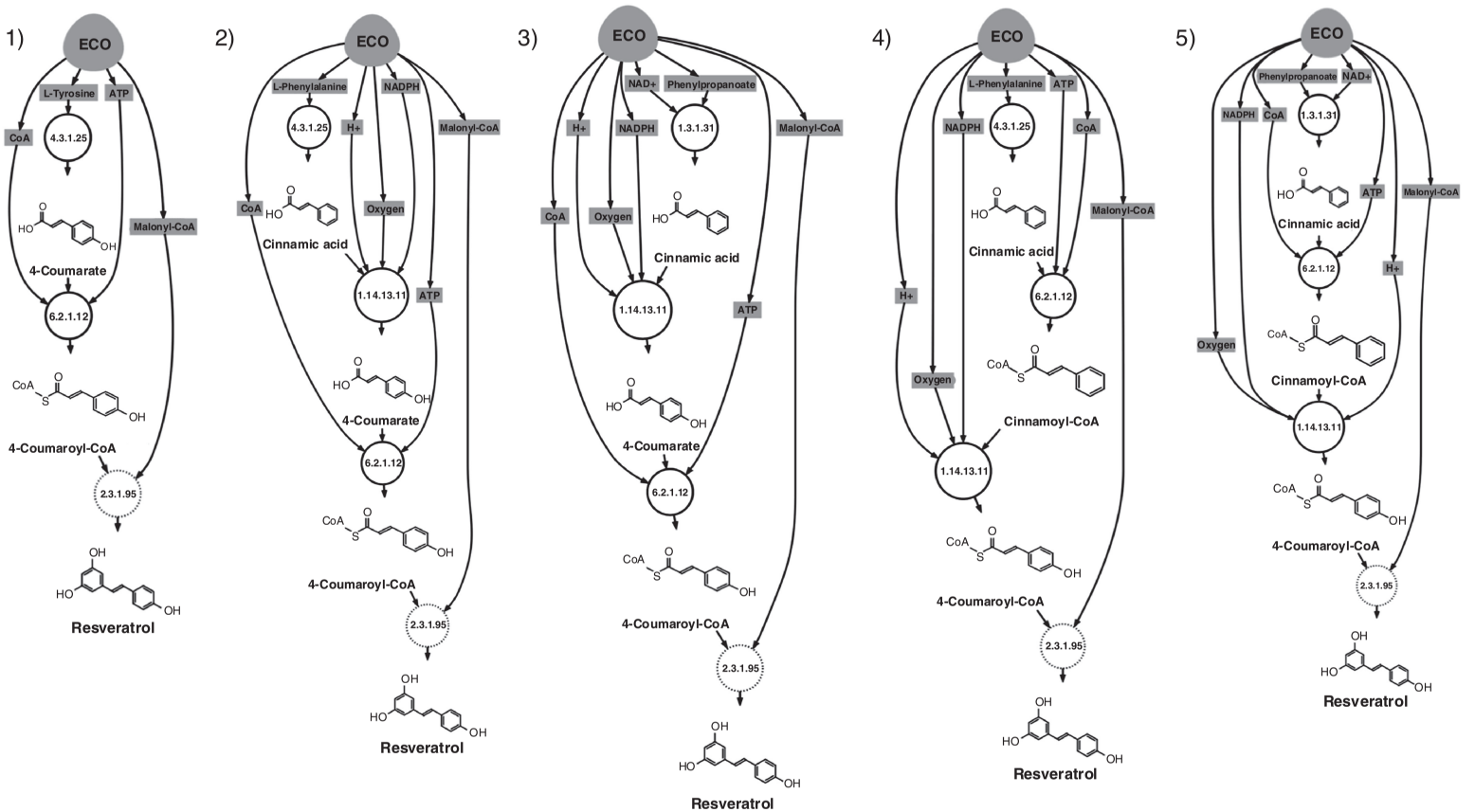

Fig. 4. Five alternative resveratrol biosynthetic pathways computed by the MetaHype server. 1) Pathway 1 contains three enzymes leading to the target. The L-phenylalanine/tyrosine ammonia lyase (PAL/TAL, EC 4.3.1.25) produces the precursor 4-coumarate from L-tyrosine, and the 4CL (EC 6.2.1.12) produces 4-coumaroyl-CoA that is converted into resveratrol by STS (EC 2.3.1.95). 2) In Pathway 2, which contains four enzymes, PAL produces cinnamic acid from L-phenylalanine. Cinnamic acid is then converted by 4CH (EC 1.14.13.11) into the precursor 4-coumarate, which is further processed into resveratrol as in Pathway 1. 3) Pathway 3 differs from Pathway 2 only in the first step producing cinnamic acid, which in this case in accomplished by a promiscuous reaction of found in 2-enoate reductase (EC 1.3.1.31) that produces cinnamic acid from phenylpropanoic acid. 4) In Pathway 4, cinnamic acid produced as in Pathway 2 from PAL is used to synthetize cinnamoyl-CoA, which is further transformed into 4-coumaroyl-CoA through promiscuous reactions from 4CL and C4H, as discussed in Subheading 3.3.) Pathway 5 uses 2-enoate reductase to produce the precursor cinnamic acid, which is processed downstream through cinnamoyl-CoA into resveratrol in the same way as in Pathway 4.

In a simple approach, the score can be built as a weighted sum of the considered factors, such as GC content or phylogenetic distance. Because of the multiplicity of factors that can influence enzyme expression, it might be difficult to blindly assign weighting priorities to each factor. One possible approach to address this issue is to build a statistical learning predictor based on techniques such as multilinear regression, support vector machines, or decision trees (32). The training set consists of the selected sequence properties with positive data formed by the list of enzyme sequences in the chassis, while the negative set has to be chosen as a significantly diverse selection of heterologous enzyme sequences (see Note 9).

Example of Gene Selection in Resveratrol Production. Besides its production from stilbene synthase (STS), production of resveratrol has also been observed as a cross-reaction from chalcone synthase (CHS, EC 2.3.1.74) (33). Interestingly, CHS is ubiquitous in plants and is also found in bacteria, while STS is only found in plant species that accumulate resveratrol and other related compounds (22). Selecting a prokaryotic CHS gene from an organism closer to *E. coli* could, thus, ease its successful expression. To that end, the retrosynthesis methodology can be applied in order to select best gene candidates expressing CHS enzyme, with the goal of converting it into an efficient resveratrol-producing factory. Table 1 provides the score of CHS genes as candidates for promiscuously producing resveratrol in *E. coli*.

3.6. Estimating Yield and Drains

The next step in pathway design corresponds to metabolic analysis of the enumerated pathways in order to get an estimation of growth and yield of the target product associated with each metabolic intervention. This step is typically performed for the steady state through flux balance analysis (FBA) (3). To that end, an *in silico* model of the metabolic network of the chassis organism, normally given in SBML representation, needs to be obtained from the literature or from databases such as BIGG. Several software packages for FBA like the COBRA toolbox (11) are available. The following steps should be followed to perform flux balance analysis of the heterologous pathways:

1. Consistency check between the *in silico* SBML model and the metabolic database: Because they contain only reactions reconstructed from high-throughput *omics* data, genome-wide reconstructed *in silico* models do not contain such level of detail in the pathways as the one in metabolic databases like MetaCyc or KEGG. Therefore, a minimum level of consistency needs to be guaranteed between the metabolic network model from databases described in Subheading 3.3 and the *in silico* SBML model for FBA, since it might happen otherwise that the heterologous pathway obtained from pathway enumeration in Subheading 3.4

Table 1
Sequence features and organism compatibility for top CHS genes according to RetroPath for gene insertion in *E. coli*

Gene_id	Cost	L	GC	ibody	pI	hyd	helix	sheet	turns	coil	d	Organism
Bind_3897	0.99	406	63.79	0.54	6.4	−86.9	41.5	22.6	17.2	22.8	0	*Beijerinckia indica*
Bind_2602	1.02	354	68.46	0.6	45.4	10.2	32.0	28.1	16.0	28.7	0	*Beijerinckia indica*
Ping_0256	1.29	362	63.4	40.52	6.6	−10.4	56.9	15.9	13.9	17.9	0	*Psychromonas ingrahamii*
Gbem_1028	1.47	349	68.86	0.66	6.5	54.5	42.0	30.0	15.6	17.1	0	*Geobacter bemidjiensis*
Psyc_0421	1.66	362	63.44	0.56	6.3	−5.1	63.3	14.5	12.7	14.2	0	*Psychrobacter arcticum*
Mrad2831_4712	1.94	357	69.47	0.58	7.5	19.5	46.9	24.9	13.8	19.1	0	*Methylobacterium radiotolerans*
Msil_3391	2.27	360	70.46	0.57	6.5	78.6	44.8	27.9	12.8	19.2	0	*Methylocella silvestris*
sce2182	2.87	371	71.16	0.63	6.8	95.2	46.2	25.1	11.0	22.3	0	*Sorangium cellulosum*
RB8853	3.27	367	69.6	60.5	34.6	38.8	31.1	29.9	20.2	23.4	1	*Rhodopirellula baltica*

Input features consisted of sequence length, GC content, probability to be expressed as inclusion bodies, isoelectric point (pI), hydrophobicity, secondary structure distribution, and distance to prokaryotes

is fully or partially disconnected from the chassis in the *in silico* model. Therefore, it is necessary to verify that the endogenous precursors in the pathway are present in the *in silico* model, because they are being produced either by some enzymatic reaction or by adding the ones missing to the medium.

2. Inserting the heterologous pathway and their corresponding transport and exchange processes: Next, the heterologous pathway has to be imported into the model by adding as many reactions as enzymatic steps are present in the pathway. The target product is exported out of the cell through the use of the relevant transport reactions (see Note 10). In addition, any side product of the reaction steps, which is not being further degraded by any reaction, has to be exported out of the cell. *In silico* models generally make distinction between the exchange reaction between the extracellular medium and the periplasmic space and the net exchange reaction of the metabolite with the system (see Note 11). For all reactions present in the model, constraints need to be defined by placing bounds in their reaction fluxes. For a nonreversible reaction, a bound between 0 and infinity might suffice, although more accurate constraints could be used based on empirical data. Similarly, bounds in exchange reactions have to be defined.
3. Setting the objective of the flux balance analysis: Constraint-based FBA is a technique that allows computing the optimal steady-state fluxes *in silico* once bounds in fluxes and an overall objective function have been established. Generally, the objective function is defined by a linear combination of fluxes experimentally determined to correlate with biomass growth. In the case of organisms engineered through genetic modification to produce a target compound in order to estimate the effect of the pathway insertion, several goals can be established: (a) comparison of the optimal growth before and after pathway insertion, (b) computation of maximum yield (generally leading to zero growth) by setting the goal to produce the desired compound, and (c) computation of the optimal biomass-product coupled yield goal (34). These optimal values provide an overall overview of what can be achieved by the modified strain as a cell factory of the desired compound.

Example of Yield Estimation of Resveratrol in E. coli. Using the COBRA toolbox, fluxes maximizing resveratrol and biomass yields were computed for the five alternative pathways in Fig. 4, as shown in Table 2. Optimal flux for biomass in wild-type strain is 0.737 (a.u.). A similar value is obtained in the strain that has been engineered with Pathway 1, while the maximum yield for resveratrol is 1.951. The strain with Pathway 2 shows an increase in biomass (1.111), indicating that some of the by-products can be used to

Table 2
Optimal steady-state fluxes for the five pathways in Fig. 3 maximizing biomass (first column) or the production of resveratrol (second column)

Strain	Biomass (a.u.)	Resveratrol (a.u)
WT	0.737	–
Path 1	0.737	1.591
Path 2	1.111	3.589
Path 3	0.798	1.907
Path 4	1.110	1.959
Path 5	0.830	3.951

increase growth. Maximum production of resveratrol (3.589) is significantly increased in this pathway. Pathways 3 and 4 correspond to biomass and resveratrol yields that are in between the maximum values obtained by Pathways 1 and 2. Finally, Pathway 5 is the one that yields the maximum production of resveratrol (3.951).

3.7. Toxicity Effects of Heterologous Pathways

In metabolic engineering, the importance of compound toxicity has been pointed out by several authors (35, 36). Indeed, for pathway performance, the less toxic molecule is usually desired, while conversely the highly toxic molecule is wanted when producing therapeutics such as antimicrobials. In both cases, detailed information on compound toxicity is important in the design of metabolic pathways, and toxicity is a parameter that should be included in the computer-aided pathway design framework (15). In order to establish a reference database of toxicity data in the chassis organism, a library of MIC (minimal inhibitory concentration) or IC_{50} (half maximal inhibitory concentration) experimental values should be built (14). These experimental values can then be used to develop a quantitative structure–activity relationship (QSAR) (37) model for toxicity of chemicals towards the chassis organism. The process consists of the following steps:

1. Firstly, a library should be designed in a way to provide a representative set of chemicals with maximal chemical diversity. The selection of compounds can be done using a method based on the optimal clustering of the chemical space determined by the distances defined as the chemical dissimilarity between compounds (see Note 12). A significant region of the chemical space has to be covered in order to maximize the spectrum of toxicity values.

2. The bacteria used for the toxicity assay must have been identified at the genus and species level. Standardization for accuracy of results and reproducibility is crucial, and as an example, one important parameter to control is the inoculum size (usually 5.10^5 colony-forming units (cfu) ml^{-1} for broth dilution).
3. A fresh pure culture of *E. coli* strain (such as *E. coli* ATCC 25922 usually chosen for toxicity assay) is used for the inoculum. Bacteria are grown in liquid medium at 37°C and bacterial growth determined at the stationary phase after incubation for a defined period (e.g., 18 h). Toxicity assay can be performed in 96-well microtiter plates. The chemicals are screened by serial dilution to assess their toxicity towards *E. coli*. Controls are important to add for each experiment. A positive control (as a triplicate) consists to bacterial culture without any chemical, and a negative control (as a triplicate) to monitor the absence of contamination consists to the media only.
4. Bacterial cell growth is monitored by measuring the turbidity (at 600 nm) at the stationary phase, and then data analyses are carried out to determine MIC or IC_{50}. Dose–response curves are built for each compound and fit using the sigmoidal equation:

$$y = \frac{OD_{\max}}{1 + \left(\frac{C}{IC_{50}}\right)^{p}} \tag{1}$$

where the OD_{max} represents the maximal OD_{600nm}, C is the concentration of the compound, IC_{50} is the molecule concentration that inhibits 50% of the bacterial growth, and p is the Hill slope describing the steepness of the curve. Only IC_{50} extracted from the curve fitting having a coefficient of determination $R^2 > 0.9$ should be used in the training set. The Levenberg–Marquardt least squares fitting algorithm for the sigmoidal curve can be used.

5. In order to predict toxicity values of intermediates, a quantitative structure–activity relationship (QSAR) model of toxicity has to be developed from the experimental dataset by using a statistical software package like the pls library in R (38). Descriptors for the compounds in the dataset might be chosen in the same fashion as the ones selected for performing the clustering of chemicals. For regularization purposes, IC_{50} values should be transformed into $\log(IC_{50})$. Two basic statistical methods can then be applied in order to build the QSAR model:
   (a) Principal component analysis (PCA) in order to reduce the dimension of the signature vectors by keeping only components whose variance is above some given cutoff ratio of the total variance in the set.

Table 3
Predicted toxicity in *E. coli* of metabolite intermediates of the resveratrol pathways in Fig. 3

Pathway	Compound	Predicted toxicity (IC_{50})
1,2,3	4-Coumarate	0.69 g/l
4,5	4-Coumaroyl-CoA	0.25 g/l
1,2,3,4,5	Resveratrol	0.42 g/l
2,3,4,5	Cinnamic acid	0.18 g/l
4,5	Cinnamoyl-CoA	0.16 g/l

(b) Model fitting by the partial least squares (PLS) regression method (38). PLS decomposes the principal components of the molecular signature descriptors into several latent variables that correlate best with the toxicity values log (IC_{50}).

6. Validation of the QSAR is typically accomplished through two steps:
   (a) Internal validation through the leave-one-out method
   (b) External validation by using a list of experimental values that have not been used before for training and validation

Example of Toxicity Estimation of Resveratrol Intermediates in E. coli. Table 3 lists predicted toxicity values for the intermediate heterologous metabolites involved in the production of resveratrol, as computed by the QSAR model of the EcoliTox web server for prediction of toxicity in *E. coli* (14). Typically, inhibition values for *E. coli* endogenous metabolites are found between $IC_{50} = 0.1$ g/l and 50 g/l. Predicted values for the resveratrol intermediates were also found within this range. Based on these estimations, high inhibition effects might not be expected due to the insertion of the pathways. Cinnamoyl-CoA, the heterologous intermediate of Pathway 4 and Pathway 5 in Fig. 4, is the most toxic compound in the list.

3.8. Defining a Cost Function for the Pathways

As presented in previous sections, several aspects are to be considered when estimating the cost of pathway insertion. A quantitative definition of a cost function associated with the pathway provides the possibility of ranking enumerated pathways so that top pathways can be selected by the designer for implementation. The process is as follows:

1. A simplified scheme consists on dividing the effects of pathway insertion into three main factors: enzyme compatibility and

reaction efficiency (Subheading 3.3 and Subheading 3.5), expected yield (Subheading 3.6), and metabolite toxicity (Subheading 3.7). A possible definition of the pathway cost function is as follows (15):

$$W(c,\rho) = -\lambda_{flux}\nu_c(\rho) + \lambda_{path}\sum_{r\in p}^{N} K(S(r)) + \lambda_{tox}\sum_{r\in p}^{N}\sum_{p\in r} T(p)$$
$$\lambda_{flux},\lambda_{path},\lambda_{tox} \geq 0$$
$$\lambda_{flux}+\lambda_{path}+\lambda_{tox} = 1 \qquad (2)$$

where $\nu_c(\rho)$ is the flux for pathway ρ producing compound c, as described in Subheading 3.6; $K(S(r))$ is the cost associated with sequence S of the enzyme catalyzing the reaction r in the pathway r; and $T(p)$ is the toxicity ($-\log_{10}(IC_{50})$) associated with the metabolite p product of reaction r, as defined in Subheading 3.7.

2. The cost for the sequence $K(S(r))$ has to take into account the fact of whether reaction is found annotated in databases for the given sequence or it is found based on a prediction as described in Subheading 3.3. In the case of a putative reaction, a penalty is added to the cost as follows:

$$K(S(r)) = \Gamma_{pred}(r) + \Delta(S) \qquad (3)$$

where ΔS is the compatibility for sequence S as defined in Subheading 3.5 and $\Gamma_{pred}(r)$ is defined in order to assign an additional cost to those enzyme sequences S where the corresponding reaction r is either catalyzed as a promiscuous or side reaction or its assignment is only putative. Therefore,

$$\Gamma_{pred}(r) = \begin{cases} \Gamma_{penalty} & promiscuous/predicted \\ 0 & annotated \end{cases} \qquad (4)$$

with penalty constant $\Gamma_{penalty}$ arbitrarily set to a value that is a upper bound for the score of sequence compatibility: $\Delta(S) \leq {}_{penalty}$.

3. The choice of values for parameters (λ_{flux}, λ_{path}, λ_{tox}, $\Gamma_{penalty}$) depends on each experimental setup as well as on the preferences set up by expert designers. A first approach is to set the values so that the cost function assigns less cost to those pathways that contain only enzymes annotated in databases (no putative enzymes). In ref. (15), parameters were fitted in this way to (0.025, 1.0, 0.398, 5.0).

Example of Ranking Resveratrol Pathways. For the resveratrol pathways, gene costs have been computed from the RetroPath server (15), and their values are shown in Table 4. The total cost associated with each pathway, according to Eq. 2 and to the weighting

parameters is shown in Table 5. From these results, Pathway 2 appears finally as the best candidate pathway to engineer in *E. coli* for the production of resveratrol, a result that is due both to the fact that the pathway contains less putative enzymatic steps and to the higher expected yield. Pathway 1 appears as the second-ranked pathway because even if it involves only three enzymes in comparison with the four enzymes from the other pathways, it contains two predicted or putative reactions (cinnamic acid production from PAL and again STS). Pathway 3, which is similar to Pathway 2 except for the first step (2-enoate reductase), predicted as a promiscuous reaction, appears next in the ranking. Finally, Pathways 5 and 4, containing three out of four predicted reactions (promiscuous activity predicted for 4CL and CH4, and STS), appear at the end of the ranking.

3.9. Pathway Implementation

Validation of the retrosynthetic design is performed through pathway implementation of the top-ranked heterologous pathways. The process consists typically of the following steps (Fig. 1):

1. Gene amplification: Genes can be amplified from genomic DNA, when available, or synthesized. The ability to synthesize genes in whole novel genetic pathways is now routinely used for metabolic engineering. Software and websites to facilitate the execution of oligonucleotide assembly into long custom sequences are available (39). Chemical synthesis allows synthesizing oligonucleotides of up to 120–150 nucleotides in length. Numerous methods have been developed to assemble relatively short synthetic oligonucleotides into longer gene sequences through ligation or PCR-mediated assembly. Also, codon usage varies by organism and has implications for heterologous expression of proteins. Codon optimization, which consists to render a nucleotide sequence with suitable codon usage for the expression host, might also help the gene expression but will not necessarily maximize the protein expression level.
2. Expression strain: Common expression strains are obtained from resources such as *E. coli* Genetic Stock Center (New Haven, CT) or companies as Life TechnologiesTM (Paisley, UK) and New England Biolabs (Ipswich, MA, USA). *E. coli* strains specially developed for gene expression are chosen according to the expression system needed: tightly controlled expression or expression level modulation, and depending on the type of the promoter used.
3. Promoter selection: Popular promoters are the lac promoter, allowing gene expression modulation, and the promoters pBAD of the arabinose operon and pRHA of the rhamnose operon that offer a tight control of gene expression. Tight expression control prior to target protein induction can be crucial for expression of host-toxic proteins to avoid

Table 4
Gene costs $K(S(r))$ associated with each gene in the pathway in Fig. 3

Pathway	EC	Gene	Organism	Substrate	Product	Penalty	Cost
1	4.3.1.25	RSP_3574	*Rhodobacter sphaeroides*	Tyrosine	4-Coumarate	5.0	5.20
1,2,3	6.2.1.12	RPA4421	*Rhodopseudomonas palustris CGA009*	4-Coumarate	4-Coumaroyl-CoA	0.0	0.99
2,3	1.14.13.11	4338409	*Oryza sativa japonica*	Cinnamic acid	4-Coumarate	0.0	4.77
2,4	4.3.1.25	4336415	*Oryza sativa japonica*	Phenylalanine	Cinnamic acid	0.0	4.64
3,5	1.3.1.31	CKL_1689	*Clostridium kluyveri DSM 55*	Phenylpropanoate	Cinnamic acid	5.0	3.84
4,5	6.2.1.12	RPA4421	*Rhodopseudomonas palustris CGA009*	Cinnamic acid	Cinnamoyl-CoA	5.0	0.99
4,5	1.14.13.11	4336415	*Oryza sativa japonica*	Cinnamoyl-CoA	4-Coumaroyl-CoA	5.0	4.65
1,2,3,4,5	2.3.1.95	Bind_3897	*Beijerinckia indica*	4-Coumaroyl-CoA	Resveratrol	5.0	1.00

Table 5
Cost of each of the five resveratrol pathways according to the cost function in Eq. 2

	Sequence compatibility	Expected yield	Metabolites toxicity	Total cost
Pathway 1	7.19	1.640	1.110	17.603
Pathway 2	11.40	2.350	1.290	16.858
Pathway 3	10.60	1.353	1.290	21.080
Pathway 4	11.28	1.535	1.010	26.644
Pathway 5	10.48	2.391	1.010	25.822

deleterious events ending in mutations that may affect target protein function or cell death.

4. Plasmid construction: Constructing a multiple enzyme biosynthetic pathway implies to combine several genes into a single plasmid or to use compatible expression plasmids. Commercially available plasmids as pETDuet-1, pACYCDuet-1, and -pCDFDuet-1 (Novagen, (40)) are widely used in metabolic engineering. Other systems allowing the cloning of multiple genes into one single plasmid such as pQlink vectors are also available (41) from repository sources such as Addgene (Cambridge, MA, USA). Gene assembly methods are also an alternative to combine multiple genes into a plasmid that have their expression under the control of their own promoter (42, 43). Nowadays, a wide selection of expression vectors is available, which differ in their origins of replication, promoters, translation initiation regions, antibiotic resistance markers, and transcription terminators.
5. Bacterial culture: Bacterial culture is commonly carried out at 30°C or 37°C in rich medium, although optimization might be needed. It might be necessary, however, to address problems related to protein misfolding and solubility. For example, to limit protein aggregation, the temperature can be decreased to 25°C, and the use of minimal medium can be more favorable for metabolite production (44). Optimization of growth temperature and induction conditions, chaperone-coexpression system, and fusions to solubilizing partners are among numerous solutions to increase product yields.
6. Verification of protein expression: Preparation of total cell protein samples is followed by the separation of protein samples by SDS-PAGE (sodium dodecyl sulfate polyacrylamide gel electrophoresis) and eventually Western blot if antibodies

recognizing the protein target are available. For high-throughput screening for protein expression, fluorescent partners can be used. Fluorescent proteins can be used to monitor the expression level of soluble or membrane-embedded proteins (45) and coupled with flow cytometry to allow large fluorescence-based library screening.

7. Identification of the target compound: Once the protein expression has been successful, the target compound must be identified using analytical techniques. Chemical production can be determined using the intrinsic spectroscopic properties of the chemical. Metabolites contain a high chemical diversity, and the two main analytical methods that can provide structural data are the nuclear magnetic resonance (NMR) and mass spectrometry (MS) with different ion sources and mass detectors (46). To detect the metabolite of interest, MS is a robust technique when coupled with chromatography. Gas chromatography (GC–MS) has the main advantage of providing high separation efficiency. However, a major drawback of GC–MS is that the compound must be volatile. Liquid chromatography coupled to mass spectrometry (LC–MS) represents an attractive alternative to GC–MS because of its versatile separation technique (hydrophilic interaction liquid chromatography (HILIC) MS, reverse phase LC–MS) (46, 47). LC–MS has emerged as a popular and powerful tool. Following this step, preparative methods such as HPLC coupled with spectrometry are usually used to quantify the metabolite production. For example, metabolite can be specifically separated on a C18 column with a determined acetonitrile/water gradient. A large number of purification methods exist and need to be optimized for each compound.

4. Notes

1. The *in silico* model should contain at least a reconstructed stoichiometric network of the organism substantially covering their main metabolic routes. Transcription regulation, thermodynamics, kinetics as well as other information are increasingly becoming available in these models and will bring in the future the design to finer levels of detail.
2. There is a basic difference between the information that is required in the model in order to design heterologous metabolic pathways and to estimate steady-state fluxes. In the former case, the most essential information is the knowledge about the metabolites that are endogenous to the organism and therefore can be used as precursors in the heterologous pathway. In the latter case, the accuracy of the stoichiometric

relationship between those reactions that directly influence the pathway is required, while partial knowledge about upstream reactions with low influence into the pathway can be tolerated.

3. In bond–electron matrices (BEM), each row and column corresponds to one atom of the compound, and each entry is the order of the covalent bond between the atoms. The BEM of a reaction is defined as the difference between the end (right) and begin (left) BEMs.
4. The reason why we need to enumerate all pathways instead of searching for the shortest one is because not always the shortest is the best in terms of the cost associated to the pathway, as described in Subheading 3.8.
5. We are only interested in enumerating minimal hyperpaths, loosely meaning cycle-free hyperpaths (see ref. (10) for proper definitions).
6. The basic limitation of the elementary modes approach is its computational complexity, which can make slow the computation in case of large heterologous networks.
7. The hypergraph representation is used in order to require all substrates to be present for the reaction to be activated.
8. The hypergraph approach can lead to solutions that are not stoichiometrically balanced, since stoichiometry is not taken into account. These solutions need to be filtered out from the output of the algorithm.
9. A detailed description of such type of predictor can be found in (15) as well as in the patented method from DNA 2.0 (30).
10. As *in silico* models become more detailed; higher attention needs to be paid in order to describe accurately the process inside the cell. Cell compartmentalization, for instance, might imply the need for enzyme co-localization in order for the pathway to proceed. This aspect is especially relevant for plant metabolism. Similarly, exporting the metabolite out of the cell might involve several transport processes through different cell compartments (11).
11. The flux balance might require for metabolites to be taken outside of the extracellular medium in order to make the net flux zero, avoiding accumulation.
12. Several clustering methods can be applied, depending first on the type of molecular descriptors used to define molecular similarity and on the clustering algorithm. Hierarchical agglomerative clustering should be preferred, since it allows building libraries of variable size.

Acknowledgements

This work was funded by Genopole® (ATIGE grant) and Agence Nationale de la Recherche (ANR Chaire d'excellence).

References

1. Oberhardt MA, Palsson BO, Papin JA (2009) Applications of genome-scale metabolic reconstructions. Mol Syst Biol 5:320
2. Yadav VG, De Mey M, Lim CG et al (2012) The future of metabolic engineering and synthetic biology: towards a systematic practice. Metab Eng 14:233–241
3. Curran KA, Crook NC, Alper HS (2012) Using flux balance analysis to guide microbial metabolic engineering. Methods Mol Biol 834:197–216
4. Planson AG, Carbonell P, Grigoras I et al (2012) A retrosynthetic biology approach to therapeutics: from conception to delivery. Curr Opin Biotechnol. doi:10.1016/j.copbio.2012.03.009
5. Caspi R, Altman T, Dreher K et al (2012) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. Nucleic Acids Res 40: D742–D753
6. Kanehisa M, Goto S, Sato Y et al (2012) KEGG for integration and interpretation of large-scale molecular data sets. Nucleic Acids Res 40: D109–D114
7. Chang A, Scheer M, Grote A et al (2009) BRENDA, AMENDA and FRENDA the enzyme information system: new content and tools in 2009. Nucleic Acids Res 37: D588–D592
8. Schellenberger J, Park J, Conrad T et al (2010) BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions. BMC Bioinformatics 11:213
9. Li C, Donizelli M, Rodriguez N et al (2010) BioModels database: an enhanced, curated and annotated resource for published quantitative kinetic models. BMC Syst Biol 4:92
10. Carbonell P, Fichera D, Pandit SB et al (2012) Enumerating metabolic pathways for the production of heterologous target chemicals in chassis organisms. BMC Syst Biol 6:10
11. Schellenberger J, Que R, Fleming RMT et al (2011) Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nat Protoc 6:1290–1307
12. Rocha I, Maia P, Evangelista P et al (2010) OptFlux: an open-source software platform for in silico metabolic engineering. BMC Syst Biol 4:45
13. Hoops S, Sahle S, Gauges R et al (2006) COPASI—a COmplex PAthway SImulator. Bioinformatics 22:3067–3074
14. Planson AG, Carbonell P, Paillard E et al (2012) Compound toxicity screening and structure-activity relationship modeling in *Escherichia coli*. Biotechnol Bioeng 109:846–850
15. Carbonell P, Planson AG, Fichera D et al (2011) A retrosynthetic biology approach to metabolic pathway design for therapeutic production. BMC Syst Biol 5:122
16. Feist AM, Herrgard MJ, Thiele I et al (2008) Reconstruction of biochemical networks in microorganisms. Nat Rev Microbiol 7:129–143
17. Menzella H, Reeves C (2007) Combinatorial biosynthesis for drug development. Curr Opin Microbiol 10:238–245
18. Ajikumar PK, Xiao WH, Tyo KEJ et al (2010) Isoprenoid pathway optimization for taxol precursor overproduction in *Escherichia coli*. Science 330:70–74
19. Orth JD, Conrad TM, Na J et al (2011) A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism-2011. Mol Syst Biol 7:535
20. Krivoruchko A, Siewers V, Nielsen J (2011) Opportunities for yeast metabolic engineering: lessons from synthetic biology. Biotechnol J 6:262–276
21. Boghigian BA, Seth G, Kiss R et al (2010) Metabolic flux analysis and pharmaceutical production. Metab Eng 12:81–95
22. Halls C, Yu O (2008) Potential for metabolic engineering of resveratrol biosynthesis. Trends Biotechnol 26:77–81
23. Beekwilder J, Wolswinkel R, Jonker H et al (2006) Production of resveratrol in recombinant microorganisms. Appl Environ Microbiol 72:5670–5672
24. Lim CGG, Fowler ZL, Hueller T et al (2011) High-yield resveratrol production in engineered *Escherichia coli*. Appl Environ Microbiol 77:3451–3460

25. Brunk E, Neri M, Tavernelli I et al (2012) Integrating computational methods to retrofit enzymes to synthetic pathways. Biotechnol Bioeng 109:572–582
26. Cho A, Yun H, Park JHH et al (2010) Prediction of novel synthetic pathways for the production of desired chemicals. BMC Syst Biol 4:35
27. Limem I, Guedon E, Hehn A et al (2008) Production of phenylpropanoid compounds by recombinant microorganisms expressing plant-specific biosynthesis genes. Process Biochem 43:463–479
28. Kamp A, Schuster S (2006) Metatool 5.0: fast and flexible elementary modes analysis. Bioinformatics 22:1930–1931
29. Gerlee P, Lizana L, Sneppen K (2009) Pathway identification by network pruning in the metabolic network of *Escherichia coli*. Bioinformatics 25:3282–3288
30. Welch M, Villalobos A, Gustafsson C et al (2011) Designing genes for successful protein expression. Methods Enzymol 498:43–66
31. Tamura K, Peterson D, Peterson N et al (2011) MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods. Mol Biol Evol 28:2731–2739
32. Kingsford C, Salzberg SL (2008) What are decision trees? Nat Biotechnol 26:1011–1013
33. Yamaguchi T, Kurosaki F, Suh D et al (1999) Cross-reaction of chalcone synthase and stilbene synthase overexpressed in *Escherichia coli*. FEBS Lett 460:457–461
34. Patil K, Rocha I, Forster J et al (2005) Evolutionary programming as a platform for in silico metabolic engineering. BMC Bioinformatics 6:308
35. Ma SM, Garcia DE, Redding-Johanson AM et al (2011) Optimization of a heterologous mevalonate pathway through the use of variant HMG-CoA reductases. Metab Eng 13:588–597
36. Pitera DJ, Paddon CJ, Newman JD et al (2007) Balancing a heterologous mevalonate pathway for improved isoprenoid production in *Escherichia coli*. Metab Eng 9:193–207
37. Tropsha A, Golbraikh A (2010) Predictive quantitative structure-activity relationship modeling. Development and validation of QSAR models. In: Faulon JL, Benders A (eds) Handbook of chemoinformatics algorithms. Chapman and Hall/CRC, London, pp 211–232
38. Mevik BH, Wehrens R (2007) The pls package: principal component and partial least squares regression in R. J Stat Softw 18:1
39. Hughes RA, Miklos AE, Ellington AD (2011) Gene synthesis: methods and applications. Methods Enzymol 498:277–309
40. Tolia NH, Joshua-Tor L (2006) Strategies for protein coexpression in *Escherichia coli*. Nat Methods 3:55–64
41. Scheich C, Kummel D, Soumailakakis D et al (2007) Vectors for co-expression of an unrestricted number of proteins. Nucleic Acids Res 35:e43
42. Tsvetanova B, Peng L, Liang X et al (2011) Genetic assembly tools for synthetic biology. Methods Enzymol 498:327–348
43. Gibson DG (2011) Enzymatic assembly of overlapping DNA fragments. Methods Enzymol 498:349–361
44. Santos CNS, Koffas M, Stephanopoulos G (2011) Optimization of a heterologous pathway for the production of flavonoids from glucose. Metab Eng 13:392–400
45. Makino T, Skretas G, Georgiou G (2011) Strain engineering for improved expression of recombinant proteins in bacteria. Microb Cell Fact 10:32
46. Roux A, Lison D, Junot C et al (2011) Applications of liquid chromatography coupled to mass spectrometry-based metabolomics in clinical chemistry and toxicology: a review. Clin Biochem 44:119–135
47. Garcia DE, Baidoo EE, Benke PI et al (2008) Separation and mass spectrometry in microbial metabolomics. Curr Opin Microbiol 11:233–239

Part II

Genome Engineering Tools

Chapter 10

Customized Optimization of Metabolic Pathways by Combinatorial Transcriptional Engineering

Yongbo Yuan, Jing Du, and Huimin Zhao

Abstract

Introduction of a heterologous metabolic pathway into a platform microorganism for applications in metabolic engineering and synthetic biology is often technically straightforward. However, the major challenge is to balance the flux in the pathway to obtain high yield and productivity in a target microorganism. To address this limitation, we recently developed a simple, efficient, and programmable approach named "*c*ustomized *o*ptimization of *m*etabolic *pa*thways by *c*ombinatorial *t*ranscriptional *e*ngineer*i*ng" (COMPACTER) for balancing the flux in a pathway under distinct metabolic backgrounds. Here we use two examples including a cellobiose-utilizing pathway and a xylose-utilizing pathway to illustrate the key steps in the COMPACTER method.

Key words: Pathway engineering, Synthetic biology, Metabolic engineering, Lignocellulosic biofuels

1. Introduction

The interest in engineering recombinant microorganisms for cost-effective production of value-added bio-based compounds has been rapidly increasing in the past few years. Achievement of the desirable capabilities in the production hosts typically requires the introduction of several genes from one or more different organism(s) (1–4). The heterologous multigene pathway, however, may affect the existing balanced intracellular pools of metabolites, proteins, and cofactors and result in a metabolic burden and accumulation of toxic intermediates (5–11). Therefore, it is highly desirable to optimize the heterologous pathway in the production host to minimize the effects from those bottlenecks. Much effort has been made to identify the bottleneck gene in the heterologous pathway and subsequently debottleneck the pathway by overexpression of the identified key genes, deletion of competing genes, or engineering of key proteins for improving desirable functions. However, these

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_10,

traditional approaches met with limited success mainly due to the complex gene regulation of the microorganism (12).

In recognition that the interactions between the structural genes and the regulatory elements of the heterologous pathway also need to be considered and optimized, several new strategies have been developed to optimize metabolic pathways on a global level, including the perturbation of global transcription machinery (13), genomic-scale mapping of fitness altering genes (14, 15), and multiplex automated genome engineering (16). In addition, a number of approaches have been developed to balance a target pathway by tuning the expressions of genes in the pathway through engineering of the promoters (13), ribosome binding sites (7), and intergenic regions (11). Note that all these new approaches allow the simultaneous optimization of a heterologous pathway to a certain extent; however, the optimized pathways in laboratory strains may not be directly transferred to industrial platform hosts due to the distinct metabolic and regulatory backgrounds between laboratory strains and industrial strains (17).

Recently, we developed a novel strategy named "*c*ustomized *o*ptimization of *m*etabolic *pa*thways by *c*ombinatorial *t*ranscriptional *e*ngineer*i*ng" (COMPACTER) which enables simultaneous optimization of multiple genes in a heterologous pathway and tailors the target pathway to the host of interest in *Saccharomyces cerevisiae* (18). Briefly, nucleotide analog mutagenesis (19) was used to create a series of promoter mutants with varying strengths for each promoter, and the resulting promoter mutants were assembled with their corresponding genes in the target pathway into a single-copy vector to generate a library of mutant pathways with various expression levels for each gene via DNA assembler (20) (Fig. 1). In order to avoid any repetitive sequences which may cause undesired recombination during assembly, one distinct pair of promoter and terminator is carefully designed for each individual gene in the pathway.

Here we use two important heterologous pathways in *Saccharomyces cerevisiae*, the cellobiose-utilizing pathway (Fig. 2) and the xylose-utilizing pathway (Fig. 3) to illustrate the experimental procedures of the COMPACTER method. The cellobiose-utilizing pathway consists of a cellobiose transporter and an intracellular β-glucosidase (21). Introduction of this heterologous pathway into a xylose-utilizing yeast strain led to a recombinant yeast capable of simultaneously co-fermenting cellobiose and xylose (22, 23). To balance the flux through this pathway, a cellobiose transporter (CDT) gene (*cdt-1*) and a β-glucosidase (BGL) gene (*gbl-1*) from *Neurospora crassa* were assembled into a single-copy expression vector under the control of the ENO2 and PDC1 promoters, respectively (Fig. 2c). Eleven ENO2 promoter mutants and ten PDC1 promoter mutants with varying strengths were generated by nucleotide analog mutagenesis and sorted by the use of green

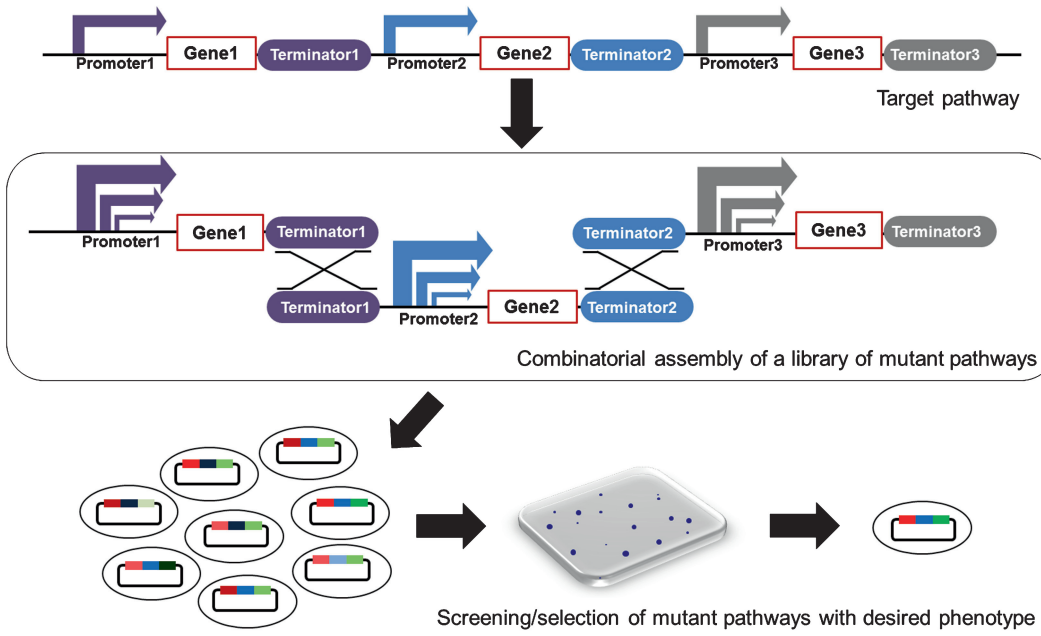

Fig. 1. General scheme of the COMPACTER method for combinatorial pathway design.

fluorescent protein (GFP) as a reporter (see Subheading 3.1). The cellobiose transporter gene controlled by eleven ENO2 promoter mutants and the β-glucosidase gene controlled by ten PDC1 promoter mutants were combined to create a library of mutant cellobiose-utilizing pathways in both industrial and laboratory *S. cerevisiae* strains.

The xylose-utilizing pathway consists of a xylose reductase, a xylitol dehydrogenase, and a xylulokinase. Introduction of this heterologous pathway into industrial and laboratory *S. cerevisiae* strains resulted in strains capable of simultaneously utilizing xylose and glucose. To balance the flux through this pathway, a xylose reductase from *Candida shehatae* (csXR), a xylitol dehydrogenase from *Candida tropicalis* (ctXDH), and the xylulokinase from *Pichia pastoris* (ppXKS) were assembled into a single-copy expression vector under the control of the PDC1, TEF1, and ENO2 promoters, respectively (Fig. 3). Similar to the cellobiose-utilizing pathway, ten mutants with varying strengths for each promoter were generated and combined to create a library of the mutant xylose-utilizing pathways in the industrial and laboratory *S. cerevisiae* strains. A colony size-based high-throughput screening approach was used for both the cellobiose- and xylose-utilizing pathways to identify the most optimized pathways from the library in *S. cerevisiae*.

In addition to the two pathways shown in this chapter, other metabolic pathways may be optimized by this strategy given that a

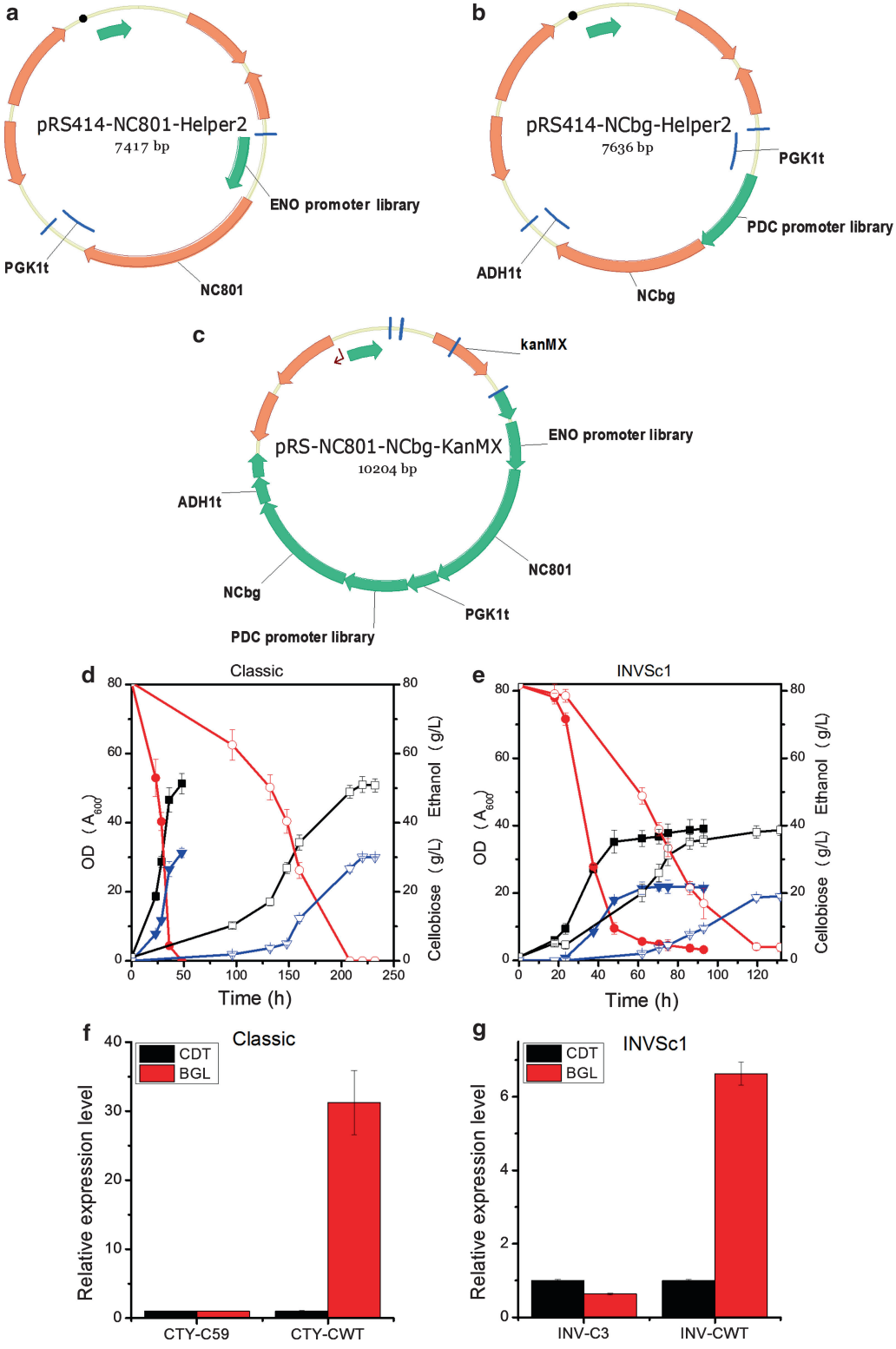

Fig. 2. (**a**) The vector map of pRS414-NC801-helper2. (**b**) The vector map of pRS414-NCbg-helper2. (**c**) The vector map of pRS414-NC801-NCbg-KanMX. (**d**) Comparison of cellobiose consumption and ethanol production in 250 mL shake-flask

proper screening or selection method is available. Moreover, this newly developed strategy can also enable host strain-specific pathway optimization for tailor-making pathways for special strains with a particular metabolic background or under a specific growth condition (see Notes 1–3).

2. Materials

Prepare all solutions using ultrapure water (ddH_2O), prepared by purifying deionized water to attain a resistivity of 18.2 MΩ cm at 25°C. Prepare and store all reagents at room temperature unless indicated otherwise.

2.1. DNA Preparation

1. pRS414 (see Notes 4 and 5): Obtained from New England Biolabs (Beverly, MA, USA) and served as the backbone for the helper plasmids for pathway assembly.
2. pRS425-nc801: Constructed previously in our laboratory (22). Used as a template for cloning *cdt-1*, *ghl-1*, a PGK1 terminator, and an ADH1 terminator.
3. The *cdt-1* gene (GenBank Accession number XM_958708) was cloned from *Neurospora crassa*.
4. The *gh1-1* gene (GenBank Accession number XM_951090) was cloned from *Neurospora crassa*.
5. pRS414-NC801-Helper: pRS414 with a *cdt-1* gene, a PGK1 terminator, and a unique *EcoR*I restriction site between the *cdt-1* gene and the plasmid backbone. The ENO2 promoter mutants were then cloned into the plasmid digested with *EcoR*I to generate the CDT gene cassettes for pathway assembly.
6. pRS414-NC801-Helper2 (Fig. 2a): pRS414 with an ENO2 promoter mutant, a *cdt-1* gene, and a PGK1 terminator (see Note 6).
7. pRS414-NCbg-Helper: pRS414 with a PGK1 terminator, a *bgl-1*gene, an ADH1 terminator, and a unique *EcoR*I restriction site between the *bgl-1* gene and the ADH1 terminator.

Fig. 2. (continued) fermentations in the industrial strain. *Open symbol*: the reference industrial strain CTY-CWT. *Solid symbol*: the mutant industrial strain CTY-C59. (**e**) Comparison of cellobiose consumption and ethanol production in 250 mL flask fermentations in the laboratory strain. *Open symbol*: the reference laboratory strain INV-CWT. *Solid symbol*: the mutant laboratory strain INV-C3. (**f**) Relative expression levels of CDT and BGL in the mutant industrial strain CTY-C59 and the reference industrial strain CTY-CWT measured using qPCR. The expression levels were normalized by setting the expression level of CDT as 1. (**g**) Relative expression levels of CDT and BGL in the mutant laboratory strain INV-C3 and the reference laboratory strain INV-CWT measured using qPCR. The expression levels were normalized by setting the expression level of CDT as 1.

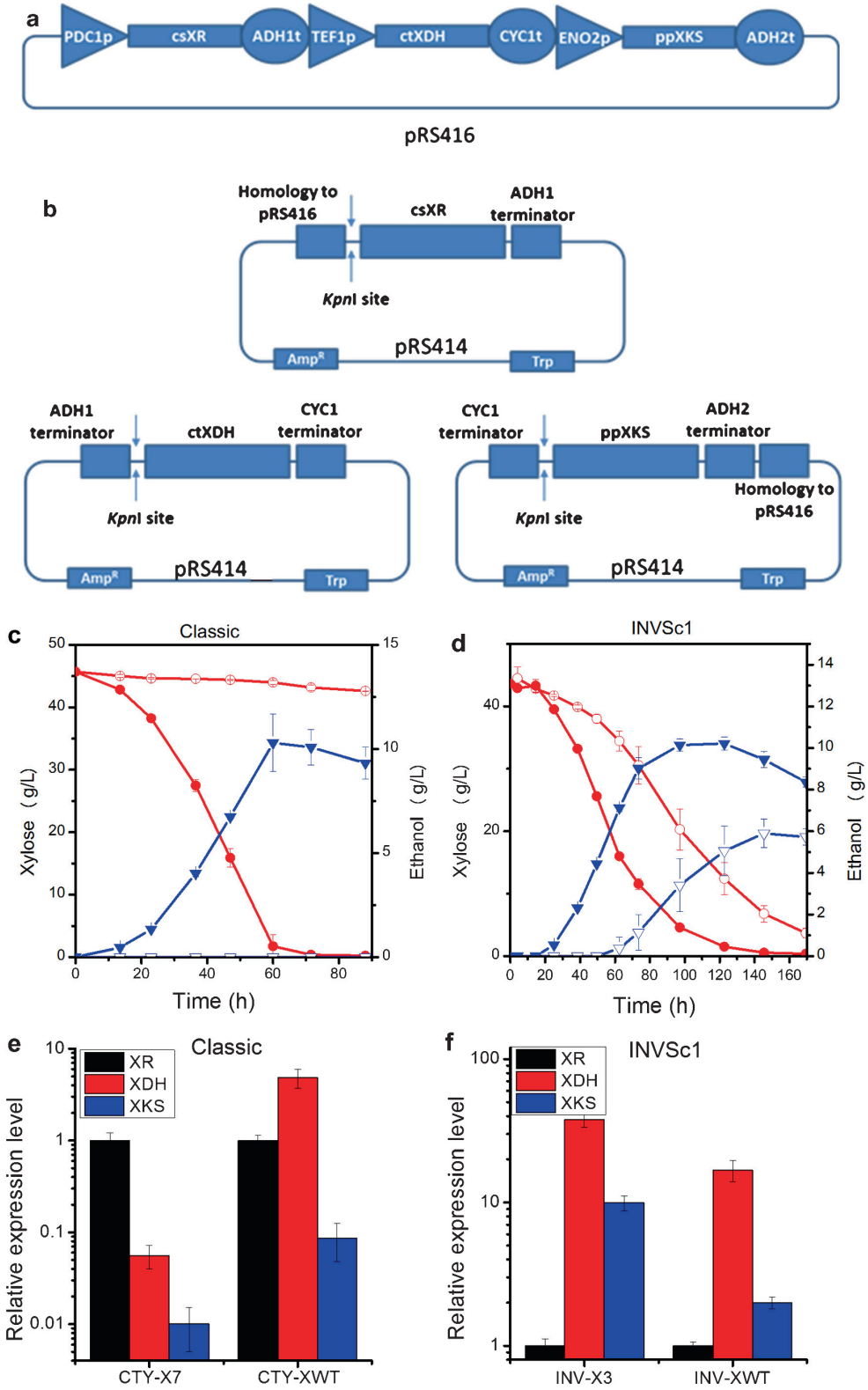

Fig. 3. (**a**) Scaffold for assembly of the xylose-utilizing pathways. The scaffold consists of a xylose reductase gene from *Candida shehatae* flanked with a PDC1 promoter and an ADH1 terminator, followed by a xylitol dehydrogenase gene

The PDC1 promoter mutants were then cloned into the plasmid digested with *EcoR*I to generate the BGL gene cassettes for pathway assembly.

8. pRS414-NCbg-Helper2 (Fig. 2b): pRS414 with a PGK1 terminator gene, a PDC1 promoter mutant, a *bgl-1*gene, and an ADH1 terminator (see Note 6).
9. pRS-kanMX: Constructed in our group by changing the URA3 marker in the pRS416 plasmid into the KanMX cassette from pUG6.
10. Strain *Candida shehatae* (NRRL Y-12858), *Candida tropicalis* (NRRL Y-5716), and *Pichia pastoris* (NRRL Y-1603): Obtained from the ARS Culture Collection (NRRL), United States Department of Agriculture, Agricultural Research Service, Peoria, IL.
11. Complementary DNA (cDNA) of *Candida shehatae*, *Candida tropicalis*, and *Pichia pastoris*: Total messenger RNA (mRNA) was isolated from each of these strains using the QIAgen RNeasy Mini Kit (QIAGEN, Valencia, CA). The resulting total mRNA was converted into cDNA using the Transcriptor First-Strand cDNA Synthesis Kit (Roche Applied Science, Branford, CT).
12. csXR: Xylose reductase homolog PCR-cloned from *Candida shehatae* using cDNA as the template.
13. ctXDH: Xylitol dehydrogenase homolog PCR-cloned from *Candida tropicalis* using cDNA as the template.
14. ppXKS: Xylulokinase homolog PCR-cloned from *Pichia pastoris* using cDNA as the template.
15. pRS426-GPM1p-GFP-GPM1t, pRS426-PDC1p-GFP-PDC1t, pRS426-TEF1p-GFP-TEF1t, and pRS426-ENO2p-GFP-ENO2t: Used as a template for cloning promoters, terminators, and the green florescent protein (GFP) gene.

Fig. 3. (continued) from *Candida tropicalis* flanked with a TEF1 promoter and a CYC1 terminator, and a xylulokinase gene from *Pichia pastoris* flanked with an ENO2 promoter and an ADH2 terminator. (**b**) Assembly of gene expression cassettes on the pRS414 helper plasmids. The helper plasmids were first linearized at the unique *KpnI* site and then co-transformed into *S. cerevisiae* with the PCR fragments of the promoter mutants. The resulting constructs were used for amplification of gene expression cassettes consisting of a promoter, the open reading frame of an enzyme homolog, a terminator, and the upstream and downstream homologous regions. (**c**) Comparison of xylose consumption and ethanol production in the industrial strain. *Open symbol*: the reference industrial strain CTY-XWT. S*olid symbol*: the mutant industrial strain CTY-X7. (**d**) Comparison of xylose consumption and ethanol production in the laboratory strain. *Open symbol*: the reference laboratory strain INV-XWT. S*olid symbol*: the mutant laboratory strain INV-X3. (**e**) Relative expression levels of XR, XDH, and XKS in the mutant industrial strain CTY-X7 and the reference industrial strain CTY-XWT measured using qPCR. The expression levels were normalized by setting the expression level of XR as 1. (**f**) Relative expression levels of XR, XDH, and XKS in the mutant laboratory strain INV-X3 and the reference laboratory strain INV-XWT measured using qPCR. The expression levels were normalized by setting the expression level of XR as 1.

16. pRS414-PDC1p-Helper: pRS414 with a unique *Kpn*I restriction site, a csXR, and an ADH1 terminator. The PDC1 promoter mutants were then cloned into the plasmid digested with *Kpn*I to generate the csXR gene cassettes for pathway assembly.
17. pRS414-TEF1p-Helper: pRS414 with an ADH1 terminator, a unique *Kpn*I restriction site, a ctXDH, and an ADH2 terminator (see Note 6). The ENO2 promoter mutants were then cloned into the plasmid digested with *Kpn*I to generate the ctXDH gene cassettes for pathway assembly.
18. pRS414-ENO2p-Helper: pRS414 with a CYC1 terminator, a unique *Kpn*I restriction site, a ppXKS, and a CYC1 terminator. The TEF1 promoter mutants were then cloned into the plasmid digested with *Kpn*I to generate the ppXKS gene cassettes for pathway assembly.
19. pRS415 (see Notes 4 and 5): Obtained from New England Biolabs (Beverly, MA, USA) and served as the backbone for characterization of promoter mutants. Promoter mutants were cloned into the plasmid digested with XhoI to measure the strengths of promoter mutants.
20. Nucleotide analogs: 8-Oxo-2′-deoxyguanosine (8-oxo-dGTP) and 6-(2-deoxy-β-D-ribofuranosyl)-3,4-dihydro-8Hpyrimido-[4,5-c][1,2]oxazin-7-one (dPTP) were purchased from Tri-Link BioTechnologies (San Diego, CA).
21. 10× Mutagenic buffer: In 40 mL of ddH_2O, add the following: 0.569 g $MgCl_2$ (70 mM), 1.491 g KCl (500 mM), 0.485 g Tris–HCl (100 mM), and 0.040 g gelatin (0.1% w/v). Add 8 M HCl to adjust pH to 8.3. Store at −20°C before use.
22. 10× Error-prone dNTPs (10× EP-dNTPs): Mix 50 μL dCTP (100 mM stock, final 10 mM), 50 μL dTTP (100 mM stock, final 10 mM), 10 μL dATP (100 mM stock, final 2 mM), 10 μL dGTP (100 mM stock, final 2 mM), and 380 μL ddH_2O to make 500 μL EP-dNTPs. Store at −20°C before use.
23. pRS415-XhoI-GFP-GPM1t: pRS415 with a unique *Xho*I site, a GFP gene, and a GPM1 terminator. Promoter mutants were then cloned into the *Xho*I site using DNA assembler method for characterization of promoter mutants (20).
24. pRS416 (see Notes 4 and 5): Obtained from New England Biolabs (Beverly, MA, USA) and served as the backbone for assembling the pathway library.
25. *S. cerevisiae* strain CEN.PK2-1c: Purchased from Euroscarf (Frankfurt, Germany) and was used for manipulation of recombinant DNA in yeast.
26. *E. coli* strain DH5α: Obtained from the cell media facility (University of Illinois at Urbana-Champaign, Urbana, IL) and used for recombinant DNA manipulation in *E. coli*.

27. Still Spirits (Classic) Turbo Distiller's Yeast: Purchased from Homebrew Heaven (Everett, WA).
28. *S. cerevisiae* strain INVSc1: Purchased from Invitrogen (Invitrogen, Carlsbad, CA).
29. Concentrated stock solution of TAE (50×): Thermo Scientific Fermentas 50× TAE buffer (Tris acetate–EDTA) was purchased from Thermo Fisher (Fisher Scientific, Pittsburgh, PA).
30. Working solution of TAE buffer (1×): Dilute the stock solution by 50-fold with deionized water. Final solute concentrations are 40 mM Tris acetate and 1 mM EDTA.
31. 1% agarose gel in 1× TAE buffer: Add 1 g of agarose into 100 mL of 1× TAE buffer and microwave until agarose is completely melted. Cool the solution to approximately 70–80°C. Add 5 μL of ethidium bromide (10 mg/mL) into the solution and mix well. Pour 50 mL of solution onto an agarose gel rack with appropriate 2-well or 8-well combs.
32. Wizard Genomic DNA Isolation Kit (Promega, Madison, WI, USA).
33. QIAquick Gel Extraction Kit (QIAGEN, Valencia, CA, USA).
34. QIAquick PCR Purification Kit (QIAGEN, Valencia, CA, USA).
35. QIAprep Miniprep Kit (QIAGEN, Valencia, CA, USA).
36. DNA polymerase: Any polymerase with high fidelity can be used. In this specific work, *Taq* DNA polymerase was used for nucleotide analog mutagenesis and Phusion high-fidelity DNA polymerase (New England Biolabs, Ipswich, MA) for the rest of the PCR reactions.
37. FailSafe PCR 2× PreMix G: Containing dNTPs and PCR reaction buffer (EPICENTRE Biotechnologies, Madison, WI).
38. Restriction enzymes: *Eco*RI, *Kpn*I, *Xho*I and *BamH*I, *HindII*I, and *Not*I restriction enzymes were purchased from New England Biolabs (Ipswich, MA, USA).
39. NanoDrop 1000: Used to measure the concentration of DNA (Thermo Scientific, Wilmington, DE, USA).
40. Benchtop centrifuges to separate cells and supernatant.
41. SYNGENE Gbox Gel imaging system: Used to check DNA on the agarose gel (Frederick, MD).

2.2. Characterization of Promoter Mutants

1. Sterilized deionized water: Sterilize ultrapure deionized water by autoclaving at 121°C for 20 min or passing through a 0.22 μm filter.
2. *S. cerevisiae* L2612 (*MATα leu2-3 leu2-112 ura3-52 trp1-298 can1 cyn1 gal*$^+$) was a gift from Dr. Yong-Su Jin (24).

3. YPAD medium: Dissolve 6 g of yeast extract, 12 g of peptone, and 60 mg of adenine hemisulfate in 500 mL of deionized water (YPA). Autoclave at 121°C for 20 min; cool down to room temperature before use. Dissolve 12 g of dextrose in 100 mL of deionized water; filter into the above-mentioned YPA medium through a 0.22 μm filter.
4. Synthetic complete dropout medium lacking tryptophan (SC-Trp): Dissolve 3 g of ammonium sulfate, 1.02 g of yeast nitrogen source without ammonium sulfate and amino acids, 0.5 g of complete synthetic medium minus tryptophan (CSM-Trp; MP Biomedicals, Solon, OH), and 12 g of dextrose in 600 mL of deionized water, sterilize by passing through a 0.22 μm filter.
5. Synthetic complete dropout medium lacking leucine (SC-Leu): Dissolve 3 g of ammonium sulfate, 1.02 g of yeast nitrogen source without ammonium sulfate and amino acids, 0.5 g of complete synthetic medium minus leucine (CSM-Leu; MP Biomedicals, Solon, OH), and 12 g of dextrose in 600 mL of deionized water. Sterilize by passing through a 0.22 μm filter.
6. Synthetic complete dropout medium lacking uracil (SC-Ura): Dissolve 3 g of ammonium sulfate, 1.02 g of yeast nitrogen source without ammonium sulfate and amino acids, 0.5 g of complete synthetic medium minus uracil (CSM-Ura; MP Biomedicals, Solon, OH), and 12 g of dextrose in 600 mL of deionized water. Sterilize by passing through a 0.22 μm filter.
7. 200 mg/mL G418 stock solution: Dissolve 1 g G418 powder into 5 mL deionized water. Sterilize by passing through a 0.22 μm filter.
8. Synthetic complete dropout medium agar plate: Make 2× concentrated SC-amino acid medium by adding half the amount of deionized water (300 mL). Add 10 g agar into 300 mL deionized water and autoclave at 121°C for 20 min. Filter the 2× concentrated medium into the autoclaved agar. Use 20–25 mL to make an agar plate in a 15 cm sterile petri dish.
9. YPAD–G418 plate: Dissolve 6 g of yeast extract, 12 g of peptone, 60 mg of adenine hemisulfate, and 10 g of agar in 500 mL of deionized water (YPA). Autoclave at 121°C for 20 min; cool down to 70–80°C. Dissolve 12 g of dextrose in 100 mL of deionized water; filter into the above-mentioned YPA medium through a 0.22 μm filter. Add 600 μL of G418 stock, mix well, and use 20–25 mL in a 15 cm petri dish.
10. 1 M lithium acetate solution: Dissolve 13.2 g lithium acetate anhydrate into 200 mL of deionized water; sterilize by passing through a 0.22 μm filter.
11. 100 mM lithium acetate solution: Add 10 mL 1 M lithium acetate solution into 90 mL sterilized deionized water.

12. 50% w/v PEG 3350 (PEG 3350): Purchased from Fisher Scientific as 50 w/v solution (Fisher Scientific, Pittsburgh, PA); sterilize by passing through a 0.22 μm filter.
13. Tris–EDTA (TE) buffer: Purchase 100× Tris–EDTA buffer from Fisher Scientific (Fisher Scientific, Pittsburgh, PA). Add 1 mL 100× Tris–EDTA buffer into 99 mL deionized water to get 1× TE buffer.
14. 2 mg/mL single-stranded carrier DNA (ssDNA): Purchase deoxyribonucleic acid sodium salt type III from salmon testes (Sigma D1626) from Sigma Aldrich (St. Louis, MO). Dissolve 200 mg single-stranded DNA into 100 mL of TE buffer. DO NOT filter to sterilize.

2.3. DNA Transformation

1. Zymoprep II Yeast Plasmid Miniprep Kit (Zymo Research, Irvine, CA).
2. Z-Competent *E. coli DH5α* was used for recombinant DNA manipulation in *E. coli* (Zymo Research, Irvine, CA).
3. *Sca*I, *Psi*I, *Sac*I, and *Asc*I restriction enzymes (New England Biolabs, Ipswich, MA, USA).
4. SOC medium: Add 20 g of Bacto-Tryptone, 5 g of yeast extract, 0.5 g of NaCl, and 186.4 mg of KCl into 980 mL of deionized water. Adjust the pH to 7.0 with NaOH. Autoclave at 121°C for 20 min. After the solution cools down to 70–80°C, add 20 mL of sterile 1 M glucose.
5. 100 mg/mL ampicillin stock solution: Dissolve 1 g of ampicillin powder in 10 mL of deionized water, sterilize by passing through a 0.22 μm filter.
6. LB broth: Add 10 g of Bacto-Tryptone, 5 g of yeast extract, and 10 g of NaCl into 1 L of deionized water. Autoclave at 121°C for 20 min.
7. LB agar: LB broth and 20 g/L agar.
8. LB–Amp$^+$ agar plates: Autoclave LB agar and when the solution cools down to 70–80°C, add 1 mL of 100 mg/mL ampicillin to 1 L of LB agar. Use 20–25 mL to make an agar plate in a 15 cm sterilized petri dish.

2.4. Screening of a Promoter Library

1. YPAC medium: Dissolve 6 g of yeast extract, 12 g of peptone, and 60 mg of adenine hemisulfate in 500 mL of deionized water (YPA). Autoclave at 121°C for 20 min, and cool down to room temperature before use. Dissolve 12 g of cellobiose (2%) or 48 g of cellobiose (8%) in 100 mL of deionized water, and filter into the above-mentioned YPA medium through a 0.22 μm filter.
2. YPAX medium: Dissolve 6 g of yeast extract, 12 g of peptone, and 60 mg of adenine hemisulfate in 500 mL of deionized

water (YPA). Autoclave at 121°C for 20 min, and cool down to room temperature before use. Dissolve 12 g of xylose in 100 mL of deionized water, and filter into the above-mentioned YPA medium through a 0.22 μm filter.

3. YPAC plate: Dissolve 6 g of yeast extract, 12 g of peptone, 60 mg of adenine hemisulfate, and 10 g of agar in 500 mL of deionized water (YPA). Autoclave at 121°C for 20 min, and cool down to 70–80°C. Dissolve 12 g of cellobiose in 100 mL of deionized water, and filter into the above-mentioned YPA medium through a 0.22 μm filter. Mix well and use 20–25 mL in a 15 cm petri dish.
4. YPAX plate: Dissolve 6 g of yeast extract, 12 g of peptone, 60 mg of adenine hemisulfate, and 10 g of agar in 500 mL of deionized water (YPA). Autoclave at 121°C for 20 min; cool down to 70–80°C. Dissolve 12 g of xylose in 100 mL of deionized water; filter into the above-mentioned YPA medium through a 0.22 μm filter. Mix well and use 20–25 mL in a 15 cm petri dish.
5. SC-Ura–xylose: SC-Ura, instead of glucose; add the same amount of xylose.
6. SC-Ura–xylose plate: SC-Ura plate, instead of glucose; add the same amount of xylose.
7. OD measurement: Using Cary 300 UV-Visible spectrophotometer (Agilent Technologies, Santa Clara, CA) at wavelength of 600 nm.
8. Culture-tube cultivation: BD Falcon round-bottom disposable polypropylene tubes with snap caps, 14 mL (Fisher Scientific, Pittsburgh, PA).
9. Synthetic complete medium with G418 (SC G418): Dissolve 3 g of ammonium sulfate, 1.02 g of yeast nitrogen source without ammonium sulfate and amino acids, 0.5 g of complete synthetic medium (MP Biomedicals, Solon, OH), and 12 g of dextrose in 600 mL of deionized water. Sterilize by passing through 0.22 μm filter. Add 600 μL of G418 stock and mix well before use.
10. SC G418 xylose plate: SC G418, replace glucose with the same amount of xylose, plus 15 g/L agar.

2.5. Quantification of Sugar, Ethanol, and Fermentation By-Products

1. High-performance liquid chromatography (HPLC) system equipped with a reflex index detector (Shimadzu Scientific Instruments, Columbia, MD).
2. HPX-87 H column (Bio-Rad, Hercules, CA).
3. Mobile phase for HPLC analysis: 5 mM H_2SO_4 solution. Add 1 mL 10 normal H_2SO_4 solution into 1 L of deionized water.

4. Standard solutions: Dissolve 500 mg D-xylose, 500 mg xylitol, 500 mg acetate, 500 mg glycerol, 500 mg D-glucose, and 500 mg ethanol to make 100 mL 5 g/L metabolite standard stock. Dilute the 5 g/L standard stock to make 1 g/L, 2 g/L, 3 g/L, and 4 g/L standards. Use these standard solutions to make a standard curve for HPLC analysis.

2.6. Determination of the Relative Expression Ratio of Optimized Pathways

1. 2 M sorbitol: Weigh 18.2 g of sorbitol (MW = 182.17) and dissolve it into 50 mL of deionized water.
2. 0.5 M ethylenediaminetetraacetic acid (EDTA) solution (pH 8.0): For a 500 mL of stock solution of 0.5 M EDTA, weigh out 93.05 g of EDTA disodium salt (MW = 372.2), and dissolve it in 400 mL of deionized water. Adjust to pH 8.0 with NaOH and correct the final volume to 500 mL. EDTA will not be dissolved completely in water unless the pH is adjusted to about 8.0.
3. 0.1% β-mercaptoethanol (β-ME): Dissolve 10 μL of β-mercaptoethanol in 10 mL of deionized water.
4. Lyticase/zymolyase: 5 U/μL zymolyase (Zymo Research, Irvine, CA).
5. Total RNA isolation from *S. cerevisiae*: Isolated from *S. cerevisiae* using RNeasy Mini Kit (QIAGEN, Valencia, CA).
6. TURBO DNA-free Kit (AMBION INC, Austin, TX) for the cleaning of RNA.
7. First-strand cDNA synthesis: The Transcriptor First-Strand cDNA Synthesis Kit (Roche, Mannheim, Germany).
8. LightCycler® 480 SYBR Green I Master (Roche, Indianapolis, IN).
9. LightCycler® 480 System (Roche, Indianapolis, IN).
10. Primers of target gene stock: Dissolve in deionized water to the final concentration of 5 pmol/μL.
11. Primers for PCR amplifying the asparagine-linked glycosylation 9 (ALG9) gene: Dissolve in deionized water to the final concentration of 5 pmol/μL.

3. Methods

3.1. Creation of Promoter Mutants Through Nucleotide Analog Mutagenesis

1. Promoter mutants with varying strengths: Create promoter mutants with varying strengths using nucleotide analog mutagenesis. Plasmids pRS426-PDC1p-GFP-PDC1t, pRS426-TEF1p-GFP-TEF1t, and pRS426-ENO2p-GFP-ENO2t are used as templates while pRS416-XhoI-Promoter-for and

Promoter-GFP-rev are used as primers (Table 2). Set up the reaction mixtures as follows (total volume 50 μL): 5 μL of 10× mutagenic buffer, 5 μL of 10× EP-dNTPs, 0.5 ng of template DNA, 1 μL of forward primer (25 pmol/μL), 1 μL of reverse primer (25 pmol/μL), 1 μL of 8-oxo-dGTP (100 mM), 1 μL of dPTP (100 mM), 0.5 μL of *Taq* DNA polymerase, and 34.5 μL of ddH_2O (25).

2. PCR condition: Fully denature at 94°C for 2 min, followed by 30 cycles of 94°C for 30 s, 50°C for 30 s, and 72°C for 30 s, with a final extension at 72°C for 5 min.
3. Digest pRS415-XhoI-GFP-GPM1t by *Xba*I and *Xho*I at 37°C for 3 h. Digestion condition: 5 μL of 10× buffer, 0.5 μL of 100× bovine serum albumin (BSA), 3 μg of pRS415-XhoI-GFP-GPM1t, 30 U of *Xba*I, and 30 U of *Xho*I. Add ddH_2O to a final volume of 50 μL.
4. Load the PCR product and the digestion product onto 1% agarose gels, and perform electrophoresis at 120 V for 20 min.
5. Gel-purify PCR product and the digestion product using the QIAquick Gel Extraction Kit.
6. Check the concentrations of the purified products using NanoDrop.
7. Mix 500 ng of the PCR product and 500 ng of the digestion product to transform the INVSc1 strain to construct the promoter mutant library (see Subheading 3.4 for the detailed yeast transformation protocol) (see Note 7).
8. Perform separate transformations for the wild-type promoters: A conventional PCR is used to amplify the wild-type promoters. Set up the reaction as follows: 50 μL of FailSafe PCR 2× PreMix G, 5 μL of forward primer (10 pmol/μL), 5 μL of reverse primer (10 pmol/μL), 1 μL of template, 1 μL of Phusion DNA polymerase, and 38 μL of ddH_2O in a total volume of 100 μL. Purify the PCR products using gel extraction, and mix with digested pRS415-XhoI-GFP-GPM1t to transform the INVSc1 strain.
9. Perform a separate transformation for the negative control: Transform 200 ng of pRS415 into the INVSc1 strain.
10. Plate the transformants on the SC-Leu plates.
11. Pick single colonies, inoculate into 3 mL of the SC-Leu media, and determine the promoter strength by measuring the GFP florescence intensity using flow cytometry or a microplate reader following the manufacturer's instructions.
12. Isolate promoter mutants with desired strengths by comparing the GFP florescence intensity of the promoter mutants with the

wild-type promoter and the negative control. Approximately 10 promoter mutants with different strengths were isolated for each promoter.

13. Confirm the strength of the promoter mutants by retransformation: Inoculate the yeast strain containing the promoter mutant of interest into 3 mL of the SC-Leu media, and isolate the plasmids using the Zymoprep II Yeast Plasmid Miniprep Kit. Transform the resulting plasmids into Z-Competent *E. coli* cells, and plate on the LB + ampicillin plate. Isolate the plasmid from *E. coli* and transfer 500 ng of the *E. coli* plasmid into the fresh INVSc1 strain. Plate the transformants on the SC-Leu plates, and pick single colonies to inoculate into 3 mL of SC-Leu liquid media. Check the GFP fluorescence intensity using flow cytometry or a microplate reader following the manufacturer's instructions. Compare with the data before retransformation, and use only the promoter mutants with consistent fluorescence intensity before and after retransformation for subsequent experiments.

3.2. DNA Preparation for Optimization of the Cellobiose Utilization Pathway

1. PCR amplify the genes *cdt-1* and *gbl-1* from the plasmid pRS425-nc801. PCR amplify the eleven ENO2 promoters and ten PDC1 promoters from the plasmids containing the ENO2 and PDC1 promoter mutants. PCR amplify the PGK1 and ADH1 terminators from the plasmid pRS425-nc801. All primers are listed in Table 1. Set up the reaction mixtures as follows: 50 μL of FailSafe PCR 2× PreMix G, 2.5 μL of forward primer (20 pmol/μL), 2.5 μL of reverse primer (20 pmol/μL), 1 μL of template (10–50 ng of the plasmid pCAR-ΔCrtX), 1 μL of DNA polymerase, and 43 μL of ddH_2O in a total volume of 100 μL.
2. PCR condition: Fully denature at 98°C for 30 s, followed by 30 cycles of 98°C for 10 s, 55°C for 30 s, and 72°C for 2 min (30 s for the promoters and terminators), with a final extension at 72°C for 10 min.
3. Load the 100 μL PCR products onto 1% agarose gels, and perform electrophoresis at 120 V for 30 min (15 min for the promoters and terminators).
4. Gel-purify PCR products using the QIAquick Gel Extraction Kit.
5. Determine the concentrations of the purified products by using NanoDrop.
6. Double digest pRS414 by *BamH*I and *Xho*I at 37°C for 3 h. Digestion condition: 5 μL of 10× buffer, 0.5 μL of 100× BSA, 3 μg of pRS414, 30 U of *BamH*I, and 30 U of *XhoI*. Add ddH_2O to a final volume of 50 μL.

Table 1
The primers used in creation of promoter mutants in cellobiose-utilizing pathway

Name	Sequence (5′→3′)
pRS414-EcoRI-nc801-for	GTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGAATTCATGTCGTCTCACGGCTCC
PGK1t-pRS414-rev	CACTAAAGGGAACAAAAGCTGGAGCTCCACCGCGGTGGCAGGAAGAATACACTATACTG
pRS414-PGK1t-for	CGACGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGATTGAATTGAATTGAAATCG
PGK1t-ncBG-rev	CCAGAGGAAATCCTTAGGAAGAGACATGAATTCCAGGAAGAATACACTATACTGG
PGK1t-ncBG-for	CTCTTTAGATCCAGTATAGTGTATTCTTCCTGGAATTCATGTCTCTTCCTAAGGATTTC
ADH1t-pRS414-rev	CACTAAAGGGAACAAAAGCTGGAGCTCCACCGCGGTGGCATGCCGGTAGAGGTGTGGTC
pRS414 NC801-ENO-08-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGTGTCGACGCTGCGGGTATAG
pRS414 NC801-ENO-08-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTATTGTATGTTATAGTGTTAG
pRS414 NC801-ENO-14-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGTGTCGACGCTGCGGGTATAG
pRS414 NC801-ENO-14-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTATTGTATGTTATAGTATTAG
pRS414 NC801-ENO-19-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGTGTCGACGCTGCGGGTATAG
pRS414 NC801-ENO-19-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTACTGTATGTTATAGTATTAG
pRS414 NC801-ENO-29-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGTGTCGACGCTGCGGGTATAG
pRS414 NC801-ENO-29-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTACTGTATGTTATAGTATTAG
pRS414 NC801-ENO-55-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGATGTCGACGCTGCGGGTATAG
pRS414 NC801-ENO-55-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTATTGTATGTTATAGTATTAG
pRS414 NC801-ENO-63-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGTGTCGACGCTGCGGGTATAG
pRS414 NC801-ENO-63-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTATTGTACGTTATAGTATTAG
pRS414 NC801-ENO-67-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGTGTCGACGCTACGGGTAC
pRS414 NC801-ENO-67-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTATTATACGTTATAGTATTAG

pRS414 NC801-ENO-76-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGTGTCGACGCTGCGGGTATGG
pRS414 NC801-ENO-76-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTATTGTATGTTATAGTATTAG
pRS414 NC801-ENO-88-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGTGTCGACGCTGCGGGTATAG
pRS414 NC801-ENO-88-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTATTGTATGTTATAGTATTAG
pRS414 NC801-ENO-100-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGTGTCGACGCTGCGGGTATAG
pRS414 NC801-ENO-100-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTATTGTATGTTATAGTATTAG
pRS414 NC801-ENO-149-for	CGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGGTGTCGACGCTGCGGGTATAG
pRS414 NC801-ENO-149-rev	GGTGCTGGCCCCGTCATGGGAGCCGTGAGACGACATTATTATTGTATGTTATAGTATTAG
pRS414 NCbg-PDC-03-for	GTACTCTTTAGATCCAGTATAGTGTATTCTTCCTGCATGCGACTGGGTGAGCATATG
pRS414 NCbg-PDC-03-rev	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATTTTGATTGATTTGACTGTGTTATTTTG
pRS414 NCbg-PDC-06-for	GTACTCTTTAGATCCAGTATAGTGTATTCTTCCTGCATGCGACTGGGTGAGCATATG
pRS414 NCbg-PDC-06-rev	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATTTTGATTGATTTGACTGTGTTATTTTG
pRS414 NCbg-PDC-08-for	GTACTCTTTAGATCCAGTATAGTGTATTCTTCCTGCATGCGACTGGGTGAGCATATG
pRS414 NCbg-PDC-08-rev	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATTTTGGTTAATTTGACTGTGTTATTTTG
pRS414 NCbg-PDC-26-for	GTACTCTTTAGATCCAGTATAGTGTATTCTTCCTGCATGCGACTGGGTGAGCATATG
pRS414 NCbg-PDC-26-rev	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATTTTGATTGATTTGACTGTGCTATTTTG
pRS414 NCbg-PDC-40-for	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATTTTGATTGATTTGACTGTGTTATTTTG
pRS414 NCbg-PDC-40-rev	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATCTTGACTGATTTGACTGTGCTATTTTG
pRS414 NCbg-PDC-60-for	GTACTCTTTAGATCCAGTATAGTGTATTCTTCCTGTATGCGACTGGGTGAGCATATG
pRS414 NCbg-PDC-60-rev	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATGTTGATTGATTTGATTGTGTTATTTTG
pRS414 NCbg-PDC-68-for	GTACTCTTTAGATCCAGTATAGTGTATTCTTCCTGCATGCGACTGGGTGAGCATATG
pRS414 NCbg-PDC-68-rev	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATTTTGATTGATTTGACTGTGTTATTTTG
pRS414 NCbg-PDC-76-for	GTACTCTTTAGATCCAGTATAGTGTATTCTTCCTGCATGCGACTGGGTGAGCATGTG

(continued)

Table 1
(continued)

Name	Sequence (5′→3′)
pRS414 NCbg-PDC-76-rev	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATTTTGATTGATTTGACTGTGTTATTTTG
pRS414 NCbg-PDC-93-for	GTACTCTTTAGATCCAGTATAGTGTATTCTTCCTGCATGCGACTGGGTGAGCATATG
pRS414 NCbg-PDC-93-rev	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATCTTGATTGATTTGACTGTGTTGTTTTG
pRS414 NCbg-PDC-100-for	GTACTCTTTAGATCCAGTATAGTGTATTCTTCCTGCATGCGACTGGGTGAGCATATG
pRS414 NCbg-PDC-100-rev	GAAGCCCCAGAGGAAATCCTTAGGAAGAGACATTTTGATTGATTTGACTGTGTTATTTTG
qPCR-Transporter-for	AATCCCCTCGCTTCCTATTTG
qPCR-Transporter-rev	TCCTGATACCGTCCCTCATC
qPCR-Beta-glucosidase-for	CTTTGGAAAGTACCCCGACTC
qPCR-Beta-glucosidase-rev	AGTTGGCTGTGTAGTGGTTC

Table 2
The primers used in creation of promoter mutants of xylose-utilizing pathway

Name	Sequence (5′→3′)
pRS416-XhoI-Promoter-for	CACGACGTTGTAAAACGACGGCCAGTGAGCGCGCGTAATACGACTCACTATAGCTCGAG
Promoter-GFP-rev	CCATCTAATTCAACAAGAATTGGGACAACTCCAGTGAAAAGTTCTTCTCCTTTACTCAT
pRS-for	CGAGGTGCCGTAAAGCACTAAATC
PDC1p-csXR-rev	CCGTTGTTCAACTTGAAAGCTGGAATTGGGCTTGGGCTCATTTTGATTGATTTGACTGTG
ADH1t-TEF1p-for	GAGGTCGCTCTTATTGACCACACCTCTACCGGCATGCATAGCTTCAAAATGTTTCTAC
TEF1p-ctXDH-rev	CAACTTTGTTAAGAACTAATGATGGGTTTGCAGTCATTTTGTAATTAAAACTTAGATTAG
CYC1t-ENO2p-for	GAGAAGGTTTTGGGACGCTCGAAGGCTTTAATTTGCGGGTGTCGACGCTGCGGGTATAG
ENO2p-ppXKS-rev	CAATGCTGAATCTCTATTTTGGATTTCTTTGGTAACCATTATTATTGTATGTTATAGTA
ADH1t-P-rev	GCATGCCGGTAGAGGTGTGGTC
ADH1t-P-for	TGGACTTCTTCGCCAGAGGTTTG
CYC1t-P-rev	CCGCAAATTAAAGCCTTCGAGC
CYC1t-P-for	ATCATGTAATTAGTTATGTCACG
pRS-rev	GGAAGCGGAAGAGCGCCCAATACG

7. Load the PCR and digestion products onto 1% agarose gels, and perform electrophoresis at 120 V for 30 min.
8. Gel-purify the PCR and digestion products using the QIAquick Gel Extraction Kit.
9. Determine the concentrations of the purified products using NanoDrop.
10. Linearize both the pRS414-NC801-helper plasmid and the pRS414-NCbg-helper plasmid pRS414 by *EcoR*I at 37°C for 3 h. Digestion condition: 5 μL of 10× buffer, 0.5 μL of 100× BSA, 3 μg of each helper plasmid, and 30 U of *EcoR*I. Add ddH_2O to a final volume of 50 μL.
11. Construction of the *cdt-1* gene cassettes using eleven ENO2 promoter mutants (Fig. 2a): Eleven ENO2 promoter mutants are assembled one by one with the linearized pRS414-NC801-helper plasmid (see Note 7). The transformation mix is set up as follows: 240 μL PEG 3350 (50% w/v), 36 μL 1.0 M LiAc, 50 μL SS-DNA (2.0 mg/ml), and 200 ng of each fragment in 34 μL deionized water. Mix the transformation mix with 10^7 competent *S. cerevisiae* L2612 cells by pipette.
12. Incubate at 42°C for 40 min.
13. Spin down the cells in a sterile tube at 5,000 × *g* for 1 min and remove the supernatant.
14. Resuspend cells with 500 μL of SC-glucose medium, and spread 100 μL of resuspended cells onto SC-Trp plates.
15. Incubate the plates at 30°C for 1–2 days until colonies appear.
16. Randomly pick five colonies from the SC-Trp plates, and inoculate each colony into 1.5 mL of SC-Trp liquid medium. Grow at 30°C overnight.
17. Purify the plasmid DNA from each 1.5 mL of the yeast culture using the Zymoprep II Kit.
18. Mix 2 μL of the isolated plasmid with 50 μL of *E. coli* competent cells using Z-Competent™ *E. coli* Transformation Kit. Store on ice for 3 min. Directly spread on an LB–Amp^+ plate. Incubate the plates at 37°C for 8–12 h until colonies appear.
19. Inoculate a single colony from each plate to 5 mL of LB supplemented with 100 μg/mL ampicillin, and grow at 37°C for 12 h.
20. Purify plasmids from each 5 mL *E. coli* culture using the QIAgen Miniprep Kit.
21. Check the plasmid concentrations by NanoDrop.
22. Verify the correctly assembled pathway through one restriction digestion reaction.

23. Digestion condition by *Hind*III at 37°C for 3 h: 1.5 μL of 10× buffer, 0.15 μL of 100× BSA, 200 ng of plasmid, and 5 U of *Hind*III. Add ddH_2O to a final volume of 15 μL. Expected bands for the correct plasmid: 3,863 bp, 3,510 bp, and 44 bp.
24. Construction of the *bgl-1* gene cassettes with ten PDC1 promoter mutants (Fig. 2b) (see Note 7): Ten PDC1 promoter mutants are assembled one by one with linearized pRS414-NCbg-helper plasmid. The transformation mixture is set up as follows: 240 μL PEG 3350 (50% w/v), 36 μL 1.0 M LiAc, 50 μL ssDNA (2.0 mg/mL), and 200 ng of each fragment in 34 μL deionized water. Mix the transformation mixture with 10^7 competent *S. cerevisiae* L2612 cells by pipette.
25. Incubate at 42°C for 40 min.
26. Spin down the cells in a sterile tube at 5,000 × *g* for 1 min and remove the supernatant.
27. Resuspend cells with 500 μL SC-glucose medium, and spread 100 μL of resuspended cells onto SC-Trp plates.
28. Incubate the plates at 30°C for 1–2 days until colonies appear.
29. Randomly pick five colonies from the SC-Trp plates, and inoculate each colony into 1.5 mL of SC-Trp liquid medium. Grow at 30°C overnight.
30. Purify plasmid DNA from each 1.5 mL yeast culture using the Zymoprep II Kit.
31. Mix 2 μL of isolated plasmid with 50 μL of *E. coli* competent cells using the Z-Competent™ *E. coli* Transformation Kit. Store on ice for 3 min. Directly spread on a LB–Amp^+ plate. Incubate the plates at 37°C for 8–12 h until colonies appear.
32. Inoculate a single colony from each plate to 5 mL of LB media supplemented with 100 μg/mL ampicillin, and grow at 37°C for 12 h.
33. Purify the plasmids from each 5 mL *E. coli* culture using the QIAgen Miniprep Kit.
34. Check the plasmid concentrations by NanoDrop.
35. Verify the correctly assembled pathway by restriction digestion reactions.
36. Digestion condition by *Hind*III at 37°C for 3 h: 1.5 μL of 10× buffer, 0.15 μL of 100× BSA, 200 ng of plasmid, and 5 U of *Hind*III. Add ddH_2O to a final volume of 15 μL. Expected bands for the correct plasmid: 4,112 bp, 2,310 bp, 1,170 bp, and 44 bp.
37. Digest pRS-kanMX by *Sal*I and *Not*I at 37°C for 3 h. Digestion condition: 5 μL of 10× buffer, 0.5 μL of 100× BSA, 3 μg of pRS414, 30 U of *Sal*I, and 30 U of *Not*I. Add ddH_2O to a final volume of 50 μL.

38. Load the PCR and digestion products onto 1% agarose gels, and perform electrophoresis at 120 V for 30 min.
39. Gel-purify the PCR and digestion products using the QIAquick Gel Extraction Kit.
40. Determine the concentrations of the purified products using NanoDrop.
41. PCR amplified all eleven *cdt-1* cassettes and ten *bgl-1* gene cassettes (see Notes 7–9). Set up the reaction mixtures as follows: 50 μL of FailSafe PCR 2× PreMix G, 2.5 μL of forward primer (20 pmol/μL stock concentration), 2.5 μL of reverse primer (20 pmol/μL stock concentration), 1 μL of template (10–50 ng of the plasmid pCAR-ΔCrtX), 1 μL of DNA polymerase, and 43 μL of ddH_2O in a total volume of 100 μL.
42. PCR condition: Fully denature at 98°C for 30 s, followed by 30 cycles of 98°C for 10 s, 55°C for 30 s, and 72°C for 3 min, with a final extension at 72°C for 10 min.
43. Load the 100 μL PCR products onto 1% agarose gels, and perform electrophoresis at 120 V for 30 min.
44. Gel-purify PCR products using the QIAquick Gel Extraction Kit.
45. Determine the concentrations of the purified products by using NanoDrop.
46. Construction of cellobiose-utilizing pathway library: Eleven *cdt-1* gene cassettes and ten *bgl-1* gene cassettes are assembled with double-digested pRS-kanMX plasmid (Fig. 2c). The transformation mixture is set up as follows: 240 μL PEG 3350 (50% w/v), 36 μL 1.0 M LiAc, 50 μL ssDNA (2.0 mg/mL), and 200 ng of each fragment in 34 μL deionized water. Mix the transformation mixture with 10^7 competent *S. cerevisiae* Classic cells by pipette.
47. Incubate at 42°C for 40 min.
48. Spin down the cells in a sterile tube at 5,000 × *g* for 1 min and remove the supernatant.
49. Resuspend cells with 500 μL YPAD medium, and shake at 250 rpm, 30°C for 4 h for recovery.
50. Spread 10 μL of cells onto YPAD plates, and store the rest of cells (with 25% of glycerol) at −80°C for library screening.

3.3. DNA Preparation for Optimization of the Xylose Utilization Pathway

1. Amplify the promoter mutants from the pRS415-Promoter-GFP-GPM1t constructs. For the PDC1 promoter mutants, use pRS-for and PDC1p-csXR-rev as primers. For the TEF1 promoter mutants, use ADH1t-TEF1p-for and TEF1p-

ctXDH-rev as primers. For the ENO2 promoter mutants, use CYC1t-ENO2p-for and ENO2p-ppXKS-rev as primers. Please see Table 2 for primer sequences. Set up the reaction as follows: 50 μL of FailSafe PCR 2× PreMix G, 5 μL of forward primer (10 pmol/μL), 5 μL of reverse primer (10 pmol/μL), 1 μL of template (10–50 ng of the pRS426-GPM1p-GFP-GPM1t plasmid), 1 μL of DNA polymerase, and 38 μL of ddH_2O in a total volume of 100 μL.

2. PCR condition: Fully denature at 98°C for 30 s, followed by 28 cycles of 98°C for 10 s, 50°C for 30 s, and 72°C for 30 s, with a final extension at 72°C for 10 min.
3. Load 2 μL of the PCR product onto 1% agarose gel, and perform electrophoresis at 120 V for 15 min to make sure the sizes of the products are correct.
4. Purify the PCR products using the QIAquick PCR Purification Kit.
5. Check the concentrations of the purified products using NanoDrop.
6. Digest pRS414-PDC1p-helper, pRS414-TEF1p-helper, and pRS414-ENO2p-helper with *Kpn*I at 37°C for 3 h. Digestion condition: 5 μL of 10× buffer, 0.5 μL of 100× BSA, 3 μg of plasmid, and 30 U of *Kpn*I. Add ddH_2O to a final volume of 50 μL.
7. Load the digested plasmids onto 1% agarose gel, and perform electrophoresis at 120 V for 22 min.
8. Gel-purify the linearized plasmids using the QIAquick Gel Extraction Kit.
9. Check the concentrations of the purified products using NanoDrop.
10. Mix 500 ng of the digested pRS414-PDC1p-helper plasmid with 500 ng of the PCR products of the PDC1 promoter mutants, and transform the mixture to CEN.PK2-1c to construct the pRS414-PDC1p-mutants-csXR plasmids (see Subheading 3.4 for the detailed yeast transformation protocol). Plate the transformants on the SC-Trp plates. Pick single colonies from the plates to inoculate into 3 mL of the SC-Trp media. Isolate the plasmids from yeast, and transform them into competent *E. coli* cells (see Subheading 3.2). Isolate plasmids and perform diagnostic PCR to confirm the constructs. Set up the PCR reaction as follows: 10 μL of FailSafe PCR 2× PreMix G, 1 μL of forward primer pRS-for (10 pmol/μL, stock), 1 μL of the reverse primer ADH1t-P-rev (10 pmol/μL, stock), 10–50 ng of template, 0.1 μL of DNA polymerase, and 7 μL

of ddH_2O in a total volume of 20 μL. PCR program: Fully denature at 98°C for 30 s, followed by 28 cycles of 98°C for 10 s, 50°C for 30 s, and 72°C for 2 min, with a final extension at 72°C for 10 min. Load the PCR product onto 1% agarose gel to confirm the construct (120 V, 22 min, expected size ~2.5 kb).

11. Follow the same procedure to clone the TEF1 and ENO2 promoter mutants into the helper plasmids. Clone TEF1 promoter mutants into pRS414-TEF1p-helper via DNA assembler to construct pRS414-TEF1p-mutants-ctXDH plasmids. Clone the ENO2 promoter mutants into pRS414-ENO2p-helper via DNA assembler to construct pRS414-ENO2p-mutants-ppXKS plasmids. Use the primer set of ADH1t-P-for and CYC1t-P-rev to confirm the pRS414-TEF1p-mutants-ctXDH constructs (expected size ~2 kb). Use the primer set of CYC1t-P-for and pRS-rev to confirm the pRS414-ENO2p-mutants-ppXKS promoter constructs (expected size ~3.5 kb).
12. Amplification of the XR, XDH, and XKS gene expression cassettes for pathway optimization: PCR amplify the XR gene expression cassettes using the pRS414-PDC1p-mutants-csXR constructs as templates with the primer set of pRS-for and ADH1t-P-rev. PCR amplify the XDH gene expression cassettes using the pRS414-TEF1p-mutants-ctXDH constructs as templates with the primer set of ADH1t-P-for and CYC1t-P-rev. PCR amplify the XKS gene expression cassettes using the pRS414-ENO2p-mutants-ppXKS constructs as templates with the primer set of CYC1t-P-for and pRS-rev. Set up the PCR reaction as follows: 50 μL of FailSafe PCR 2× PreMix G, 5 μL of the forward primer ADH1t-P-for (10 pmol/μL), 5 μL of the reverse primer CYC1t-P-rev (10 pmol/μL), 1 μL of template (20 times diluted plasmid), 1 μL of DNA polymerase, and 38 μL of ddH_2O in a total volume of 100 μL. Load 1 μL of the PCR product onto 1% agarose gel to confirm the construct (120 V, 22 min, expected size ~2.5 kb for XR cassette, ~2 kb for XDH cassette, and ~3.5 kb for XKS cassette).
13. Purify the PCR products using the QIAquick PCR Purification Kit.
14. Check the concentrations of the purified products using NanoDrop.
15. Digest pRS416 with *Xho*I and *Xba*I at 37°C for 3 h. Digestion condition: 5 μL of 10× buffer, 0.5 μL of 100× BSA, 3 μg of plasmid, 30 U of *Xho*I, and 30 U of *Xba*I. Add ddH_2O to a final volume of 50 μL.
16. Load the digested plasmids onto 1% agarose gel, and perform electrophoresis at 120 V for 22 min.

17. Gel-purify the linearized plasmids using the QIAquick Gel Extraction Kit.
18. Check the concentrations of the purified products using NanoDrop.
19. Mix 100 ng of each PCR product and 100 ng of linearized pRS416. If the volume of the final DNA mixture is smaller than 34 μL, add ddH_2O to a final volume of 34 μL. If the volume of the final DNA mixture is larger than 34 μL, perform DNA precipitation to concentrate the DNA mixture to 34 μL.
20. DNA precipitation: Add a 10× in excess amount of *n*-butanol to the DNA mixture and vortex for 5 s. For example, if the amount of the DNA mixture is 50 μL, add 500 μL of *n*-butanol to the DNA mixture. Spin at top speed in a microfuge for 10 min. Carefully remove the supernatant with a pipette without touching the DNA pellet. Add 300 μL 70% EtOH, wash the DNA pellet by gently inverting the tube several times. Spin at top speed in a microfuge for 10 min. Carefully remove the supernatant with a pipette without touching the DNA pellet. Dry the pellet in the SpeedVac for 5 min. Add 34 μL of ddH_2O to dissolve the DNA pellet.

3.4. DNA Transformation

1. Inoculate a single colony of the INVSc1 or Classic strain into 2–5 mL of YPAD medium, and grow overnight in a shaker at 30°C and 250 rpm.
2. Inoculate a fresh YPAD culture to initial OD ~0.2 (approximately 600 μL for 25 mL culture or 1.2 mL for 50 mL culture). Shake at 250 rpm, 30°C till OD ~0.8 (it will take approximately 3–5 h).
3. Harvest the culture in a sterile 50 mL centrifuge tube at 1,000 × *g* for 5 min.
4. Pour off the medium, resuspend the cells in 25 mL of sterile water, and centrifuge again (1,000 × *g* for 5 min).
5. Pour off the water, resuspend the cells in 1.0 mL 100 mM LiAc, and transfer the suspension to a clean 1.5 mL microfuge tube.
6. Pellet the cells at 4,500 × *g* for 30 s and remove the LiAc with a micropipette.
7. Resuspend the cells to a final volume of 500 μL. For 50 mL culture, add about 400 μL of 100 mM LiAc (or for 25 mL culture, add 200 μL 100 mM LiAc).
8. Boil ssDNA for 5 min and quickly chill on ice before use.
9. Mix the cell suspension and pipette 50 μL samples into microfuge tubes. Pellet the cells at 4,500 × *g* for 30 s and remove the LiAc with a micropipette.

10. Make the DNA transformation cocktail: Mix 240 μL of PEG 3350, 36 μL of 1 M LiAc, and 50 μL of ssDNA for each sample to be transformed. Mix the cocktail by vortexing.
11. Add 326 μL of DNA transformation cocktail into the cell pellet, add 34 μL of the DNA mixture, and resuspend the cell pellet gently with a micropipette. Vortex at medium speed briefly.
12. Incubate the DNA transformation mixture at 42°C for 20–40 min.
13. Pellet the cell at 4,500 × *g* for 30 s, remove the supernatant, and resuspend the cell pellet with 500 μL of ddH_2O.
14. Plate cells on YPAD–G418 to evaluate the library size and transformation efficiency (see Notes 15–16). Pellet the cell and resuspend in YPAD to recover for 4 h to overnight at 30°C with 250 rpm shaking.
15. Plate a portion of the cell suspension onto an appropriate selection plate (SC-Ura plate for the INVSc1 strain and YPAD–G418 plate for the Classic strain) or liquid media.
16. Determine the library diversity (see Note 17).

3.5. Screening for More Efficient Cellobiose-Utilizing Pathway Mutants

1. After the transformation of the library of cellobiose utilization mutant pathways, a small fraction of the transformants (less than 1% of the total volume of transformants) is plated on a YPAD–G418 plate to estimate the library size.
2. Plate the rest of transformants on a 24.5 × 24.5 cm YPAC plate evenly using glass beads.
3. Plate the control pathway consisting of the wild-type promoters at the same cell density on the same kind of the plate.
4. Incubate the plates at 30°C and check the colony size on the plates every day.
5. When the colony size difference between the library plates and the control plate is obvious, pick the 80 colonies with largest colony size on the screening plate (see Note 10).
6. Inoculate each of the 80 colonies into 2 mL YPAD–G418 media in a 14 mL round-bottom culture tube to grow a seed culture (see Note 11). Grow at 30°C with 250 rpm shaking to OD_{600} of 10–15.
7. For each colony, transfer appropriate amount of cells into 4 mL YPAC media in a 14 mL round-bottom culture tube to the final OD_{600} of 1 (see Note 12). Grow at 30°C with 250 rpm shaking.
8. For each colony, take two samples during the exponential growth phase. Measure OD_{600} and ethanol concentration.
9. Sort all 80 colonies by the ethanol concentration.

10. Select the top ten colonies in the Classic strain (top five colonies in the INVSc1 strain) with highest ethanol concentrations. Inoculate each of the selected colonies in to 2 mL YPAD–G418 media in a 14 mL round-bottom culture tube to grow a seed culture (see Note 11). Grow at 30°C with 250 rpm shaking to OD_{600} of 10–15.
11. For each colony, transfer appropriate amount of cells into 10 mL YPAC (with 8% of cellobiose) media in a 50 mL unbaffled flask to the final OD_{600} of 1. Grow at 30°C with 100 rpm shaking.
12. For each colony, take two samples during exponential phase. Measure OD_{600} and ethanol concentration (see Note 13).
13. Sort all 80 colonies by the ethanol concentration.
14. Pick the mutants with the high ethanol concentration for further analysis.
15. To confirm, isolate the plasmids from yeast, transform into *E. coli* cells, isolate the plasmids from *E. coli*, and retransform into the fresh yeast host strain before further analysis (see Note 14).

3.6. Screening for More Efficient Xylose Utilization Pathway Mutants

1. After the transformation of the library of xylose utilization pathway mutants, spread a small fraction of the transformants (less than 1% of the total volume of transformants) on a SC-Ura or YPAD–G418 plate to estimate the library size. Inoculate the rest of the transformants into 25 mL of SC-Ura or YPAD–G418 liquid medium in a 125 mL baffled flask, and shake at 30°C with 250 rpm for 12–24 h.
2. Measure the OD_{600} of the culture and estimate the number of cells by assuming 1 OD equals to 2.5×10^7 cells.
3. Plate around 10^5 cells on a 24.5 × 24.5 cm SC-Ura–xylose or SC-G418–xylose plate evenly with glass beads (add 1 mL of sterilized ddH_2O to help spreading).
4. Plate the control pathway consisting of the wild-type promoters at the same cell density on the same kind of the plate.
5. Incubate the plates at 30°C and check the colony size on the plates every day.
6. When the colony size difference between the library plates and the control plate is obvious, pick colonies with a colony size larger than the largest colony on the control plate. In this study, 80 colonies were picked from the INVSc1 strain, and 50 colonies were picked from the Classic strain (see Note 10).
7. Inoculate the colonies in to selective media to grow a seed culture. For INVSc1 strain, inoculate into 3 mL SC-Ura liquid

medium. For Classic strain, inoculate into 3 mL of SC G418 medium. Grow at 30°C with 250 rpm shaking for 2 days.

8. Take 200 μL of the seed culture and spin down at 4,500 × *g* for 30 s. Remove the supernatant with a micropipette. Resuspend with 200 μL YPAX, and use 120 μL to inoculate 3 mL of YPAX in a 14 mL round-bottom culture tube.
9. Take OD readings at 20 h and 30 h, and calculate the specific growth rate.
10. Pick the mutants with the high specific growth rate for further analysis.
11. To confirm, isolate plasmids from the yeast, transform the plasmids into *E. coli* cells, isolate the plasmids from *E. coli*, and retransform into the fresh yeast host strain before further analysis (see Subheading 3.1, step 13 for protocol for retransformation) (see Note 14).

3.7. Fermentation of the Strains Containing the Cellobiose Pathways

1. Pick a single colony from the selection plate, and inoculate into 2 mL of YPAD–G418 medium in a 15 mL round-bottom culture tube. Grow at 30°C with 250 rpm shaking to OD_{600} overnight.
2. Inoculate the seed culture into 10 mL YPAD–G418 media in a 50 mL unbaffled flask. Grow at 30°C with 100 rpm shaking to OD_{600} of 10–15 (see Note 11).
3. Transfer appropriate amount of cells into 50 mL YPAC (with 8% of cellobiose) media in a 250 mL unbaffled flask to the final OD_{600} of 1. Grow at 30°C with 100 rpm shaking (see Note 12).
4. Sample every 6 h by taking out 200 μL culture. Measure OD_{600}. Pellet the cells at 12,000 × *g* for 30 s, and store the supernatants at 4°C in a closed tube for a short period of time (up to a week) before analysis.

3.8. Fermentation of the Strains Containing the Xylose Utilization Pathway

1. Pick a single colony from the selection plate, and inoculate into 3 mL of seed medium in a culture tube. For the INVSc1 strain, both the selection plate and the seed medium would be SC-Ura. For the Classic strain, the selection plate and the seed medium would be YPAD–G418 (see Note 11). Grow at 30°C with 250 rpm shaking for 1–2 days.
2. Use the tube culture to inoculate 25 mL of seed medium in a 125 mL baffled flask, and grow at 30°C with 250 rpm shaking for another day.
3. Measure the OD_{600} of the seed culture, and inoculate 50 mL YPAX medium in a 250 mL unbaffled flask to an appropriate initial OD_{600} (see Note 12).

4. Grow the culture at 30°C with 100 rpm shaking. Sample every day (or every 12 h) by taking out at least 200 μL culture. The sample can be stored at 4°C in a closed tube for a short period of time (up to a week) before analysis.

3.9. Determination of Relative Expression Ratio of Optimized Pathways

1. Pick a single colony and inoculate into 2 mL YPAD medium (for xylose pathway, use YPAD–G418 for Classic strain and SC-Ura medium for INVSc1 strain), and grow overnight in a shaker at 30°C and 250 rpm.
2. Measure the OD_{600} of the seed culture, and inoculate the appropriate amount to 50 mL of fresh YPAC medium (for the cellobiose utilization pathway) and YPAX medium (for the xylose utilization pathway) to obtain an OD_{600} of 1.
3. Culture the cells in the shaker at 30°C and 100 rpm.
4. Harvest 10^7 cells at the middle of the exponential growth phase by centrifuging at 3,000 × *g* and 4°C.
5. Purify the total RNA from yeast using the RNeasy Mini Kit.
6. Clean purified RNA from DNA contamination using TURBO DNA-free Kit.
7. Synthesize the first-strand cDNA using the Transcriptor First-Strand cDNA Synthesis Kit.
8. Dilute the synthesized cDNA 60 times by adding 5 μL of cDNA into 295 μL water.
9. For each sample, set up the reaction as follows: 10 μL of LightCycler® 480 SYBR Green I Master, 6 μL diluted cDNA, 2 μL of forward primer (2.5 pmol/μL), and 2 μL of reverse primer (2.5 pmol/μL) to a total volume of 20 μL.
10. Purified RNA as template is used as negative control.
11. Quantitative PCR condition: Preincubation at 95°C for 10 s, followed by 45 cycles of 95°C for 10 s, 52°C for 20 s, and 72°C for 30 s, and then a melting curve at 95°C for 5 s, 65°C for 60 s, and finally a cooling at 40°C for 10 s.
12. Calculate relative expression levels using asparagine-linked glycosylation 9 (ALG9) gene as reference.

4. Notes

1. The same pathway can be optimized in different strain backgrounds. Using this method, thousands of combinations of gene expression levels for a multistep metabolic pathway can be assembled and investigated.
2. The pathway libraries generated using the strategy described in this chapter exhibit a controlled diversity. These kinds of

libraries are very useful in the understanding of metabolic pathways. Regulation and interaction of metabolic pathways can be studied through approaches such as metabolic flux analysis and DNA microarrays.

3. The pathway library consisting of the same pathway enzymes with different expression profiles can also be used to study the effect of the perturbation of expression level of a certain enzyme on the overall pathway performance. Genome-scale metabolic models can be generated using the data collected by studying mutants from the pathway library to understand and predict the response of the metabolic pathway to varying gene expression profiles.
4. Different backbone vectors are used for the helper plasmid construct and the final assembly intentionally to reduce the amount of work involved in material preparation for pathway assembly. Since the gene expression cassettes are amplified from the pRS414 helper plasmids, which contain a different selection marker than the backbone vector pRS416 used in the final assembly, it is very unlikely that the trace amount of helper plasmids in the PCR mixture would result in false-positive colonies in the assembly. Due to this, the DNA fragments with the correct size could be purified using simple PCR cleanup rather than a gel extraction method. This design greatly reduced the amount of labor required for preparation of the gene expression cassettes.
5. The single-copy vector eases the control of expression levels of genes compared to the multicopy vector which generally leads to large variations in gene expression. Furthermore, compared to the integrative vector, the single-copy vector has higher efficiency and flexibility to be transferred to different host strains.
6. For efficient assembly of the cellobiose and xylose utilization pathway libraries, the scaffold for the cellobiose and xylose utilization pathways—namely the combination of the metabolic genes and terminators for each catalytic step—remained consistent throughout this study. A fixed scaffold provides many advantages for subsequent investigation. Due to the fact that the lengths of yeast terminators have an average length of around 400 bp, the terminator of the adjacent enzyme provides a fixed DNA sequence of around 400 bp in length. In later steps, these fixed DNA sequences are included in both of the neighboring gene expression cassettes to generate longer homologous ends, which resulted in higher assembly efficiency for the library creation.
7. DNA fragments of the gene expression cassettes amplified from the helper plasmids were then mixed together with the

linearized shuttle vector at an equal DNA amount (in ng) for the combinatorial assembly of the pathway library. In this experiment, a lower molar amount of backbone was used in comparison to the amount used for enzyme homologues for two reasons. First, like in regular cloning work, more insert DNA was used than backbone DNA in order to ensure a high cloning efficiency. Second, less backbone was also used to avoid cyclization of the backbone by itself, inevitably decreasing the overall likelihood of false-positive colonies and thus increasing the overall efficiency of assembly for all three catalytic steps.

8. This approach is also restricted by the efficiency of assembly, so try to make the assembly efficiency as high as possible. Increasing the length of the overlaps between the adjacent fragments is necessary for increasing the efficiency of assembly. If possible, make the length of the overlap at least 50 bp.
9. The efficiency of assembly is also restricted by the number of fragments. Lowering the number of fragments can significantly increase the efficiency.
10. During the library screening on an agar plate, picking up a large number of colonies is necessary due to the relative low reliability of estimating the colony size by visual check.
11. In order to avoid adaptation during the library screening, the strain was always pre-cultured once in glucose medium before the main culture. Moreover, the selected optimized pathway was retransferred into the host strain for confirmation.
12. Cells from the early exponential phase are the best for the inoculation of the main fermentation. Cells at different phases may affect the reproducibility of the experiments.
13. During the screening of the best cellobiose-utilizing pathway, except the final ethanol concentration, the OD_{600} need to be considered as well to reduce the possibility of the experimental error for selecting the best strains.
14. In this chapter, only small designed libraries, about ten mutants for each promoter, were constructed and screened for both cellobiose- and xylose-utilizing pathway to prove the concept. It is possible to create large libraries of promoter mutants by nucleotide analog mutagenesis for the optimization of both pathways.
15. Estimation of the library size: Plate an appropriate amount of cells after DNA transformation onto the selection plates so that single colonies can be clearly identified and counted (100–1,000 cells on a 15 cm petri dish). The size of the library can be calculated like this:

$$Library\ size = \#\ of\ colonies\ on\ the\ plate \times \frac{Total\ volume\ of\ transformants}{Volume\ plated\ on\ the\ plate}$$

For example, if the total volume of transformants is 1 mL and when 1 μL was plated on the selection plate and 1,000 colonies showed up on the plate, the library size would be:

$$Library\ size = 1000 \times \frac{1000\ \ \mu L}{1\ \ \mu L} = 10^6$$

16. Estimation of the transformation efficiency: Transformation efficiency can be calculated like this:

$$Transformation\ efficiency = \frac{Library\ size}{Amount\ of\ DNA\ used\ in\ tranformation\ (g)}$$

For example, if 2 μg of DNA was used in the DNA transformation mentioned above, the transformation efficiency would be 5×10^5.

17. Evaluation of the library diversity: The library diversity can be assessed by randomly picking a number of transformants, isolating the plasmids from yeast, and retransforming the resulting plasmids to *E. coli*. The plasmids isolated from *E. coli* can then be used as templates for DNA sequencing to determine the diversity of the library. In the case of cellobiose and xylose utilization pathways, because they are associated with cell survival, their library diversity can be simply assessed by plating the transformants on the plates supplied with the sugar at an appropriate cell density and examine the size of colonies formed on the plate. If there is very obvious size difference on the plate, it indicates that the library diversity is good. Another way to assess the library diversity is to randomly pick up 8 colonies from SC-Ura or YPAD–G418 plate (selection plates but not linked to the pathway functionality) and perform a fermentation experiment to check if the transformants behave differently (see Subheadings 3.7 and 3.8 for detailed fermentation protocol).

References

1. Du J, Shao Z, Zhao H (2011) Engineering microbial factories for synthesis of value-added products. J Ind Microbiol Biotechnol 38:873–890
2. Pfeifer B, Ajikumar PK, Xiao WH, Tyo KEJ, Wang Y, Simeon F, Leonard E, Mucha O, Phon TH, Stephanopoulos G (2010) Isoprenoid pathway optimization for taxol precursor overproduction in *Escherichia coli*. Science 330:70–74
3. Liao JC, Atsumi S, Hanai T (2008) Non-fermentative pathways for synthesis of branched-chain higher alcohols as biofuels. Nature 451:86–89
4. Keasling JD, Ro DK, Paradise EM, Ouellet M, Fisher KJ, Newman KL, Ndungu JM, Ho KA, Eachus RA, Ham TS, Kirby J, Chang MCY, Withers ST, Shiba Y, Sarpong R (2006) Production of the antimalarial drug precursor

artemisinic acid in engineered yeast. Nature 440:940–943

5. Liao JC, Shen CR, Lan EI, Dekishima Y, Baez A, Cho KM (2011) Driving forces enable high-titer anaerobic 1-butanol synthesis in *Escherichia coli*. Appl Environ Microbiol 77:2905–2915
6. Bond-Watts BB, Bellerose RJ, Chang MCY (2011) Enzyme mechanism as a kinetic control element for designing synthetic biofuel pathways. Nat Chem Biol 7:222–227
7. Salis HM, Mirsky EA, Voigt CA (2009) Automated design of synthetic ribosome binding sites to control protein expression. Nat Biotechnol 27:946–950
8. Dueber JE, Wu GC, Malmirchegini GR, Moon TS, Petzold CJ, Ullal AV, Prather KL, Keasling JD (2009) Synthetic protein scaffolds provide modular control over metabolic flux. Nat Biotechnol 27:753–759
9. Anthony JR, Anthony LC, Nowroozi F, Kwon G, Newman JD, Keasling JD (2009) Optimization of the mevalonate-based isoprenoid biosynthetic pathway in *Escherichia coli* for production of the anti-malarial drug precursor amorpha-4,11-diene. Metab Eng 11:13–19
10. Stephanopoulos G, Luetke-Eversloh T (2008) Combinatorial pathway analysis for improved L-tyrosine production in *Escherichia coli*: Identification of enzymatic bottlenecks by systematic gene overexpression. Metab Eng 10:69–77
11. Pfleger BF, Pitera DJ, Smolke CD, Keasling JD (2006) Combinatorial engineering of intergenic regions in operons tunes expression of multiple genes. Nat Biotechnol 24:1027–1032
12. Bujara M, Panke S (2010) Engineering in complex systems. Curr Opin Biotechnol 21:586–591
13. Alper H, Stephanopoulos G (2007) Global transcription machinery engineering: a new approach for improving cellular phenotype. Metab Eng 9:258–267
14. Warner JR, Reeder PJ, Karimpour-Fard A, Woodruff LBA, Gill RT (2010) Rapid profiling of a microbial genome using mixtures of barcoded oligonucleotides. Nat Biotechnol 28:856–862
15. Warnecke TE, Lynch MD, Karimpour-Fard A, Lipscomb ML, Handke P, Mills T, Ramey CJ, Hoang T, Gill RT (2010) Rapid dissection of a complex phenotype through genomic-scale mapping of fitness altering genes. Metab Eng 12:241–250
16. Wang HH, Isaacs FJ, Carr PA, Sun ZZ, Xu G, Forest CR, Church GM (2009) Programming cells by multiplex genome engineering and accelerated evolution. Nature 460:894–898
17. Matsushika A, Inoue H, Murakami K, Takimura O, Sawayama S (2009) Bioethanol production performance of five recombinant strains of laboratory and industrial xylose-fermenting *Saccharomyces cerevisiae*. Bioresour Technol 100:2392–2398
18. Du J, Yuan Y, Si T, Li Y, Zhao H (2012) Customized Optimization of Metabolic Pathways by Combinatorial Transcriptional Engineering (COMPACTER). Nucleic Acids Res 40(18):e142
19. Gherardi E, Zaccolo M (1999) The effect of high-frequency random mutagenesis on in vitro protein evolution: a study on TEM-1 beta-lactamase. J Mol Biol 285:775–783
20. Shao Z, Zhao H, Zhao H (2009) DNA assembler, an in vivo genetic method for rapid construction of biochemical pathways. Nucleic Acids Res 37:e16
21. Galazka JM, Tian C, Beeson WT, Martinez B, Glass NL, Cate JH (2010) Cellodextrin transport in yeast for improved biofuel production. Science 330:84–86
22. Li SJ, Du J, Sun J, Galazka JM, Glass NL, Cate JHD, Yang XM, Zhao HM (2010) Overcoming glucose repression in mixed sugar fermentation by co-expressing a cellobiose transporter and a beta-glucosidase in *Saccharomyces cerevisiae*. Mol Biosyst 6:2129–2132
23. Ha SJ, Galazka JM, Kim SR, Choi JH, Yang XM, Seo JH, Glass NL, Cate JHD, Jin YS (2011) Engineered *Saccharomyces cerevisiae* capable of simultaneous cellobiose and xylose fermentation. Proc Natl Acad Sci U S A 108:504–509
24. Jin YS, Ni HY, Laplaza JM, Jeffries TW (2003) Optimal growth and ethanol production from xylose by recombinant *Saccharomyces cerevisiae* require moderate D-xylulokinase activity. Appl Environ Microbiol 69:495–503
25. Alper H, Fischer C, Nevoigt E, Stephanopoulos G (2005) Tuning genetic control through promoter engineering. Proc Natl Acad Sci U S A 102:12678–12683

Chapter 11

Adaptive Laboratory Evolution for Strain Engineering

James Winkler, Luis H. Reyes, and Katy C. Kao

Abstract

Complex phenotypes, such as tolerance to growth inhibitors, are difficult to rationally engineer into industrial model organisms due our poor understanding of their underlying molecular mechanisms. Adaptive evolution circumvents this issue by exploiting the linkage between growth rate and inhibitor resistance to select for mutants with enhanced tolerance. In order to aid experimentalists in the design and execution of adaptive laboratory evolution, we present detailed protocols for batch, continuous, and visualizing evolution in real-time (VERT) approaches to this technique.

Key words: Adaptive evolution, Evolutionary engineering, Complex phenotypes

1. Introduction

Adaptive evolution is a powerful method to improve certain features of common industrial strains (e.g., inhibitor tolerance, substrate utilization, growth temperature) without requiring knowledge of any underlying genetic mechanisms, as long as the desired trait can be coupled with growth. At its core, adaptive evolution simply involves the extended propagation of a microbial strain, typically for hundreds of generations, under the influence of the desired selective pressure. Mutants with enhanced growth rates, both due to increased tolerance or faster consumption of limiting nutrients, will occasionally arise and expand within the population over time. Enhanced strains are therefore obtained by repeated isolation, characterization, and sequencing of these mutant isolates.

Serial batch transfer and continuous bioreactors have been commonly used for evolution experiments. Serial batch transfer has the advantage of being simple to set up and utilize but are more susceptible to random drift due to repeated population bottlenecks. Continuous systems, though requiring a greater initial investment in equipment, avoid this downside by obviating the

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_11,

need for repeated inoculations. The relative merits of these distinct experimental setups for different situations are debated below. In addition to how the experiments are carried out, the strain of interest is also a factor. Though any culturable microbial strain can be used in laboratory evolution, the traditional workhorses (*Escherichia coli* and *Saccharomyces cerevisae*) are most often used given their extensive characterization.

Despite the simplicity of laboratory evolution, there are many factors that affect the outcomes of evolution experiments. Factors such as population size, rate of mutation, and frequency of beneficial mutations determine the population structure during evolution and may result in complex population structures resulting from clonal interference, a phenomenon where beneficial mutants compete within a population. Some lineages (individuals descending from specific mutants) are often lost from the population as a result; depending on when mutants are isolated from the population, adaptive mutations present in these mutants may not be identified. If the ultimate goal of the experiment is to use adaptive evolution as a tool to identify the genotypic basis of a particular trait, then the loss of beneficial mutants imposes a limit on the detectable adaptive mechanisms. The visualizing evolution in real time (VERT) pioneered in our laboratory, detailed in this chapter, can be used to observe this phenomenon and assist in the isolation of adaptive mutants. The mutation rate can be enhanced via random whole-cell mutagenesis to increase the genetic diversity of evolving populations so that adaptation can proceed more quickly. Other important variables include the strength of the selective pressure, the population size, and the length of the experiment, all of which will, in combination, influence the evolutionary outcome.

Before the initiation of an evolution experiment, the desired end goal, in terms of phenotypic improvement, should be determined and effective, high-throughput assays for analyzing isolates from the experiment developed. The methods described below are primarily for either *E. coli* or yeast with nutrient limitation as the selective pressure but can be readily modified for any other organism that can be cultured and any other trait that can be coupled with growth.

2. Materials

2.1. Biological and Chemical Materials

1. Target strain for evolution.
2. Defined media (although rich media may be used, defined media is preferred as it allows better control over the condition) in a volume sufficient for the anticipated experimental run.

3. Inhibitor stocks should be sterilized. Antibiotics should be freshly made and stored at −20°C until needed. Fresh supplies of chemical inhibitors should be ordered and stored appropriately.
4. 50% (v/v) glycerol for making frozen stocks of evolving populations.

2.2. Equipment

1. Isothermal incubators, shakers, and water baths.
2. Spectrophotometer capable of absorbance measurements.
3. Microplate reader (e.g., TECAN Infinite) for high-throughput analysis.
4. Optical microscope (e.g., Zeiss Axioscope.A1) for visual inspection of morphology. For yeast, a 40× phase contrast objective (for a total of 400× magnification combined with the 10× magnification of the eye piece) is sufficient. For bacteria, a 100× phase contract objective is needed.
5. Vacuum pumps, lines, and glass bioreactors for continuous experiments.
6. Culture tubes or flasks (sealable if required) for batch experiments.
7. Fluorescent-activated cell sorter (FACS). Commonly used fluorescent proteins such as green fluorescent protein (GFP), yellow fluorescent protein (YFP), and DsRED (RFP) can be excited using a blue laser (excitation wavelength of 488 nm). The exact specifications of the flow cytometer will depend on the properties of the fluorescent proteins used.

3. Methods

We introduce the two main methods for adaptive laboratory evolution: serial batch and continuous systems. General considerations for the design, execution, and analysis of both approaches are presented in the following section. The VERT method, which enables the visualization of competition between different genotypes during evolution, is also detailed. Finally, some additional notes concerning how to use protoplast fusion and mutagenesis to enhance evolutionary outcomes and to prevent contamination are included in Subheading 4.

3.1. Serial Batch Evolution

As mentioned previously, serial batch experiments involve the serial transfer of a population into media containing the inhibitor of interest. Batch experiments may be used if the selection environment does not need to be constant, as culture pH and nutrient composition will change during the course of microbial growth.

External stressors such as nonoptimal temperature may also be used. When evolving for tolerance of volatile chemicals (ethanol, butanol), screw-cap tubes should be used to prevent excessive evaporation during incubation. Be aware that potentially excessive buildup of CO_2 may occur during growth for some organisms. Before the initiation of the experiment, decide whether or not to use mutagenesis to increase genetic diversity (see Note 1). With that caveat in mind, the outline of a general serial batch experiment is given below:

1. Streak out the strain of interest onto agar plates. Grow overnight or until single colonies are visible.
2. Pick one colony for every set of technical replicates and inoculate it into the evolution media. Again, grow the culture until a sufficient cell density is achieved. Replicate cultures from independent colonies (biological replicates) are used to limit the impact of jackpot mutations.
3. Make a glycerol stock of the overnight cultures as the parental strain for subsequent analysis (sequencing, detection of spontaneous mutants, etc.), as mutants may have arisen in the inoculum.
4. Prepare *N* tubes or flasks with *M* ml of media.
   (a) The number of replicates (*N*) should be at least four (two independent colonies as starting parents with two replicates each). Since evolutionary processes are highly stochastic, more replicates are needed if the adaptive evolution experiment is meant to identify distinct molecular mechanisms associated with a trait of interest. If small population sizes are acceptable (small volumes of ~1 ml), 96 replicates can be placed in a 96-well deep-well plates.
   (b) The media volume (*M*) depends on the glassware used in the experiment.
   (c) The inhibitor of interest should already be added to the media, if possible. If not, add it now.
5. Inoculate the replicate cultures with the overnight using 1–2% inoculum (e.g., for a 5 ml culture, use 50 μl of overnight).
6. Incubate at the appropriate evolution condition until the desired benchmark (e.g., cell density, residual substrate) is achieved.
7. During growth, the growth kinetics can be monitored generally via plating, optical density measurements, or cell counting. Measurements for final cell density and other important parameters of interest (e.g., extracellular metabolites, pH of the media, residual nutrients) should be taken before each serial passaging.

(a) A spectrophotometer is best used for a small number of replicates. Microplate readers (e.g., Infinite series from TECAN) are useful for larger (>24) sets).

(b) Keep in mind that optical density is affected by cell size, so mutants with abnormally large or small cell volumes will influence OD readings.

(c) The approximate number of generations can be calculated based on the ratio of initial and final density: Gen = ln $(OD_{final}/OD_{initial})/\ln(2)$.

(d) Prepare another *N* tubes with new evolution media.

8. Perform a tube-to-tube transfer of the inoculum into the new media (replicate 1 to 1). Use a 1–2% inoculum.
   (a) If the cultures are showing signs of growth but are still at a low optical density, allow the overnights to grow for longer (12–24 more hours).
9. Incubate the new replicates until desired benchmark is reached. Keep the overnights at 4°C for one day in case of growth failure, contamination, or other issues; these can be used to establish a new set of replicates if needed.
10. Prepare glycerol stocks of the evolving populations using 67% culture and 33% by volume 50% glycerol. Store at −80°C for later analysis.
11. Repeat steps 7–10 until the desired experimental phenotypic goals have been reached or for a set number of generations or serial passages.

Phenotypic assays can be performed on a routine (e.g., weekly) basis to determine if any desirable mutants have arisen within the population, prior to isolation and detailed characterization. While the initial characterization can be done on a population level to simplify screening, clonal isolates should be used for any sort of detailed analysis. In addition, cell morphology, cell viability, and if relevant, residual media composition (concentrations of residual substrate and/or extracellular products) assays should also be done on a routine basis to detect any unusual changes. Notes 2–4 detail steps that may be taken to avoid contamination of the experiment.

3.2. Continuous Systems

The use of continuous systems, or chemostats, is recommended when a constant physiological state of the microbial system is desired. Continuous systems also offer advantages such as larger population sizes and smaller bottlenecks and allow control of the specific growth rate of the evolving population. The growth kinetics of the culture is dependent on the concentration of the limiting substrate and any growth inhibitors; using a material balance for biomass around the chemostat, the volumetric flow rate (q) over

the total volume (V) can be shown to be equal to the net specific growth rate of the culture (see Box 1 for derivation). This ratio of $\mu = q/V$ is called dilution rate (D). Therefore, the average specific growth rate (μ) of the population can be determined by changing the volumetric flow rate (q) of the feed.

Box 1

$$q[x_o] - q[x] + V\mu_g x - Vk_d x = V\frac{dx}{dt}$$
$[x_o] = 0$: *concentration of biomass in feed*
$[x]$: *concentration of biomass in bioreactor*
$\mu = \mu_g - k_d$
$\frac{dx}{dt} = 0$: *steady state system*
$\frac{q}{V} = \mu = D$

3.2.1. Establishing the Limiting Nutrient

Carbon source is generally used as the limiting nutrient in the chemostat, although other essential nutrients (e.g., nitrogen, phosphorous, amino acids for auxotrophic strains) can also be used. If carbon source is the limiting nutrient, keep in mind that glucose-limited cultures seem to be most sensitive to changes in the dilution rate, as lower dilution rates provoke more respiration while higher dilution rates favor fermentation.

1. Inoculate an overnight culture for the strain of interest from a single colony on solid media.
2. Aliquot equal volumes (1–2% inoculum) into a series of appropriate shake flasks that contain different concentrations of the nutrient to be used as the limiting nutrient in the same media (rich or defined) that will be used in the evolution experiments.
3. Determine the growth kinetics by measuring the optical density at different time points at the same temperature that will be used in the evolution experiments.
4. Based on the maximum biomass yield measured from the growth kinetics (optical density at stationary phase), determine the highest concentration of the limiting nutrient that leads to a reduced maximum biomass concentration (earlier onset of deceleration growth phase). Concentration of the substrate lower than or equal to this is limiting.
5. The exact concentration (within the limiting range) of the nutrient is determined by the desired size of the population inside the chemostat. The approximate size of the population

inside the chemostat needs to be measured using different concentrations of the limiting nutrient (within the limiting range) and measured after the system reaches physiological steady state (generally after at least 6–7 volume replacements).

3.2.2. Calculation of Experimental Dilution Rate

Before the start of any continuous cultures, the growth kinetics under the experimental conditions (under the selective pressure to study) must be analyzed. As stated earlier, once the system has achieved steady state, the specific growth rate (μ) of the microorganism corresponds to the dilution rate (D). The dilution rate cannot be set higher than the maximum specific growth rate of the culture, as this will lead to wash out. To determine the maximum specific growth rate, grow the strain under the condition of interest in batch culture with an excess of the limiting nutrient.

1. Streak out the strain of interest on selective solid agar plate to pick a single colony.
2. Inoculate a single colony in 3 ml of selective media at temperature that will be used in the evolution experiments to establish the overnight inoculum.
3. Start batch cultures using 1–2% (v/v) of the overnight inoculum in shake flask or microplate reader. Note: the use of the microplate reader generally underestimates the maximum specific growth rate of the culture.
4. Measure growth kinetics. Calculate the maximum specific growth rate (μ_{max}) of the culture during the exponential growth phase (μ_{max} = slope of the ln (biomass concentration) vs. time plot).
5. Use a μ that is below μ_{max}. The volumetric flow rate (q) can be calculated, since $\mu = D = q/V$, where V represents the volume of the culture. Generally V is a known parameter.

3.2.3. Feed Media

1. Once the volumetric flow rate (q) to be used has been determined, calculate the volume of feed necessary to supply the system for at least 1–2 days (depending on whether the components in the feed are stable and on the ramp-up schedule of the selective pressure, if applicable). It is important to reduce the frequency of feed tank changes to decrease the chances of contamination.

3.2.4. Chemostat Setup

1. Thoroughly clean all the experimental equipment; avoid the use of detergents, as their residues can negatively impact experiments.
2. Assemble the equipment. Sterilize via autoclaving or other sterilization techniques all components that will become in physical contact with feed or cells (see Notes 2–4).

3.2.5. Chemostat Inoculation and Sampling

1. Streak out the strain of interest on selective solid agar plate for single colonies.
2. Inoculate a volume of selective media corresponding to one sixth of *V* from a single colony and allow growing until stationary phase (overnight) at the experimental conditions. Similar to what was described for serial batch transfers, replicate experiments started from different single colonies are important.
3. Transfer starter culture to chemostats using sterile pipettes using inoculation ports (dependent on the setup of the chemostats).
4. Adjust instruments for experimental conditions (pH meters, thermocouples, flow rates, etc.).
5. Initiate flow rate to start the chemostat.
6. Samples should be taken from the chemostats at least daily. A common sampling regimen includes:
   (a) Prepare glycerol stocks as described in Subheading 3.1. Store at −80°C.
   (b) Inspect cultures under the microscope to determine changes in cell morphology.
   (c) Measure optical density.
   (d) Determine cell viability. Plating on solid media at various dilutions or the LIVE/DEAD® BacLight™ Bacterial Viability Kit (Invitrogen) or propidium iodide staining can be used for this purpose.
   (e) Quantify residual substrate concentration and extracellular metabolite/product concentrations.
7. When it is necessary to change the media, try to do it after sampling to avoid perturbations in the system.

3.3. VERT

3.3.1. Generation of Fluorescent Strains

VERT relies on the use of different fluorescent strains to visualize expansions of different fluorescently marked subpopulations. The adaptive events (changes in proportions of different subpopulations) represent the rise and expansion of adaptive mutants with higher fitness advantage in comparison with the background. VERT requires that the strain to be used for adaptive evolution be marked with different fluorescent proteins. Note that certain fluorescent proteins require oxygen for proper folding; thus, if the evolution were to be carried out under anaerobic conditions, fluorescent proteins that are functional in the absence of oxygen must be used. These fluorescent strains can be generated using classical cloning methods. Some general considerations to take into account when generating fluorescent strains for use with VERT follow:

1. It is best to work with plasmid-free strains to eliminate the need for antibiotics by integrating each fluorescent protein coding sequence into the genome. Integration can be performed using any available technique (Red/ET recombination, CRIM plasmids (1), homologous recombination for yeast, etc.). The use of antibiotics for plasmid maintenance also creates an additional selective pressure for the experiment.
2. Constitutive expression of the fluorescent proteins is preferred (e.g., a protein fusion with a housekeeping gene or use of a constitutive promoter). Chemically inducible promoters may cause extra metabolic burden due to excess expression of the protein, imposing an undesired selective pressure on the strain. This may result in the selection of promoter mutants. In addition, the use of inducers increases experimental cost.
3. It is important to determine whether there are fitness differences between different fluorescently marked strains to ensure an unbiased starting culture. Pairwise competition experiments can be used to determine if such fitness biases exist.

   Day 1

   (a) Inoculate each fluorescent strain from single colonies in evolution media, and grow overnight under the experimental conditions.

   Day 2

   (b) Measure the optical density.

   (c) Prepare different known ratios of each pair of fluorescently marked strains (20:80, 50:50, and 80:20 for every pair of fluorescent strains or just one 50:50 ratio if there are technical limitations) on a cell-density basis. Be sure to have biological (from different starting colonies) and technical (from the same colony, but two inoculations) replicates.

   (d) Measure proportions of the prepared mixtures using FACS. The measured proportions should be close to the known proportions.

   (e) The relative fitness between the two fluorescently marked strains can be determined using either batch cultures or continuous cultures depending on which type of bioreactors the evolution experiment will be carried out. If batch cultures are to be used, prepare 25 ml cultures in baffled flasks of fresh evolution media, inoculating the prepared mixtures in a 1–2% inoculum, and grow overnight under the experimental conditions. If continuous cultures are to be used, then use the mixed culture to seed chemostats under the experimental condition.

Day 3

(f) Measure proportions of the cultures using FACS every 8–12 h (or more frequently if the relative fitness of the two strains are very different).

(g) If batch cultures are used, once the culture reaches mid to late exponential phase, inoculate fresh media using a 1–2% inoculum.

Repeat the suggested protocol to get at least five data points.

(h) If no significant changes (<1% change per generation) are observed in the relative proportions of the two fluorescently marked strains, then they can be assumed to be neutral. If a single replicate shows a consistent bias toward one strain, then it is likely that a jackpot mutation is the cause. However, if significant and consistent differences in the relative fitness of the two strains are observed in all replicates (the same fluorescently marked strain is always expanding against the other one), then these two strains likely do not have equivalent fitness and should not be used together in the evolution experiment if other fluorescent markers can be substituted.

(i) If fitness advantages between the different fluorescent strains are detected, it may be possible to correct the bias by modifying the gene expression. Suggested alternatives include:

1. Modify the extension of the UP elements of used constitutive promoter.
2. Engineering of ribosome binding site (RBS) to tune protein expression.
3. Use alternative constitutive promoters.

3.3.2. Additional Steps

Regardless of whether batch or continuous systems are the selected platform for the evolution experiments, the following steps need to be added to the experimentation when using VERT:

1. *Sampling.* During sampling, it is necessary to measure the proportions of the different fluorescently marked strains using flow cytometer at least daily. More frequent sampling may also be used to provide more granular data.
2. *Isolation of mutants.* VERT facilitates the isolation of adaptive mutants, reducing the size of the population to screen for adaptive mutants. Once the generation that should be sampled has been determined (using the data analysis step described below), do the following:

(a) Streak out cells from the glycerol stock from the specific generation on solid agar media and incubate until single colonies are visible.

(b) Pick single colonies selecting for the correct fluorescent marker (the color of the expanding subpopulation). When selecting colonies, it is advisable to use a dark reader transilluminator instead of a UV lamp to avoid DNA damage. Inoculate colonies in the appropriate media.

(c) Verify the fluorescence of the selected colonies using FACS.

(d) Carry out a pairwise competition as indicated above. Useful comparisons include isolated mutants vs. parental strain and isolated mutant vs. previously isolated mutant. Take into account that the two strains involved in the comparison should express different fluorescent proteins.

(e) An alternative to steps a–b above is to inoculate a large matchstick-sized amount of frozen stock sample from the appropriate generation into the selective media. A FACS sorter can then be used to sort out the desired fluorescently marked subpopulation. Isolation of clones then proceeds as described in steps c and d above to identify the adaptive mutant of interest.

3.3.3. Data Analysis

VERT produces a series of measurements that indicate the relative abundance of cells expressing the different fluorophores (e.g. GFP, YFP, and RFP) within an evolving population. The user can identify frequency changes that correspond to adaptive events manually or computationally using an algorithm developed in our laboratory (2). The latter requires some information concerning variability in FACS measurements, annotated training data denoting adaptive events in real VERT systems, and data from the actual evolution experiments. Example datasets for variability and training calculations are included in the model.

1. Combine the entire series of measurements into one comma-separated text file.
2. Use the first two lines of the file as comments (experiment name and date) and the third as column headers: generation, challenge (inhibitor level), label_1, label_2, label_3.
   (a) Generation refers to the number of generations since the previous measurement.
   (b) Challenge can be fixed at zero if desired.
   (c) The other columns represent frequencies of various fluorophores in the population.
   (d) If only two colors are used, the label_3 column may be omitted.
   (e) Input of data containing four or more colors requires a slight modification of the file input system (see fileRead.m).
3. Place your data in the chemostat_data/experiments folder in the file distribution.

4. Open testVERT.m and change the *filename* variable to match your file.
5. Run the script. The data output will consist of an image with adaptive events' highlight and a line graph of the selective pressure in the system.

The MATLAB package can be downloaded from our laboratory website (http://research.che.tamu.edu/groups/Kao/Webpage/Home.html) or the Journal of Biological Engineering (2), including full instructions on its usage. Octave may also be used to analyze the data without graphical output if desired.

4. Notes

1. Techniques such as protoplast fusion or chemical mutagenesis can be used to speed up adaptation by either inducing recombination between beneficial mutants or increasing the mutation rate. It is best to use mutagenesis sparingly to avoid introducing too many deleterious mutations in the population.
2. In order to detect microbial contamination directly, PCR can be used to amplify strain-specific markers. Sequencing of 16S (prokaryotes) and 18S (eukaryotes) rDNA can also be used.
3. Maintenance of a sterile environment is needed to prevent contamination. For both continuous and batch experiments, all accessories that come into physical contact with the bioreactor must be sterilized. Continuous systems should filter (0.22 μm pore size) inlet air. Media containers should be closely watched for signs of contamination, as foreign organisms will consume nutrients required for growth and may secrete toxins.
4. Sampling from batch and continuous experiments should be done in a sterile environment if possible. In continuous systems, contamination of the effluent tubing is unavoidable but can be reduced by keeping good sterile practices like flaming sampling ports or sterilizing using 70% isopropanol (or 10% bleach) prior to sampling.

References

1. Haldimann A, Wanner BL (2001) Conditional-replication, integration, excision, and retrieval plasmid-host systems for gene structure-function studies of bacteria. J Bacteriol 183(21): 6384–6393
2. Winkler J, Kao KC (2012) Computational identification of adaptive mutants using the VERT system. J Biol Eng 6(1):3

Chapter 12

Trackable Multiplex Recombineering for Gene-Trait Mapping in *E. coli*

Thomas J. Mansell, Joseph R. Warner, and Ryan T. Gill

Abstract

Recent advances in homologous recombination in *Escherichia coli* have enabled improved genome engineering by multiplex recombineering. In this chapter, we present trackable multiplex recombineering (TRMR), a method for gene-trait mapping which creates simulated knockdown and overexpression mutants for virtually all genes in the *E. coli* genome. The method combines oligonucleotide synthesis with multiplex recombineering to create two libraries comprising of over 8,000 *E. coli* strains in total that can be selected for traits of interest via high-throughput screening or selection. DNA barcodes included in the recombineering cassette allow for rapid characterization of a naïve or selected population via DNA microarray analysis. Important considerations for oligonucleotide design, DNA library construction, recombineering, strain characterization, and selection are discussed.

Key words: Genome engineering, Recombineering, Trackable multiplex recombineering, Oligonucleotide synthesis, DNA barcode, Strain engineering, Genome editing

1. Introduction

Recombineering refers to the use of phage-based recombination machinery to increase the efficiency of homologous recombination of linear DNA in bacteria for genetic engineering (1–7). Over the past decade, the efficiency of recombination of ssDNA or dsDNA has increased by several orders of magnitude via recombineering advances with ssDNA efficiencies now reported as high as 30% (8) to 70% (9). As such, Lambda Red recombineering (10, 11) has become an effective technique for engineering the genome of *Escherichia coli* (9, 12). Mosberg et al. (13) asserted that recombineering of double-stranded DNA in *E. coli* proceeds via a single-stranded intermediate. Sawitzke et al. showed that recombineering in *E. coli* does not require any native mechanisms but solely the addition of the Lambda Red proteins. The authors also explored a

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_12,

variety of conditions and oligo compositions to optimize recombineering to an unprecedented 70% yield in one experiment. These and prior reports enabled the development of multiplex recombineering, which is based on the use of synthetic DNA that can now be created massively in parallel using various approaches (14, 15).

A prominent example of multiplex recombineering is the multiplex automated genome engineering (MAGE) approach reported by Wang et al. in 2009 (8). In the first MAGE report, the expression levels of 24 genes were engineered in parallel to improve lycopene production as much as any previously reported efforts, but in considerably less time (a total of over three billion combinatorial mutants were engineered and evaluated). This demonstration was enabled by a priori knowledge of which genes to modify. However, engineering complex phenotypes, such as chemical tolerance or optimized metabolic flux, depends on mechanisms and pathways that are either not obvious or unknown. A tool that links the contributions of individual genes to a targeted phenotype on a genome scale is therefore important for informing the use of combinatorial genome-engineering strategies.

Here we describe a protocol for trackable multiplex recombineering (TRMR), a method that provides this capability for simultaneously mapping genetic modifications onto a trait of interest. The method combines parallel DNA synthesis, multiplex recombineering, and molecular barcode technology to enable rapid alteration of the genomic context of all *E. coli* genes. We have demonstrated this general approach through the construction of two comprehensive *E. coli* genomic libraries comprising over 8,000 distinct mutations and gene-trait mapping of these libraries in a range of environments (16). However, by changing the genomic targets or the synthetic insert, the multiplex recombineering approach may be altered to accomplish a range of trackable genomic alterations.

The essence of the method is the use of synthetic DNA constructs in multiplex to alter expression of each gene in the *E. coli* genome. First, we design and synthesize a library of DNA oligonucleotides with homology to every gene in the *E. coli* genome with unique molecular barcodes. This synthetic DNA library is joined to a DNA cassette containing an antibiotic-resistance marker and mutations that either remove the native ribosome-binding site (RBS), simulating knockdown, or add a strong promoter and RBS, simulating overexpression. The resulting library is a substrate for Lambda Red recombineering, a powerful technique that uses DNA homology to insert the cassettes into the genome in multiplex. The library of over- and underexpression strains can then be subjected to any standard selection procedure. Finally, cells isolated from selections for a given phenotype can be quickly analyzed for their molecular barcodes by hybridization to a DNA microarray, and the contributions of given genes to the phenotype of interest can be rapidly identified.

2. Materials

2.1. Labware

1. 15 mL and 50 mL conical tubes
2. 1.5 mL microcentrifuge tubes
3. 200 µL (or equivalent) PCR tubes
4. Centrifuge (ideally with temperature control)
5. Petri dishes
6. Shaking temperature-controlled water bath

2.2. DNA Recovery and Purification

1. TE buffer: 5 mM Tris–CL pH 8.0, 0.5 mM EDTA.
2. Sodium acetate: 3 M at pH 5.2. Dissolve 40.8 g sodium acetate in 70 mL deionized water. Adjust pH by dropwise addition of glacial acetic acid.
3. Oyster glycogen.
4. Ethanol, 100% and 70% v/v.
5. Magnesium chloride: 2 M. Dissolve 19 g $MgCl_2$ to 90 mL deionized mix until dissolved, and then raise volume to 100 mL with deionized water.
6. QIAquick Gel Extraction/PCR Purification Kits (Qiagen).

2.3. SynDNA Construction

1. Deoxyuracil-containing primers as in Table 1
2. PfuTurbo Cx hotstart polymerase (Stratagene)
3. Thermal cycler
4. USER enzyme (New England Biolabs)
5. DpnI (New England Biolabs)
6. T4 DNA ligase (New England Biolabs)
7. Phi29 polymerase (New England Biolabs), exo-resistant random hexamer, phi29 DNA polymerase buffer (1×), dithiothreitol (10 mM), bovine serum albumin (200 µg/mL), dNTPs (1 mM each), phi29 DNA polymerase (9 units, New England Biolabs), and yeast inorganic pyrophosphatase (0.1 units, New England Biolabs)
8. Amicon YM30 filter (Millipore)
9. AscI (NEB), NEB buffer 4
10. Mung bean nuclease (NEB), 10 mM EDTA, and SDS 0.01%
11. Plasmids: pEM7/bsd (Invitrogen), pKD13 (3)

2.4. Recombineering

1. Base strain of choice (e.g., MG1655).
2. Plasmid: pSIM5 (17).
3. Agarose.

Table 1
Sequences of primers used in TRMR library creation (5′-3′, * denotes phosphorothioate linkage)

1	GTAAGCGGGGCATTTTTCTTCCTGTTATGTTTTTAATCAAACATCCTGCGTAGCACA CGAGGTCTCTTTAGTCACGCCACTGGTTCGTCTCCCTATAGTGAGTCGTATTAG TGTAGGCTGGAGCTGCTTC
2	TCTGTGACAGAGAAAAAGTAGCCGAAGATGACGGTTTGTCACATGGAGTTATTCC GGGGATCCGTCGACC
3	GCTGCTGGCTTACCATGTC
4	CAGTCATAGCCGAATAGCCT
5	CGGTGCCCTGAATGAACTGC
6	CAGTTATATGTAAGGAATATGACAG
7	GGAACTTCACGCTAGGGATAACAGGGTAATTGTTGACAATTAATCATCGGCA
8	GAAGTTCCTATACTTTCTAGAGAATAGGAACTTCTATATTACCCTGTTATCCCTATTA GCCCTCCCACACATAAC
9	GGAACTTCACGCTAGGGATA
10	TCACTGATAGGGATGTCAATCTCTATCACTGATAGGGAGAAGTTCCTATACTTT CTAGAG
11	CCAATGCTATGTCTGCAGAATGATTAGTTAGAAGTTCCTATACTTTCTAGAG
12	GTAGACCTCAGCGAAGTTCCTATTCTCTAGAAAGTATAGGAACTTCACGCT AGGGATA
13	TAGGTCAGTGCGTCCTGCTGATGTGCTCAGTATCTCTATCACTGATAGGGAT GTCAATC
14	CCAATGCTATGTCTGCAGAATG
15	GTAGACATCAGCTAGCTCTCCCTATAGTGAGTCGTATTAGTAGACCTCAGCG AAGTTCC
16	CCAATGCATTGTCTGCAGTCCTCCTTAGGTCAGTGCGTCCTGCT
17	GTAGACATCAGCTAGCTCTCCCTTTAGTGAGGGTTAATTGTAGACCTCAGCG AAGTTCC
18	AAGCGGAUAGTAGACCTCAGCGAAGTTC
19	AGGTCAGUGCGTCCTGCTG
20	AGGTCAGUCTGCAGAATGATTAGTTAGAAG
21	CTTCTAATACGACTCACTATAGGGAGA
22	GTAGGTACGTAAGGAGGTGATAAATG
23	GACCTCTCCGTTATCTCCTCCATG
24	ATCCGCUTCUAATACGACTCACTATAGGGAGA
25	ACTGACCUAAGTAGGTACGTAAGGAGGTGATAAATG
26	ACTGACCUCTCCGTTATCTCCTCCATG

(continued)

Table 1
(continued)

27	NNNN*N*N
28	TTTAATGATGATTATTTCCAGCC
29	CAGACGAGGCGCGATGAC
30	TGCACTCCTGCTAACTTCTTATC
31	GACCAGCGTAATGGGCGTC
32	TCGTTATCCATTTCATGCACG
33	AAATTGCGACTGAACCAGCG
34	CTGCAAAAACTGACGGAAGG
35	TAATTCAGCCGACGCTGTTC
36	ATGCATTAATTCTTAACATTAATTGATC
37	AGCGAGCTAACGCACATAC
38	TGATGAATATTCAGGAGATGGC
39	TGAGGTGATCGTCTATAAGG
40	CATTCTTATCCTCAAACTCATTG
41	AGCATGGCGCAACTGTGTC
42	CTTCTGTAGTTAGAGGACAG
43	TCTCTTCTTGCTTGCCTGC
44	ACTAAGTCCATGCTCTTATTGC
45	AAATCGTCGAGGGATTTACC
46	GTGAACCGTATACTGATGCG
47	CGTCTTTCTCTTCGATTAACTC
48	GCTACCTGCTTTACCTTGG
49	GGTTTTCTGAAGACATCAGC
50	CGACGAAGTTGAAGCTAAATTC
51	ACTTAACCGTAAATATTGGCGC
52	GGCTTGTTAAAAGCCAGCTTG
53	AAT ATT CAG CGT CCC CGT TC
54	GACAACCCATATAAACCGG
55	ACC TTC TTT CCA TGT AGA GGG
56	ACCTACAGGCGCTGGAACAG
57	CGA TTT ACT GAC CGA CAG CG
58	CAAGGCGAGCCACATAAAAATC
59	TAC TTG ATA TGA TTT CAG ACG CTC G

(continued)

Table 1
(continued)

60	GCTTTGGCGTGTGGCAATTC
61	TGA CTT CCA CAC CCA GCG AG
62	CACAGTTTATTAGCGGTGACG
63	AAT GTT GGT ATG CCC GCC GAC
64	TTGAGACACAAGGCGAGAC
65	GAA TTC CGT CAC AAA TAA AGG CTT C
66	TTGTGGTGCCGAAACTGCTG
67	TTT TCC GGT CAG CGC CAT G
68	CAATGGACTGTATTGCGCTC
69	CGC TCA GGT CGA TAA TCT TC
70	CTACGACTGGAACCACGC
71	GTT AAA CCG GCG TAG CTG TC
72	GTGGGGTTCCTGGATATTCG
73	GAA AAC CCT TAA GTC TGT GCG
74	CTATTACTTTCATGGCTGGCG
75	CAT AGC TCA TCG GGC GCT C
76	AGCGCACTGTCTACCAGCAG
77	CAC TCA ACA AGG AAA CGC GC
78	GCATTTTCAATCGCGGCAAC
79	GAA TAA ACT CCT CGG TAA TCG G
80	AGCGAGAATTAATCTGTACGCG
81	GAG GAT TTC CAC CAT TTG GC
82	CAACGTTGATCTGCAGATCC
83	TGC ATG GTG TCG TTC AGC AC
84	CTACTGATGATTTCTACCGTCTG
85	CAG AAG TTA GTC GAT AAA GCG
86	GCTTTTACTGGAGCAGGAAG
87	ATT TTA CGC AGC AGG TCA GAC
88	CGATTTCAAGACAAGGTGCG
89	AAT ACC TTC TTT TTC CAG CTC TTC
90	ACTTGACCTGGCTAAAGGTG
91	CAT TTA AAA GCA CCT TTG AAA GCG
92	GCGATAAATTTGGTGCCGTC

(continued)

Table 1
(continued)

93	GAG TTA ATC CCG CTG GCT GC
94	CACCGGATGAACTTTCACTTATCC
95	AGG TCA CCG TTG TTC AGC AC
96	CAGGCTTTACACTTTATGCTTCCG
97	TAAGTTGGGTAACGCCAGGG
98	CTGACCGCAGAACAGGCAGC
99	GCGCCTGAATGGTGTGAGTG
100	CACGAGTTCTGCACAAGGTC
101	GTAGCACACGAGGTCTCT

4. TAE gel electrophoresis buffer: (50× stock) Dissolve 242 g Tris base in approximately 750 mL deionized water. Add 57.1 mL glacial acetic acid and 100 mL of 0.5 M EDTA (pH 8.0), and adjust the solution to a final volume of 1 L.
5. SOB media: Dissolve in 900 mL deionized water 20 g tryptone, 5 g yeast extract, 0.5 g NaCl, 0.186 g KCl, and 0.952 g $MgCl_2$. Adjust volume to 1,000 mL. Adjust pH to 7.0; autoclave to sterilize.
6. SOC media: Add 20 mL of 1 M glucose to 1 L SOB media.
7. Electroporator, 0.2 cm electroporation cuvettes.
8. MA salts: Dissolve 52.5 g potassium phosphate (dibasic), 22.5 g potassium phosphate (monobasic), 5 g ammonium sulfate, and 2.5 g sodium citrate hydrate in 1 L deionized water. Autoclave to sterilize.
9. Blasticidin-S (Research Products International Corp.).
10. LB media: Dissolve 10 g tryptone, 5 g yeast extract, and 10 g NaCl in 800 mL deionized water. Adjust pH to 7.0 with 1 N NaOH. Adjust volume to 1 L with water. Autoclave to sterilize.
11. Agar.
12. Low-salt LB: Same as LB broth, except 5 g NaCl instead of 10 g NaCl.
13. Glycerol.

2.5. Barcoded Genomic Analysis

1. Purelink Genomic Mini Kit (Invitrogen).
2. Taq polymerase (New England Biolabs).

3. Custom Oligonucleotide Array Geneflex Tag4 16KV2 array (Affymetrix).
4. Hybridization oven 640 (Affymetrix, part no. 800138) This Affymetrix instrumentation is typically available at most university core facilities.
5. GeneChip Fluidic Station 450 (Affymetrix, part no. 00–0079) This Affymetrix instrumentation is typically available at most university core facilities.
6. MES (2-(*N*-morpholino)ethane sulfonic acid)-free acid monohydrate.
7. MES sodium salt.
8. MES stock solution: 12 × 0.70 g of MES-free acid monohydrate, 1.9 g of MES sodium salt, and 8 mL of H_2O, and adjust volume to 10 mL. Adjust pH to between 6.5 and 6.7. Shield from light by wrapping bottle in foil. Store at 4°C. Replace if solution becomes visibly yellow or after 1 month.
9. 2× Hybridization buffer: 8.3 mL of 12× MES stock, 17.7 mL of 5 M NaCl.
10. 4.0 mL of 0.5 M EDTA, 0.1 mL of 10% Tween 20 (v/v), and 19.9 mL of filtered dH_2O. Filter to remove dust particles that would interfere with scanning. Shield from light by wrapping bottle in foil. Store at 4°C. Replace as for MES stock solution.
11. Denhardt's solution: Dissolve 10 g Ficoll 400, 10 g polyvinylpyrrolidone, and 10 g bovine serum albumin (BSA) in 900 mL deionized water. Adjust volume to 1 L. Filter solution prior to storage through a 0.2 μM filter. Store at 4°C.

3. Methods

3.1. General Strategy for TRMR Library Creation and Selection

TRMR combines oligonucleotide synthesis with recombineering for the creation of libraries of overexpressed (up) and underexpressed (down) genes in *E. coli*. Briefly, oligonucleotides containing (1) homology regions corresponding to each gene in *E. coli* and (2) short DNA barcodes are synthesized. These oligos are ligated to a synthetic DNA cassette containing a blasticidin-resistance marker (bsd) and either a strong promoter (pL-TetO (18)) and strong ribosome-binding site (RBS) or no promoter and no RBS. The ligated products are rearranged by circular ligation, rolling-circle amplification, and restriction digestion to arrange the homology regions, bsd cassette, and promoter/RBS sequence for recombineering. PCR on the resulting constructs provides the double-stranded, linear substrates for Lambda Red recombineering (17).

The recombineering process occurs in multiplex on the chosen base strain (in this case, MG1655), resulting in the creation of "up" and "down" genotypes for each gene in *E. coli*. Once the library is created, selections can be performed using appropriate challenges and selection techniques. Genomic DNA from the isolated cells resulting from the selection process then be rapidly characterized using an Affymetrix DNA microarray corresponding to the assigned barcodes or by high-throughput DNA sequencing. The rapid creation of libraries and post-selection characterization make TRMR a powerful tool for genomic analysis in *E. coli*.

3.2. Oligonucleotide Design

Targeting oligos have been designed for all protein-coding genes in *E. coli* K12, gene annotation from the Ecogene database version 2.20 (http://www.ecogene.org/). Sequences of RNA genes, pseudogenes, and insertion elements were excluded. Each 189-mer targeting oligo contains flanking regions for PCR priming (43 b), downstream homology region (48 b), AscI site (8 b), upstream homology region (52 b), Tag priming site P3 (18 b), and barcode sequence (20 b) (see Fig. 1). In all, 8,154 targeting oligos were designed to create two possible expression alleles for 4,077 genes. Targeting regions were chosen such that DNA cassettes would insert upstream of genes and replace the translation start codon. In 584 instances gene overlap occurs where allelic replacement may result in disruption of the upstream gene. In these cases up to five nucleotides including the start codon are replaced such that a stop codon (UAG) is inserted in-frame of the upstream gene. Because gene overlap is usually minimal (1–4 nucleotides), cassette insertion usually results in no change to the upstream gene product or a single amino acid change at the C-terminus. In 4% of instances, a small truncation of the upstream gene product has occurred. Once designed, the set of targeting oligos, each 189 nucleotides in length, were purchased through limited access to the oligonucleotide library synthesis product from Agilent. The sequences of these oligos have been published and are available in the supplementary material of Warner et al. (16).

Molecular barcodes, known as "barcodes" or "tags", were chosen from the experimentally verified set used in the yeast deletion collection (19). Barcode tag sequences were downloaded from http://chemogenomics.stanford.edu/supplements/04tag/download.html and converted to 5′-3′ orientation. This collection of tags was further narrowed by excluding sequences that would lead to cleavage of the DNA during library synthesis (see below) and sequences that may hybridize with the regions used to amplify the tag sequences. Tags in this set were excluded if they have the following sequences (M = A or C, K = G or T): GGCGCGCC, CCGCGCGG, CCTCAGA, GCTGAGG, GTMKAC, GGCGCG, CACGAG, GAGGTC, GCACGA, TAGTGA, GAGTCG, TCC CTA, CTCGTG, GACCTC, TCGTGC, TCACTA, CGACTC,

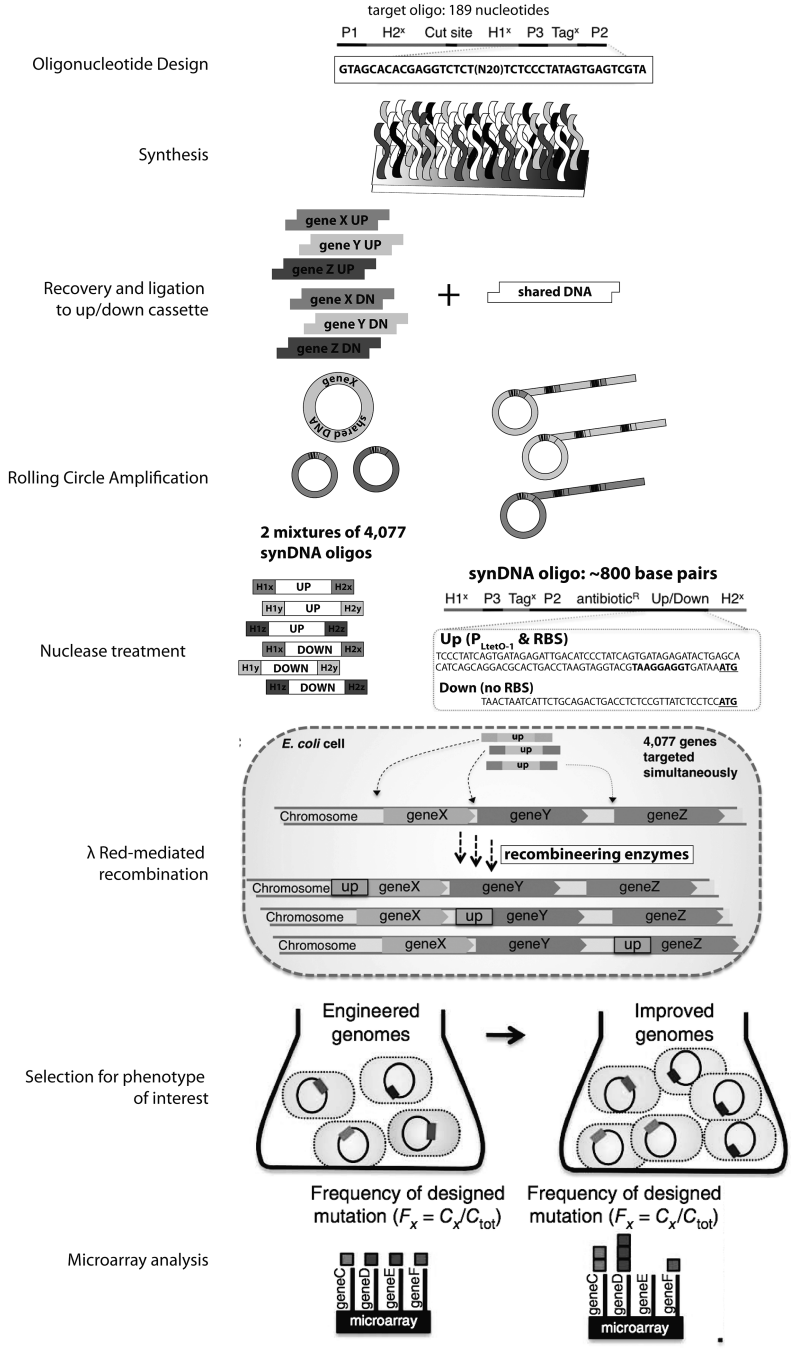

Fig. 1. Schematic of TRMR library creation, selection, and analysis.

TAGGGA. Tags were then assigned randomly to the 8,154 targeting oligos.

Ligation sequences (8 base, non-palindrome) are added to "target" oligos and "shared" DNA during PCR amplification. The downstream ends of targeting oligos have 10-base 3′ overhangs, which leaves a 2 base gap on the non-template strand after ligation of "shared" DNA with "target" oligos. The purpose of the gap is to hypothetically improve rolling-circle amplification (RCA) by allowing the two strands of DNA circles to unwind, separate, and allow unhindered/constrained RCA. These 3′ overhang sequences are targeting oligos, upstream, AGGTCAGT-3′; downstream, AGAAGCGGAT-3′; shared DNA, upstream, ATCCGC TT-3′; and downstream, ACTGACCT-3′.

3.3. Preparation of "Shared" DNA: Up and Down Blasticidin® Cassette Construction

To select for recombineered genomes, we used resistance to the antibiotic blasticidin-S as a selection marker. Blasticidin-S deaminase was chosen as a reporter gene because of its small size (approximately 400 bp) to optimize recombineering efficiency of the cassette (see Note 1). The construct was designed to contain flanking FRT sites, which allow for *flp*-recombinase-based removal of the antibiotic-resistance gene using a plasmid such as pCP20 (3) if desired. As a base vector, we used the pEM7/bsd plasmid (Invitrogen) and inserted the cassette via nested PCR with primers 7–16 in Table 1 and cloning by standard methods (16). The synthetic DNA cassettes required can be constructed by a number of standard procedures or can be provided commercially (see Note 2).

The final sequences of the shared DNA cassettes are:

"Up" shared DNA sequence prior to uracil excision
5′-AAGCGGATAGTAGACCTCAGCGAAGTTCCTATTCT
CTAGAAAGTATAGGAACTTCACGCTAGGGATAACAGGGT
AATTGTTGACAATTAATCATCGGCATAGTATATCGGCATA
GTATAATACGACAAGGTGAGGAACTAAACCATGGCCAAG
CCTTTGTCTCAAGAAGAATCCACCCTCATTGAAAGAGCA
ACGGCTACAATCAACAGCATCCCCATCTCTGAAGACTAC
AGCGTCGCCAGCGCAGCTCTCTCTAGCGACGGCCGCAT
CTTCACTGGTGTCAATGTATATCATTTTACTGGGGGACCT
TGTGCAGAACTCGTGGTGCTGGGCACTGCTGCTGCTGC
GGCAGCTGGCAACCTGACTTGTATCGTCGCGATCGGAAA
TGAGAACAGGGGCATCTTGAGCCCCTGCGGACGGTGCC
GACAGGTGCTTCTCGATCTGCATCCTGGGATCAAAGCC
ATAGTGAAGGACAGTGATGGACAGCCGACGGCAGTTGG
GATTCGTGAATTGCTGCCCTCTGGTTATGTGTGGGAGGG
CTAATAGGGATAACAGGGTAATATAGAAGTTCCTATTCTC
TAGAAAGTATAGGAACTTCTCCCTATCAGTGATAGAGATT
GACATCCCTATCAGTGATAGAGATACTGAGCACATCAGCA
GGACGCACTGACCT-3′

"Down" shared DNA sequence prior to uracil excision
5′-AAGCGGATAGTAGACCTCAGCGAAGTTCCTATTCT
CTAGAAAGTATAGGAACTTCACGCTAGGGATAACAGGGT
AATTGTTGACAATTAATCATCGGCATAGTATATCGGCATA
GTATAATACGACAAGGTGAGGAACTAAACCATGGCCAAG
CCTTTGTCTCAAGAAGAATCCACCCTCATTGAAAGAGCA
ACGGCTACAATCAACAGCATCCCCATCTCTGAAGACTAC
AGCGTCGCCAGCGCAGCTCTCTCTAGCGACGGCCGCAT
CTTCACTGGTGTCAATGTATATCATTTTACTGGGGGACCT
TGTGCAGAACTCGTGGTGCTGGGCACTGCTGCTGCTGC
GGCAGCTGGCAACCTGACTTGTATCGTCGCGATCGGAAA
TGAGAACAGGGGCATCTTGAGCCCCTGCGGACGGTGCC
GACAGGTGCTTCTCGATCTGCATCCTGGGATCAAAGCCA
TAGTGAAGGACAGTGATGGACAGCCGACGGCAGTTGGG
ATTCGTGAATTGCTGCCCTCTGGTTATGTGTGGGAGGG
CTAATAGGGATAACAGGGTAATATAGAAGTTCCTATTCTC
TAGAAAGTATAGGAACTTCTAACTAATCATTCTGCAGACT
GACCT-3′

3.4. Amplification and Uracil Excision of "Up" and "Down" Cassettes for Ligation to "Target" Oligos (Fig. 2)

1. Using PfuTurbo Cx hotstart DNA polymerase (Stratagene) (see Note 3), amplify the "up" and "down" cassettes using primers 18 and 19 and 18 and 20 in a 100 μL reaction following the manufacturer recommended polymerase conditions.

   PCR conditions:

   2 min at 95°C

   30 cycles of 30 s at 96°C, 30 s at 56°C, and 1 min at 72°C

   5 min at 72°C
2. Treat unpurified PCR products with two units of USER enzymes (New England Biolabs) at 37°C for 2 h to generate single-stranded 3′ overhangs (see Note 4).
3. Purify the reaction products with QIAquick PCR Purification Kit (Qiagen) and elute in 50 μL water. Treat purified products with 20 units DpnI (New England Biolabs) (see Note 5) in a total of 60 μL for 2 h at 37°C.
4. Separate products by agarose gel electrophoresis and excise the correct band corresponding to the major product (approximately 600 bp) using the QIAquick gel extraction protocols (Qiagen). These "up" and "down" cassettes will be ligated to the "target" DNA libraries.

3.5. Amplification of Target Oligos

To construct the original TRMR library, an oligonucleotide library containing a theoretical total of 10 pmol of 8,154 unique "target" sequences was purchased from Agilent as described in Subheading 3.1. The following protocols assume an identical oligonucleotide library:

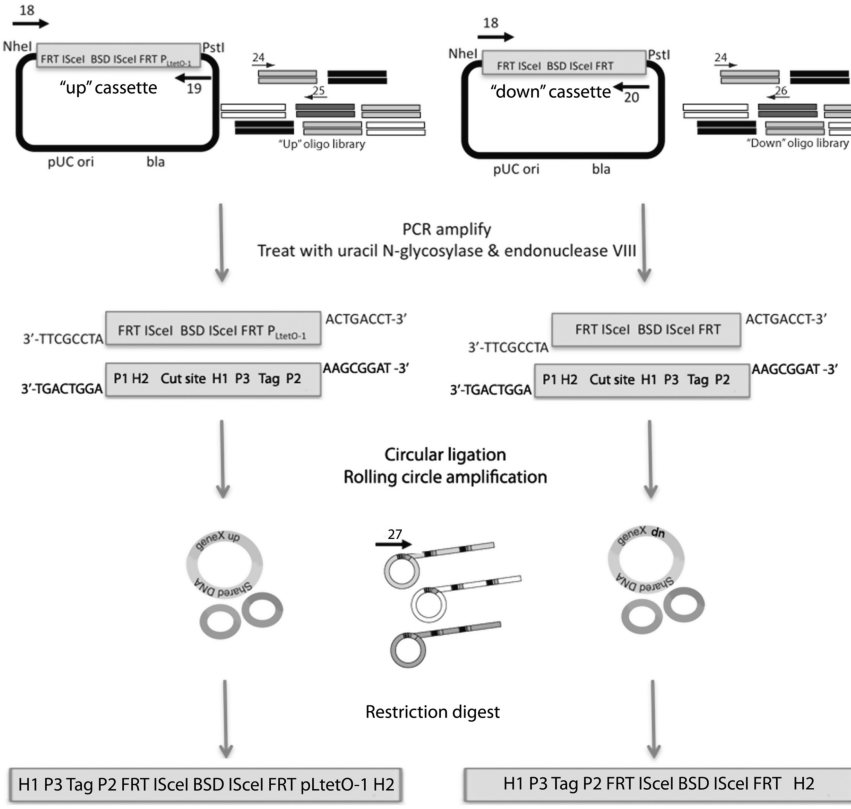

Fig. 2. Preparation and ligation of shared DNA with synthetic oligo libraries and rearrangement by rolling-circle amplification and digestion.

1. Dissolve lyophilized library in 50 μL of Buffer TE.
2. To amplify the "up" library, add 3 μL of library DNA as the template for PCR in a 50 μL solution containing PAGE-purified primers 21 and 22 (0.5 mM each), 0.2 mM each dNTP, 1× PfuTurbo Cx reaction buffer, and 2.5 units PfuTurbo Cx hotstart DNA polymerase (Stratagene) (see Note 6).

   PCR conditions:

   95°C for 2 min.

   17 cycles of 30 s at 95°C, 30 s at 58°C, and 30 s at 72°C.

   5 min at 72°C.

The "down" library DNA is amplified under the same reaction conditions using PAGE-purified primers 21 and 23.

3. Purify reactions by the QIAquick PCR purification protocols (Qiagen) and elute in 50 μL of 5 mM Tris–Cl, pH 8.5.
4. A portion of the "up" library DNA from the first amplification is PCR amplified a second time to attach the sequences used for subsequent ligations. Prepare a 250-μL solution containing PAGE-purified primers 24 and 25 (0.5 mM each), approximately 60 ng "up" PCR product, 0.2 mM each dNTP, 1× PfuTurbo Cx reaction buffer, and 12.5 units PfuTurbo Cx hotstart DNA polymerase.

   PCR conditions:

   95°C for 2 min

   18 cycles of 30 s at 95°C, 30 s at 60°C, and 30 s at 72°C

   5 min at 72°C

   Likewise, the "down" library DNA was PCR amplified a second time under the same reaction conditions using PAGE-purified primers 24 and 26.
5. Add four units USER enzyme (New England Biolabs) and EDTA to 0.1 mM to each reaction. Incubate at 37°C for 2 h to generate the 3′ single-stranded overhangs.
6. Separate products by agarose gel electrophoresis (1.5% agarose) (see Note 7) and excise the correct band corresponding to the major product (approximately 200 bp) using the QIAquick gel extraction protocols (Qiagen), eluting in 60 μL of 5 mM Tris–Cl, pH 8.5. These "up" and "down" cassettes will be ligated to the "shared" DNA constructs created in Subheading 3.3. The sizes of the constructs should be 199 bp and 207 bp for the "up" after the first and second PCRs, and 200 bp and 215 bp "up" after the first and second PCRs.

3.6. Ligation of "Target" DNA to "Shared" DNA and Circularization

1. In a total volume of 100 μL of 1× T4 ligase buffer containing 200 units T4 DNA ligase (New England Biolabs), mix approximately 0.5 pmol of "target" DNA with 1.5 pmol "shared" DNA.
2. Carry out ligations separately but in parallel for the "up" DNA and "down" DNA (see Note 8): Heat to 37°C for 10 min then allow to react for ~15 h at 18°C to generate DNA circles with a 2 base gap on the sense strand.
3. Inactivate ligase by heating at 65°C for 10 min.

3.7. Rolling-Circle Amplification of Ligation Products

The circular products of ligation are copied into linear concatemers by multiply primed rolling-circle amplification. This circularization and subsequent digestion put the H1 and H2 homology

regions, barcode, and bsd gene in the correct orientation for recombineering.

1. Divide the ligation product for the "up" library into five aliquots of 20 μL.
2. Add 20 nmol exo-resistant random hexamer (primer 27) with 6 μL phi29 buffer (New England Biolabs) to each reaction.
3. Heat to 96°C for 4 min, then cool on ice.
4. Add 160 μL of the following mixture (final concentration in 200 μL reactions): Phi29 DNA polymerase buffer (1×), dithiothreitol (10 mM), bovine serum albumin (200 μg/mL), dNTPs (1 mM each), phi29 DNA polymerase (9 units, New England Biolabs), and yeast inorganic pyrophosphatase (0.1 units, New England Biolabs). Incubate at 32°C overnight.
5. Stop the reactions by addition of EDTA to 10 mM and inactivation of the polymerase at 65°C for 20 min.
6. Desalt DNA by diluting to 450 μL with deionized water (see Note 9) and concentrating to 50 μL by spinning in an Amicon YM30 filter (Millipore) for 5 min at 10,000 × *g*.
7. Repeat step 6 until dilution and concentration steps are carried out four times in total. The total volume of each aliquot should be 200 μL.
8. These procedures are also carried out in parallel to amplify the "down" library.

3.8. Nuclease Treatment of RCA Products and Ethanol Precipitation of DNA

1. Add 22 μL of buffer 4 (New England Biolabs) to an aliquot each of the RCA product of the "up" and "down" libraries.
2. Add 20 units AscI restriction endonuclease (New England Biolabs) and incubate at 37°C for 4–8 h.
3. Inactivate enzyme by heating to 65°C for 15 min.
4. Precipitate DNA by addition of sodium acetate (0.3 M final, pH 5.2) and 480 μL cold ethanol for 1 h to overnight at −20°C.
5. Collect DNA pellet by centrifugation at 13,000 × *g* for 5 min. Discard supernatant.
6. Wash twice by addition of 500 μL cold 70% ethanol and centrifugation as above.
7. After discarding supernatant from the second ethanol wash, allow sufficient time for pellet to dry.
8. Dissolve pellet in 100 μL water (see Note 10).
9. Add 11 μL mung bean nuclease reaction buffer (New England Biolabs) and mung bean nuclease (1 unit/μg DNA, New England Biolabs) to remove 5′ terminal extensions (see Note 11). Incubate at 32°C for 40 min.

10. Stop reaction by addition of EDTA to 10 mM and SDS to 0.01%.
11. Separate products by agarose gel electrophoresis and excise the correct band corresponding to the major product (approximately 800 bp) using the QIAquick gel extraction protocols (Qiagen), eluting in 100 μL of 5 mM Tris–Cl, pH 8.5.
12. Concentrate DNA by ethanol precipitation (steps 4–8 above) to 0.5 μg/mL in 20 μL water. Each RCA aliquot carried though these steps yields 3–10 μg of synDNA. Approximately 7–10 μg DNA was used to make each allele library.

3.9. Recombineering Optimization

Transformation of the double-stranded cassette into cells expressing the Lambda Red genes leads to incorporation of the cassette into the genome by homologous recombination. Before constructing the library, it is advisable to perform recombineering operations on a small scale to optimize recombineering efficiency.

1. Grow *E. coli* cells (see Note 12) containing the recombineering plasmid pSIM5 (17) (see Note 13) in a 50 mL SOB cultures with 34 μg/mL chloramphenicol at 30°C to an OD_{600} of 0.7.
2. Divide the contents into two 50 mL conical tubes at 25 mL each.
3. Incubate one tube in a shaking water bath at 42°C for 15 min to induce the Lambda Red enzymes (see Note 14).
4. Transfer tubes to an ice water bath for 10 min (see Note 15).
5. Spin cells at 5000 rpm = approximately 3200 × g for 4 min and resuspend with 1 mL cold deionized water by swirling in an ice water bath. Dilute each aliquot to 12 mL with cold water.
6. Spin cells at 5,000 rpm for 4 min and resuspend in 1 mL dH_2O.
7. Dilute each aliquot to 15 mL.
8. Spin cells at 5,000 rpm for 4 min and resuspend in 300 mL dH_2O.
9. Transform aliquots of cells (50 μL) in a 0.1 cm electrocuvette with varying amounts (e.g., 50–500 ng) of "up" or "down" synDNA and a pulse of 12.5 kV cm^{-1}.
10. Recover cells from each transformation in 1 mL SOC medium for 2 h at 37°C.
11. Plate cells on low-salt LB agar and low salt with and without 90 μg/mL blasticidin-S. For the plate without antibiotic, dilute cells by a factor of $10^{4–5}$ to facilitate colony counting.
12. Count colonies on each plate to estimate number of total viable colonies (nonselective plate) to total recombineered colonies (blasticidin-S plate). Expect a ratio of between $1:10^4$ and $1:10^5$ engineered colonies to total viable colonies.

13. Calculate total number of engineered colonies upon scale-up. For each "up" or "down" library, the number of recombinants needed for complete library coverage of 4,077 genes at a probability of 0.99 is approximately 53,000 as calculated by the GLUE web interface (20).

3.10. Scale-up of Recombineering for Full Library Construction

1. Grow *E. coli* cells containing the recombineering plasmid pSIM5 (17) in 2 × 400 mL SOB cultures with 34 μg/mL chloramphenicol at 30°C to an OD_{600} of 0.7.
2. Transfer contents of flasks to 50 mL conical tubes in water baths at 42°C for 15 min to induce the Lambda Red enzymes.
3. Transfer tubes to an ice-water bath for 10 min.
4. Spin cells at 5,000 rpm for 4 min and resuspend with 1 mL cold deionized water by swirling in an ice water bath.
5. Dilute each aliquot to 12 mL and combine to a total of 4 × 50 mL conical tubes.
6. Spin cells at 5,000 rpm for 4 min and resuspend in 1 mL dH_2O.
7. Dilute each aliquot to 15 mL and combine into 2 × 50 mL conical tubes.
8. Spin cells at 5,000 rpm for 4 min and resuspend in 3.3 mL dH_2O.
9. Transform aliquots of cells (400 μL) in a 0.2 cm electrocuvette with approximately 1 μg of "up" or "down" synDNA (see optimization in Subheading 3.9) and a pulse of 12.5 kV cm^{-1}. Transformation was carried out eight times to generate the "up" allele library and eight times to generate the "down" allele library.
10. Recover cells from each transformation in 12 mL SOC medium for 2 h at 37°C.
11. Spin cells at 5,000 rpm for 4 min and resuspend in 30 mL MA salts.
12. Repeat centrifugation and resuspension twice more with the final resuspension of each aliquot (up and down) to a volume of 2 mL in MA salts.
13. Spread the "up" and "down" allele libraries separately (100 μL for each plate) onto a total of 40 low-salt LB agar plates containing blasticidin-S (90 μg/mL) and allowed to grow at 37°C for 22 h (see Note 16).
14. Scrape colonies from the agar plates and "up" and "down" allele libraries and suspend in a total of 35 mL LB.
15. Concentrate cells by centrifugation and suspension to 3 × 10^9 cells/mL in LB medium containing 16% glycerol and 90 μg/mL blasticidin-S.

16. Store aliquots of the "up" or "down" cell libraries (1 mL) at −80°C.

3.11. Creation of a Barcoded Wild-Type Control Strain

To estimate the effect of TRMR mutations in comparison to the wild type, it is necessary to create a control strain containing a barcode similar to the barcodes present in TRMR libraries. Here we describe the construction of a barcoded version of our strain of choice, MG1655. This control strain, called JWKAN, is used during library selections.

1. Use primers 1 and 2, which contain 50 bp regions of homology to the *E. coli* chromosome near the attTn7 site to PCR amplify the kanamycin-resistance gene from the pKD13 plasmid (primer 1 also contains the molecular barcode tag 7661 from the yeast deletion collection flanked by the same sequences used for amplification of the tags in the TRMR libraries.)
2. Using the above recombineering procedure (steps 1–9 in Subheading 3.9), insert the PCR product into the MG1655 chromosome.
3. Select for recombinants on LB agar plates containing kanamycin (30 μg/mL) at 37°C. The desired recombinants replace nucleotide 3,909,796 with the tagged kanamycin cassette.
4. Recombinants are verified by PCR amplification of the flanking regions of the cassette inserted into the chromosome using primers 3 and 4 or 5 and 6.
5. Sequence this region to confirm correct insertion of the tag.

3.12. Growth Selections of TRMR Libraries

Growth selections on the libraries can be performed on solid or liquid media, using any number of chemical growth inhibitors or antibiotics. Examples of selection and screen conditions can be found in Warner et al. (16).

1. From freezer stock aliquots, inoculate 50 mL low-salt LB medium containing 90 μg/mL blasticidin-S with 2×10^9 TRMR "up" and 2×10^9 TRMR "down" cells.
2. Grow with shaking at 37°C to an optical density at 600 nm of 0.8 (see Note 17).
3. Spin cells at 4,500 × *g* for 6 min, decant, and suspend in 30 mL of MA salts (or in whatever media the growth selection will occur).
4. Collected once more by centrifugation and suspend in desired selection media to a concentration of 5×10^8 cells/mL.
5. Add barcoded wild-type control (JWKAN) cells to a final concentration of 7.7×10^4 cells/mL.

6. Save and freeze a 1.7 mL aliquot of the cell library (called recovery culture) for microarray analysis.
7. Use the remainder of the culture for various growth selections.
8. As an example, liquid selections can be carried out with shaking at 37°C in 600 mL of MOPS minimal medium containing 2 mM phosphate and 4% w/v glucose or in 600 mL LB medium. Each medium is inoculated with 2.4×10^8 cells from a recovery culture and allowed to grow to an optical density at 600 nm of 1.0–1.2. Cells are collected from each culture by centrifugation of 10 mL aliquots at $4{,}500 \times g$ for 6 min, decanted, and stored at −80°C for microarray analysis.
9. Growth on various selective agars can be carried out by spreading a total of 0.7×10^8 cells of the allele mixtures recovered from freezer aliquots on five plates for each selective condition. Plates are incubated at 37°C until colonies are visible (1–3 days), depending on selection conditions.
10. After selections have concluded, recover cells for genomic DNA preparation.

3.13. Preparing DNA for Barcode Genotyping

1. Extract genomic DNA from ~10^9 *E. coli* cells using Purelink Genomic Mini Kit (Invitrogen).
2. Amplify barcode tags in 300 µL PCR reactions (final concentrations: 1× PCR buffer, 2.5 mM $MgCl_2$, 0.2 mM each dNTP, 1 mM each primer 5′-GTAGCACACGAGGTCTCT and 5′-Biotin-TACGACTCACTATAGGGAGA, 0.6 U μL^{-1} Taq polymerase and 0.5 µg genomic DNA or 30 pg synDNA).

   PCR conditions:
   2 min at 95°C

   25 cycles of 30 s at 96°C, 30 s at 55°C, and 1 min at 72°C

   5 min at 72°C
3. Purify by agarose gel electrophoresis and extraction using the QIAquick gel extraction protocols (Qiagen, substitute buffer QX1 for QG) (see Note 18).

3.14. Microarray Analysis

Analysis of barcoded genotypes can be performed via DNA microarray scanning. This process requires specialized equipment from Affymetrix but is available at most university core facilities. For our experiments, a custom Geneflex Tag4 16 K V2 array based on yeast barcode arrays (see Subheading 3.1 for design specifications) was used (see Note 19).

1. The procedure for microarray hybridizations to the Geneflex Tag4 16 K V2 array (Affymetrix) has been described elsewhere (REF). As it is generally carried out by staff of core facilities, we will present the steps required for DNA preparation. In our

experiments, 600 ng of purified tags (combined "up" and "down" tags) were hybridized along with 10–20 tags (amplified and purified as above) included at known concentrations (0.5 pM–10 nM). The 10–20 control tag sequences were chosen randomly from the narrowed collection of unused tags, and the template was purchased from Integrated DNA Technologies with flanking primer regions P3 and P2 (shown in Fig. 1). The templates of the control tag sequences were separately PCR amplified under the same conditions as the sample DNA. The concentration of each control tag sequence was measured, and tags were spiked into samples for chip hybridization.

2. Control tags are amplified from 2 nM final concentration of each oligo template in a 60 μL PCR reaction as described above.
3. Desalt PCR products by three rounds of dilution/concentration using Microcon YM30 filter units by adding 0.5× hybridization buffer to a final volume of 450 μL and centrifugation until the retentate is less than 100 μL or purified by gel electrophoresis and extraction. Final concentrations of each tag are measured (concentrations typically are between 10–20 ng/μL).
4. Prepare four arrays worth of hybridization mix (HM) as 225 μL 2× hybridization buffer, 1.2 μL of 100 nM dilution of oligo 5′ biotin-CTG AAC GGT AGC ATC TTG AC-3′, 36 μL mixed oligos MO (MO = 1.25 μL each of oligos 105, 127, 128, and 129 diluted to 50 μL in water), and 9 μL 50× Denhardt's solution.
5. For each array aliquot 60 μL HM, add up to 10 μL control tags and up to 30 μL sample tags. Dilute each aliquot to 100 μL and submit for processing as in Pierce et al. (21).
6. Background hybridization is calculated from the average intensity of 1,642 unused tag probes; threshold intensity was set to background hybridization +2 stdev. The intensities of the 10–20 spiked tags were used to calculate allele concentrations and correct for array saturation. Allele frequencies are calculated by dividing allele concentrations by the total concentration of all alleles detected on the array.
7. Microarray analysis of the recovery culture (before selection) confirms successful creation of the TRMR library (i.e., coverage of down/up genotypes). Analysis of the post-selection microarray is used to determine enrichment of particular phenotypes relevant to the selection at hand.

3.15. Confirmation of TRMR Colonies

To confirm successful creation of the TRMR library, it is advisable to analyze individual colonies to assess the diversity of the library.

Plating of the library before selection gives colonies that can be tested for various up or down alleles (including barcode tags) by colony PCR. Individual colonies that survive selections can be also isolated and characterized.

1. Perform colony PCR using 1× Taq polymerase buffer, 2.5 mM $MgCl_2$, 0.2 mM each dNTPs, 0.3 mM each primer 100 and 101, 5% DMSO, and 1.5 units Taq polymerase (Invitrogen). Add colonies to the reactions by toothpick.

   PCR conditions:
   95°C for 3 min

   35 cycles of 94°C for 40 s, 55°C for 30 s, and 72°C for 30

   72°C for 5
2. Analyze products by agarose gel electrophoresis, with the correct size product appearing at 386 bp.
3. Sequence PCR products using primer 52. The incorporated tag sequences are obtained from the reverse complement sequence and cross-referenced to identify alleles.

PCR can be used to identify chromosome insertion sites by using primers that hybridize to chromosomal DNA approximately 450 bp upstream or downstream of the insertion site and primers (101 or 100, respectively) within the synthetic cassette. Example loci and primers (upstream, downstream) tested are the following: sodC (28,29), ampH (30,31), yeiG (32,33), yobD (34,35), leuL (36,37), ilvN (38,39), emrD (40,41), xylA (42,43), hns (44,45), gloA (46,47), pbpG (48,49), ihfB (50,51), ydeK (52,53), ybgQ (54,55), tnaC (56,57), ydeM (58,59), cyaA (60,61), yhaM (62,63), paaI (64,65), puuE (66,67), ptsI (68,69), ugpE (70,71), ydiH (72,73), ygaZ (74,75), yciV (76,77), rluF (78,79), lsrA (80,81), panF (82,83), lpp (84,85), ahpC (86,87), talA (88,89), elaD (90,91), yqhC (92,93), pykF (94,95), lacZ (96,97), and galK (98,99).

4. Notes

1. While we used blasticidin-S, any antibiotic-resistance or auxotrophic marker could be used if the strain is appropriate. Kanamycin resistance (NeoR) is a common choice for recombineering experiments.
2. If desired, the "up" and "down" cassettes could be synthesized by a company such as DNA 2.0 and GenScript for on the order of a few hundred USD and 2–3 weeks.
3. PfuTurbo Cx polymerase is high fidelity, yet tolerant of deoxyuracil primers. Use of other polymerases may result in decreased or no yield from PCRs involving deoxyuracil.

4. NEB USER enzymes are functional in the PCR buffer, so a purification step is not necessary prior to uracil excision.
5. DpnI selectively digests dsDNA that is methylated at 5′-GATC-3′ sites. It therefore is used to remove plasmid DNA (the parent from the original PCR reaction), which, if transformed, would lead to bsd-resistant colonies in the absence of recombineering.
6. Amplification of targeting oligos may need to be optimized, especially in the first round of amplification. A number of polymerases may be tested for optimal PCR yield.
7. For electrophoresis of small (100–200 bp) PCR products, e.g., the oligo libraries, 1.5–2% agarose is recommended to reduce diffusion of bands.
8. Be sure to mix "up" target DNA with "up" shared DNA and "down" target DNA with "down" shared DNA.
9. The concatameric DNA product of rolling-circle amplification can be rather viscous and jellylike, necessitating dilution before further processing.
10. Proteins are also precipitated in this process. Yields may be improved by phenol/chloroform extraction of proteins prior to ethanol precipitation.
11. Treatment with mung bean nuclease prevents the formation of recombineering side products that are thought to have arisen from in vivo ligation of AscI site overhangs.
12. For our libraries, we used the wild-type-like strain MG1655, although theoretically any strain of *E. coli* that tolerates Lambda Red recombineering could be used.
13. There exists an array of plasmids expressing the Lambda Red recombinase genes (11). We used plasmid pSIM5, but if chloramphenicol resistance is not desired, many other plasmids could accomplish the task. A complete list is available at http://redrecombineering.ncifcrf.gov/Plasmids.html.
14. On pSIM5, the Lambda Red enzymes *gam*, *bet*, and *exo* are controlled by a heat-inducible promoter. Incubation times of 15–20 min are optimal for expression of these genes at 42°C (11).
15. For optimum cell viability, keep cells as close to 4°C as possible during electrocompetent cell preparation.
16. Some wild-type cells may also grow under these conditions. Blasticidin-S is less effective under high-salt conditions, and this may be a factor in growth of wild-type cells under these selection conditions. Optimizing blasticidin-S concentration on plates may alleviate this potential problem.
17. The optical density of recovering cells initially drops, possibly due to presence of wild-type cells or nonviable TRMR cells.

18. Tag purification reduces background hybridization.
19. DNA microarrays are useful for assessing the sequence of barcoded PCR products by hybridization. However, recent advances in DNA sequencing technologies make direct sequencing of a barcoded pool a viable alternative. High-throughput sequencing techniques such as Illumina sequencing, which can provide millions of runs on short (ca. 100 bp) regions of DNA, are ideal for sequencing of barcoded regions.

References

1. Zhang Y, Buchholz F, Muyrers JPP, Stewart AF (1998) A new logic for DNA engineering using recombination in *Escherichia coli*. Nat Genet 20(2):123–128
2. Murphy KC (1998) Use of bacteriophage lambda recombination functions to promote gene replacement in *Escherichia coli*. J Bacteriol 180(8):2063
3. Datsenko KA, Wanner BL (2000) One-step inactivation of chromosomal genes in *Escherichia coli* K-12 using PCR products. Proc Natl Acad Sci U S A 97(12):6640–6645
4. Fu J, Bian X, Hu S, Wang H, Huang F, Seibert PM, Plaza A, Xia L, Muller R, Stewart AF, Zhang Y (2012) Full-length RecE enhances linear-linear homologous recombination and facilitates direct cloning for bioprospecting. Nat Biotechnol 30(5):440–446
5. Murphy KC, Campellone KG, Poteete AR (2000) PCR-mediated gene replacement in *Escherichia coli*. Gene 246(1–2):321–330
6. Ellis HM, Yu D, DiTizio T, Court DL (2001) High efficiency mutagenesis, repair, and engineering of chromosomal DNA using single-stranded oligonucleotides. Proc Natl Acad Sci U S A 98(12):6742–6746
7. Zhang Y, Buchholz F, Muyrers JP, Stewart AF (1998) A new logic for DNA engineering using recombination in *Escherichia coli*. Nat Genet 20(2):123–128
8. Wang HH, Isaacs FJ, Carr PA, Sun ZZ, Xu G, Forest CR, Church GM (2009) Programming cells by multiplex genome engineering and accelerated evolution. Nature 460(7257): 894–898
9. Sawitzke JA, Costantino N, Li XT, Thomason LC, Bubunenko M, Court C, Court DL (2011) Probing cellular processes with oligo-mediated recombination and using the knowledge gained to optimize recombineering. J Mol Biol 407(1):45–59
10. Court DL, Swaminathan S, Yu D, Wilson H, Baker T, Bubunenko M, Sawitzke J, Sharan SK (2003) Mini-lambda: a tractable system for chromosome and BAC engineering. Gene 315:63–69
11. Datta S, Costantino N, Court DL (2006) A set of recombineering plasmids for gram-negative bacteria. Gene 379:109–115
12. Yu D, Ellis HM, Lee EC, Jenkins NA, Copeland NG, Court DL (2000) An efficient recombination system for chromosome engineering in *Escherichia coli*. Proc Natl Acad Sci U S A 97 (11):5978–5983
13. Mosberg JA, Lajoie MJ, Church GM (2010) Lambda red recombineering in *Escherichia coli* occurs through a fully single-stranded intermediate. Genetics 186(3):791–799
14. Gibson DG, Benders GA, Andrews-Pfannkoch C, Denisova EA, Baden-Tillson H, Zaveri J, Stockwell TB, Brownley A, Thomas DW, Algire MA (2008) Complete chemical synthesis, assembly, and cloning of a *Mycoplasma genitalium* genome. Science 319(5867):1215
15. Kosuri S, Eroshenko N, LeProust E, Super M, Way J, Li JB, Church GM (2010) A scalable gene synthesis platform using high-fidelity DNA microchips. Nat Biotechnol 28(12): 1295
16. Warner JR, Reeder PJ, Karimpour-Fard A, Woodruff LBA, Gill RT (2010) Rapid profiling of a microbial genome using mixtures of barcoded oligonucleotides. Nat Biotechnol 28 (8):856–862
17. Sharan SK, Thomason LC, Kuznetsov SG (2009) Recombineering: a homologous recombination-based method of genetic engineering. Nat Protoc 4(2):206–223

18. Lutz R, Bujard H (1997) Independent and tight regulation of transcriptional units in Escherichia coli via the LacR/O, the TetR/O and AraC/I1-I2 regulatory elements. Nucleic Acids Res 25(6):1203–1210
19. Pierce SE, Fung EL, Jaramillo DF, Chu AM, Davis RW, Nislow C, Giaever G (2006) A unique and universal molecular barcode array. Nat Methods 3(8):601–603
20. Patrick WM, Firth AE, Blackburn JM (2003) User-friendly algorithms for estimating completeness and diversity in randomized protein-encoding libraries. Protein Eng 16(6):451–457
21. Pierce SE, Davis RW, Nislow C, Giaever G (2007) Genome-wide analysis of barcoded Saccharomyces cerevisiae gene-deletion mutants in pooled cultures. Nat Protoc 2(11): 2958–2974

Part III

Systems-Level ‘Omics Tools

Chapter 13

Identification of Mutations in Evolved Bacterial Genomes

Liam Royce, Erin Boggess, Tao Jin, Julie Dickerson, and Laura Jarboe

Abstract

Directed laboratory evolution is a common technique to obtain an evolved bacteria strain with a desired phenotype. This technique is especially useful as a supplement to rational engineering for complex phenotypes such as increased biocatalyst tolerance to toxic compounds. However, reverse engineering efforts are required in order to identify the mutations that occurred, including single nucleotide polymorphisms (SNPs), insertions/deletions (indels), duplications, and rearrangements. In this protocol, we describe the steps to (1) obtain and sequence the genomic DNA, (2) process and analyze the genomic DNA sequence data, and (3) verify the mutations by Sanger resequencing.

Key words: Directed evolution, Mutations, Single nucleotide polymorphisms, Insertions/deletions, Duplications, Rearrangements, Sequencing, Sequence data processing, Systems metabolic engineering

1. Introduction

Bacteria acting as biocatalysts for production of biorenewable fuels and chemicals are often faced with product-mediated inhibition. For example, ethanol was shown to negatively impact growth and structure of *Escherichia coli* and yeast (1, 2); the effects of succinate were revealed on the membrane and enzymes of yeast (3, 4); and butanol was shown to inhibit the growth and sugar uptake rate of *Clostridium acetobutylicum* (5, 6).

Lignocellulosic biomass has been extensively utilized as a source of carbon and energy for the fermentative production of ethanol and other biorenewable fuels (7–9). However, the sugar streams released from this biomass frequently contain inhibitory contaminants that restrict the growth and substrate utilization of the microorganism (10–12).

Thus, the fermentative production of biorenewable fuels and chemicals is associated with both inhibitory contaminants in the feedstock and inhibitory products. In these cases, it can be sometimes useful to increase the tolerance of the biocatalyst to these

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_13,

inhibitory compounds. Metabolic or directed evolution is frequently used to increase the tolerance of bacteria to inhibitory compounds by selecting for random mutations under appropriate selective pressure. While metabolic evolution is sufficient to acquire a strain with the desired phenotype, it is often of interest to identify the mutations acquired during the evolutionary process.

Reverse engineering can yield a roadmap for reproducing the desired phenotype or behavior in other biocatalysts. This method begins with whole genome sequencing using high-throughput sequencing technology, such as Illumina's sequencing by synthesis technique. Bioinformatics methods known as de novo assembly and mapping (or alignment) are used to analyze the short-read data and reconstruct the genome (13–15). By obtaining DNA sequences of the parent and evolved organism genomes, it is possible to perform a comparative analysis and identify variations in the evolved strain. Isolation of bacteria genomes is a standard procedure. Sequencing platforms are changing rapidly in their throughput and chemistry to increase availability and fidelity of sequence data (16). As sequencing data becomes more readily available, there remain many challenges to the processing and analysis of sequence data, which are costly and time-consuming. Automation by computer programs alleviates the burden of manual analysis. The finishing step and gap filling in DNA sequence analysis are bottlenecks in automation (17). In the recent decade, there has been a great amount of improvement in automating the process with computer programs; however, this step still requires human intervention.

As the genotype of the evolved strain is defined, hypotheses are formed regarding the roles of mutations in the context of the phenotype. As researchers elucidate which mutations improve fitness, the intent is to infer the mechanisms that lead to the increased tolerance to toxicity and then proceed with rational engineering techniques (12, 18). This can also enable researchers to identify functions of undercharacterized enzymes and pathways (19). However, the focus of this chapter is to describe the use of genome sequence analysis to identify the mutations acquired in an evolved strain. Determination of which of these mutations impact the phenotype and understanding the mechanism of the mutation's function is outside the scope of this chapter.

2. Materials

All materials used are standard kits and reagents. Software for high-throughput sequence analysis generally requires UNIX/Linux operating systems with a large amount of memory and storage.

Both free and commercial software packages are available for analyzing high-throughput sequencing data. All software included in this protocol is open source unless otherwise noted.

2.1. Genomic DNA Purification and Sequencing

1. Luria broth (LB) for growing bacteria cells: Dissolve at 25 g/L in nanopure water and filter-sterilize using a 0.22 CA bottle top filter.
2. 1.5 mL microfuge tubes and 50 mL centrifuge tubes for sample processing.
3. QIAGEN DNeasy® Blood & Tissue Kit for genomic DNA isolation and purification. Buy RNase A and 100% ethanol separately.
4. AccuBlock Digital Dry Bath for temperature-controlled incubation.
5. NanoDrop spectrophotometer for genomic DNA quantification and quality control.
6. Illumina cBot System and Illumina TruSeq PE Cluster Kit-GA for cluster generation and Illumina GAII sequencing instrument for short-read whole genome sequencing (available at a university core facility, prices vary).

2.2. Bioinformatics Software for High-Throughput Sequence Data

1. Galaxy is a scientific workflow system for high-throughput sequence data preprocessing, integration, and analysis. A free public server is available, but most users will need to download and install the open-source Galaxy software locally due to the upload limitations and to preserve data privacy. UNIX/Linux and Mac OS X are supported, and a recent version of Python must be installed (20–22).
2. FastQC provides quality control checks for raw sequence data and generates summary graphs and basic statistics. FastQC is available through the Galaxy interface or for download and independent installation (23).
3. FASTX-Toolkit is a collection of scripts for manipulating raw sequence data. It includes conversion, trimming, and filtering tools and will generate some quality statistics. The FASTX-Toolkit is distributed with Galaxy or can be downloaded and installed independently (24).
4. Mapping software: Bowtie, Bowtie 2, and BWA are popular short-read aligners that are distributed under the GPLv3 license. Bowtie and BWA are distributed with Galaxy. Memory requirements vary by algorithm and input data, but at least 2 GB memory is required and at least 4 GB is recommended. Multiple processors can also improve alignment speed. It is critical to read the manual for mapping software because different parameters will generate different alignments.

5. De novo assembly software: Velvet and ABySS (available for download and distributed under the GPLv3 license) are examples of the many available de Bruijn graph-based assemblers (25, 26). Other assemblers that use an overlap/layout/consensus approach are available, but take considerably longer to assemble short reads and are not considered for this protocol. While many assemblers support 32-bit platforms, a 64-bit machine is recommended, and memory requirements vary by algorithm, short-read data, and selected *k*-mer length. It is critical to read the assembler manuals because different parameters generate different contigs/scaffolds.
6. Basic Local Alignment Search Tool (BLAST) is the most widely used sequence similarity tool. A web interface is available through NCBI, but a local installation of the BLAST + open-source applications provide a command line usage (27).
7. SAMtools is a collection of utilities for manipulating alignments. BCFtools, which is distributed with SAMtools, performs variant calling. SAMtools is distributed with Galaxy and can also be independently installed (14, 15).

2.3. Mutation Verification

1. Primer3 software (distributed under GPLv2) for primer design and primers (28).
2. Plate Spinner Centrifuge.
3. Commercial 10 mM Tris–HCl, pH = 8.5 buffer.
4. 96-well PCR plates.
5. Polymerase chain reaction (PCR) materials: QIAGEN® Taq PCR Master Mix Kit or QIAGEN® LongRange PCR Kit and Strain Genomic DNA (*from Subheading* 2.1, item 1).
6. Gel loading materials: Blue (6×) gel loading dye and ethidium bromide, 1% solution/molecular biology, for visualization of PCR products and 1 Kb plus DNA Ladder for size determination.
7. 50× TAE: 242 g Tris base, 57.1 mL glacial acetic acid, and 18.6 g EDTA dissolved in 900 mL nanopure water. Add makeup nanopure water to 1 L.
8. TAE DNA gel for separating DNA fragments: Dissolve 1% W/V agarose in 1× TAE.
9. Gel electrophoresis equipment.
10. PCR Purification Kit to purify PCR products.
11. DNA sequence finishing software Phred/Phrap/Consed or CodonCode Aligner (17, 29, 30).
12. Thermal cycler for generating PCR products.

3. Methods

Obtaining the evolved strain and interpretation of mutation function is outside the scope of this chapter. Here we restrict this protocol to DNA purification, genome sequencing, analysis, and verification.

3.1. Obtain Sequence Data

1. After obtaining an evolved bacteria colony isolate, prepare to use the QIAGEN DNeasy® Blood & Tissue Kit. Other commercial kits can also be used to isolate the genomic DNA. First, grow the parent strain (before the evolution experiment) and the evolved strain overnight in 25 mL LB.
2. Follow the QIAGEN DNeasy® Blood & Tissue Kit protocol for gram-negative bacteria.
   (a) Harvest cells (maximum 2×10^9 cells) in 50 mL centrifuge tube by centrifuging for 20 min at 4°C, ~5,000 × *g*. Discard supernatant (see Note 1).
   (b) Resuspend pellet in 180 µL Buffer ATL *and transfer to a microcentrifuge tube.*
   (c) Add 20 µL proteinase K. Mix thoroughly by vortexing, and incubate at 56°C in a temperature-controlled water bath until the cells are completely lysed (3 h). Vortex *every hour.*
   (d) Add 20 µL RNase A, briefly vortex, and incubate at room temperature for 2 min.
   (e) Add 200 µL Buffer AL, and mix thoroughly by vortexing. Then add 200 µL 100% ethanol, and mix again thoroughly by vortexing (see Note 2).
   (f) Pipet the sample into the DNeasy Mini spin column, and centrifuge at *maximum speed* for 1 min. Discard flow-through (see Note 3).
   (g) Add 500 µL Buffer AW1, and centrifuge at *maximum speed* for 1 min.
   (h) Add 500 µL Buffer AW2, and centrifuge at *maximum speed for 1 min*. Discard flow-through, and centrifuge again at *maximum speed for 1 min* to dry the column (see Note 4).
   (i) Place the DNeasy Mini spin column in a clean 1.5 mL microfuge tube, and pipet 100 µL Buffer AE directly onto the DNeasy membrane. *Incubate at room temperature for 1 min, and then centrifuge at maximum speed for 1 min to elute. Add another 100 µL Buffer AE, incubate for 1 min, and then centrifuge at maximum speed for 1 min* (see Note 5). Freeze DNA at −20°C, or proceed directly to the next step.

3. Check the quality of the genomic DNA on a NanoDrop. First, blank the spectrophotometer with 1 μL nanopure water. Wipe away the water, then add the sample. One should see a smooth profile with a minimum at 230 nm and a maximum at 260 nm. Typical values should be ~20 μg genomic DNA, 280/260 value of ≥1.8, and a 260/230 value of ≥2. If the quality is too low, repeat the wash steps 2.7–2.9 with a new column.
4. Submit ≥2 μg/sample genomic DNA (at least one parent strain sample and one evolved strain sample) to a core facility for whole genome sequencing. There are many options to choose which sequencing instrument and which sequencing method; currently the DNA core facility at Iowa State University has a GAII sequencer from Illumina, Inc. 75-cycle paired-end sequencing is recommended as the researcher obtains more reads at a higher quality. To date, Illumina offers 150-cycle paired-end data with the GAII sequencer and 100-cycle paired-end data on their HiSeq instrument. If submitting more than one sample, indexing is the best option as one pays only for one sequencing lane. Indexing allows to a maximum of 12 samples in a single lane. The workflow of the Illumina platform is shown in Fig. 1: Illumina sample preparation protocol, adapted from the Illumina guide *Preparing Samples for Sequencing Genomic DNA*. See the Illumina guidebooks for their detailed protocols. Refer to the Illumina website for their sequencing technology: http://www.illumina.com/technology/sequencing_technology.ilmn (see Note 6).

3.2. Preprocess Sequence Data

High-throughput sequence data is most commonly stored in FASTQ format. FASTQ format represents each read as a set of lines: header, sequence, sequence ID (optional), and quality scores in ASCII encoding. These text files typically have a `.fq`, `.fastq`, or `.txt` extension.

1. Before beginning analysis, identify what quality scoring encoding is associated with the raw data. Different Illumina genome analyzer pipeline software versions use different scoring scheme variations (e.g., Illumina 1.3+, Illumina 1.5+, and Illumina 1.8+). If there is difficulty identifying which encoding is used, FastQC includes this in its output. Software user manuals will specify if a particular scoring scheme is expected as input, and it may be necessary to perform a conversion prior to analysis using Galaxy, the FASTX-Toolkit, or using a "Bio*" library in the language of your choice (e.g., *BioPerl*, *Biopython*, *BioRuby*, *BioJava*).
2. Use FastQC to perform an initial quality assessment of raw data. Launch the FastQC GUI, and open FASTQ data files to

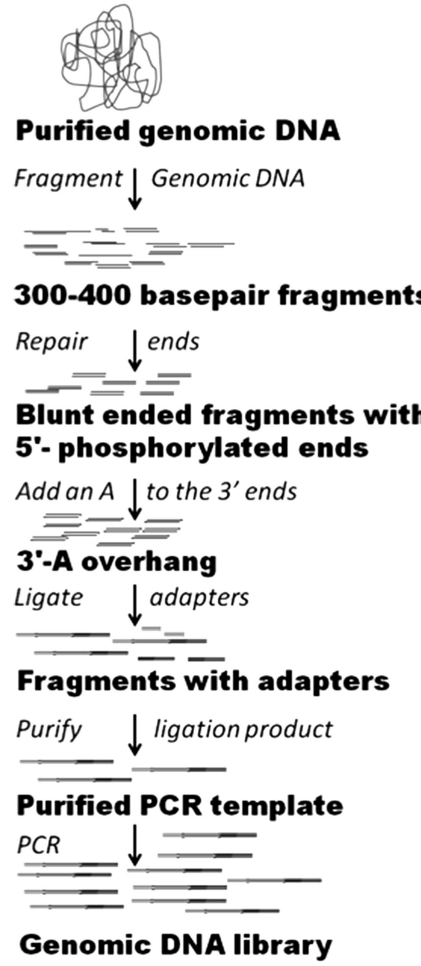

Fig. 1. Illumina sample preparation protocol, adapted from the Illumina guide *Preparing Samples for Sequencing Genomic DNA*. See the Illumina guidebooks for their detailed protocols.

generate all FastQC reports at once. Reports and graphs are presented in HTML format and can be saved for reference.

Examine per-base quality, per-sequence quality, per-base content, and length distributions (not applicable for Illumina reads). Also check for overrepresentation of sequences and if they correspond to contaminants or PCR artifacts (in addition to common artifacts provided by FastQC, users may supply sequences of potential contaminants to screen for).

Use the summary icons (green, normal; orange, slightly abnormal; and red, very unusual) as guidelines in the following preprocessing steps. It is important to acknowledge that not all preprocessing steps will be necessary for all data and also that having small abnormalities may be acceptable in the context of the data and should not prevent a researcher from proceeding with analysis.

3. Perform read trimming using the FASTX-Toolkit if necessary. Read quality deteriorates with position, and base calls near the end of a read are more prone to error. An appropriate length to trim may be determined from FastQC output. Use the `fastx-trimmer` command from the FASTX-Toolkit:

   ```
   $ fastx_trimmer [-f N] [-l N] [-i INFILE] [-o OUTFILE]
   ```

   where `[-f N]` specifies the first base to keep (default is 1), `[-l N]` specifies the last base to keep (default is entire read), `INFILE` specifies the FASTQ file, and `OUTFILE` is the name to give the trimmed data file. More advanced techniques allow for adaptive read trimming; however, reads of varied length may not be acceptable as input for all analysis software.
4. Filter reads by overall quality with the FASTX-Toolkit:

   ```
   $ fastq_quality_filter [-q N] [-p N] [-i INFILE] [-o OUTFILE]
   ```

   The minimum quality score to keep is `[-q N]`, and `[-p N]` is the minimum percentage of bases that must have `[-q]` quality.
5. Remove sequencing artifacts, described as reads that are predominantly one base (e.g., `AAAAAAAAAAAAAAAACAAACA`), using the FASTX-Toolkit:

   ```
   $ fastx_artifacts_filter [-i INFILE] [-o OUTFILE]
   ```

   where `INFILE` specifies the FASTQ file and `OUTFILE` is the name to give the filtered data file.
6. Remove adapter sequences (identified as overrepresented sequences in the FastQC report or defined in protocol) with the FASTX-Toolkit:

   ```
   $ fastx_clipper [-a ADAPTER] [-l N] [-i INFILE] [-o OUTFILE]
   ```

   where `[-a ADAPTER]` is the adapter sequence that is to be removed from 3′ end of sequences, `[-l N]` is the minimum length of reads to keep in the dataset (default is 5), `INFILE` specifies the FASTQ file, and `OUTFILE` is the name to give the filtered data file.
7. Resubmit filtered and trimmed data to FastQC to verify improved data quality, and recalculate data summary statistics before proceeding with analysis.

3.3. Map Short Reads to Reference Genome

1. Build a reference index (using the reference genome in FASTA format) using the alignment algorithm of your choice, e.g.,

   ```
   $ bwa index [-p prefix] [-a algoType] ref.fa # BWA
   $ bowtie-build [options]* ref.fa <prefix> # Bowtie
   $ bowtie2-build [options]* ref.fa <prefix> # Bowtie 2
   ```

where `ref.fa` is the reference genome in FASTA format and `prefix` is the prefix of the output database and also the database filename. Additional options are defined in the corresponding user manuals.

Using the genome of the parent strain as the reference yields the best alignments. If the genome of the parent strain has not been sequenced, download the genome of the wild-type laboratory strain from a public online database such as NCBI RefSeq. One benefit of using the wild-type genome as reference is the ability to easily leverage existing annotation in publically available databases (e.g., BioCyc).

2. Align reads to the reference and generate a SAM file. The SAM file format is a TAB-delimited text file that contains information such as alignment position (or "*" for unaligned reads) and mapping quality for each read and is the common output format for aligners. SAMtools performs conversions between SAM and a compressed and indexed binary format called BAM.
3. Assess overall alignment quality by reviewing the summary statistics generated by mapping software such as the percentage of reads that aligned to the reference genome. Use SAMtools to calculate read depths for each position of the genome:

```
$ samtools depth aln.sorted.bam > depth.txt
```

where `aln.sorted.bam` is the sorted BAM file. The output file, `depth.txt`, contains one line for each position in the reference genome. The second column is the coordinate, and the third column is the number of reads that cover that position. The SAMtools depth utility does not report positions where read depth is zero; thus, the number of lines in the file is equal to the number of bases where coverage is nonzero. Alternative, specify a depth cutoff to ignore very small read depths (i.e., do not consider depth = 1 as genome coverage). Calculate base coverage with one of the following:

```
$ wc -l depth.txt # depth > 0
$ awk '$3 > $N{i++}END{print i}' depth.txt # depth > N
```

Divide base coverage by genome size to obtain the percentage of the genome covered by reads.

Map quality scores can also be examined by investigating column 5 (MAPQ) of the SAM file.

3.4. De Novo Assembly

De novo assembly and mapping of short reads to a reference sequence are fundamentally different analysis procedures. Assembly of the genomes of evolved bacterial strains can be used to search for novel insertions and complex mutations that are difficult for mapping software to identify. Additionally, results from assembly methods can provide support for proposed alignments.

Assembly of short-read data does not use a reference sequence and instead tiles reads to generate sequences called contigs. Incorporating the average distance between paired-end reads (called the insert size) is used to join contigs into scaffolds. The most important parameter in de Bruijn graph-based assembly algorithms is the hash length, which is also known as the *k*-mer length. Large *k*-mer values require longer overlap between reads in order for them to be assembled (therefore, the *k*-mer value may not be larger than the read length). Conversely, small *k*-mer values require short overlap which results in increased sensitivity, but decreased specificity. The experimenter must provide the *k*-value parameter, and there is no method to find the optimal value. Because of this, it is recommended that researchers test multiple *k*-mers and then compare several assemblies before proceeding.

1. Assemble short-read data with the assembler of your choice. Test multiple *k*-mer values, and calculate the total number of contigs, N50, and N90 for each assembly (typically reported by assembly software).
2. Proceed with the "best" assemblies such that the number of contigs is minimized and the N50 and N90 are maximized.

3.5. Identify Variations in Evolved Strains

1. Identify single nucleotide polymorphisms (SNPs) and short insertions/deletions (indels) for an alignment using SAMtools/BCFtools:

   ```
   $ samtools mpileup -uf ref.fa aln.bam | bcftools view -bvcg - > var.raw.bcf
   ```

   where `ref.fa` specifies the reference genome in FASTA format and aln.bam is a binary alignment file. The output is a binary file (BCF) for Variant Call Format (VCF) TAB-delimited files. VCF is standard for storing information about variants in alignment data.
2. Find large deletions by inspecting read depth (the number of reads mapped to a specific position on the reference genome). Calculate read depth values using SAMtools:

   ```
   $ samtools idxstats aln.bam
   ```

   Regions with zero or very low read depth may indicate deletions. Determining what qualifies as "low" read depth may be aided by examining the read depth distribution.
3. Use assembly results to distinguish complex mutations such as large insertions, duplications, and inversions that are difficult for mapping algorithms to identify. Align contigs to a reference genome or an alignment consensus sequence.

   First, generate the consensus sequence from an alignment file with SAMtools:

```
$ samtools mpileup -uf ref.fa aln.bam | bcftools
view -cg - | vcfutils.pl vcf2fq > cns.fq
```

where `cns.fq` is the output consensus sequence. Next, BLAST contigs against these sequences to reveal sequence variations. Syntenic dotplots can also be used to visually identify discontinuities.

4. If possible, leverage reference genome annotation to form hypotheses about the effects of mutation. Verification by targeted sequencing can be used to confirm mutations. More advanced experimentation is necessary to confirm hypothesized effects.

3.6. Verify Mutations

1. Obtain a list of mutations from the above analysis and the sequences of the regions of interest.
2. There are two approaches for obtaining primers for PCR: for genes and for noncoding regions.
   (a) For mutated regions containing open reading frames (ORFs or genes), first, note how large the gene is and round up to the nearest 1,000. Add the additional sequences upstream and downstream of the gene equally. Then split the sequence into 1,000 bp segments (see Notes 7 and 8). This method will give you room to pick optimal primers to include the entire sequence of interest. Use the Primer3 program to design optimal primers whose PCR product size range is 851–1,000 bp in length (see Note 9). Paste in the first 1,000 bp sequence, use the other default values, and click "Pick Primers." Select any of the suggested primers, noting where they bind to the template and the product size. Add the next 1,000 bp block of sequence and repeat until complete.
   (b) For mutated regions within a noncoding region (NCR), take a 1,000 bp segment of DNA sequence, and set the suspected mutation in the middle (~500 bp from the first base). This ensures good sequencing data of this region. Use the Primer3 program for NCRs the same way for ORFs (see Note 10).
3. For long sequencing regions (>1 kb), the above method will have gaps in the total sequence. In order to fill in the gaps, repeat the process with a 500 bp offset, and choose the reverse complement primers that bind in the middle of the sequence. This will also increase the fidelity of the sequence data (see Fig. 2: Schematic for designing sequencing primers) for long sequencing regions.
4. After choosing the primers, order them from an oligo synthesis company. If you have multiple primers, a 96-well plate format may be convenient. Resuspend them in either nanopure water

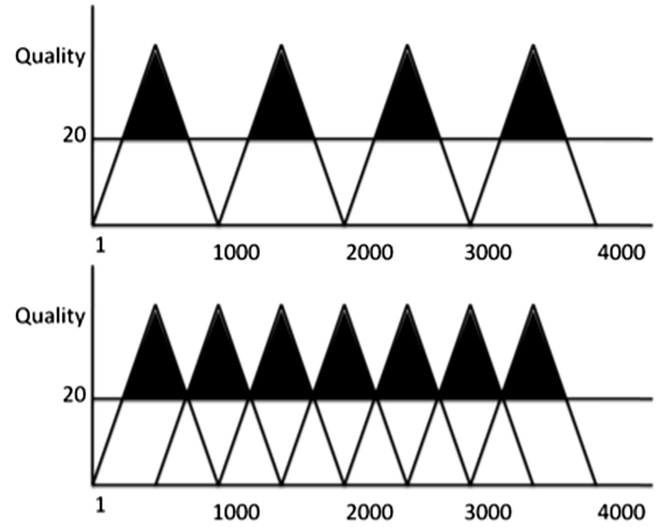

Fig. 2. Schematic for designing sequencing primers for long sequencing regions. The average quality score is plotted for each individual base along the template. A quality score of 20 or greater is considered acceptable. Choose the forward primer binding to the lagging strand to cover the general area (*top*). To fill the gaps, use the reverse primer binding to the leading strand, with a 500 bp offset. In this example, the forward sequencing primers will bind to bases 1, 1,000, 2,000, and 3,000. The reverse sequencing primer will bind to bases 3,500, 2,500, and 1,500.

or 10 mM Tris–HCl, pH = 8.5 at 100 μM, vortex, and centrifuge briefly.

5. Keep all PCR materials on ice, and set up your PCR in a 96-well plate according to Table 1. Reserve one well for a negative-control PCR (no template, choose any primer pair) to check for contamination. Run PCR using a thermocycler according to Table 2. If the sequencing region is longer than 1 kb, it is possible to make a long PCR product and then submit multiple sequencing primers for a single template (up to 5 kb) (see Note 11). For higher fidelity, especially at longer sequencing templates (>5 kb) or difficult templates (high GC content), use the QIAGEN® LongRange PCR Kit according to Table 3 and Table 4.
6. Check the concentration of the PCR products using a NanoDrop (see Subheading 3.1, step 3). Check the size of the PCR product on a 1% TAE agarose gel.
   (a) Melt 1× TAE with 1% agarose gel in a standard microwave. Add 25 mL with 2 drops of ethidium bromide to a 8.5 × 10 cm gel casting tray in a gel casting tray holder with either an 8- or 15-sample comb (see Note 12). For more samples, use a 17 × 10 gel casting tray with a 26-sample comb. In this case, use 50 mL of 1% TAE agarose

Table 1
A typical 20 μL PCR

Material	Stock concentration	Amount to add	Final concentration
Nuclease-free water	–	Variable	–
Primer A	100 μM	0.1 μL	500 nM
Primer B	100 μM	0.1 μL	500 nM
Template	Variable	Variable	50–500 ng
2× QIAGEN® Taq PCR Master Mix	2×	10 μL	1× or 2.5 U Taq +200 μM dNTP

Table 2
Typical PCR conditions

Step	Time	Temperature	Comments
1. Denaturation	4 min	94°C	Denaturation of template and primer-dimers
2. Denaturation cycle	0.5 min	94°C	
3. Annealing cycle	0.5 min	55°C	Or 5°C below the lowest primer melting temperature
4. Extension cycle	1 min/kb	72°C	
5. Repeat steps 2–4			Repeat 30 times
6. Final extension	10 min	72°C	
7. End	Infinite	4°C	

Table 3
20 μL QIAGEN® LongRange PCR Kit (up to 10 kb) setup

Material	Stock concentration	Amount to add	Final concentration
Nuclease-free water	–	Variable	–
Primer A	10 μM (diluted tenfold)	0.8 μL	400 nM
Primer B	10 μM (diluted tenfold)	0.8 μL	400 nM
Template	Variable (diluted tenfold)	Variable	0.1–10 ng
dNTP mix	10 mM of each base	1 μL	500 μM of each base
QIAGEN® LongRange PCR buffer	10×	2 μL	1× or 2.5 mM Mg^{2+}
QIAGEN® LongRange PCR enzyme mix	100 U (total enzyme mix)	0.16 μL	0.8 U

Table 4
QIAGEN® LongRange PCR conditions

Step	Time	Temperature	Comments
1. Denaturation	3 min	93°C	Denaturation of template and primer-dimers
2. Denaturation cycle	15 s	93°C	
3. Annealing cycle	0.5 min	62°C	Or 5°C below the lowest primer melting temperature
4. Extension cycle	1 min/kb	68°C	
5. Repeat steps 2–4			Repeat 35 times
6. End	Infinite	4°C	

Table 5
Recipe for mixing standard and samples

Standard		Samples	
Standard	1 μL	PCR product	8 μL
Dye	2 μL	Dye	2 μL
Nuclease-free water	7 μL	Nuclease-free water	–

gel with a few drops of ethidium bromide. Allow 30 min for the gel to solidify.

(b) Remove the comb and the gel casting tray. Place the gel casting tray into the gel box. Add 1× TAE until the surface of the gel is covered evenly.

(c) Mix standard and samples according to Table 5. Mix the standard and load into the first well; perform the same with each sample. Set the voltage to 100, put the top on, and click "run" (see Note 13). Wait 45 min–1 h for the dye to reach the bottom. Turn the system off when finished.

(d) Use the UV camera to visualize the PCR products. Match the standard with the PCR product to determine the approximate size (see Note 14).

7. If the PCR worked as expected, submit samples for sequencing by a core facility or company. The sample may need to be purified (use a standard PCR Purification Kit protocol) before submission (check the submission requirements). Use the sequencing primers as described in Fig. 2 (see Note 15). The sequencing data is returned as `.ab1` trace files and `.seq` files. One can view the `.seq` files in any text editor program. More advanced analysis requires the use of `.ab1` trace files and DNA sequence finishing software.

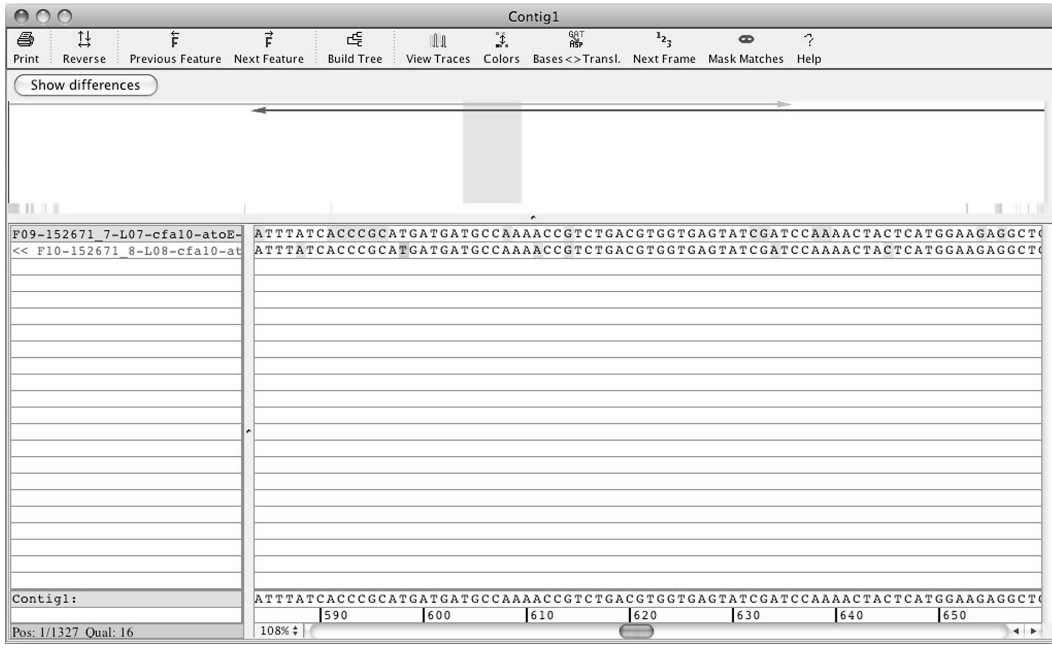

Fig. 3. Contig assembly in CodonCode Aligner.

8. In CodonCode Aligner (or any sequence finishing software), load the forward and reverse sequence of the samples. The first 20 bases and the last few bases (depends on the sequence length) have low quality scores. Highlight the samples and choose "clip ends" using the default parameters. Highlight the overlapping samples and assemble them into contigs (see Fig. 3). The consensus sequence is shown at the bottom, where the base with the highest quality score is chosen. Here one can manually edit the sequence and call individual bases that are difficult. If there are discrepancies, open the trace files again to determine which is correct (see Note 16).
9. Use the BLAST alignment tool (choose "Align two or more sequences" option) to align the consensus sequence to the parent strain and wild-type sequence. Sequences that are not matching or are unknown can be found using the NCBI nucleotide BLAST database.

4. Notes

1. The methods described here are developed in our lab, unless it is a published protocol from Illumina, Inc. or commercial kit protocols. The steps using commercial kits are the published protocols of the kit manufacturer, where special deviations are

in *italics*. Harvesting cells at 4°C, 5,000 x*g* prevents lysis and increases DNA yields. Do not overload the DNA column. Overloading the column causes blockage of the membrane and decrease yields. To obtain the maximum cell count per DNA column, first obtain a correlation of OD (we use 550 nm for *E. coli*) to *C* cells/mL (outside the scope of this protocol). Next, use Eq. 1 to calculate the amount of cells you need:

$$\frac{2 \times 10^9 cells}{c^{cells/mL} \times 10mL} = x \quad OD_{550} \tag{1}$$

For example, if $C = 1.69 \times 10^8$, we have $\frac{2 \times 10^9 cells}{1.69 \times 10^8 \ cells/mL \times 10mL} = 1.18 \quad OD_{550}$. Therefore, 10 mL, OD_{550} 1.18 is required to obtain 2×10^9 cells.

2. The ethanol, sample, and Buffer AL need to be mixed immediately and thoroughly by vortexing. Otherwise, local precipitation may occur in the sample, which will decrease yields.
3. Buffer AL and Buffer AW1 are not compatible with bleach and may form decomposition products.
4. The column must be dried before eluting the DNA. Residual ethanol will decrease yields.
5. Subsequent elution steps will increase DNA yields, but decrease concentration. Do not elute more than 200 μL into a single 1.5 mL microfuge tube.
6. The insert sizes are less than 800 bp. We typically use 400–500 bp.
7. Due to possible polar effects from upstream mutations, the researcher may want to include the complete sequence from the promoter to the stop codon of the gene of interest. The sequencing length may be prohibitive and costly, so it is up to the researcher to include the upstream sequences along with the gene of interest. This is especially true if there are many genes in between the annotated promoter sequence and the gene of interest.
8. For example, the *E. coli* gene *carB* is 3,222 bp; therefore, round up to 4,000 bp by using this formula: 4,000−3,222 = 778/2 = 389 bp. Add 389 bp upstream and downstream of the *carB* gene. The total sequence is therefore 4,000 bp with the *carB* gene in the middle. This is enough to include a sequencing primer region and the promoter sequence 42 bp from the translational start.
9. The limit of good quality Sanger sequence reads is about 1,000 bp. The Primer3 program chooses optimal primers and

performs *in silico* PCR to obtain PCR products in the desired range. This ensures that each primer has approximately the same length and melting temperature. It is good to also select alternate primers in case the primers weakly bind to the template. If the region is heavily mutated, it may be difficult choosing the correct primers.

10. NCRs may include long blocks of A–T-rich sequences, and therefore, optimal primers may not be available. In this case, adjust the target sequence so that the mutated region is closer to either the 5′ end or 3′ end. This way one can obtain optimal primers that can be used for sequencing this region.
11. QIAGEN® Taq DNA Polymerase is for general applications. For longer PCR products, the probability for incorporation of the incorrect base increases (false SNP); therefore, the use of a higher-fidelity enzyme (i.e., QIAGEN® LongRange PCR Kit) is recommended. Higher-fidelity PCR enzymes are recommended for SNP identification and resequencing applications. It depends on the researcher which option is best. For extremely long PCR (10–40 kb), the researcher is referred to the QIAGEN® LongRange PCR Handbook for an alternate PCR protocol.
12. Ethidium bromide is toxic and mutagenic. Always wear proper protection equipment.
13. Make sure that the diode colors match (black with black and red with red) and that the black one is at the top. This ensures that the DNA samples will run through the gel in the correct direction. Also the dye should not run off the gel, otherwise one may lose the samples.
14. If DNA bands are not visible, soak the gel for 1 h in 1× TAE with ethidium bromide.
15. Use the following formula to calculate the number of sequencing reactions:

$$\#\ \textit{Quality sequences} = \frac{bp}{1000} \times 2 - 1 \qquad (2)$$

16. Common mismatches occur when the local sequence contains blocks of the same base (i.e., A block of 6 As in a row) or the ends are overlapping with one sample containing poor quality bases. This step may not be necessary as the consensus sequence is called according to quality. If SNPs or indels are discovered, this step becomes much more difficult, especially if there are duplication events.

Acknowledgement

We would like to thank Michael Baker at the DNA facility of the Iowa State University Office of Biotechnology for his input on next-generation sequencing using the Illumina platform. We also thank Emily Rickenbach, an undergraduate student who helped automate the method for picking primers for verifying mutations. Funding was provided for this work by the NSF Engineering Research Center for Biorenewable Chemicals (CBiRC), NSF award number EEC-0813570 and NSF Energy for Sustainability award number CBET-1133319.

References

1. Ingram LO, Buttke TM (1984) Effects of alcohols on micro-organisms. Adv Microb Physiol 25:253–300
2. Trinh CT, Huffer S, Clark ME, Blanch HW, Clark DS (2010) Elucidating mechanisms of solvent toxicity in ethanologenic *Escherichia coli*. Biotechnol Bioeng 106(5):721–730
3. Duro AF, Serrano R (1981) Inhibition of succinate production during yeast fermentation by de-energization of the plasma membrane. Current Microbiology 6:111–113
4. Smith EH, Janknecht R, Maher LJ 3rd (2007) Succinate inhibition of alpha-ketoglutarate-dependent enzymes in a yeast model of paraganglioma. Hum Mol Genet 16(24):3136–3148
5. Winkler J, Kao KC (2011) Transcriptional analysis of *Lactobacillus brevis* to N-butanol and ferulic acid stress responses. PLoS One 6 (8):e21438
6. Schwarz KM, Kuit W, Grimmler C, Ehrenreich A, Kengen SW (2012) A transcriptional study of acidogenic chemostat cells of *Clostridium acetobutylicum*—Cellular behavior in adaptation to n-butanol. J Biotechnol 161(3):366–77
7. Jarboe LR, Grabar TB, Yomano LP, Shanmugan KT, Ingram LO (2007) Development of ethanologenic bacteria. Adv Biochem Eng Biotechnol 108:237–261
8. Li C, Wang Q, Zhao ZK (2008) Acid in ionic liquid: an efficient system for hydrolysis of lignocellulose. Green Chemistry 10:177–182
9. Jorgensen H, Kristensen JB, Felby C (2007) Enzymatic conversion of lignocellulose into fermentable sugars: challenges and opportunities. Biofuels Bioproducts & Biorefining-Biofpr 1:119–134
10. Zaldivar J, Ingram LO (1999) Effect of organic acids on the growth and fermentation of ethanologenic *Escherichia coli* LY01. Biotechnol Bioeng 66(4):203–210
11. Zaldivar J, Martinez A, Ingram LO (1999) Effect of selected aldehydes on the growth and fermentation of ethanologenic *Escherichia coli*. Biotechnol Bioeng 65(1):24–33
12. Miller EN, Jarboe LR, Turner PC, Pharkya P, Yomano LP, York SW, Ingram LO (2009) Furfural inhibits growth by limiting sulfur assimilation in ethanologenic *Escherichia coli* strain LY180. Appl Environ Microbiol 75(19): 6132–6141
13. Langmead B, Trapnell C, Pop M, Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10(3):R25
14. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25(14):1754–1760
15. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Durbin R (2009) The sequence alignment/Map format and SAMtools. Bioinformatics 25(16):2078–2079
16. Metzker ML (2010) Sequencing technologies—The next generation. Nat Rev Genet 11(1):31–46
17. Gordon D, Abajian C, Green P (1998) Consed: a graphical tool for sequence finishing. Genome Res 8(3):195–202
18. Miller EN, Jarboe LR, Yomano LP, York SW, Shanmugam KT, Ingram LO (2009) Silencing of NADPH-dependent oxidoreductase genes (yqhD and dkgA) in furfural-resistant ethanologenic *Escherichia coli*. Appl Environ Microbiol 75(13):4315–4323
19. Jarboe LR (2011) YqhD: a broad-substrate range aldehyde reductase with various applications in production of biorenewable fuels and

chemicals. Appl Microbiol Biotechnol 89 (2):249–257

20. Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, Taylor J (2010) Galaxy: a web-based genome analysis tool for experimentalists. Curr Protoc Mol Biol Chapter 19(Unit 19.10):1–21
21. Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Nekrutenko A (2005) Galaxy: a platform for interactive large-scale genome analysis. Genome Res 15(10):1451–1455
22. Goecks J, Nekrutenko A, Taylor J (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 11(8):R86
23. Andrews S (2012) FASTQC: a quality control tool for high throughput sequence data. Computer program distributed by the author, website http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/
24. Gordon A (2012) FASTX-Toolkit. Computer program distributed by the author, retrieved from http://hannonlab.cshl.edu/fastx_toolkit/index.html
25. Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJ, Birol I (2009) ABySS: a parallel assembler for short read sequence data. Genome Res 19(6):1117–1123
26. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res 18(5):821–829
27. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL (2009) BLAST+: architecture and applications. BMC Bioinformatics 10:421
28. Rozen S, Skaletsky H (2000) Primer3 on the WWW for general users and for biologist programmers. Methods Mol Biol 132:365–386
29. Ewing B, Green P (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Res 8(3):186–194
30. Ewing B, Hillier L, Wendl MC, Green P (1998) Base-calling of automated sequencer traces using phred. I. Accuracy assessment. Genome Res 8(3):175–185

Chapter 14

Discovery of Posttranscriptional Regulatory RNAs Using Next Generation Sequencing Technologies

Grant Gelderman and Lydia M. Contreras

Abstract

Next generation sequencing (NGS) has revolutionized the way by which we engineer metabolism by radically altering the path to genome-wide inquiries. This is due to the fact that NGS approaches offer several powerful advantages over traditional methods that include the ability to fully sequence hundreds to thousands of genes in a single experiment and simultaneously detect homozygous and heterozygous deletions, alterations in gene copy number, insertions, translocations, and exome-wide substitutions that include "hot-spot mutations." This chapter describes the use of these technologies as a sequencing technique for transcriptome analysis and discovery of regulatory RNA elements in the context of three main platforms: Illumina HiSeq, 454 pyrosequencing, and SOLiD sequencing. Specifically, this chapter focuses on the use of Illumina HiSeq, since it is the most widely used platform for RNA discovery and transcriptome analysis. Regulatory RNAs have now been found in all branches of life. In bacteria, noncoding small RNAs (sRNAs) are involved in highly sophisticated regulatory circuits that include quorum sensing, carbon metabolism, stress responses, and virulence (Gorke and Vogel, Gene Dev 22: 2914–2925, 2008; Gottesman, Trends Genet 21:399–404, 2005; Romby et al., Curr Opin Microbiol 9: 229–236, 2006). Further characterization of the underlying regulation of gene expression remains poorly understood given that it is estimated that over 60% of all predicted genes remain hypothetical and the 5′ and 3′ untranslated regions are unknown for more than 90% of the genes (Siegel et al., Trends Parasitol 27: 434–441, 2011). Importantly, manipulation of the posttranscriptional regulation that occurs at the level of RNA stability and export, trans-splicing, polyadenylation, protein translation, and protein stability via untranslated regions (Clayton, EMBO J 21:1881–1888, 2002; Haile and Papadopoulou, Curr Opin Microbiol 10:569–577, 2007) could be highly beneficial to metabolic engineering.

Key words: Next generation sequencing, Deep sequencing, Illumina, Small RNA discovery

1. Introduction

1.1. Overview of Next Generation Sequencing Technologies

Next generation platforms have completely shifted our paradigm for producing and organizing large-scale data of various types of genomes as well as for developing the downstream bioinformatics methods that are required for high data utility and proper interpretation. The major advantage of these methods is the ability to

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_14,

Table 1
Comparison of NGS methods

Platform	Method of sequencing	Detection method	Read length	Advantages	Disadvantages
Roche 454 Genome Sequencer	Pyrosequencing	Optical	230–400	Long read lengths and shorter run times than other platforms	Suspect to error accumulation and difficulty reading repetitive regions
Illumina/ Solexa HiSeq	Sequencing by synthesis	Fluorescent/ optical	2 × 150	High throughput	Limited ability to distinguish multiple samples
SOLiD	Sequencing by ligation	Fluorescent/ optical	25–35	Reduced error rate compared to HiSeq	Long run times

produce large volumes of data cheaply, beyond just determining the order of bases, as in traditional "First Generation Sequencing" approaches (7). As noted below, some applications are better suited for some technologies than others and careful thought is needed to determine a proper platform to use. Here, we briefly introduce the three most commonly used, commercially available Next Generation Sequencing Platforms: Illumina/Solexa, 454 pyrosequencing, and SOLiD (see Table 1, Comparison of NGS methods). The increasingly useful Ion Torrent sequencing technique, based on the detection of hydrogen ions released during each cycle of DNA polymerization via semiconductor approaches (8), is not discussed in this chapter. (As a historical note, First Generation Sequencing refers to the dideoxy chain-termination method, where four different colors were used to denote each of the four DNA bases based on the migration of different DNA fragments in automated capillary electrophoresis systems. This technique is also referred to as fluorescent Sanger sequencing (9, 10). This was the technique used to sequence the first 5,386 bp genome, bacteriophage phiX 174, in 1977).

This chapter outlines the preparation of high quality cDNA libraries and of experimental challenges associated with these protocols, in the context of the Illumina/Solexa sequencing platform. The preparation of cDNA libraries for next generation sequencing (NGS) is important given that recent data have shown that the library preparation method, more than the sequencing technology, can highly impact the diversity and abundance of the sRNAs that are reported by sequencing (29).

Illumina/Solexa HiSeq. This short read sequencing approach is incorporated into a fluidic flow cell design, an 8-channel sealed glass microfabricated device that is part of a so-called Cluster Station. The flow cell surface is populated with capture

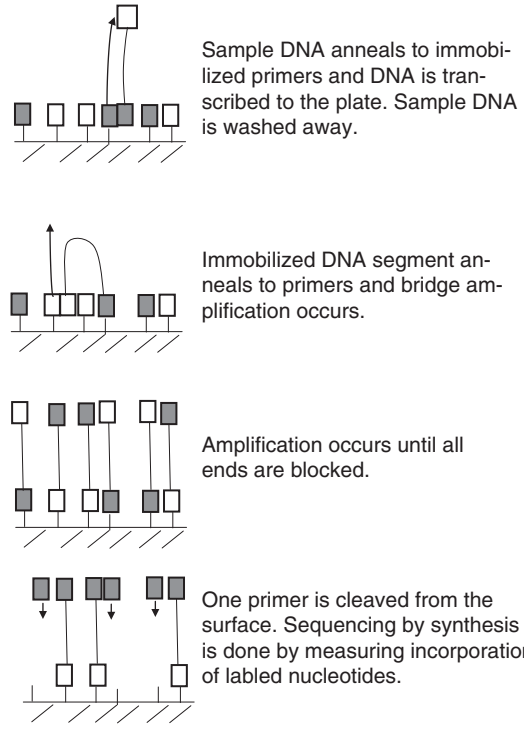

Fig. 1. Branch amplification of cDNA.

oligonucleotide anchors, which hybridize the appropriately modified DNA segments of a sequencing library generated from a genomic DNA sample. By a "branch (or bridge) amplification" process (see Fig. 1: Branch Amplification of cDNA), captured DNA are amplified in the flow cell by "arching" over and hybridizing to an adjacent anchor oligonucleotide primer (11). The actual sequencing is performed by hybridizing a primer complementary to the adapter sequence, and then cyclically adding DNA polymerase and a mixture of all four nucleotides; the nucleotides carry a base specific fluorescent label and have the 3′OH group chemically blocked so that every incorporation is a unique event. By this approach, unmodified DNA fragments and unincorporated nucleotides are washed away, while captured DNA fragments are extended one nucleotide at a time. After each nucleotide-coupling cycle, an imaging step takes place, where the flow cell is scanned and digital images are acquired to record the locations of fluorescently labeled nucleotide incorporations. Following imaging, the fluorescent dye and the terminal 3′blocker are chemically removed from the DNA before the next sequencing cycle. Laser excitation and total internal reflection optics are used by Illumina for high-sensitivity fluorescence detection. Although this technique is the most widely used NGS platform, it is limited by relatively low multiplexing (the ability to sequence different samples by use of a "barcode"

sequence that distinguishes them). This technology has been applied to gene discovery, whole exome analysis, and single nucleotide polymorphism detection (12). Given that the Illumina platform can generate more than 20 million raw short sequencing reads per sequencing lane, this technology is highly suitable for sequencing small RNA (sRNA) species (13).

454 pyrosequencing. This platform relies on highly parallel PCR reactions taking place in small emulsions. These oil–water emulsions are composed of a primer-coated bead with a single captured DNA template encased with the DNA polymerase, primers, and nucleotide triphosphates (NTPs) necessary for PCR. PCR amplification results in approximately one million copies of each DNA fragment on the surface of each bead so that each bead is associated with a single DNA fragment. The emulsions are broken and the DNA-coated beads are loaded onto an array of picoliter wells for the sequencing reaction (14, 15). Pyrosequencing is performed over the picoliter well array and the nucleotide additions are visualized and located by a fiber optic-coupled imaging camera. In pyrosequencing, the individual incorporation of one nucleotide by DNA polymerase results in the release of pyrophosphate, which initiates a series of downstream reactions that ultimately produce light by the firefly enzyme luciferase (11). The strength of this system is its ability to provide longer reads and relatively fast running times. However, this platform is limited by potential contamination problems associated with emulsion PCRs and with error rates in genetic regions that are rich in homopolymer repeats. This technology has been applied to targeted capture sequencing and whole genome mapping of plants and insects (15–17).

SOLiD (*Supported Oligonucleotide Ligation and Detection*) *sequencing*. This platform is based on an adapter-ligated fragment library (similar to the other approaches) and uses PCR amplification in an emulsion format with small magnetic beads (11, 17). However, unlike the other approaches, this technique uses DNA ligase and a unique method to sequence the amplified fragments based on primer hybridization to a special immobilized adapter sequence. Fluorescent signals are captured by a CCD camera imaging before enzymatic cleavage of the ligated probes and, after washing, the sequencing process is repeated. This approach has been used in applications very similar to the Illumina platform. An advantage of this approach is a reduction in sequencing error rates relative to the Illumina. However, this technique requires relative long run times and complex data analysis.

1.2. Applications of NGS

It is expected that additional advances in NGS technologies will continue to lower overall cost, increase the breadth of genome sequencing, speed up the turnaround time, and become applicable for the analysis of smaller samples (18). Some of the applications

of NGS include the following: (1) *Studies of gene expression by sequencing the transcriptome* (18). These approaches are more sensitive than traditional microarray approaches and do not depend on *a priori* knowledge of the genome. Methods for both mapping and quantifying transcriptomes are often referred to as RNA sequencing (RNA-seq) (19). (2) *Discovery of noncoding RNAs (ncRNAs) to elucidate novel aspects of posttranscriptional regulation*. Given the quantitative readouts of NGS, ncRNA studies can include detection of differential expression levels that correlate with environmental factors. As addressed in this chapter, one of the major challenges here has been the ability to convert the unique sequences of noncoding RNAs into next generation sequence libraries (20). (3) *Attempts to characterize fossil-derived DNAs from ancient genomes* (21). However, the co-isolation of bacterial DNA and the degraded nature of these genomes remains a challenge for NGS technologies. (4) *Characterization of the diversity found in bacterial infections, in the human biota, and in all species on Earth to enhance the field of metagenomics*. While traditionally these studies were addressed by characterization of 16S ribosomal RNA (rRNA) for taxa classification, DNA (or RNA) from a population can now be isolated, subcloned, and then deep-sequenced by Next Generation Sequencing platforms to obtain the full spectrum of taxa present. This is highly useful to determine the number and diversity of species and to analyze the breath of diversity. In this way it is possible to analyze various microorganisms including bacteria, viruses, fungi and parasites with a single protocol (11, 22) (5) *Analysis of other important aspects of cell biology using NGS technologies combined with other established biochemical techniques*. A powerful example is represented by the ChIP-chip approach (23) that has been developed to describe DNA–protein interactions through chromatin immunoprecipitation sequencing. Future applications of this technology are likely to include studies that uncover the relation between mRNA expression and protein-bound DNA populations from the same cells as well as studies of epigenomic variations to help understand regulatory patterns of nucleic acid methylation (7, 11, 24).

1.3. Challenges

1.3.1. Sample Preparation

While the unique combination of specific protocols differentiates one sequencing platform from the other, all rely on robust methods that produce a representative, high-quality, and non-biased source of nucleic acid material from the genome of interest. Additionally, amplified templates are required, since most methods are unable to detect single fluorescent events. Two common methods that are currently used in amplification are (1) clonally amplified templates (in a cell-free system) and (2) single-molecule templates (that do not require PCR). An additional consideration concerning library construction is deciding whether or not to prepare strand-specific libraries that have the advantage of yielding information about the orientation of transcripts (25, 26). This is valuable information,

particularly for transcriptome annotation of regions with overlapping transcription from opposite directions. However, in this chapter we only consider the preparation of cDNAs without strand information given the fact that strand-specific libraries are laborious to produce as they require many steps or inefficient RNA–RNA ligations. Although there are different challenges with these specific amplification methods regarding the required amounts of genomic DNA material, amplification bias of AT-rich and GC-rich templates, and the potential of introduced mutations, this chapter focuses on addressing in details the challenges associated with template preparation.

1.3.2. Data Analysis

As a technology that can produce higher than one billion sequences, NGS is highly dependent on sophisticated bioinformatic analysis programs and faces significant data management and interpretation challenges. In contrast to alternative microarray technologies, NGS can achieve base-pair-resolution and a much higher dynamic range of expression levels. However, since sequences obtained from the common platforms discussed in this chapter are relatively short (35–500 bp) (7), it is necessary to reconstruct the full-length transcripts by transcriptome assembly (18), except in the case of smaller classes of RNAs (e.g., micro RNAs, small interfering RNAs, small nucleolar RNAs, and PIWI-interacting RNAs). To date, the Optical Mapping System is the only system capable of strategically guiding, validating and completing the sequencing assembly of whole, complex genomes (27). In addition, systematic ways are needed to conduct image analysis, signal processing, background subtraction, base calling, and quality assessment to report the final sequence reads for each run (11, 18, 28). Aberrant nucleotide incorporation rates driven by polymerase errors (as in the case of Illumina systems) add to the importance of computational techniques to interpret sequencing results.

1.3.3. Sequencing and Imaging

Given that clonal amplification results in a population of identical templates, while single molecules are susceptible to multiple nucleotide (or probe) additions in a given cycle that can create sample heterogeneity, there are different challenges associated with the efficiency of the template synthesis extension and signal detection in these approaches. These are outside the scope of the focus of this chapter and have been reviewed in detail (7).

2. Materials

Although we have listed the suppliers that we typically use for our reagents, substitutions can be made from other available commercial sources.

2.1. General Materials and Reagents

1. Disposable RNase-free pipette tips, PCR tubes, siliconized 1.5 mL nonstick microfuge tubes.
2. RNaseZAP® (Ambion) for preparing RNase-free materials, diapers to cover work space while working with RNA (see Note 1).
3. Sodium chloride (NaCl): filtered 0.4 M solution immediately prior to use from a 5 M solution in ultrapure water.
4. 70% ethanol prepared with highest-grade ethanol and ultrapure water, stored at −20°C.
5. Thermocycler.
6. 100% high-grade ethanol.
7. Spectrophotometer and appropriate cuvettes.
8. 37°C incubator, 65°C incubator.
9. Blue-light transilluminator or 360 nm UV transilluminator.
10. FlashGel™ DNA Cassette (Lonza).
11. Vortexer.
12. Centrifuge.

2.2. Total RNA Preparation

1. Sterile 3 M sodium acetate (NaOAc) made with ultrapure water and adjusted to pH 5.2 with acetic acid.
2. Isopropanol, stored at room temperature.
3. 20% w/v SDS solution, prepared with filter-sterilized ultrapure water.
4. 5:1 acid phenol: chloroform pH 4.5 (Ambion), stored at 4°C.
5. RNase-free treated water (Ambion), stored at room temperature.
6. TRIzol® reagent (Invitrogen).
7. Zirconia acid washed glass beads (Sigma).
8. Siliconized screw cap tubes (VWR).
9. BeadBeater (Biospec).
10. Chloroform–isoamyl alcohol mix (24:1).
11. Isopropanol.
12. 95% ethanol.
13. 0.8 M sodium citrate + 1.2 M sodium chloride.
14. 2 mL siliconized microcentrifuge tubes.

2.3. Polyacrylamide Gel Casting, Electrophoresis, and Visualization

1. Mini-PROTEAN® Tetra Cell electrophoresis system (Bio-Rad).
2. 1× TBE for electrophoresis (For a 1 L 10× TBE stock solution: 55 g boric acid, 108 g Tris base, 40 mL 0.5 M EDTA pH 8.0, 960 mL ultrapure/deionized water, stored at room temperature).

3. Polyacrylamide gels: 40% w/v acrylamide (19:1 acrylamide–bisacrylamide (Bio-Rad)), urea, *N,N,N,N*–tetramethylethylenediamine (TEMED), 10% w/v ammonium persulfate solutions in ultrapure water (can be used within 1 month of solution preparation, store at 4°C) (see Note 2).
4. PCR 20 bp low ladder (SIGMA) mixed with standard DNA loading buffer (~0.5 μg per lane), used to estimate RNA size.
5. 100 bp Quick load DNA ladder (NEB) mixed with standard DNA loading buffer (~0.5 μg per lane).
6. Gel Loading Buffer II (Ambion), stored at −20°C. Thaw on ice prior to use.
7. SYBR® Gold (Invitrogen), stored wrapped in foil at −20°C (dye is light sensitive). Prior to use, completely thaw at room temperature. Ethidium bromide staining could be used as an alternative visualization strategy (see Note 3).
8. Razor blades.
9. Syringe with 18-gage needle.
10. RNase-free tray for RNA visualization.

2.4. RNA Gel Extraction and Precipitation

1. 21-gage needle.
2. 0.5 mL RNase-free microfuge tubes, nonstick (e.g., siliconized) round-bottom 2.0 mL RNase-free microfuge tubes.
3. GlycoBlue™ (Applied Biosystems), for clear detection of precipitate RNA as blue pellets.
4. 0.3 M sodium chloride (NaCl) solution.
5. Spin-X Cellulose acetate filter (2 mL, 0.45 μm, Sigma).
6. Isopropanol (HPLC grade).
7. Ethanol (HPLC grade).

2.5. Ligation of 5′ and 3′ Adaptor Linkers (Specific for Illumina/ Solexa HiSeq)

1. 5′ end adaptor RNA oligonucleotides. It is possible to attach a specific stretch of four or five "barcode" nucleotides to the 3′end of adaptor sequence. This approach makes it feasible to sort two or more samples that are loaded onto one lane of the Illumina/Solexa flow cell. It is important to note that all the primers and adaptors used in this protocol are specific to the Illumina/Solexa sequencing platform.
2. *5′end adaptor sequence*:
   5′-5AmMC6 GUU CAG AGU UCU ACA GUC CGA CGA UC-3′, where 5AmMC6 denotes a 5′Amino Modifier C6.
3. *3′end adaptor sequence*:
   5′-5Phos UCG UAU GCC GUC UUC UGC UU 3AmMO-3′, where 5phos denotes 5′ phosphorylation and 3AmMO denotes a 3′ Amino Modifier.
4. RNasin (Promega).

5. T4 RNA ligase and 10× ligase buffer (Promega) (see Note 4).
6. Nuclease-free water.
7. Low range ssRNA ladder (NEB).
8. RNA gel loading Buffer II (Ambion).
9. RNase free tray in gel staining.

2.6. Depletion of tRNAs and rRNAs

1. Single-stranded ~30 nucleotide long oligonucleotides (kept in 1 mM Tris–HCl, pH 8.0, stored at −20°C). Oligonucleotides could be designed by complementarity to the almost identical 3′ends of tRNA sequences or by designed mismatches that allow binding to multiple tRNA sequences (see Note 5).
2. A 100 μM solution of all Depletion Oligonucleotides mixed at a 1:1 ratio stored at −20°C (see Note 6).
3. Depletion Buffer: 100 mM Tris–HCl pH 7.8, 600 mM KCl, 20 mM $MgCl_2$, 20 mM DTT.
4. RNase H.

2.7. Reverse Transcription

1. 12.5 mM dNTPs mixed at a 1:1:1:1 ratio, stored at −20°C. Solution is made by diluting 100 mM dNTPs (NEB) with ultrapure water.
2. RT/REV primer (*Primer 1*: 5′-CAA GCA GAA GAC GGC ATA CGA-3′) suspended in 1 mM Tris–HCl, pH 8.0, and stored at −20°C. Make a 100 μM solution. Working solutions of 10 μM are made fresh prior to use.
3. Superscript III Reverse Transcriptase (Invitrogen), 5× First-Strand Buffer and 0.1 M DTT as provided by supplier.
4. RNasin (Promega).

2.8. PCR Amplification and Final Library Preparation

1. *AmpliTaq* gold DNA polymerase with 10× Buffer (Applied Biosystems) or Phusion DNA polymerase kit with 5× Phusion high-fidelity buffer (New England Biolabs).
2. *Primer 1*: 5′-CAA GCA GAA GAC GGC ATA CGA-3′.
3. *Primer 2*: 5′-TAA TGA TAC GGC GAC CAC CGA CAG GTT CAG AGT TCT ACA GTC CG-3′.
4. 12.5 mM dNTPs mixed at a 1:1:1:1 ratio, stored at −20°C. Solution is made by diluting 100 mM dNTPs (NEB) with ultrapure water.
5. 6× DNA loading Dye: 0.03% bromophenol blue 0.03% xylene cyanol FF, 10 mM Tris–HCl pH 7.5, 15% Ficoll 400, and 50 mM EDTA pH 8.0.
6. *n*-Butanol.
7. QIAprep® 96 Turbo Miniprep Kit (Qiagen).
8. TOPO® TA Cloning Kit for Sequencing (Invitrogen).

9. Resuspension Buffer for final library: 10 mM Tris–HCl, pH 8.5 (same as elution buffer in many mini and medi prep kits which can be used as long as they are RNase free).
10. 3 M sodium acetate
11. Ethanol.
12. GlycoBlue™ (Ambion).

3. Methods

3.1. General Principles of Next Generation Sequencing Technologies

In stark contrast to genome-wide microarray analysis, NGS methods can determine the cDNA sequence in a high throughput, relatively inexpensive, and quantitative way. In general, current NGS methods involve random breaking of genomic DNA into smaller sizes (<1 kb). From these pieces fragmented templates or double stranded circular "mate-pair" templates are made by circularizing sheared DNA of a selected given size (to bring distal ends into close proximity) (7). These templates can then be attached or immobilized onto a solid surface, where billions of sequencing reactions can be performed simultaneously. Amplified templates are required because most imaging systems are unable to detect single fluorescent events. Two major ways for template amplification include clonally amplified templates (e.g., via emulsion PCR and solid-phase amplification (30, 31)) and single-molecule templates.

Clonally amplified templates. For clonally amplified templates via emulsion PCR, a cell-free system is used where the library fragmented pairs or mate-pair targets are created and ligated to adaptors that contain universal priming sites at their ends; this allows the amplification of both unknown and known genomes with common PCR primers. After ligation, DNA is separated into single strands and captured onto beads, whereby one DNA molecule is attached per bead. Following the successful amplification of the beads, millions of DNA molecules can be immobilized. This immobilization can take place by a variety of strategies, depending on the sequencing platforms, that include: (1) the involvement of a polyacrylamide gel on microscope slides, (2) deposition on individual plate wells, or (3) chemical cross-linking to amino-coated glass surfaces. In the case of solid-phase amplification (e.g., in Illumina/Solexa), randomly distributed, clonally amplified DNA clusters from either fragment or mate-pair templates bound to an adapter sequence are immobilized onto a glass slide via universal sequencing primers that can initiate the sequencing reaction via the so-called "bridge-amplification" (or two dimensional PCR) technique (Fig. 1: Branch amplification of cDNA). Briefly, the DNA target is anchored to the glass slide via hybridization to a forward

primer containing a complementary sequence to the adapter on the DNA molecule. Following PCR elongation, the newly synthesized molecule extends (still anchored to the glass surface while linked to the forward primer) and bends to hybridize to a second (reverse) PCR primer that is also anchored on the glass slide, forming a "bridge"; the original template DNA molecules washes away. After repeated cycles of PCR extensions and denaturing of the two DNA strands, strand duplication continues until the process is terminated by blockage of all the 3′ends. A drawback that is associated with clonal amplification is that that some of the protocols are not straight forward to implement and require high levels of genomic DNA material (3–20 μg) (7). An important consideration in the preparation of templates is the type of adapter linker that will be ligated to the DNA template to mediate attachment or immobilization; this is highly dependent on the type of platform being used.

Single-molecule templates. Three advantages of single-molecule template preparation are that some of the protocols are more straightforward to implement, less DNA material is required (<1 μg), and PCR is not required (which can lead to mutations in clonally amplified templates appearing as false sequence variants or to biases in amplification of AT-rich or GC-rich target sequences). Before sequencing reactions are carried out, single molecule templates are typically immobilized on solid supports via highly similar methods to those used for clonally amplified templates (discussed above).

As indicated by the protocols suggested in this chapter, several experimental modifications can be introduced to ease application of template preparation and to improve its efficiency for high yields and purity of the desired DNA libraries. Our own efforts have focused on the use of these technologies for the discovery of regulatory RNA elements in bacterial cells.

3.2. Experimental Overview of RNA-seq

As illustrated in Fig. 2: Flow Diagram of cDNA Preparation, the general preparation of a cDNA library for RNA-seq includes the following steps by which a population of total RNA (or fractionated RNA) is converted to a library of cDNA fragments with adaptors attached to one or both ends: (1) total RNA preparation, (2) polyacrylamide gel purification of sRNAs, (3) ligation of small RNA adaptor linkers, (4) depletion of tRNAs and rRNAs; this strategy depletes total RNA samples of highly abundant tRNAs and small subunit rRNAs to enrich the sample for noncoding sRNA transcripts, (5) reverse transcription, and (6) PCR amplification. At the end of this cycle, the recovered amplified DNA is submitted for sequencing and bioinformatics analysis.

Subheadings 3.3–3.11 present and evaluate the experimental protocols involved in Illumina sequencing.

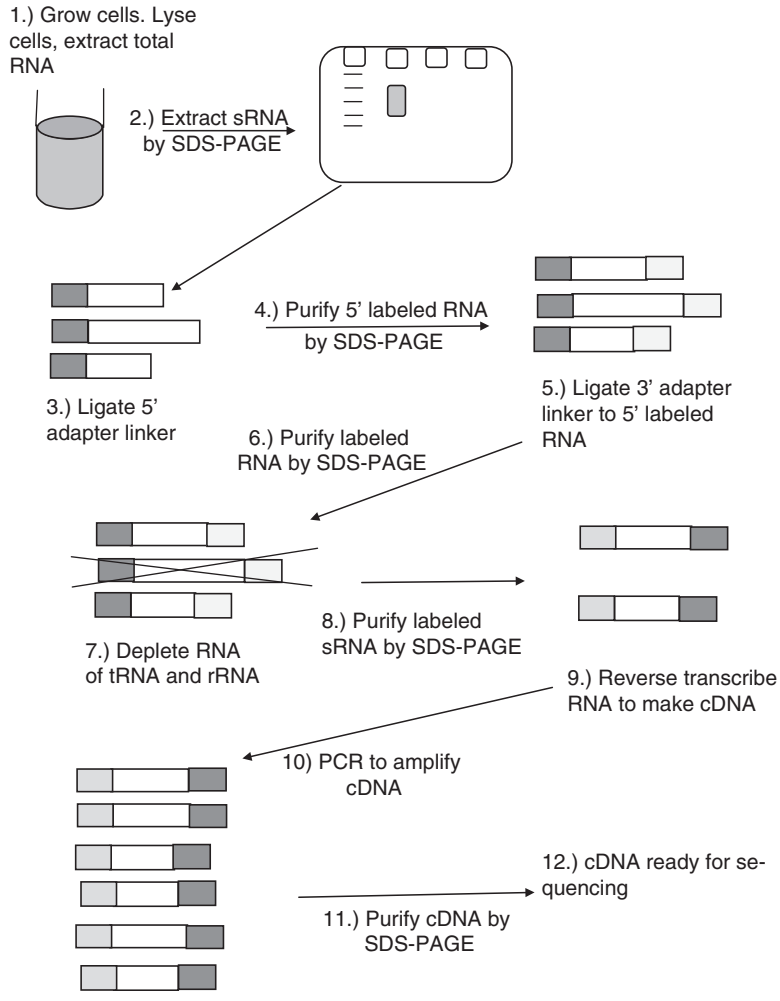

Fig. 2. Schematic of cDNA creation.

3.3. Total RNA Preparation

Given that the total amount of RNA varies at different growth stages etc., the amount of total RNA may need to be optimized for different applications. However, we recommend phenol–-chloroform-based RNA extraction procedures for preparing total RNA samples since they are known to provide the highest RNA yields. It is important to note that lithium-chloride precipitated RNA samples are not optimal for sRNA isolation, since sRNAs do not co-precipitate with longer RNA transcripts (see Notes 7–9).

1. Harvest cells in a centrifuge tube at the desired growth phase. Collect samples of ~4–5 mL per each individual centrifuge tube. We typically grow and spin ~50–100 mL of cells and harvest them in about 10–20 centrifuge tubes that we use for RNA preparation. At this point, cells could be frozen at −80°C. To minimize contamination, we recommend preparing RNA samples from different cells on separate batches.

2. Thaw out the pellets on ice and resuspend each pellet in 1 mL TRIzol® at room temperature (RT) by gently pipetting up and down. The TRIzol® can be taken out of the refrigerator in advance to ensure that it is at room temperature at the time of use.
3. Transfer the solution to a screw cap tube and add Zirconia beads to fill about 1/4 of the tube.
4. Incubate the solution with the beads at RT for 5 min.
5. Lyse cells in Beadbeater for 100 s (set the time setting at 10) and immediately place on ice to cool for 5–10 min. Repeat the process a second time to complete two full cycles. When processing multiple cycles, lyse all samples in the Beadbeater once and then place on ice. Continue onto the second round after all samples are lysed once and placed on ice (lyse the samples in the same order both times so that they get roughly the same amount of time on ice).
6. Spin in a precooled table centrifuge at maximum speed (14,000 rpm) for 2 min at 4°C. Transfer each supernatant to a 2 mL siliconized tube.
7. To extract the RNA phase, add 300 μL chloroform–isoamyl alcohol mix (24:1) (CIA) to separate cell lysate. Due to the toxicity of chloroform, this addition step should be done under the hood. Mix by vortexing for 15 s.
8. Incubate at RT for 3 min.
9. Spin in a precooled table centrifuge at maximum speed (14,000 rpm) for 10 min at 4°C. Transfer the supernatant to individual 2 mL siliconized tube.
10. Separate (nearly all) the top aqueous phase into a fresh siliconized tube. Be careful not to get any phenol/chloroform (bottom phase) in the sample. When separating the aqueous RNA layer from the organic layer, ensure that you do not accidentally carry over the white precipitate by leaving some of the aqueous layer behind.
11. Add 270 μL isopropanol and 270 μL of a solution of 0.8 M sodium citrate and 1.2 M NaCl. DO NOT vortex or flip. Salt will precipitate the RNA. The protein fraction will precipitate out in phenol/chloroform phase.
12. Chill on ice for 10 min.
13. Spin at 14,000 rpm for 15 min at 4°C and carefully pour out the supernatant.
14. To wash pellet, add 1 mL of cold 95% EtOH on top of the pellet. NEVER resuspend the pellet. (sRNAs dissolve in 70% EtOH).

15. Spin at 14,000 rpm for 5 min at 4°C; discard supernatant carefully by tilting the tube carefully. Be careful with pellet, since the pellet can be loose.
16. Air-dry pellet for 5 min at RT to remove the remaining EtOH.
17. Resuspend in 30 μL of water. To resuspend, add water and let it stand for 5 min; pipette it up and down carefully.
18. Make a 1:10 dilution (to make a more accurate reading) and quantitate the RNA in a spectrophotometer at OD_{260}. Pure RNA has an OD_{260}/OD_{280} ratio around 2.0 (a ratio of 1.8–2.0 is acceptable) (see Note 10). Our final preps range from concentrations of 5–10 μg per prep. However, this is highly dependent on the types of cells and might need to be optimized for the specific cell sample.
19. Check the quality of the RNA by running ~1 μg on a 10% polyacrylamide in TBE gel (could be a native gel) and visualize (as described in Subheading 3.4) by staining with 0.25 μg/mL ethidium bromide solution in 1× TBE. This gel should show three distinct bands for the 23S, 16S, and 5S rRNA and tRNAS (which will co-migrate).
20. Store the RNA prep in 500 μg aliquots at −80°C until ready to use for RNA-seq cDNA library preparation.

3.4. Polyacrylamide Gel Isolation of sRNAs from Total RNAs

This protocol is intended to enrich for small (non-tRNA) transcripts 80–200 nt and less than 80 nt long. Note that bacterial sRNAs are in general larger in size than microRNAs. Starting material for this protocol is approximately a total of 500 μg total RNA.

1. Prior to gel preparation, clean all Mini-PROTEAN® Tetra Cell plates, spacers (0.75 mm), and the comb with RNaseZAP. Clean again with 95% ethanol. Make sure to dry everything well with lint free paper towels before pouring gel.
2. For each RNA sample, prepare a denaturing 10% acrylamide/bis mini gel in a Mini-PROTEAN® Tetra Cell electrophoresis system. For one mini gel preparation (~1 mm thick): dissolve 2.1 g urea in RNase-free water to a volume of 3.2 mL, add 1.25 mL 40% acrylamide/bis, 500 μL 10× TBE, 50 μL 10% ammonium persulfate (APS), and 2 μL TEMED. Immediately pour the gel and add the comb to avoid premature solidification. Binder clips can be used to hold the comb against the larger plate to create better defined wells. The gel mixture can be stored without APS and TEMED at 4°C for several weeks, and a stock solution can be made. The SequaGel system (National Diagnostics) could also be used as an alternative to make the 10% gel, using the instructions provided by the manufacturer. As another alternative, commercially available

gels may be used. (We suggest 10 wells, 0.75 mm or 1 mm width). To save time, gels can be poured earlier (see Note 11).

3. Let the gel polymerize for ~30 min or until it completely solidifies.
4. Pre-run the gel in 1× TBE buffer for 30 min at 200 V. Prior to pre-running the gel, rinse the wells carefully with 1× TBE buffer to remove excess urea and extra pieces of acrylamide. A needle and 10 mL syringe could be used for this step to squirt buffer in and out of the wells. Also, ensure that there are no air bubbles at the bottom of the gel; air bubbles can be removed by adding 1× TBE buffer with a bent needle across the edge of the gel between the glass plates (see Note 12).
5. To prepare loading sample: mix 1 μg of total RNA with 10 μL of 2× Gel Loading Buffer II. Heat the sample to 65°C for 5 min, then immediately place on ice to cool. Quick spin to collect the sample at the bottom of the tube (see Note 13).
6. To prepare marker: load 10 μL of the denatured oligo ladder (PCR 20 bp low ladder (Sigma)) (see Note 14).
7. Load and run the gel at 100 V until the bromophenol blue reaches the bottom of the gel (~2 h).
8. Disassemble the gel apparatus. A razor can be used to crack open the gel cassette.
9. Stain each gel separately for about 15–30 min in the dark at room temperature using ~40 mL of SYBR® Gold staining solution (1× TBE with ~5 μL of SYBR® Gold). We typically place a tray containing SYBR® Gold and our gel inside an empty incubator that we can close as a way to avoid light. Nucleic acids can then be visualized with the following filter set: Ex 465 nm/Em 535 nm. A 0.25 μg/mL ethidium bromide solution in 1× TBE can be used (as a less efficient alternative). Visualize the RNA by using a blue-light transilluminator to minimize potential DNA and RNA damage (see Note 15).
10. Use a clean (RNase-free) razor blade to cut out the desired gel slice from the sample lane that corresponds to the desired size range. Cut out the RNA between 80 and 200 nt and the RNA below 80 nt. Use the razor or a small spatula to transfer the gel slice to an RNase-free microfuge tube. Use a clean blade for each sample. Make sure to avoid taking tRNAs and rRNAs from the gel; in bacteria, we typically avoid the ~80 nt range since this is where we typically see tRNAs.
11. Proceed to RNA extraction and ethanol precipitation (as described in Subheading 3.5).
12. Dissolve the pellet in a final volume of 5.7 μL RNase-free water. The purified sRNA fraction can be stored at −80°C.

3.5. RNA Gel Extraction and Precipitation

1. Puncture the bottom of a 0.5 mL microfuge tube with a 21-gage needle three or four times. Place the 0.5 mL punctured microfuge tube into a 2.0 mL RNase-free round-bottom non-stick (e.g., siliconized) microfuge tube and spin down in a tabletop centrifuge for 2 min at maximum speed. The gel piece will now be shredded in the 2.0 mL tube. If there are remaining pieces of the gel left in the 0.5 mL tube, puncture the tube again and spin down a second time. If there is only one small piece, use a pipette tip to transfer it to the 1.5 mL tube.
2. Remove and discard the empty 0.5 mL tube. Add 500 μL of sterile-filtered 0.3 M NaCl to the 2.0 mL tube containing the shredded gel to extract the RNA from the gel. Rotate overnight at 4°C.
3. The next day, transfer both the eluate and the gel debris into a Spin-X Cellulose acetate filter (2 mL, 0.45 μm, Sigma) using a 1 mL pipette tip (cut the tip off with a razor to increase the size of the opening prior to pipetting). Centrifuge at a tabletop centrifuge at maximum speed (~14,000 rpm) for 2 min.
4. Add another 100 μL 0.3 M NaCl on top of the gel pieces to wash the gel debris once more and centrifuge again at maximum speed (~14,000 rpm) for 2 min.
5. Collect the 600 μL eluate in the same 2.0 mL tube. Remove and discard the Spin-X column containing the gel debris.
6. Add to the flow through an equal volume (600 μL) of 100% isopropanol and 3 μL of GlycoBlue™ to precipitate the purified sRNA fraction present within the eluate. Incubate at −80°C for 30 min or at −20°C for at least 1 h. This precipitation step can also be done overnight at −20°C.
7. Spin down at 4°C in a tabletop centrifuge for 30 min at maximum speed (~14,000 rpm). At this point, a clear blue pellet should be visible. Remove the supernatant by carefully tilting the tube sideways (as the RNA pellet can be relatively soft and can be easily eluted).
8. Add 750 μL 70% ethanol (room temperature) to the tube on top of the pellet to wash it (do not dissolve the pellet). Centrifuge again at maximum speed for 15 min. Carefully discard the supernatant and allow the pellet to air-dry.
9. Dissolve the pellet in the desired volume of RNase-free water. The purified sRNA fraction can now be stored at −80°C.

3.6. 5′ Ligation of RNA Adaptor Linkers

The ligation of adaptors is the most important step in the preparation of the cDNA libraries. Although the protocol presented here uses T4 RNA ligase 1, there have been reports ((32), reviewed) that indicate the use of T4 RNA ligase 1 is sensitive to the nucleotide

base composition, the ligation site, and potential sRNA modifications (since not all sRNA can act as donor substrates for the enzyme). The use of T4 RNA ligase 2 has been suggested as a more efficient strategy for the ligation of several specific 3′-adapter.

1. Set up a 10% denaturing acrylamide/bis gel (see Subheading 3.4).
2. Heat the purified RNA (as isolated in Subheading 3.5) for 30 s at 90°C to completely denature the RNA and snap-cool on ice. Quick spin to collect contents at the bottom of the microfuge tube.
3. Set up the following reaction in a 1.5 mL RNase-free siliconized microfuge tube.

10× Ligase Buffer	1.0 μL
5′ RNA adaptor nucleotide (10 μM)	1.3 μL
purified RNA (from Subheading 3.5)	5.7 μL
T4 RNA ligase (10 U/μL)	1.0 μL
RNasin (40 U/μL)	1.0 μL

   Flick the tube, quick spin, and incubate at 37°C for 1 h. (The ligation can also take place at 20°C for 6 h followed by 4°C incubation overnight). As a ligation control, set up an additional reaction without the RNA by substituting it with 5.7 μL RNase-free water. The reaction can be carried out in a thermocycler if the reaction is set in a 0.2 mL thermocycler tube.
4. To stop the ligation reaction, add 10 μL of 2× Gel Loading Buffer II (see Note 13). As an optional step, incubate at 65°C for 5 min and snap cool on ice; skip this step if you will immediately proceed to run the 10% polyacrylamide gel.
5. Pre-run the 10% gel for 30 min at 100 V. After pre-running and before loading, wash the wells carefully with 1× TBE using a syringe and make sure to remove any bubbles at the bottom of the gel (see Subheading 3.4).
6. To prepare the ladder, mix 0.5 μL low-range ladder with 9.5 μL RNase-free water and 10 μL RNA loading buffer. Incubate both the sample and the ladder at 65°C for 5 min, then snap-cool on ice. Centrifuge to collect the sample to the bottom of the tube.
7. Load the 10 μL ladder followed by two empty lanes and then the 5′ adapter-ligated sample (see Note 14).
8. Run the gel at 100 V until the xylene cyanol dye (the lighter blue dye) reaches the bottom of the gel. This corresponds to approximately the 55 nt size marker on a 10% gel.

9. Disassemble the gel apparatus and stain the gel in an RNase-free tray with 45 mL 1× TBE and 4.5 μL of SYBR® Gold. Stain for 15–30 min in the dark. Alternatively, the gel can be stained with 45 mL of 1× TBE containing 0.25 μg/mL ethidium bromide for 5 min (see Subheading 3.4 for additional suggestions).
10. Visualize the RNA and the ladder on the gel using a blue-light transilluminator (see Note 15).
11. Using a clean razor blade, excise a gel slide in the sample lane that corresponds to 96–220 nt. For Illumina/Solexa sequencing, the adaptor that is now annealed to the sRNA fraction is 26 nt long, shifting the size of the sRNA fraction on the gel. This gel slice should contain 5′ adapter ligated samples. Make sure not to cut below 40 nt to avoid contamination from the 5′adaptor oligonucleotide. These size estimates for excision are based on specific Illumina adapters. When using other NGS technologies, the predicted size after ligation (after addition of the required adapter) needs to be calculated for modification of the gel excision parameters and choice of the oligo ladder.
12. Place the gel pieces in a punctured 0.5 mL tube and proceed to RNA gel extraction and precipitation (see Subheading 3.5).
13. Dissolve the final air-dried pellet in 6.4 μL of RNase-free water.

3.7. 3′ Ligation of RNA Adaptor Linkers

1. Set up a 10% denaturing acrylamide/bis gel (see Subheading 3.4).
2. Heat the purified 5′adaptor ligated RNA (as prepared from Subheading 3.6) for 30 s at 90°C to completely denature the RNA and snap-cool on ice. Quick spin to collect contents at the bottom of the microfuge tube.
3. Set up the following reaction in a 1.5 mL RNase-free siliconized microfuge tube:

10× Ligation Buffer	1.0 μL
3′ RNA adaptor nucleotide (10 μM)	0.6 μL
purified 5′ adaptor ligated RNA (from Subheading 3.6)	6.4 μL
T4 RNA ligase (10 U/μL)	1.0 μL
RNasin (40 U/μL)	1.0 μL

Flick the tube, quick spin, and incubate at 37°C for 1 h (The ligation can also take place at 20° for 6 h followed by 4° incubation overnight). As a ligation control, set up an additional reaction without the RNA by substituting it with 6.4 μL

RNase-free water. The reaction can be carried out in a thermocycler if the reaction is set in a 0.2 mL thermocycler tube.

4. To end the ligation reaction, add 10 μL of 2 × Gel Loading Buffer II (see Note 13). As an optional step, incubate at 65°C for 5 min and snap cool on ice; skip this step if you will immediately proceed to run the 10% polyacrylamide gel.
5. Pre-run the 10% gel for 30 min at 100 V. After pre-running and before loading, wash the wells carefully with 1× TBE using a syringe and make sure to remove any bubbles at the bottom of the gel (see Subheading 3.4).
6. To prepare the ladder, mix 0.5 μL low-range ladder with 9.5 μL RNase-free water and 10 μL RNA loading buffer. Incubate both the sample and the ladder at 65°C for 5 min, then snap-cool on ice. Centrifuge to collect the sample to the bottom of the tube.
7. Load the 10 μL ladder and the 3′ adapter-ligated sample (see Note 14).
8. Run the gel at 100 V until the xylene cyanol dye (the lighter blue dye) reaches the bottom of the gel. This corresponds to approximately the 55 nt size marker on a 10% gel.
9. Disassemble the gel apparatus and stain the gel in an RNase-free tray with 45 mL 1× TBE and 4.5 μL of SYBR® Gold. Stain for 15–30 min in the dark. Alternatively, gel can be stained with 45 mL of 1× TBE containing 0.25 μg/mL ethidium bromide for 5 min (see Subheading 3.4 for additional suggestions).
10. Visualize the RNA and the ladder on the gel using a blue-light transilluminator (see Note 15).
11. Using a clean razor blade, excise a gel slide in the sample lane that corresponds to 117–241 nt. For Illumina/Solexa sequencing, the 3′ adaptor that is now annealed to the sRNA fraction is 21 nt long, further shifting the size of the 5′ adapter-ligated RNA fraction on the gel. This gel slice should contain 5′ and 3′ adapter ligated samples. Make sure not to cut below 60 nt to avoid contamination with the 3′adaptor oligonucleotide. These size estimates for excision are based on specific Illumina adapters. When using other NGS technologies, the predicted size after ligation (after addition of the required adapter) needs to be calculated for modification of the gel excision parameters and choice of the oligo ladder.
12. Place the gel pieces in a punctured 0.5 mL tube and proceed to RNA gel extraction and precipitation, as described below. Note that some portions of the protocol described in Subheading 3.5 have been modified to include the addition of 1.5 nmol of the

Oligonucleotide mix containing the multiple oligonucleotides in the ethanol precipitation step.

13. Puncture the bottom of a 0.5 mL microfuge tube with a 21-gage needle three-four times. Place the 0.5 mL punctured microfuge tube into a 2.0 mL RNase-free round-bottom non-stick (e.g., siliconized) microfuge tube and spin down in a tabletop centrifuge for 2 min at maximum speed. The gel piece will now be shredded in the 2.0 mL tube. If there are remaining pieces of the gel left in the 0.5 mL tube, puncture the tube again and spin down a second time. If there is only one small piece, use a pipette tip to transfer it to the 1.5 mL tube.
14. Remove and discard the empty 0.5 mL tube. Add 500 μL of sterile-filtered 0.3 M NaCl to the 2.0 mL tube containing the shredded gel to extract the RNA from the gel. Rotate overnight at 4°C.
15. The next day, transfer both the eluate and the gel debris into a Spin-X Cellulose acetate filter (2 mL, 0.45 μm, Sigma) using a 1 mL pipette tip (cut the end of the tip with a razor to increase the size of the opening). Centrifuge at maximum speed (~14,000 rpm) for 2 min.
16. Add another 100 μL 0.3 M NaCl on top of the gel pieces to wash the gel debris once more and centrifuge again at maximum speed (~14,000 rpm) for 2 min.
17. Collect the 600 μL eluate in the same 2.0 mL tube. Remove and discard the Spin-X column containing the gel debris.
18. Add to the flow through 15 μL of the 100 mM Oligonucleotide Mix, an equal volume (600 μL) of 100% isopropanol and 3 μL of GlycoBlue™ to precipitate the purified sRNA fraction present within the eluate. Incubate at −80°C for 30 min or at −20°C for at least 1 h. This precipitation step can also be done overnight at -20°C.
19. Spin down at 4°C in a tabletop centrifuge for 30 min at maximum speed (~14,000 rpm). At this point, a clear blue pellet should be visible. Remove the supernatant by carefully titling the tube sideways (as the RNA pellet can be relatively soft and can be easily eluted).
20. Add 750 μL 70% ethanol (room temperature) to the tube on top of the pellet to wash it (do not dissolve the pellet). Centrifuge again at maximum speed for 15 min. Carefully discard the supernatant and allow the pellet to air-dry.
21. Dissolve the final air-dried pellet in 10 μL 1× Depletion Buffer to proceed with depletion of tRNAs in Subheading 3.8.

3.8. Depletion of tRNAs and rRNAs

To further remove any contaminating RNAs (particularly tRNAs and 5S rRNA) from the linkered-sRNA library, we use the

following protocol. Note that bacterial tRNAs (~80 nt) and 5S rRNAs (~120 nt) fall within the size range of bacterial sRNAs (~30–300 nt) and therefore cannot be easily separated during sRNA extraction and purification. Bioinformatics approaches represent an alternative method to do the removal of these RNAs, once NGS results are obtained. Since bacterial sRNAs are significantly larger than eukaryotic miRNAs, their direct isolation and amplification is hampered by abundant tRNA and rRNA sequences. Importantly, the common strategy used to deplete tRNA and rRNA from the eukaryotic transcriptome by various RNA-seq technologies, based on the cloning of polyadenylated (polyA), fails for bacterial sRNAs since these are not polyadenylated. The treatment of RNA that we present in this section has proven to be highly efficient in enriching the starting pool of RNA for regulatory sRNAs in bacterial cells (33).

1. Prepare a 10% polyacrylamide gel, (see Subheading 3.4).
2. Take 10 μL of the 5′ and 3′ linkered RNA pool that was dissolved in 1× Depletion Buffer with the Depletion Oligonucleotide Mix Solution (see Subheading 3.7) and transfer to a PCR tube (see Note 16).
3. Using a thermocycler, heat up the sample to 65°C for 5 min and then cool to 37°C slowly (0.1°C/s). When the sample reaches 37°C, add 0.5 μL RNase H to hydrolyze the targeted RNAs and incubate the reaction at 37°C for 30 min. Repeat this step once.
4. To end the hydrolysis reaction, add 10 μL of 2× RNA gel-loading Buffer II (see Note 13). As an optional step, incubate at 65°C for 5 min and snap cool on ice; skip this step if you will immediately proceed to run the 10% polyacrylamide gel.
5. Pre-run the 10% gel for 30 min at 100 V. After pre-running and before loading, wash the wells carefully with 1× TBE using a syringe and make sure to remove any bubbles at the bottom of the gel (see Subheading 3.4).
6. To prepare the ladder, mix 0.5 μL low-range ladder with 9.5 μL RNase-free water and 10 μL RNA loading buffer. Incubate both the sample and the ladder at 65°C for 5 min, then snap-cool on ice. Centrifuge to collect the sample to the bottom of the tube.
7. Load the 10 μL ladder followed by two empty lanes and then the RNA depleted sample (see Note 14).
8. Run the gel at 100 V until the xylene cyanol dye (the lighter blue dye) reaches the bottom of the gel. This corresponds to approximately the 55 nt size marker on a 10% gel.
9. Disassemble the gel apparatus and stain gel in an RNase-free tray with 45 mL 1× TBE and 4.5 μL of SYBR® Gold. Stain for

15–30 min in the dark. Alternatively, gel can be stained with 45 mL of 1× TBE containing 0.25 μg/mL ethidium bromide for 5 min (see Subheading 3.4 for additional suggestions).

10. Visualize the RNA and the ladder on the gel using a blue-light transilluminator (see Note 15).
11. Using a clean razor blade, excise a gel slide in the sample lane that corresponds to 117–241 nt. For Illumina/Solexa sequencing, the 3′ adaptor that is now annealed to the sRNA fraction is 21 nt long, further shifting the size of the 5′ adapter-ligated RNA fraction on the gel. This gel slice should contain 5′ and 3′ adapter ligated samples. Make sure not to cut below 60 nt to avoid contamination with the 3′adaptor oligonucleotide. These size estimates for excision are based on specific Illumina adapters. When using other NGS technologies, the predicted size after ligation (after addition of the required adapter) needs to be calculated for modification of the gel excision parameters and choice of the oligo ladder.
12. Place the gel pieces in a punctured 0.5 mL tube and proceed to RNA gel extraction and precipitation (see Subheading 3.5).
13. Dissolve the final air-dried pellet in 8 μL of RNase-free water. Do not use DEPC-treated water to avoid possible interference with the PCR step due to left over traces of ethanol (formed during the autoclaving of DEPC-treated water).

3.9. Reverse Transcription of sRNA-Enriched Fraction

This protocol reverse transcribes the adapter-ligated sRNAs to cDNA using an oligonucleotide that is antisense to the 3′end of the adapter to prime the reaction.

1. In a PCR tube add the following:

linkered sRNA-enriched RNA from Subheading 3.8	8.0 μL
RT/REV (Primer 1) (0.1 mM).	1.0 μL

2. Incubate at 70°C for 10 min to denature and quick spin to cool.
3. Add the following reagents in this order:

5× First-strand buffer	3.0 μL
dNTP mix (12.5 mM)	0.5 μL
DTT (100 mM)	1.0 μL
RNasin (40 U/μL)	0.5 μL

4. Incubate at 50°C for 3 min and then add 1.0 μL of Superscript III RT (200 U/μL). Flick the tube and quick spin.
5. Incubate at 50°C for 1 h.
6. The cDNA can be stored at −20°C.

3.10. PCR Amplification of the cDNA Library

The concentration of the amplified sRNA library (as well as the quality) is critical for NGS and depends on the platform utilized. Consult the center where sequencing will take place regarding the necessary concentrations.

1. Amplify the cloned sRNA cDNA library using the following recipe. This protocol uses 7 μL (~half) of the reverse transcription reaction product to allow for a second try in case of failure.

RT reaction from Subheading 3.9	7.0 μL
5× High-fidelity buffer	16.0 μL
dNTP's (12.5 mM)	1.5 μL
PCR Primer 1 (0.1 mM)	0.5 μL
PCR Primer 2 (0.1 mM)	0.5 μL
Phusion polymerase (NEB)	1.0 μL
Nuclease-free water	53.5 μL

2. Vortex to mix well. Quick spin and aliquot this 80 μL reaction into four separate PCR tubes (for four separate PCR reactions).
3. Place in the thermocycler to amplify with the following settings: 1 cycle (98°C for 30 s); 14 cycles (98°C for 10 s, 58°C for 30 s, 72°C for 30 s); 1 cycle (72°C for 7 min). Immediately place the reaction on ice (or set up the thermocycler to 4°C indefinitely).
4. To confirm the presence of DNA library, add 1 μL FlashGel™ loading dye to 4 μL of the PCR product and run on a FlashGel™ DNA cassette for 6 min at 200 V. Use the FlashGel™ DNA marker (50–1,500 bp) as a ladder.
5. Check for the presence of a band above the adapter-dimer band that could form (~70 bp). There should be a second higher library band. The number of cycles for the PCR reaction might need to be determined empirically by trying different cycle amounts for amplification. 22–25 cycles is the maximum recommended amount, since over-amplification of redundant sRNA species can lead to poor quality libraries.

6. Precipitate the PCR product by adding 10 μL 3 M sodium acetate, 250 μL 100% ethanol, and 1.5 μL GlycoBlue™. Store at −20°C for at least 30 min. Spin down in a tabletop centrifuge at maximum speed (14,000 rpm) for 30 min at 4°C. Remove the supernatant by carefully tilting the tube and add 750 μL cold 95% ethanol on top of the visibly blue pellet to wash it, taking care to retain the pellet.
7. Dissolve the final pellet in 20 μL nuclease free water.

3.11. Final Library Purification

1. Prepare a 10% acrylamide/bis gel and pre-run (see Subheading 3.4). Use 100 bp ladder as a marker.
2. Run the entire sample prepared in Subheading 3.10, following the same protocol described in Subheading 3.4.
3. Stain the gel, visualize, and excise the desired band using the same protocol described in Subheading 3.4.
4. Extract and precipitate the DNA use the same procedure described in Subheading 3.5, with the following modifications: (1) instead of using GlycoBlue™ in the precipitation step, use Pellet Paint NF Co-precipitant (Novagen) as GlycoBlue™ can interfere with the DNA concentrations that are measured during fluorescent readouts in NGS, and (2) resuspend the final pellet in 15 μL of 10 mM Tris-HCl, pH 8.5 (this the same buffer as the elution buffer in many mini- and medi- prep kits). (Thaw Pellet Paint at room temperature and vortex 3–5 min prior to use).
5. The quality of the library can be checked by subcloning (by TOPO® cloning) the library and checking ~100 clones to get a sense of the library composition. A QIAprep® 96 Turbo Miniprep kit is highly recommended for the preparation of these clones for sequencing. This sequencing could be highly useful to optimize the Oligo Mix that is used for tRNA depletion, as the oligonucleotide ratios and the types of oligonucleotides that are used can be adjusted based on the tested adequacy of depletion.

4. Notes

1. It is important to handle all RNA samples with extreme care throughout these protocols to avoid degradation and RNase contamination. A few things to keep in mind are (1) to handle mRNA with RNase-free tips and RNase-free equipment, (2) to wear gloves and change them often, (3) to always keep mRNA frozen or on ice when in use, (4) to clean all equipment (including the bench in use) with RNaseZAP® (Ambion), (5) to sterile-filter all solutions and to bake glassware at 210°C for

3–4 h, as opposed to autoclaving glassware and solutions, (6) to store RNA at −80°C at all times, (7) to use filter tips, nonstick (e.g., siliconized) RNase-free microfuge tubes, and (8) to continually apply RNaseZAP® (Ambion) to remove RNase contamination from bench, pipettes, and all other equipment being used.

2. Unpolymerized acrylamide and bisacrylamide are strong neurotoxins and should be handled with gloves at all times.
3. Ethidium bromide is a toxic mutagen and should be handled with the appropriate gloves. We prefer the use of SYBR® Gold since this is a more sensitive and less mutagenic dye than ethidium bromide and can reduce risks of nucleotide modifications during gel exposure.
4. The presence of possible ATP contaminating traces within the T4 ligase can result in unwanted RNA-RNA ligation or RNA circularization, since the ATP can charge the 5′end of phosphorylated RNA/DNA. To avoid this problem, the T4 RNA ligase is often diluted at a 1:10 or 1:5 dilutions with 1× Ligase Buffer.
5. The tRNA sequences for many organisms are tabulated in the Genomic tRNA database (http://gtrnadb.ucsc.edu/). Scripts can also be written to design oligonucleotide sets that are complementary to a given sequence set, incorporating mismatches to account for similarities in tRNAs and for natural mismatches. In our experience, two or three mismatches allow for one oligonucleotide sequence to base pair with several different tRNAs, reducing the number of oligonucleotide sequences that need to be synthesized and designed. It should also be noted that an alternative strategy could involve the design of oligonucleotides that are complementary to the 3′ends of tRNAs, since many tRNA 3′ends are highly similar. Reportedly, only ~25–30 nt oligonucleotides were designed to deplete all 98 tRNAs in the *V. cholerae* genome (33).
6. The amount and composition of the tRNA/rRNA Depletion Oligonucleotide Mix is something that might need to be tested empirically.
7. Best results are obtained when starting with high-quality RNA and DNA. Since the protocols are designed to amplify and transcribe small quantities of genomic material, quality is more important than quantity. However, if quality is suspected to be a problem, increasing the amount of starting material can be helpful.
8. The quality of the final library preparation is strongly tied to the quality of the starting RNA material. High levels of RNA degradation can end up in the sequencing library as false sRNA reads.

9. If the presence of RNase is suspected, treat the DNA with Proteinase K (100 μg/mL) and SDS (0.5%) in 50 mM Tris–HCl (pH 7.5), 5 mM $CaCl_2$ for 30 min at 37°C.
10. If possible, the use of a bioanalyzer will provide more accurate measurements of the sample and quality of the RNA (and DNA) samples relative to a UV spectrophotometer.
11. Polyacrylamide gels can be made the day before (to ensure complete polymerization) and, after 40 min of polymerization, stored at 4°C wrapped (with the comb intact) in a wet paper towel (with 1× TBE) and SARAN wrap.
12. The comb should be removed slowly by pulling straight upwards from both sides of the comb to avoid disturbing the wells.
13. As an alternative, Gel Loading Buffer II solution can be substituted with a gel-loading solution made of 10 mL of deionized formamide, 200 mL 0.5 M EDTA (pH 8.0), 1 mg xylene cyanol FF, and 1 mg bromophenol blue (buffer can be stored at 4°C for up to 1 year).
14. Loading equal volumes in the gel is important to ensure that samples run at a uniform pace, and not tilted in a "smile" shape through the gel. Every time that multiple samples are processed, it is advisable to run separate purification gels with their own ladder to avoid contamination. Alternatively, keep two lanes empty between each sample and the ladder.
15. To minimize contamination when visualizing the gel, the gel can be wrapped in SARAN wrap. Alternatively, gels can be visualized on a 360 nm UV transilluminator. A blue-light transilluminator may not be as sensitive as the UV transilluminator.
16. The concentration of Depletion Oligonucleotides used in the mix for this protocol might need to be optimized for individual cases. Here, we present only a potential starting point.

Acknowledgments

This work is supported by a Defense Threat Reducing Agency (DTRA) Young Investigator Award to LMC and by support from the Welch Foundation to LMC.

References

1. Gorke B, Vogel J (2008) Noncoding RNA control the making and breaking of sugars. Genes Dev 22:2914–2925
2. Gottesman S (2005) Micros for microbes: noncoding regulatory RNAs in bacteria. Trends Genet 21:399–404

3. Romby P, Vandenesch F, Wagner EGH (2006) The role of RNAs in the regulation of virulence-gene expression. Curr Opin Microbiol 9:229–236
4. Siegel G et al (2011) Gene expression in *Trypansoma brucei*: lessons from high throughput RNA sequencing. Trends Parasitol 27: 434–441
5. Clayton CE (2002) Life without transcriptional control? From fly to man and back again. EMBO J 21:1881–1888
6. Haile S, Papadopoulou B (2007) Developmental regulation of gene expression in trypanosomatid parasitic protozoa. Curr Opin Microbiol 10:569–577
7. Metzker ML (2010) Sequencing technologies—The next generation. Nat Rev Genet 11:31–46
8. Rusk N (2011) Torrents of sequence. Nat Methods 8:44
9. Sanger F, Nicklen S, Coulson AR (1977) DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci U S A 74(12): 5463–5467
10. Franca LT, Carrilho E, Kist TB (2002) A review of DNA sequencing techniques. Q Rev Biophys 35(2):169–200
11. Mardis ER (2008) Next-generation DNA sequencing methods. Annu Rev Genomics Hum Genet 9:387–402
12. Margulies E et al (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature 437:376–380
13. Kuchenbauer M et al (2008) In-depth characterization of the microRNA transcriptome in a leukemia progression model. Genome Res 18:1787–1797
14. Schweiger MR, Kerick M, Timmermann B et al (2011) The power of NGS technologies to delineate the genome organization in cancer: from mutations to structural variations and epigenetic alterations. Cancer Metastasis Rev 30:199–2010
15. Tawfik DS, Griffiths AD (1998) Man-made cell-like compartments for molecular evolution. Nat Biotechnol 16:652–656
16. Slamon DJ, Leyland-Jones B, Shak S et al (2001) Use of chemotherapy plus monoclonal antibody against HER2 for metastatic breast cancer that overexpresses HER2. N Engl J Med 344:783–792
17. Shendure J, Porreca GJ, Reppas NB et al (2005) Accurate multiplex polony sequencing of an evolved bacterial genome. Science 309:1728–1732
18. Martin JA, Wang Z (2011) Next-generation transcriptome assembly. Nat Rev Genet 12: 671–682
19. Wang Z, Gerstein M, Snyder M (2009) RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet 10:57–63
20. Lu C, Meyers BC, Green PJ (2007) Construction of small RNA cDNA libraries for deep sequencing. Methods 43:110–117
21. Noonan H et al (2006) Sequencing and analysis of Neanderthal genomic DNA. Science 314:1113–1118
22. Nakamura S, Nakaya T, Iida T (2011) Metagenomic analysis of bacterial infections by means of high-throughput DNA sequencing. Exp Biol Med 236:968–971
23. Ren R et al (2000) Genome-wide profiles of STAT1 of DNA binding proteins. Science 290: 2306–2309
24. Niedringhaus M et al (2011) Landscape of next-generation sequencing technologies. Anal Chem 83:4327–4341
25. Lister O'M et al (2008) Highly integrated single-base resolution maps of the epigenome in *Arabidopsis*. Cell 133:523–536
26. Cloonan F et al (2008) Stem cell transcriptome profiling via massive-scale mRNA sequencing. Nat Methods 5:613–619
27. Lin Q et al (1999) Whole-genome shotgun optical mapping of *Deinococcus radiodurans*. Science 285(5433):1558–1562
28. Li H et al (2009) The sequence alignment/map format and SAMtools. Bioinformatics 25: 2078–2079
29. Linsen SE, de Wit E et al (2009) Limitations and possibilities of small RNA digital gene expression profiling. Nat Methods 6:474–476
30. Dressman Y et al (2003) Transforming single DNA molecules into fluorescent magnetic particle for detection and enumeration of genetic variations. Proc Natl Acad Sci USA 100: 8817–8822
31. Feurco R et al (2006) BTA, a novel reagent for DNA attachment on glass and efficient generation of solid-phase amplified DNA colonies. Nucleic Acids Res 34:e22
32. McCormick KP, Willmann MR, Meyers BC (2011) Experimental design, preprocessing, normalization and differential expression analysis of small RNA sequencing experiments. Silence J 2:2
33. Liu L et al (2009) Experimental discovery of sRNAs in Vibrio cholera by direct cloning, 5S/tRNA depletion and parallel sequencing. Nucleic Acids Res 37:e46

Chapter 15

^{13}C-Based Metabolic Flux Analysis: Fundamentals and Practice

Tae Hoon Yang

Abstract

Isotope-based metabolic flux analysis is one of the emerging technologies applied to system level metabolic phenotype characterization in metabolic engineering. Among the developed approaches, ^{13}C-based metabolic flux analysis has been established as a standard tool and has been widely applied to quantitative pathway characterization of diverse biological systems. To implement ^{13}C-based metabolic flux analysis in practice, comprehending the underlying mathematical and computational modeling fundamentals is of importance along with carefully conducted experiments and analytical measurements. Such knowledge is also crucial when designing ^{13}C-labeling experiments and properly acquiring key data sets essential for in vivo flux analysis implementation.

In this regard, the modeling fundamentals of ^{13}C-labeling systems and analytical data processing are the main topics we will deal with in this chapter. Along with this, the relevant numerical optimization techniques are addressed to help implementation of the entire computational procedures aiming at ^{13}C-based metabolic flux analysis in vivo.

Key words: ^{13}C-based metabolic flux analysis, ^{13}C-labeling system, Mass isotopomer distribution analysis, Constraint-based numerical flux estimation, Systems biology

1. Introduction

Biological systems provide endless possibilities for producing useful materials on the basis of the metabolic engineering. Recently, the state-of-the-art techniques have become capable of not only overproducing biologically occurring substances but also nonnatural commodity chemicals from renewable feedstocks, e.g., the first bio-based 1,4-butanediol production through metabolic engineering (1). Herein, metabolic engineers seek to understand characteristics of metabolic phenotypes, involving diverse analytical and computational modeling techniques. Such techniques are applied to the systems level characterizations of biological components such as genes, proteins, metabolites, and enzyme reactions and

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_15,

their interactions in the metabolic network. Among those, the metabolic fluxes quantitating enzyme reaction rates in vivo are the end products of any conceivable interactions between biological components and, therefore, the key parameters in systems metabolic engineering.

Typically, a realistic metabolic network that comprises catabolic and anabolic reactions of biological pathways represents an underdetermined system from the stoichiometric viewpoint. Therefore, additional constraints are required to fully quantify the fluxes of the entire network. This additional information is typically gained from isotopic labeling measurements, and modeling of mathematical relations between metabolic fluxes and isotopic labeling patterns is the framework of isotope-based metabolic flux analysis. Hence, this approach necessitates mathematical modeling efforts as well as analytical measurement techniques including isotopic measurement data processing. In this concern, it is crucial to comprehend the relevant computational and analytical data processing procedures in order to implement in vivo flux analysis successfully, and these are the main topics addressed in the present chapter.

1.1. Mathematical Notations

A matrix is represented by a bold capital letter such as "**A**" and a vector by a bold lowercase letter such as "**a**". If not specified, the vectors are in a column format, e.g., $\mathbf{a} = (a_1, a_2, \ldots, a_n)^{\mathrm{T}}$, where "$a_i$" signifies each scalar entry, the superscript "T" the transpose of the row array $(a_1, a_2, \ldots, a_n)$, and "n" the vector dimension. Bold numbers such as "**0**" is the column vector or matrix with all zero entries, e.g., $\mathbf{0} = (0, 0, \ldots, 0)^{\mathrm{T}}$ and "**1**" the column vector whose entries are ones such as $\mathbf{1} = (1, 1, \ldots, 1)^{\mathrm{T}}$. The operator diag(**a**) and diag(**A**) denotes a diagonal matrix with the vector **a** on the main diagonal and the main diagonal of **A**, respectively. Hence, if $\mathbf{A} = \mathrm{diag}(\mathbf{a})$, then $\mathbf{a} = \mathrm{diag}(\mathbf{A})$. The operator dim(**a**) or dim(**A**) is the dimension of a vector or a matrix, e.g., $\dim(\mathbf{a}) = n \times 1$ or $\dim(\mathbf{A}) = n \times m$, and [**a**, **b**] or [**A**, **B**] denotes concatenation of two vectors or matrices.

1.2. Metabolic Fluxes and ^{13}C-Based Metabolic Flux Analysis

The metabolic fluxes are defined as the rates at which material are processed through metabolic pathways in vivo by cellular reactions. From the algebraic viewpoint, a metabolic reaction network is a system of linear equations, and, theoretically intracellular fluxes can be determined from extracellular flux (efflux) measurements and reaction stoichiometry by linear regression. However, a typical metabolic network consisting of catabolic and anabolic pathways comprises parallel, cyclic, and reversible reactions, representing an underdetermined system from the algebraic viewpoint (2). Due to this, not all intracellular fluxes can be calculated solely from efflux measurements unless additional information is available. Hereto, scientists and engineers have found an elegant way to gain the additional measurements required, i.e., the application of isotopic

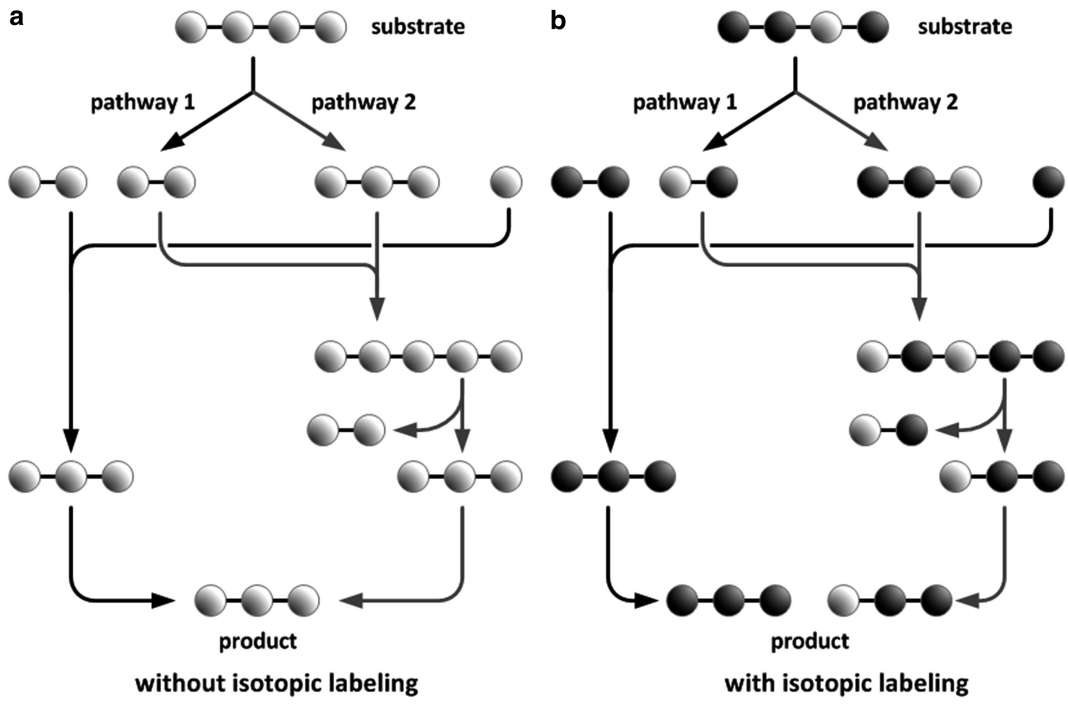

Fig. 1. Two alternative pathways from a substrate to a product without (**a**) and with (**b**) isotopic labeling.

labeling such as stable carbon isotopes. The characteristic isotopic labeling patterns formed in different metabolic products from specifically labeled isotopic tracers helps determine the fluxes which are immeasurable solely based on stoichiometry. An example is depicted in Fig. 1, in which a metabolite is produced from two alternative pathways through metabolic reactions. On the stoichiometry basis, the flux distribution between pathway 1 and 2 are not distinguishable (Fig. 1a). In contrast, the flux distribution can be quantified when the substrate elements are specifically labeled (Fig. 1b).

Along with progress in analytical instrumentations such as nuclear magnetic resonance spectroscopy (NMR) and mass spectrometry (MS), stable isotopic labeling techniques have widely been applied in bioscience to unveil in vivo mechanisms and activities of metabolic pathway reactions. This relies on the fact that atomic level transfers of elements from substrates to products are strictly governed by biochemical algorithms given by the specificity of enzymes. Hence, the metabolic fate of elements throughout a metabolic reaction network is decided by metabolic flux distributions, and this can be quantitatively traced using specifically isotope-labeled substrates. The first elegant way to formulate this elemental transfers throughout the entire central metabolic pathways was conceived by Zupke and Stephanopoulos (1994) (3),

and now genome-scale transfers of elements became available for bacterial metabolism (4).

Among others, ^{13}C-labeling is the typical choice because biochemical carbon transfer mechanisms between metabolites are well-studied, well-defined by the specificity of the enzymes, and not subject to nonenzymatic exchanges. As also demonstrated in Fig. 1b, The ^{13}C-labeling patterns in both metabolic intermediary and final products mainly depend on (1) ^{13}C-labeling positions in tracer substrates, (2) carbon transfer mechanisms of enzyme reactions, and (3) metabolic flux distribution. Functionally describing this relationship between metabolic fluxes and ^{13}C-labeling patterns over reaction stoichiometry is the framework of ^{13}C-based metabolic flux analysis (^{13}CMFA).

1.3. Isotopic Labeling Patterns and Mass Isotopomers

To determine the fluxes which are immeasurable by stoichiometry, ^{13}CMFA relies on the ^{13}C-labeling patterns of intermediary metabolites and/or metabolic end products formed from specifically ^{13}C-labeled substrates through in vivo metabolic reactions. In particular, the fractional enrichment of each subspecies of a metabolite with a distinctive ^{13}C-labeling pattern is the quantity utilized in ^{13}CMFA. Thus, it is useful to understand different terms utilized for isotopic species of a chemical compound (5). *Isotopologues* (isotopic homologues) are molecular species having identical elemental and chemical compositions but differ in isotopic content. Among those, *isotopomers* (isotopic isomers) differ by the location of isotopes on the compound, and *mass isotopomers* are groups of isotopomers classified according to their nominal mass. An example of all conceivable isotopomers and the isotopomers surjectively mapped into mass isotopomers is shown for two stable carbon isotopes (^{12}C and ^{13}C) in conjunction with a three carbon compound in Fig. 2.

Computing the number of isotopomers is a repeated permutation problem: there are n^p isotopomers for a compound containing p specific atoms having n stable isotopes. For the stable carbon isotopes of ^{12}C and ^{13}C, there are 2^3 carbon isotopomers for a molecule consisting of 3 carbon atoms. Typically, positional isotopomers are expressed using binary numbers to declare the positional labeling patterns, "0" for ^{12}C and "1" for ^{13}C , e.g., "000", "001", "010", "011", "100", "101", "110", and "111" for the above positional carbon isotopomers (6). When converting the binary code into decimal numbers, the carbon isotopomers are indexed from 0, 1, 2, …, $2^n - 1$ for the entire 2^n isotopomers (Fig. 2). Positional carbon isotopomers are observable by ^{13}C NMR in terms of ^{13}C multiplet analysis, and can be mapped onto $n + 1$ mass isotopomers according to the mass shift associated with isotopic content (Fig. 2).

Compared to the positional isotopomer analysis using NMR, each group of isotopomers having the same mass can be detected

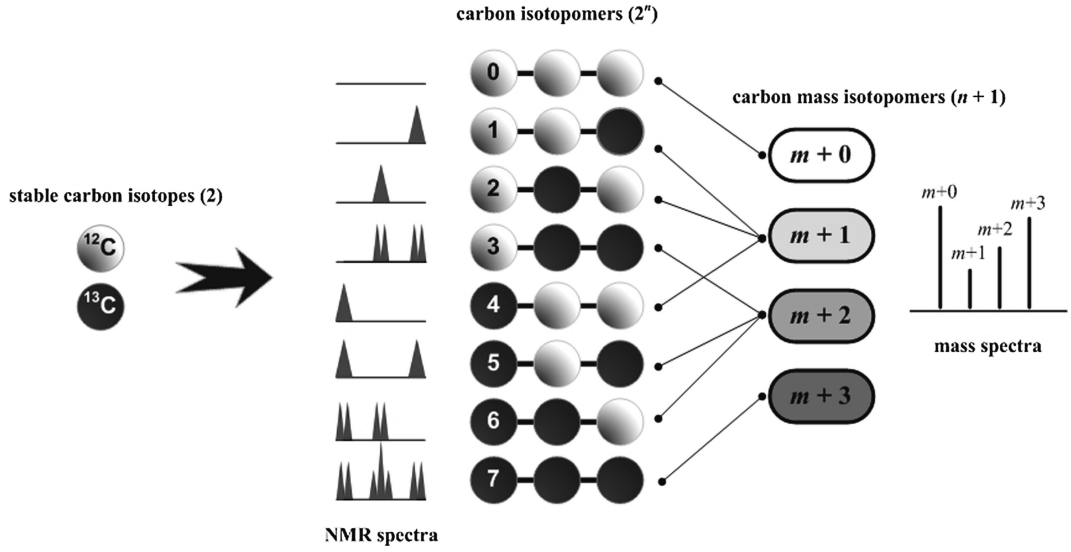

Fig. 2. An example of carbon isotopomers occurring from two stable isotopes of carbon (^{12}C and ^{13}C) for a three carbon compound and their surjective mapping onto mass isotopomers: *n* denotes the number of carbons in a compound and *m* the monoisotopic mass.

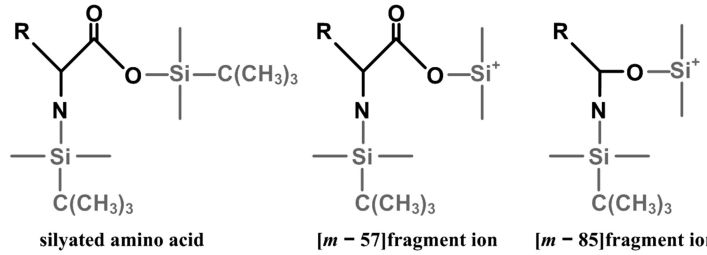

Fig. 3. *t*-Butyldimethylsilyl-derivatives of amino acid (TBDMS-silylation) and typical fragment ions utilized for ^{13}CMFA.

by MS with a higher sensitivity. Gas chromatography/mass spectrometry (GC/MS) with chemical derivatization, e.g., silylation using *N*-methyl-*N*-(*t*-butyldimethylsilyl)trifluoroacetamide) is a typical choice for mass spectrometry-based ^{13}CMFA (7). This is because the chemical derivatization results in defined fragment patterns during ionization process (Fig. 3), and measuring different fragment ions provides richer information for ^{13}CMFA compared to the case just measuring mass isotopomers of intact ions. Recently, a method using GC tandem MS (also known as MS/MS) has been developed for ^{13}CMFA in which further fragments are obtained for a selected mass isotopomer by collision-induced dissociation. This approach is quite impressive because it drastically expands the information content by additionally using the daughter mass isotopomer distributions for flux estimation without loss of the better sensitivity of MS over NMR (8).

As shown in Fig. 3, *t*-butyldimethylsilyl-derivative ions contain not only skeleton carbons of a metabolite but also carbons from the derivatization agent applied and other elements such as H, O, N, S, and Si. Carbons and other elements have naturally occurring stable isotopes of which abundance is well known (9). For ^{13}CMFA, the mass shift occurring due to the stable isotopes of the elements other than skeleton-carbons have to be taken into account and corrected. This requirement however depends on instrumental mass resolution. For instance, the carbon mass isotopomers of a compound that is not chemically derivatized can directly be measured using a high resolution instrument such as Fourier transform ion cyclotron resonance MS derivatization (10). If metabolites are derivatized for MS measurements, only non-skeleton carbons are subject to correction. With a lower resolution MS instruments such as GC/MS, the mass shift cannot be resolved. Therefore, this mass shift effect needs to be corrected for ^{13}CMFA implementation, and the correction involves mathematical modeling efforts (11). In Subheading 3.2, the related mathematical modeling details are discussed based upon the method reported in Yang et al. (2009) (12).

For isotope-based metabolic flux analysis, mass isotopomers of a compound are supplied or computed as fractional abundance (discrete distribution), i.e.,

$$ {}^{m+i}f = \frac{{}^{m+i}I_{m/z}}{\sum_{k=0}^{n} {}^{m+k}I_{m/z}} \quad \text{where} \sum_{i=0}^{n} {}^{m+i}f = 1 \text{ and } 0 \le \forall {}^{m+i}f \le 1, \tag{1} $$

where ${}^{m+i}f$ with $i = 0, 1, \ldots, n$ denotes the fractional abundance of the mass isotopomer with a mass shift of i from the monoisotopic mass m, $I_{m/z}$ the measured intensity, and $n + 1$ the number of mass isotopomers counted for the abundance calculation. The mass isotopomer distribution (MID) of a compound is a discrete distribution consisting of each fraction (${}^{m+i}f$) of which sum gives the unity. To describe the mass shift effect modeling and correction in Subheading 3.2 and Note 2, we define two different but interconvertible MIDs:

- $\mathbf{MID_{MS}}$. Mass isotopomer distribution that includes all mass effects from naturally occurring stable isotopes. This is directly obtained by normalizing measured MS signals by the sum. The symbol used for this quantity is $\tilde{\mathbf{y}}$.
- $\mathbf{MID^{C}_{SIM}}$. Carbon mass isotopomer distribution that only includes the mass shift effect by stable carbon isotopes (^{12}C and ^{13}C) in the carbon skeleton of a compound. This quantity is signified by $\mathbf{x}_C$.

1.4. Basic Material Balances and ^{13}C-Labeling System

To begin with modeling of ^{13}C-labeling system that can be utilized for in vivo flux estimation, one needs to understand basic material balances given for intracellular metabolite pools. Based on the mass conservation law and assumption of constant intracellular volume, the mass accumulation given for an intracellular metabolite with a pool size of c_i can be formulated by its net formation through p incoming and q outgoing reactions along with the dilution occurring due to cell growth with a specific growth rate of μ:

$$\frac{dc_i}{dt} = \left(\sum_{j=1}^{p} s_j v_{in,j} - \sum_{k=1}^{q} s_k v_{out,k} \right) - \mu c_i. \tag{2}$$

The rates of incoming and outgoing reactions equal the products of fluxes (v_{in} and v_{out}) with the corresponding stoichiometric coefficients (s), i.e., rate $= sv$. The stoichiometric coefficients define how many moles of the metabolite for which the balance is set up are converted from reactants (s_j) and into other metabolites (s_k). All fluxes are physically and physiologically constrained such that $v = \{v \in \Re \mid v \geq 0\}$. This non-negativity implies the separate use of forward and backward fluxes for reversible reactions instead of using net fluxes.

Usually, it is accepted that the turnover of most intracellular metabolite pools is fast and intracellular pool sizes are quasi time-invariant (pseudo-steady-state), i.e., $dc_i/dt = 0$ in (Eq. 2). In addition to this, the dilution rate is typically much smaller than net formation rate and thus considered negligible. Hence, (Eq. 2) can be expressed as a linear equation of the net formation:

$$\frac{dc_i}{dt} = \left(\sum_{j=1}^{p} s_j v_{in,j} - \sum_{k=1}^{q} s_k v_{out,k} \right) = 0. \tag{3}$$

Consequently, a metabolic reaction network can easily be constructed as a linear system by setting up each balance for every metabolite defined in the network, i.e.,

$$\mathbf{S} \cdot \boldsymbol{\nu} = 0. \tag{4}$$

Here, $\boldsymbol{\nu} = (v_1, v_2, \ldots, v_n)^{\mathrm{T}}$ is the flux vector comprising all the fluxes defined in the metabolic network and $\mathbf{S}$ the $m \times n$ stoichiometric matrix, linearly mapping n fluxes for m metabolites.

The mass conservation law is also valid for each isotopomer pool belonging to a metabolite. Therefore, the material balance (Eq. 3) is expandable to each distinctive ^{13}C-labeling component (e.g., positional carbon isotopomers, cumomers, elementary metabolite units, etc.; Subheading 1.5) belonging to a metabolite. For each iith component of which fraction is x_{ii}, we can set up the following balance equation by assuming constant intracellular

metabolite pool size, i.e., $dc_i/dt = 0$ (metabolic steady-state) and multiplying x_{ii} on each side of (Eq. 3):

$$c_i \frac{dx_{ii}}{dt} = x_{ii} \sum_{j=1}^{p} s_j v_{in,j} - x_{ii} \sum_{k=1}^{q} s_k v_{out,k}, \quad ii = 1, 2, 3, \ldots, n. \tag{5}$$

Here, n equals the number of ^{13}C-labeling components belonging to the ith metabolite. Note that each isotopomer pool size can be change with respect to time under the time-invariant metabolite pool size assumption, which is implied by the term on the left-hand side of (Eq. 5). This term becomes zero at the so-called isotopic steady-state.

Further, the fraction x_{ii} in the first term on the right-hand side of (Eq. 5) can be decomposed into individual subcomponents produced from each jth incoming reaction, i.e., $x_{ii} = x_{ii,1} + x_{ii,2} + \ldots + x_{ii,p}$, where $x_{ii,j}$ is the fraction of x_{ii} converted by the jth reaction with a rate of $s_j v_{\mathrm{in},j}$. Accordingly, (Eq. 5) can be reformulated as follows:

$$c_i \frac{dx_{ii}}{dt} = \sum_{j=1}^{p} s_j v_{in,j} x_{ii,j} - x_{ii} \sum_{k=1}^{q} s_k v_{out,k}. \tag{6}$$

It would be very cumbersome if one would set up the above balance equation for every ^{13}C-labeling component of each metabolite separately. Therefore, individual balance equations are rather written for the group of the entire ^{13}C-labeling components belonging to the ith metabolite $\mathbf{x}_i = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$, which are made from p different reactions and further converted into other metabolites by q different reactions.

$$c_i \frac{d\mathbf{x}_i}{dt} = \sum_{j=1}^{p} s_j v_{in,j} \mathbf{h}(\mathbf{x}_{\forall \in j}) - \mathbf{x}_i \sum_{k=1}^{q} s_k v_{out,k} \tag{7}$$

Here, $\mathbf{x}_{\forall \in j}$ denotes the ^{13}C-labeling components of all reactants involved in the jth reaction. The function $\mathbf{h}(\mathbf{x}_{\forall \in j})$ maps the entries of $\mathbf{x}_{\forall \in j}$ to the corresponding entries of $\mathbf{x}_i$ by considering reaction mechanisms. For instance, $\mathbf{h}(\mathbf{x}_{\forall \in j})$ equals a linear mapping for a monomolecular reaction, e.g., but bilinear mapping for a bimolecular (condensation) reaction if positional isotopomers are applied (6). In contrast, $\mathbf{h}(\mathbf{x}_{\forall \in j})$ is simply a linear mapping for any reaction mechanisms when isotopic enrichments in individual carbon positions are balanced (3). A similar but more complex modeling concept as the latter was employed to resolve the computational challenges associated with the bilinear structure of the isotopomer system (11, 12).

1.5. Progress in ^{13}C-Labeling System Modeling

Around the 1980s, several approaches have been developed to utilize isotopic labeling for flux calculation (13–19), yet those proto-type models were limited to a small part of metabolism and not sophisticated for a universal application. The first mathematical

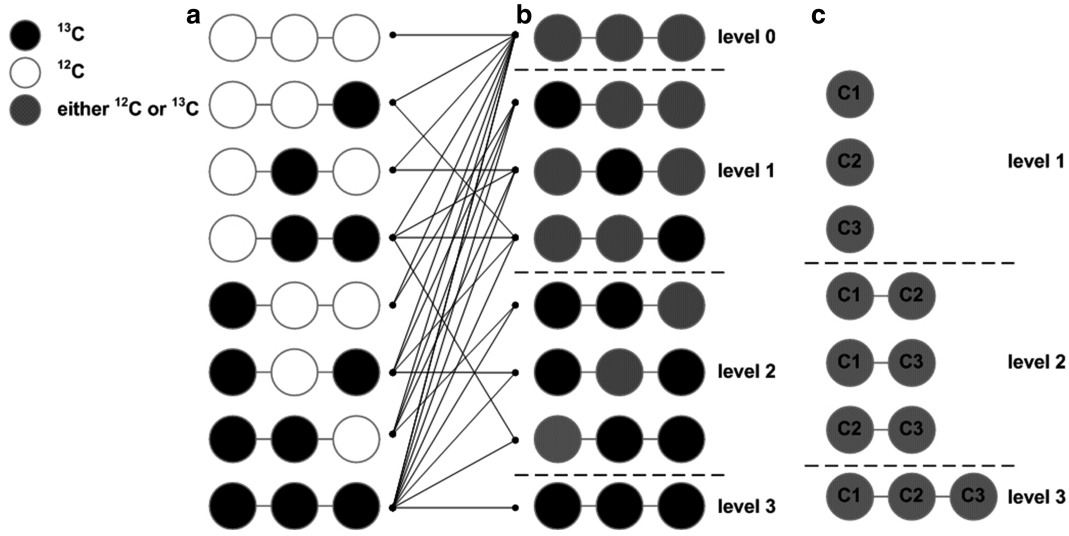

Fig. 4. Positional carbon isotopomers (6) (**a**), cumomers (21) (**b**), elementary metabolite units (22) (**c**), and their relations.

model for ^{13}CMFA that was capable of dealing with the entire central metabolic pathways was conceived by Zupke and Stephanopoulos (1994). They introduced the so-called atom mapping matrices and facilitated modeling of positional carbon transfer mechanisms in a metabolic network (3). Here, material balances such as (Eq. 7) are set up for the positional ^{13}C-enrichments in individual carbon atoms, and elemental transfer of carbon atoms is linearly mapped between metabolites by the use of the so-called atom mapping matrices, i.e.,

$$c_i \frac{d\mathbf{x}_i}{dt} = \sum_{j=1}^{p} s_j v_{in,j} \mathbf{M}_{j\to i} \cdot \mathbf{x}_j - \mathbf{x}_i \sum_{k=1}^{q} s_k v_{out,k}. \qquad (8)$$

Here, $\mathbf{M}_{j\to i}$ is the atom mapping matrix mapping positional carbon transfer from the jth to the ith metabolite and $\mathbf{x}_j$ the positional ^{13}C-labeling enrichment of the jth precursor. By setting up such balance for each metabolite, the entire ^{13}C-labeling system can be expressed as $\mathbf{Ax} + \mathbf{b} = \mathbf{0}$ under isotopic steady-state condition. However, the method is not directly applicable to MS data unless the so-called summed fractional labeling (20) is applied, which is much less informative than the direct use of MS data.

On the basis of the above approach, Schmidt et al. (1997) expanded the concept to positional isotopomer level, i.e., 2^n carbon isotopomers (Fig. 4a) instead of n carbons of a metabolite (6). The isotopomer model also enabled to apply MS data without loss of information gained from mass isotopomer measurements, as their surjective mapping depicted in Fig. 2. The positional isotopomer balance is linear for a monomolecular reaction and can be expressed similarly to (Eq. 8), but it becomes bilinear for a bimolecular

(condensation) reaction. For a bimolecular reaction A + B → C, the function $\mathbf{h}(\mathbf{x}_{\forall \in j})$ in (Eq. 7) would be $(\mathbf{M}_{A\to C}\cdot\mathbf{x}_A) \bigotimes (\mathbf{M}_{B\to C}\cdot x_B)$, where $\bigotimes$ denotes the element-wise multiplication of two vectors and $\mathbf{M}_{A\to C}$ and $\mathbf{M}_{B\to C}$ the isotopomer mapping matrices. The positional isotopomer model expedited in vivo application of MS-based ^{13}CMFA, yet it was computationally expensive and occasionally instable due to its nonlinear structure necessitating a numerical algorithm such as relaxation or root-finding methods.

This motivated Wiechert et al. (1999) to find a mathematical work-around by introducing the so-called cumomers (cumulative isotopomers) (21). There, the nonlinear isotopomer system is transformed into a cascade linear system, resulting in an explicit solution for the positional isotopomer system. By definition, the cumomers are a set of isotopomers with particular positional ^{13}C-labels, grouped into different levels. Any positions can be either ^{12}C or ^{13}C at level 0, one particular position is ^{13}C but others either ^{12}C or ^{13}C at level 1, two specific positions are ^{13}C but others either ^{12}C or ^{13}C at level 2, and so forth (Fig. 4b).

Followed by this definition, cumomer balances are set up for each level separately. Any 0th level cumomers comprise all possible isotopomers (Fig. 4), and, therefore, its fraction is unity. Accordingly, (Eq. 7) for the 0th level is simply $\mathbf{S}\cdot\boldsymbol{\nu} = \mathbf{0}$ at metabolic steady-state. The 1st level involves only one carbon of each position, which is equivalent to the positional carbon transfer model described above (3). Thus, the entire 1st level can be formulated such that $\mathbf{Ax} + \mathbf{b} = \mathbf{0}$ under isotopic steady-state assumption. Therefore, the 1st level cumomer fractions $\mathbf{x}$ are obtained using matrix inversion. From the 2nd level, balances are set up for cumomers only involving those belonging to the same or lower levels. By doing so, all the isotopomer reactions are counted, and the cumomer reactions from a higher to a lower level completely vanish since they are redundant (21). The reactions from the same level cumomers are monomolecular (A → B), and those from the lower levels bimolecular (condensation: A + B → C). Thus, only the reactions involving lower level cumomers introduce bilinear term in (Eq. 7), of which values are already computed in the previous levels of the cascade system. A generalized cumomer cascade system with n levels can be expressed as follows:

$$\left(\begin{array}{l} \text{level } 0 : \mathbf{S}\cdot v = 0 \text{ for metabolic steady-state} \\ \text{level } 1 : \mathbf{D}\dfrac{d\mathbf{x}_1}{dt} = \mathbf{A}_1\cdot\mathbf{x}_1 + \mathbf{b}_1 \\ \text{level } 2 : \mathbf{D}\dfrac{d\mathbf{x}_2}{dt} = \mathbf{A}_2\cdot\mathbf{x}_2 + \mathbf{b}_2 + \mathbf{h}_2(\mathbf{x}_1) \\ \vdots \\ \text{level } n : \ \mathbf{D}\dfrac{d\mathbf{x}_n}{dt} = \mathbf{A}_n\cdot\mathbf{x}_n + \mathbf{b}_n + \mathbf{h}_n(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_{n-1}) \end{array}\right) = 0 \text{ for isotopic steay-state} \quad (9)$$

Here, $\mathbf{x}_i$ with $i = 1, 2, \ldots, n$ is the entire cumomer fractions belonging to each ith level of the cascade system and $\mathbf{D}$ the diagonal matrix representing intracellular metabolite pool sizes. $\mathbf{A}_i$ is the system matrix stoichiometrically mapping cumomer reactions and $\mathbf{b}_i$ the vectors with known values such as the cumomer fractions of ^{13}C-substrate(s). The vector function $\mathbf{h}_j$ with $j = 2, 3, \ldots, n$ is the bilinear expression for the bimolecular reactions from the lower level cumomers. Hence, the solution for $\mathbf{x}$ can be obtained analytically by sequentially solving the cascade system from the 0th to the nth if isotopic steady-state is achieved. The cumomer fractions $\mathbf{x}$ can then be converted to positional isotopomer distributions and MID. Such a cascade system is also advantageous for a isotopic non-steady-state system because the isotopomer problem becomes simply the first-order linear ordinary differential equation (ODE) system.

Followed by the cumomer concept, Antoniewicz et al. (2007) introduced elementary metabolite units (EMU) and drastically reduced the dimension of isotopomer labeling systems without loss of information (22). EMU of a compound is defined to be a moiety comprising any distinct subsets of the compound's atoms (Fig. 4c). The minimum set of EMUs required for modeling is identified by back-tracking the reaction routes to the measured ^{13}C-species upwards ^{13}C-substrates applied. The underlying idea is that any metabolites' MIDs can be calculated either from MIDs of precursor EMUs either directly or using the polynomial convolution (see Note 2). The convolution is only required for bimolecular reactions. An EMU system is also constructed analogously to that of a cumomer model, yet yields much less equations due to the back-tracking. This gives a very similar cascade structure given by (Eq. 9) that can be solved analytically using matrix inversion. Here, the nonlinear expression $\mathbf{h}_j$ contains convolution terms involving lower level EMU variables.

Lately, Srour et al. (2011) has lately introduced a concept called fluxomers which combine fluxes and isotopomer abundances into one and improved computational efficiency for steady-state ^{13}CMFA (23), but this is not covered in this chapter.

1.6. Generalized ^{13}C-Labeling System Formulation

Using a model discussed in Subheading 1.5, ^{13}C-labeling values such as positional isotopomer or mass isotopomer distributions can be generated for intermediary metabolites as well as metabolic products (excreted metabolites, proteinogenic amino acids, etc.). In a model of ^{13}C-labeling system, these ^{13}C-labeling values of metabolites are the dependent variables, and metabolic fluxes and ^{13}C-labeling patterns of substrates are the independent variables. Thus, one can generalize this relation under metabolic steady-state condition as follows:

$$\mathbf{D}\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{n}(\Theta), \mathbf{x}, \mathbf{x}_0). \tag{10}$$

Here, $\mathbf{f}$ is the system of balance equations such as (Eq. 7) or (Eq. 9) set up for all the metabolites defined in the metabolic network model, $\mathbf{D}$ the diagonal matrix representing intracellular pool sizes, $\mathbf{x}$ the ^{13}C-labeling values given for the metabolites, and $\mathbf{x}_0$ the known ^{13}C-substrate labeling patterns. The function $\mathbf{n}(\boldsymbol{\Theta})$ signifies the parametrized form of (Eq. 4) and $\boldsymbol{\Theta}$ the independent flux variables (or often referred as free fluxes). The free fluxes are the design parameters numerically determined from efflux and ^{13}C-labeling measurements (Subheading 1.7). Once $\boldsymbol{\Theta}$ is estimated from available experimental data, one can compute other fluxes simply by computing $\mathbf{n}(\boldsymbol{\Theta})$. We will discuss the parametrization procedure, i.e., transformation of an underdetermined system $\mathbf{S}\cdot\boldsymbol{\nu} = \mathbf{0}$ into $\boldsymbol{\nu}_{\text{depend}} = \mathbf{n}(\boldsymbol{\Theta})$ in Subheading 3.3, and the generation of values for $\mathbf{x}_0$ (see Note 1).

To determine $\boldsymbol{\Theta}$ numerically using a gradient-based optimization (24) or design ^{13}C-labeling experiments (25), the analytically calculated sensitivity matrix of (Eq. 10) is useful, which is the partial derivatives of $\partial\mathbf{x}/\partial\boldsymbol{\Theta}$. For the isotopic steady-state approaches, i.e., $\mathbf{f}(\mathbf{n}(\boldsymbol{\Theta}),\ \mathbf{x}) = \mathbf{0}$, the partial derivatives are obtained straightforwardly based on the chain rule and implicit function differentiation:

$$\frac{\partial \mathbf{x}}{\partial \Theta} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1} \cdot \left(\frac{\partial \mathbf{f}}{\partial v_{\text{depend}}} \cdot \frac{\partial \mathbf{n}(\Theta)}{\partial \Theta} + \frac{\partial \mathbf{f}}{\partial \Theta}\right) \text{ with } v_{\text{depend}} = \mathbf{n}(\Theta). \tag{11}$$

For the isotopic non-steady-state case, computing $\partial\mathbf{x}/\partial\boldsymbol{\Theta}$ can be computationally expensive since the partial derivatives are numerically integrated while solving the ODE system of (Eq. 10). It adds additional dim($\mathbf{x}$) × dim ($\boldsymbol{\Theta}^*$) variables to dim ($\mathbf{x}$) while solving (Eq. 10) numerically.

$$\begin{aligned}\frac{\mathrm{d}(\partial \mathbf{x}/\partial \Theta^*)}{\mathrm{dt}} = \Bigg[&\mathbf{D}^{-1}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \cdot \left(\frac{\partial \mathbf{x}}{\partial \Theta}\right) + \frac{\partial \mathbf{f}}{\partial v_{\text{depend}}} \cdot \frac{\partial \mathbf{n}(\Theta)}{\partial \Theta} + \frac{\partial \mathbf{f}}{\partial \Theta}\right), \\ &-\left(\mathbf{D}^{-1}\right)^2 \cdot \mathrm{diag}(\mathbf{f})\Bigg] \\ &\text{where } \Theta^* = [\Theta,\ \text{pool sizes}]\end{aligned} \tag{12}$$

The second matrix $-(\mathbf{D}^{-1})^2\cdot\mathrm{diag}(\mathbf{f})$ represents the partial derivatives of $\mathbf{x}$ with respect to pool sizes since metabolite pool sizes are also the parameters to be optimized using the corresponding measurements and/or measured ^{13}C-labeling dynamics.

1.7. Experimental Approaches and Numerical Flux Estimation

Experimental approaches applied to ^{13}CMFA predominantly utilize isotopic steady-state ^{13}C-labeling patterns formed in different metabolites from ^{13}C-labeled substrates such as hydrolyzed cell proteins, extracellular products, or intracellular metabolites. There are also approaches of isotopic non-steady-state conditions, which is both

computationally and experimentally more expensive, involving measurements of time-domain ^{13}C-labeling dynamics and information on intracellular metabolite pool sizes (26–29), as shown in (Eq. 10). Metabolite pool sizes can be measured or limitedly estimated during ^{13}CMFA (27). However, measuring intracellular concentrations accurately can be nontrivial often due to difficulties associated with sample preparation (30). In either case, the metabolic steady-state is the prerequisite, i.e., time-invariant intracellular metabolite pool sizes (Subheading 1.4). The metabolic steady-state is attained in a chemostat or for exponentially growing cells (balanced growth condition). There was a theoretical study concerning ^{13}C-labeling experiments under the metabolic non-steady-state condition (31), yet its practical application is not fully established.

There are other ^{13}CMFA approaches such as the flux ratio concept (32–34), yet the majority of techniques developed for ^{13}CMFA involve a constrained nonlinear optimization problem to determine intracellular fluxes from experimentally measured isotopic labeling patterns and effluxes, i.e.,

$$\begin{aligned} \min_{\Theta \in \Re^m} f(\Theta) &= \frac{1}{2}(\boldsymbol{\eta} - \mathbf{F}(\Theta))^{\mathrm{T}} \cdot \Sigma_{\boldsymbol{\eta}}{}^{-1} \cdot (\boldsymbol{\eta} - \mathbf{F}(\Theta)) \\ &\text{subject to } \mathbf{LB} \le \Theta \le \mathbf{UB} \wedge \mathbf{a} \le \mathbf{n}(\Theta) \le \mathbf{b} \end{aligned} \tag{13}$$

In the above constrained nonlinear least-squares minimization problem (NLSP), $f(\boldsymbol{\Theta})$ is the objective to be minimized by finding an optimal set of independent variables of $\boldsymbol{\Theta}$ (often referred as free fluxes or design parameters), and $\mathbf{F}(\boldsymbol{\Theta})$ signifies the model function generating the values corresponding to the measured data set $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_n)^{\mathrm{T}}$ consisting of ^{13}C labeling data and effluxes. The measurements $\boldsymbol{\eta}$ are typically assumed to have random errors with the normal distribution such that $\boldsymbol{\varepsilon} \in N(\mathbf{0}, \boldsymbol{\Sigma}_\eta)$ with $\boldsymbol{\Sigma}_\eta$ denoting the covariance matrix. The design parameters ($\boldsymbol{\Theta}$) typically have lower and upper bounds ($\mathbf{LB}$ and $\mathbf{UB}$), and the inequality constraint given for the flux variables ($\boldsymbol{\nu}$) can be obtained, e.g., by solving a linear programming problem using stoichiometry. We will discuss how to generate these inequality constraints in Subheading 3.4 and the computational implementation of numerical ^{13}CMFA in Subheading 3.5.

To obtain the optimal solution of (Eq. 13), one can choose either a gradient-based local optimization such as sequential quadratic programming (SQP, see Note 5) or gradient-free global optimization such as simulated annealing or genetic algorithms. Such algorithms are described elsewhere in detail (35–37). The stochastic global optimization methods can be computationally expensive due to the time required to obtain the so-called asymptotic convergence or reachability in high dimensional parameter spaces (38–44). Moreover, such algorithms of random nature may

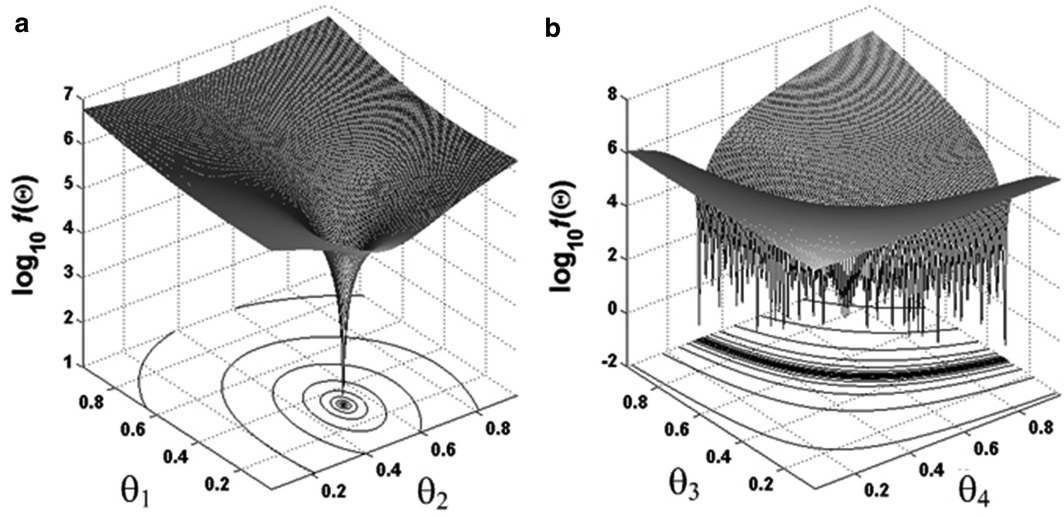

Fig. 5. Objective function behavior in space of non-correlated parameters of θ_1 and θ_2 (**a**) and nonlinearly correlated parameters of θ_3 and θ_4 (**b**) (reproduced from ref. 24)).

fail to locate the global solution unless the number of samplings tends to infinity, which is often impossible in practice (44, 45).

In comparison, the gradient-based local optimizations are much superior in term of convergence speed, especially, when analytical gradients and Hessian are supplied (see Note 3). However, the solution quality depends heavily on starting points (36, 42), and reaching the global optimum is ensured only for convex problems, whereas it is nontrivial to determine the convexity of general nonlinear problems with nonlinear constraints (46). Thus, one may obtain solutions that are not necessarily global (36) and that might vary depending on starting points.

During numerical flux estimation, some independent variables can have nonlinear correlations (24). For those parameters that are not correlated with others (Fig. 5a), their unique solutions can be located using an optimization algorithm. If two or more parameters have nonlinear correlations with each other, there can be infinite number of global solutions (Fig. 5b). Such nonlinear parameter correlations in a high-dimensional nonlinear model such as ^{13}CMFA problem can be identified only by a tedious process of executing parameter estimation using different sets of starting values a posteriori (47). Together with the starting-point-dependency, this actually motivates the use of Monte Carlo method for flux estimation and statistical quality analysis, of which implementation is described in Subheading 3.5.

In practice, the constrained problem of (Eq. 13) is formulated as the Lagrangian function, a linear combination of the objective function and the constraints (36). Therefore, a simple nonlinear regression model cannot be applied to determine confidence

intervals given for flux estimates. Based on this fact, Antoniewicz et al. (2006) (48) suggested a statistical method that numerically determines the confidence intervals given for flux estimates by solving a series of nonlinear optimization problems, which can be selected as an alternative to the Method of Monte Carlo.

2. Materials

In order to implement ^{13}CMFA, experimental measurements of effluxes and ^{13}C-labeling patterns of different metabolites are required, of which measurements can be conducted, e.g., by HPLC and GC/MS, respectively. Relevant information on experimental and analytical procedures is well described elsewhere (49, 50). From the experimental measurement data, intracellular fluxes are estimated by means of a computational tool that supports mathematical operations such as linear algebra and numerical optimization. Of course, one can build own model from scratch, but there are also user-friendly open source tools available such as EMU-based Metran (www.che.udel.edu/mranton/metran.html) or OpenFLUX (openflux.sourceforge.net) and cumomer-based 13CFlux (www.13cflux.net). For the steady-state ^{13}CMFA implementation, the following data sets and tools are necessitated:

1. Effluxes obtained at metabolic steady-state, e.g., cumulative or instantaneous yield coefficients of all detectable extracellular species including biomass.
2. ^{13}C-labeling data such as GC/MS mass isotopomer signals of metabolic products and/or intermediary metabolites.
3. A computational algorithm to correct the mass shift effect associated with naturally occurring isotopes of elements.
4. A computational tool to carry out numerical estimation of in vivo fluxes and statistical analysis.

Ideally, replicate data are required to take uncertainties associated with effluxes and ^{13}C-labeling measurements into account for statistical analysis. The uncertainties associated with effluxes can simply be obtained from replicate measurements, whereas those given for mass spectrometric ^{13}C-labeling measurements can be obtained by linearized regression analysis (12) (see Subheading 3.2.2 and Note 3).

3. Methods

The overall experimental as well as computational procedure of ^{13}CMFA is depicted in Fig. 6. On one hand, the cultivation using ^{13}C-labeled substrates is conducted to obtain experimental data

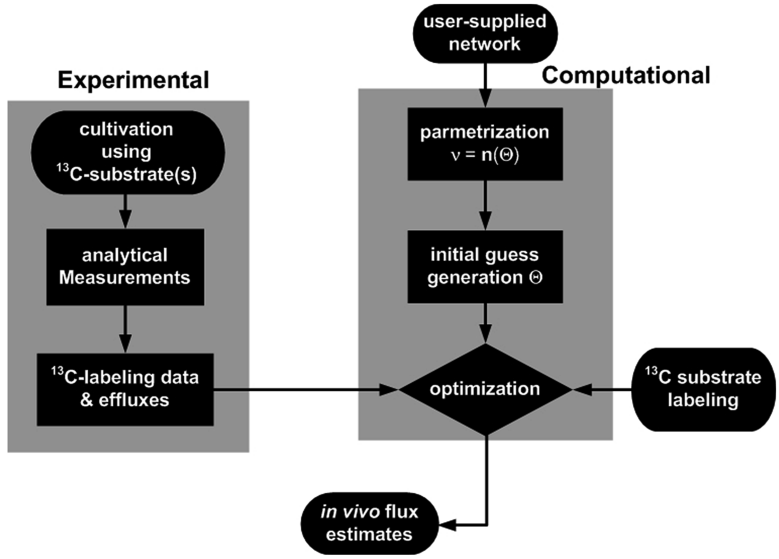

Fig. 6. Overall procedure of ^{13}C-based metabolic flux analysis: experimental data preparation and computational procedure.

sets comprising effluxes and ^{13}C-labeling patterns. On the other hands, a computational algorithm is required to select free fluxes from the user-supplied metabolic reaction network (parametrization) and perform numerical optimization. Using the optimization algorithm, the difference between the user-supplied experimental data and the corresponding model values are numerically minimized by iteratively altering the free fluxes and generating the model values. Hereby, the model values are the function of the free fluxes and the ^{13}C-labeling patterns of the tracer substrates applied to the cultivation, as defined in (Eq. 10). At the end of this process, in vivo fluxes are estimated and also their statistical properties, e.g., using the method of Monte Carlo.

3.1. Experimental Implementation

A general procedure of ^{13}C-labeling experiments begins with a cultivation using one or more ^{13}C-substrates in a chemically defined medium. In this context, the first question one may have is the choice of ^{13}C-substrates concerning their labeling position. Hereto, a *D*-optimality-based computational algorithm has been developed, which is the so-called computer-aided optimal design (25, 51). The algorithm basically maximizes the invertiability of the model's partial derivatives (see Note 3) in order to increase the overall flux observability. Once a proper set of ^{13}C-substrates is selected, one of the following two procedures can be conducted to obtain experimental data required for a steady-state ^{13}CMFA implementation depending on experimental questions and conditions:

3.1.1. Growth Phase ^{13}CMFA

This method is applicable for balanced growth conditions under which macromolecules such as cell proteins are synthesized from carbon sources through steady-state fluxes of catabolic and anabolic reactions. The balanced growth condition is obtainable in a chemostat or during exponential cell growth. The experimental procedure of the approach is as follows:

1. Grow cells on selected ^{13}C-tracer(s) under metabolic steady-state condition, e.g., exponential growth or chemostat. For batch cultures, minimize the inoculum size so that the proportion of non-labeled cells is negligible at cell harvest.
2. Obtain metabolic products at isotopic steady-state condition, e.g., amino acids from cell protein hydrolysis and/or secreted metabolic products. Five generation time (residence time in chemostat) is typically considered sufficient to obtain the isotopic steady-state condition for biomass constituents.
3. From concentration measurements, calculate cumulative yield coefficients of all the extracellular species produced and consumed by the cells including biomass yield coefficients.
4. Measure ^{13}C-labeling patterns of different metabolic products such as amino acids hydrolyzed from cell proteins, e.g., using chemical derivatization.

It should be noted that this method is not applicable, e.g., to the cells in multi-phase processes (separate growth and production phase) or non-growing but metabolically active cells. In this case, the following approach can be considered.

3.1.2. Instantaneous ^{13}CMFA

For those cells which do not necessarily grow exponentially or are non-growing but metabolically active, one can assume a local steady-state for a short period of time under certain circumstances. Globally fluxes are subject to variation with respect to time during a fermentation process, i.e., metabolic non-steady-state. Locally, there can be a certain time frame during which no mass accumulation occurs and metabolic steady-state can be attained or approximated, e.g., instantaneously quasi-invariant yield coefficients with closed carbon balance within that time frame. Moreover, one may obtain isotopic steady-state for certain intracellular metabolites within that time frame. If this is the case, one can determine instantaneous flux distributions given for the time frame in terms of steady-state ^{13}CMFA. A similar approach has been done elsewhere, which applied dynamic CO_2 labeling measurements extrapolated to isotopic steady-state together with instantaneous efflux measurements (52). However, a dynamic ^{13}CMFA approach is required, in case isotopic steady-state cannot be attained or easily extrapolated, which will be discussed in the following chapter of this book. For an isotopic steady-state approach, the following steps can be implemented:

1. Incubate cells with ^{13}C-labeled substrate(s) until isotopic steady-state is achieved for those intracellular metabolites applied to flux estimation. It should be noted that the time required for isotopic steady-state has to be determined a priori, e.g., using universally ^{13}C-labeled substrate(s) and by tracing isotopic dynamics of intracellular metabolites.
2. Harvest cells at isotopic steady-state, and extract intracellular metabolites for ^{13}C-labeling measurements such as intermediary metabolites of the central metabolic pathways and free intracellular amino acids. Different extraction protocols are available elsewhere (30, 53, 54).
3. From cultivation data, calculate instantaneous yield coefficients at the event of sampling for all the extracellular species produced and consumed by the cells including biomass yield coefficients. Check if instantaneous elemental balances such as carbon and degree of reduction are closed.
4. Measure isotopic steady-state ^{13}C-labeling patterns of the intracellular metabolites extracted from cells.

The above approach may involve spiking ^{13}C-tracer(s) or switching feed in the middle of fermentation at which non-labeled substrates and products are present in fermentation broth. Due to this, there can be non-negligible carbon fluxes or exchanges from/with extracellular non-labeled species, and this has to be taken into account. A modeling approach to this issue is addressed in detail elsewhere (55).

3.2. Mass Isotopomer Data Processing

The effluxes required for ^{13}CMFA, i.e., instantaneous yield coefficients are straightforwardly obtainable from concentration measurements. In contrast, preparing the ^{13}C-labeling data for ^{13}CMFA is not trivial, especially, when a lower mass resolution MS instruments are employed for mass isotopomer analysis. For instance, MS signals measured with conventional GC/MS always implicate mass shift effects from naturally occurring stable isotopes of elements. This so-called mass shift effect has to be corrected when preparing the ^{13}C-labeling data ($\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$ for GC/MS) and measurement uncertainties. The MID of any chemical compound ($\mathbf{MID}_{\mathbf{MS}}$) can be described as a polynomial convolution of the MID given for the carbon skeleton ($\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$) and that for non-skeleton moiety (see Note 2). This is the fundamental of the modeling procedure discussed in this section.

3.2.1. Mass Isotopomer Distribution Analysis

For a compound C_n–$C_\alpha H_\beta O_\delta N_\gamma$ with Cn denoting the n carbons in the skeleton and $C_\alpha H_\beta O_\delta N_\gamma$ the non-skeleton moiety, there are $n + \alpha + \beta + 2\delta + \gamma + 1$ mass isotopomers corresponding to $\mathbf{MID}_{\mathbf{MS}}$ due to naturally occurring stable isotopes of ^{12}C, ^{13}C, ^{1}H, ^{2}H, ^{16}O, ^{17}O, ^{18}O, ^{14}N, and ^{15}N. Isotopes of Si (^{28}Si, ^{29}Si, ^{30}Si) can also be counted when metabolites are silylated for GC/MS measurement.

Just for the carbon skeleton, there are $n + 1$ mass isotopomers only consisting of carbon isotopes, which corresponds to $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$.

In practice, not all mass isotopomers of a compound can be measured, e.g., due to signal detection limits, and, also, $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$ is not directly measurable unless a high mass resolution MS instruments are employed. Therefore, $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$ has to be computed from $\mathbf{MID}_{\mathbf{MS}}$ measurement, which is a nonlinear problem although it has been typically approximated as a linear problem (56–59). To get $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}(\mathbf{x}_C)$ from $\mathbf{MID}_{\mathbf{MS}}$ $(\tilde{\mathbf{y}})$ the following steps can be implemented for a fragment ion with an elemental composition of C_q–$C_\alpha H_\beta O_\delta N_\gamma P_\phi S_\kappa Si_\lambda$ with C_q being the carbon skeleton and $C_\alpha H_\beta O_\delta N_\gamma P_\phi S_\kappa Si_\lambda$ the non-skeleton moiety.

1. Acquire mass isotopomer peaks of the fragment ion, and integrate peak areas. At least $q + 1$ mass isotopomer peaks are necessary to determine $q + 1$ entries of $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$. A useful guideline for an accurate mass isotopomer assessment can be found elsewhere (60).
2. Normalize peak intensities as given by (Eq. 1), and form a nonlinear regression model by denoting $\tilde{\mathbf{y}}$ the normalized MID corresponding to $\mathbf{MID}_{\mathbf{MS}}$, $\mathbf{y}$ the mass isotopomer signal intensities, $\mathbf{x}_C$ the MID given for the carbon skeleton corresponding to $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$, and $\boldsymbol{\varepsilon}_{\tilde{\mathbf{y}}}$ the normalized measurement noise with the normal distribution with its covariance $\boldsymbol{\Sigma}_{\tilde{\mathbf{y}}}$:

$$\tilde{\mathbf{y}} = \frac{\mathbf{y}}{\mathbf{1}^{\mathrm{T}} \cdot \mathbf{y}} = \mathbf{F}(\mathbf{x}_C) + \varepsilon_{\tilde{\mathbf{y}}} \quad \text{with } \varepsilon_{\tilde{\mathbf{y}}} \in \mathrm{N}(\mathbf{0}, \Sigma_{\tilde{\mathbf{y}}}). \tag{14}$$

3. Formulate the model function $\mathbf{F}(\mathbf{x}_C)$ from (Eq. 37) by considering the above normalization of $\mathbf{y}$. For $\tilde{\mathbf{y}} = \mathbf{y}/(\mathbf{1}^{\mathrm{T}}\mathbf{y}) = ({}^{m+0}\tilde{y},\ {}^{m+1}\tilde{y}, \ldots,\ {}^{m+p}\tilde{y})^{\mathrm{T}}$ and $\mathbf{x}_C = ({}^{m+0}x_C,\ {}^{m+1}x_C, \ldots,\ {}^{m+q}x_C)^{\mathrm{T}}$ with $p \geq q$, we get

$$\mathbf{F}(\mathbf{x}_C) = \frac{1}{\mathbf{1}^{\mathrm{T}} \cdot \mathbf{M}_C \cdot \mathbf{x}_C} \cdot \mathbf{M}_C \cdot \mathbf{x}_C \quad \text{with}$$

$$\mathbf{M}_C = \begin{pmatrix} {}^{m+0}\rho & 0 & \cdots & 0 \\ {}^{m+1}\rho & {}^{m+0}\rho & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ {}^{m+p}\rho & {}^{m+p-1}\rho & \ldots & {}^{m+p-q}\rho \end{pmatrix}. \tag{15}$$

The correction matrix $\mathbf{M}_C$ given for the non-skeleton moiety can be formed as described in Note 2, which has a dimension of $(p + 1) \times (q + 1)$. Its element ${}^{m+i}\rho$ denotes the fractional abundance of the non-skeleton moiety with a mass shift of $+i$. Thus, multiplying the jth row with the vector $\mathbf{x}_C$ gives a function value that corresponds to the jth element of $\tilde{\mathbf{y}}$ with a mass shift of $+j$.

4. Formulate a constrained NLSP to get $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$ from $\mathbf{MID}_{\mathbf{MS}}$.

$$\min_{\mathbf{x}_C} f = \frac{1}{2}(\tilde{\mathbf{y}} - \mathbf{F}(\mathbf{x}_C))^T \cdot (\tilde{\mathbf{y}} - \mathbf{F}(\mathbf{x}_C))$$
$$\text{subject to} \sum_{k=1}^{q+1} \mathbf{x}_C(k) = 1 \wedge \forall k : 0 \le \mathbf{x}_C(k) \le 1 \tag{16}$$

5. Once $\mathbf{x}_C$ is estimated from the above minimization and if the number of degree of freedom is larger than zero, i.e., dim $(\tilde{\mathbf{y}}) >$ dim$(\mathbf{x}_C)$, perform statistical analysis to get the uncertainties associated with $\mathbf{x}_C$ as given in the following section.

3.2.2. Statistical Analysis

During ^{13}CMFA implementation, the statistical qualities of flux estimates are determined by the uncertainties associated with $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$ through nonlinear error propagation. Typically, measurement noise associated with MS signals is a priori unknown or even difficult to measure directly. This is because the amount of sample injection into MS is simply not reproducible, whereas the MID measurement itself is highly reproducible. Due to this, deviations in MS signals ($\mathbf{y}$) are typically large in comparison to minute deviations in its normalized data ($\tilde{\mathbf{y}}$). The measurement noise associated with MS signals is, however, required for the statistical quality assessment of $\mathbf{x}_C$. Hereto, the following steps can be implemented:

1. Compute the measurement noise σ associated with MS signals ($\mathbf{y}$) using the following approximation once $\mathbf{x}_C$ is numerically determined (see Note 3):

$$\sigma^2 = \frac{1}{\dim(\mathbf{y}) - \dim(\mathbf{x}_C)} \cdot \left[\left(\frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{y}}\right)^{-1} \cdot (\tilde{\mathbf{y}} - \mathbf{F}(\mathbf{x}_C))\right]^T \cdot \left[\left(\frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{y}}\right)^{-1} \cdot (\tilde{\mathbf{y}} - \mathbf{F}(\mathbf{x}_C))\right] \tag{17}$$

where

$$\frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{y}} = \frac{1}{\left(\mathbf{1}^T \cdot \mathbf{y}\right)^2}\left[\frac{\partial \mathbf{y}}{\partial \mathbf{y}} \cdot \left(\mathbf{1}^T \cdot \mathbf{y}\right) - \mathbf{y} \cdot \frac{\partial\left(\mathbf{1}^T \cdot \mathbf{y}\right)}{\partial \mathbf{y}}\right]. \tag{18}$$

2. Prepare the measurement covariance matrix $\mathbf{\Sigma}_\mathbf{y}$ by considering that MS signals ($\mathbf{y}$) are independent from each other and assuming that signals have a unit variance:

$$\mathbf{\Sigma}_\mathbf{y} = \begin{pmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{pmatrix} \tag{19}$$

The covariance matrix has a dimension of dim$(\mathbf{y}) \times$ dim$(\mathbf{y})$ with all off-diagonal elements zeros.

3. Compute the covariance matrix given for the normalized MID ($\tilde{\mathbf{y}}$) from $\mathbf{\Sigma_y}$ based on the Gaussian error propagation:

$$\Sigma_{\tilde{y}} = \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{y}} \cdot \Sigma_y \cdot \left(\frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{y}}\right)^T \tag{20}$$

Note that the normalization of $\mathbf{y}$ introduces dependencies between entries of $\tilde{\mathbf{y}}$. Accordingly, there are non-zero off-diagonal elements in $\mathbf{\Sigma}_{\tilde{\mathbf{y}}}$.

4. Finally, compute the covariance given for $\mathbf{x}_C$ estimates by linearizing the regression model of (Eq. 14) and considering the equality constraint given for $\mathbf{x}_C$ in (Eq. 16), i.e., $\mathbf{x}_C = \{\mathbf{x}_C \in \Re^{n=\dim(\mathbf{x}_C)} | \Sigma_{k=1}^{n} \mathbf{x}_C(k) = 1\}$:

$$\Sigma_{x_C} = \mathbf{T}_\Sigma \cdot \left[\left(\frac{\partial \mathbf{F}(\mathbf{x_C})}{\partial \mathbf{x_C}} \cdot \mathbf{T}_\Sigma \right)^T \cdot \Sigma_{\tilde{y}}^{-1} \cdot \frac{\partial \mathbf{F}(\mathbf{x_C})}{\partial \mathbf{x_C}} \cdot \mathbf{T}_\Sigma \right]^{-1} \cdot (\mathbf{T}_\Sigma)^T \tag{21}$$

Here, the matrix $\mathbf{T}_\sum$ is a linear mapping between $\mathbf{x}_C = (x_1, x_2, \ldots, x_{n-1}, x_n)^T$ and $\mathbf{z}_C = (x_1, x_2, \ldots, x_{n-1})^T$ such that $\mathbf{x}_C = \mathbf{T}_\sum \cdot \mathbf{z}_C + \mathbf{c}$ with $\mathbf{c} = (0, 0, \ldots 0, 1)^T$ or, equivalently, $\mathbf{T}_\sum = \partial \mathbf{x}_C / \partial \mathbf{z}_C$. Note that only the nth entry of $\mathbf{c}$ is 1 and entries other than the nth are zeros, and $\dim(\mathbf{z}_C) + 1$ equals $\dim(\mathbf{x}_C)$ and $\dim(\mathbf{c})$. The matrix $\mathbf{T}_\sum$ only contains 0, 1, and −1 (see Note 3). The diagonal elements of $\mathbf{\Sigma_{xC}}$ is the variance given for $\mathbf{x}_C$.

Using the modeling and statistical approaches introduced in this section, one can obtain the discrete mass isotopomer distributions given for the carbon skeleton labeling ($\mathbf{MID}_{\mathbf{SIM}}^{\mathbf{C}}$) including their statistical qualities. These quantities obtained for different metabolites are further applied to in vivo flux estimation along with efflux measurements.

3.3. Parameterization of Stoichiometric System

Prior to the numerical flux estimation, independent flux variables ($\mathbf{\Theta}$, free fluxes) have to be identified from the overall stoichiometry of (Eq. 4) to formulate the NLSP given by (Eq. 13). There are two approaches to the identification of $\mathbf{\Theta}$, that is, based on (1) the reduced row echelon form of the stoichiometric matrix or (2) the general solution of a linear system.

3.3.1. Parameterization using Reduced Row Echelon Form

The reduced row echelon form of a matrix can be obtained using the Gauss-Jordan elimination and partial pivoting. The following procedure can be implemented for the parametrization using the reduced row echelon form.

1. Compute the reduced row echelon form of the stoichiometric matrix $\mathbf{S}$ in (Eq. 4), i.e., $\mathbf{S}_{\text{rref}}$, and identify which columns of

$\mathbf{S}_{\text{rref}}$ contain only one non-zero entry of which value is 1, called leading ones, e.g., $(1, 0, 0, 0,\ldots, 0)^{\text{T}}$ or $(0, 1, 0, 0,\ldots, 0)^{\text{T}}$.

2. Rearrange these columns such that $\mathbf{S}_{\text{rref}} = [\mathbf{I}, \mathbf{A}]$ and the corresponding flux variables, where $\mathbf{I}$ denotes the identity matrix and $\mathbf{A}$ the matrix consisting of remaining columns:

$$\mathbf{S}_{\text{rref}} \cdot v = \mathbf{0} \Rightarrow (\mathbf{I} \quad \mathbf{A}) \begin{pmatrix} v_{\text{depend}} \\ \Theta \end{pmatrix} = \mathbf{0} \tag{22}$$

3. As a consequence of $\mathbf{S}_{\text{rref}}$ and flux vector rearrangement, the following explicit expression can be obtained for (Eq. 4).

$$v_{\text{depend}} = \mathbf{n}(\Theta) = -\mathbf{A} \cdot \Theta \tag{23}$$

3.3.2. Parameterization from the General Solution of a Linear System

A linear system can be decomposed into its particular solution and the so-called homogenous part, e.g., the general solution for $\mathbf{A}\cdot\mathbf{x} = \mathbf{b}$ equals $\mathbf{x} = (\mathbf{A}^{\text{T}}\cdot\mathbf{A})^{-1}\cdot\mathbf{A}^{\text{T}}\cdot\text{b} + \text{null}(\mathbf{A})\ \boldsymbol{\beta}$ with $\boldsymbol{\beta}$ being a vector with arbitrary non-zero values. The first term is the particular solution, while the second term the homogenous part with null ($\mathbf{A}$) being the null space of the system matrix $\mathbf{A}$. For an underdetermined system, only the entries of $\mathbf{x}$ corresponding to the empty null space, i.e., zero entries in the corresponding rows of null($\mathbf{A}$) result in unique solutions from $\mathbf{b}$. This theorem can also be utilized for the parameterization of (Eq. 4).

1. Decompose the stoichiometric system of (Eq. 4) into intracellular and extracellular parts, i.e.,

$$(\mathbf{S}_{\text{intra}} \quad \mathbf{S}_{\text{extra}}) \cdot \begin{pmatrix} v_{\text{intra}} \\ v_{\text{extra}} \end{pmatrix} = \mathbf{0} \text{ or } \mathbf{S}_{\text{intra}} \cdot v_{\text{intra}} = -\mathbf{S}_{\text{extra}} \cdot v_{\text{extra}}. \tag{24}$$

Here, $\boldsymbol{\nu}_{\text{intra}}$ and $\boldsymbol{\nu}_{\text{extra}}$ are the intracellular and extracellular fluxes with corresponding stoichiometry $\mathbf{S}_{\text{intra}}$ and $\mathbf{S}_{\text{extra}}$, respectively. $\boldsymbol{\nu}_{\text{extra}}$ equals the effluxes and typically measurable.

2. Compute the null space of $\mathbf{S}_{\text{intra}}$ to obtain the general solution of (Eq. 24) and identify the flux variables in $\boldsymbol{\nu}_{\text{intra}}$ corresponding to the non-empty rows in the null space.

$$\begin{aligned} v_{\text{intra}} = &-(\mathbf{S}_{\text{intra}}{}^{\text{T}} \cdot \mathbf{S}_{\text{intra}})^{-1} \cdot \mathbf{S}_{\text{intra}}{}^{\text{T}} \cdot \mathbf{S}_{\text{extra}} \cdot v_{\text{extra}} \\ &+ \text{null}\,(\mathbf{S}_{\text{intra}}) \cdot \beta \end{aligned} \tag{25}$$

3. Together with the effluxes in $\boldsymbol{\nu}_{\text{extra}}$, the intracellular fluxes corresponding to the non-empty null space are independent variables ($\boldsymbol{\Theta}$), while other intracellular fluxes corresponding to empty null space are dependent variables ($\boldsymbol{\nu}_{\text{depend}}$). Similarly to (Eq. 22), (Eq. 24) can be rewritten as follows:

$$\mathbf{S} \cdot v = \mathbf{0} \Rightarrow (\mathbf{S}_1 \quad \mathbf{S}_2) \cdot \begin{pmatrix} v_{\text{depend}} \\ \Theta \end{pmatrix} = \mathbf{0}. \tag{26}$$

4. Now, solving the above equation for $\boldsymbol{\nu}_{\text{depend}}$ gives the following explicit expression given for the stoichiometric reaction network:

$$v_{\text{depend}} = \mathbf{n}(\Theta) = -\left(\mathbf{S}_1{}^{\text{T}} \cdot \mathbf{S}_1\right)^{-1} \cdot \mathbf{S}_1{}^{\text{T}} \cdot \mathbf{S}_2 \cdot \Theta. \tag{27}$$

Using either of the above approaches of (Eq. 23) or (Eq. 27), one can generate any flux states that obey the stoichiometry of $\mathbf{S}\cdot\boldsymbol{\nu} = \mathbf{0}$ during a numerical optimization process iterating $\boldsymbol{\Theta}$. Subsequently, any steady-state or non-steady-state ^{13}C-labeling values of metabolites ($\mathbf{x}$) can be computed by solving (Eq. 10) for a certain ^{13}C-substrate labeling ($\mathbf{x}_0$) at a given flux state.

3.4. Computation of Inequality Constraints

The free fluxes ($\boldsymbol{\Theta}$) identified by the above parametrization are the design parameters in (Eq. 13) which are numerically determined from ^{13}C-labeling and efflux measurements. During the iterative optimization process, the free fluxes are physically and physiologically constrained such that $\boldsymbol{\Theta} = \{\boldsymbol{\Theta} \in \Re^{\dim(\boldsymbol{\Theta})} \mid \boldsymbol{\Theta} \geq \mathbf{0} \wedge \boldsymbol{\nu}_{\text{depend}} = \mathbf{n}(\boldsymbol{\Theta}) \geq \mathbf{0}\}$ and act as the inequality constraints in the Lagrangian function of (Eq. 47). However, $\mathbf{n}(\boldsymbol{\Theta}) \geq \mathbf{0}$ does not always hold for all $\boldsymbol{\Theta} \geq \mathbf{0}$. Thus, one may want to define the range of the free fluxes such that $\mathbf{LB} \leq \boldsymbol{\Theta} \leq \mathbf{UB}$, which holds $\mathbf{n}(\boldsymbol{\Theta}) \geq \mathbf{0}$. Moreover, the range given for $\mathbf{n}(\boldsymbol{\Theta})$ depends on the confidence intervals of experimentally measured effluxes. Thus, the feasible range of $\mathbf{n}(\boldsymbol{\Theta})$ can be narrowed down such that and $\mathbf{a} \leq \mathbf{n}(\boldsymbol{\Theta}) \leq \mathbf{b}$ by considering experimental measurement precision. This can be done based on the so-called flux variability analysis, which can be formulated as a constraint-based linear programming problem (61, 62). This is implemented as follows:

1. Compute the confidence intervals given for the effluxes measured from ^{13}C-cultures such that:

$$\max\left(\mathbf{0}, v_{\text{extra}} - \sigma_{\text{extra}}\sqrt{\chi^2_{1,\alpha}}\right) \leq v_{\text{extra}} \leq v_{\text{extra}} + \sigma_{\text{extra}}\sqrt{\chi^2_{1,\alpha}}, \tag{28}$$

where $\boldsymbol{\nu}_{\text{extra}}$ denotes the measured effluxes, $\boldsymbol{\sigma}_{\text{extra}}$ the measurement errors, and χ^2_1,α the inverse of χ^2-cumulative distribution function value at a certain confidence level of α. Thus, the range covers $100 \times \alpha$ % of possible experimental observations.

2. Formulate two separate linear programming problems to estimate upper and lower bounds given for each flux variable, i.e.,

$$\min_v \omega^{\mathrm{T}} \cdot v \;\&\; \min_v -\omega^{\mathrm{T}} \cdot v \;\text{ subject to}\ldots$$
$$\mathbf{S} \cdot v = \mathbf{0}$$
$$\mathbf{0} \le v < \infty \text{ or a large finite number}$$
$$\max\left(\mathbf{0}, v_{\mathrm{extra}} - \sigma_{\mathrm{extra}}\sqrt{\chi^2_{1,\alpha}}\right) \le v_{\mathrm{extra}} \le v_{\mathrm{extra}} + \sigma_{\mathrm{extra}}\sqrt{\chi^2_{1,\alpha}} \tag{29}$$

3. Repeat the above optimization $\dim(\boldsymbol{\nu}) \times$ times by setting $\boldsymbol{\omega}(k) = 1$ with other entries of $\boldsymbol{\omega}$ being zeros at the kth repetition. By doing so, the minimization of $\boldsymbol{\omega}^{\mathrm{T}} \cdot \boldsymbol{\nu}$ and $-\boldsymbol{\omega}^{\mathrm{T}} \cdot \boldsymbol{\nu}$ gives the lower and upper bound for the kth flux variable, respectively.

If respiratory parameters such as carbon dioxide evolution, oxygen uptake (or terminal electron acceptor reduction for anaerobic cultures), and respiratory quotient (RQ) are measured, these measurements can also be utilized as the constraints of (Eq. 29) to further narrow down the flux range. For instance, one mole of oxygen is reduced by two moles of reducing equivalents, e.g., $\mathrm{NAD(P)H} + \tfrac{1}{2}\mathrm{O_2} + \mathrm{H^+} \rightarrow \mathrm{NAD(P)^+} + \mathrm{H_2O}$, and we get the following aerobic redox balance (52):

$$v_{\mathrm{oxygen}} - \frac{1}{2}\left(\sum_{j=1}^{p} v_{\mathrm{ox,j}} - \sum_{k=1}^{q} v_{\mathrm{red,k}}\right) = 0. \tag{30}$$

Here, v_{oxygen} denotes the oxygen uptake flux, v_{ox} the oxidative fluxes donating electrons to reducing equivalents, and v_{red} the reductive fluxes accepting electrons from reducing equivalents. Since RQ equals the ratio from $\mathrm{CO_2}$ production per $\mathrm{O_2}$ uptake, one can set up the following linear equation by introducing r decarboxylation (v_{decarb}) and s carboxylation (v_{carb}) reactions.

$$\left(\sum_{h=1}^{r} v_{\mathrm{decarb,h}} - \sum_{i=1}^{s} v_{\mathrm{carb,i}}\right) - \frac{1}{2}\left(\sum_{j=1}^{p} v_{\mathrm{ox,j}} - \sum_{k=1}^{q} v_{\mathrm{red,k}}\right) \cdot \mathrm{RQ} = 0 \tag{31}$$

In practice, the above equation can be defined as an inequality by considering the measurement uncertainty given for RQ, e.g., (net decarboxylation) – ½ (net reducing equivalent generation) × (lower confidence bound of RQ) ≥ 0.

Without respiration measurements, one may simply rely on elemental carbon balance and/or degree of reduction balance constraining the intracellular net carboxylation and net reducing equivalent generation fluxes, respectively (62).

3.5. Implementation of Numerical Flux Estimation

Once the feasible flux space is determined through the above flux variability analysis, the numerical flux estimation can be implemented using ^{13}C-labeling data and effluxes obtained from a ^{13}C labeling experiment. The numerical flux estimation is a constrained NLSP as defined in (Eq. 13) and can be performed using a gradient-based algorithm such as SQP. In this section, the overall numerical optimization procedure will be addressed in conjunction with the method of Monte Carlo using mass isotopomer distribution data.

1. Prepare the measurement pseudo-populations for mass isotopomer distribution data ($\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$) and effluxes, i.e., their mean and standard deviation estimates.
2. Prepare the tracer substrate ^{13}C-labeling value $\mathbf{x}_0$ (see Note 1) with the format corresponding to the model selected for ^{13}C-labeling system formulation (Subheading 1.5). Isotopomer-, cumomer-, or EMU-based model can be applied for $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$.
3. Generate random numbers from the measurement pseudo-populations by assuming the normal distribution. The random numbers for the $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$ of each species ($\mathbf{x}_C$) need to be generated by holding the constraint of $\mathbf{x}_C = \{\mathbf{x}_C \in \Re^{n=\dim(\mathbf{x}_C)} | \Sigma^n_{k=1}\mathbf{x}_C(k) = 1 \wedge \forall k: \ 0 \ \leq \ \mathbf{x}_C(k) \ \leq \ 1\}$.
4. Generate initial guess for $\boldsymbol{\Theta}$, e.g., using a random number generator with the uniform distribution, subject to $\{\boldsymbol{\Theta} \in \Re^{\dim}(\boldsymbol{\Theta}) \ \mid \ \mathbf{LB} \leq \boldsymbol{\Theta} \leq \mathbf{UB} \ \wedge \ \mathbf{a} \leq \boldsymbol{\nu}_{\text{depend}} = \mathbf{n}(\boldsymbol{\Theta}) \leq \mathbf{b}\}$. Optionally, the free flux parameters can be compactified in order to increase optimization sensitivity while solving the constrained NLSP (see Note 6).
5. Set termination criteria for the optimization process: one can select small positive numbers for the termination tolerance on $\Delta\boldsymbol{\Theta}$ and objective function value. An approach with dynamic tolerance adjustment that can be applied to the NLSP with noisy measurement data is addressed in Note 7.
6. Initiate the constrained NLSP given by (Eq. 13) using a chosen algorithm such as SQP. During this iterative process, **steps 7–10** is repeated.
7. Evaluate the current flux values $\boldsymbol{\nu}_k$ at the current iterate $\boldsymbol{\Theta}_k$ from either (Eq. 23) or (Eq. 27). Note that $\boldsymbol{\nu}_k = [\mathbf{n}(\boldsymbol{\Theta}_k), \boldsymbol{\Theta}_k]$, where $\mathbf{n}(\boldsymbol{\Theta}_k)$ and $\boldsymbol{\Theta}_k$ the dependent and independent fluxes, respectively.
8. Evaluate the current ^{13}C-labeling state $\mathbf{x}_k$ corresponding to $\boldsymbol{\nu}_k$ and $\mathbf{x}_0$, e.g., by solving (Eq. 9), and get the model values

of $\mathbf{x}_{\mathrm{mdv}}$ and $\boldsymbol{\nu}_{\mathrm{efflux}}$ corresponding to the experimental $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$ and effluxes, respectively. Also, compute key partial derivatives to provide the analytical gradients involved in (Eq. 51) for an efficient optimization.

9. Evaluate the objective function value at $\boldsymbol{\Theta}_k$, that is, the covariance-weighted Euclidean distance between the model values and the experimental data given by (Eq. 13). Also, evaluate inequality function values, i.e., $\mathbf{a} - \mathbf{n}(\boldsymbol{\Theta}_k) \leq \mathbf{0}$ and $\mathbf{n}(\boldsymbol{\Theta}_k) - \mathbf{b} \leq \mathbf{0}$.
10. Compute the next iterate $\boldsymbol{\Theta}_{k+1}$ according to the optimization algorithm chosen.
11. Stop the iterative step from 7 to 10 if the termination criterion defined in step 5 is satisfied. Store the flux estimates at the termination.
12. Repeat **steps 3–11** for N trials to get N flux estimates, where at least $N \geq 50$ is typically recommended.
13. Evaluate statistical properties of the resulting flux pseudo-populations to get the confidence intervals given for the fluxes.

Due to the stochastic nature of the above approach, involving random numbers of experimental data (step 3) and random starting points (step 4), not only the nonlinear error propagation from noisy experimental data to estimated fluxes but also the starting-point-dependency intrinsic to any gradient-based algorithms can be counted.

3.5.1. Evaluation of Key Partial Derivatives

Typically, a gradient-based nonlinear optimization problem is solved more accurately and efficiently when analytical gradients are provided compared to the case relying on finite-difference approximation. To supply the analytical gradients involved in (Eq. 51) at step 8, the partial derivatives of the measurement model $[\mathbf{x}_{\mathrm{mdv}}, \boldsymbol{\nu}_{\mathrm{efflux}}] = \mathbf{F}(\boldsymbol{\Theta})$ in (Eq. 13) are required along with those of the inequality constraints $\mathbf{C} = \mathbf{a} - \mathbf{n}(\boldsymbol{\Theta}) \leq \mathbf{0} \wedge \mathbf{n}(\boldsymbol{\Theta}) - \mathbf{b} \leq \mathbf{0}$, i.e.,

$$\frac{\partial \mathbf{F}(\boldsymbol{\Theta}_k)}{\partial \boldsymbol{\Theta}} = \begin{pmatrix} \frac{\partial \mathbf{x}_{\mathrm{mdv}}(\boldsymbol{\Theta}_k)}{\partial \boldsymbol{\Theta}} \\ \frac{\partial \nu_{\mathrm{efflux}}(\boldsymbol{\Theta}_k)}{\partial \boldsymbol{\Theta}} \end{pmatrix} \text{ for model and } \frac{\partial \mathbf{C}(\boldsymbol{\Theta}_{\mathrm{k}})}{\partial \boldsymbol{\Theta}} = \begin{pmatrix} -\frac{\partial \mathbf{n}(\boldsymbol{\Theta}_{\mathrm{k}})}{\partial \boldsymbol{\Theta}} \\ +\frac{\partial \mathbf{n}(\boldsymbol{\Theta}_{\mathrm{k}})}{\partial \boldsymbol{\Theta}} \end{pmatrix} \text{ for } \mathbf{C} \leq \mathbf{0}. \tag{32}$$

Of course, $\mathbf{x}_{\mathrm{mdv}}$ is obtained after calculating $\mathbf{x}_k$ that is the ^{13}C-labeling state of the entire species defined in the model at $\boldsymbol{\Theta}_k$, and the following steps can be conducted to get the partial derivatives in (Eq. 32):

1. If required, define a matrix that transforms $\mathbf{x}_k$ into the format corresponding to $\mathbf{x}_{\mathrm{mdv}}$, e.g., isotopomer distribution (or cumomer fraction) into mass isotopomer distribution by $\mathbf{x}^*_k = \mathbf{T}_1 \cdot \mathbf{x}_k$.
2. Define mapping between $\mathbf{x}_{\mathrm{mdv}}$ and $\mathbf{x}^*_k$. e.g., using a linear mapping matrix such that $\mathbf{x}_{\mathrm{mdv}} = \mathbf{T}_2 \cdot \mathbf{x}^*_k = \mathbf{T}_2 \cdot \mathbf{T}_1 \cdot \mathbf{x}_k$, i.e., selecting measured species out of the entire species defined in $\mathbf{x}_k$.
3. Evaluate $\partial \mathbf{x}_{\mathrm{mdv}} / \partial \boldsymbol{\Theta}$ such that $\partial \mathbf{x}_{\mathrm{mdv}} / \partial \boldsymbol{\Theta} = \mathbf{T} \cdot \partial \mathbf{x}_k / \partial \boldsymbol{\Theta}$ with $\mathbf{T} = \mathbf{T}_2 \cdot \mathbf{T}_1$. $\partial \mathbf{x}_k / \partial \boldsymbol{\Theta}$ is obtained from (Eq. 11).
4. Since effluxes ($\boldsymbol{\nu}_{\mathrm{efflux}}$) are a part of $\boldsymbol{\Theta}$, $\partial \boldsymbol{\nu}_{\mathrm{efflux}} / \partial \boldsymbol{\Theta}$ is a simple matrix containing only 0 and 1, that is, injective mapping of $\boldsymbol{\nu}_{\mathrm{efflux}}$ to $\boldsymbol{\Theta}$.
5. Since $\mathbf{n}(\boldsymbol{\Theta})$ is a linear system as given by (Eq. 23) or (Eq. 27), $\partial \mathbf{n}(\boldsymbol{\Theta}) / \partial \boldsymbol{\Theta}$ equals $-$A for (Eq. 23) and $-(\mathbf{S}_1{}^{\mathrm{T}} \cdot \mathbf{S}_1)^{-1} \cdot \mathbf{S}_1{}^{\mathrm{T}} \cdot \mathbf{S}_2$ for (Eq. 27).

3.5.2. Correction of Computational Round-Off Errors

Another thing to be noted is that one may consider computational round-off errors which can result in violation of $\mathbf{x}_{\mathrm{C}} = \{\mathbf{x}_{\mathrm{C}} \in \Re^{n=\dim(\mathbf{x}_{\mathrm{C}})} | \ \Sigma_{k=1}^{n} \mathbf{x}_{\mathrm{C}}(k) = 1 \wedge \forall k: \ 0 \leq \mathbf{x}_{\mathrm{C}}(k) \leq 1\}$ in **steps 3** and **8** during the constrained NLSP implementation. Also, the same violation can occur when generating values for $\mathbf{x}_0$ (see Note 1) or correction matrix (see Note 2). Depending on the floating-point relative accuracy (eps), one may have $\Sigma_{k=1}^{n} \mathbf{x}_{\mathrm{C}}(k) \approx 1 \pm \mathrm{eps}$ or $\mathbf{x}_{\mathrm{C}}(\mathrm{k}) \approx -\mathrm{eps}$. In practice, the actual bias may be a few orders of magnitude larger than the true eps, e.g., due to the error propagation associated with other mathematical operations such as matrix inversion. However, one may conduct the bias correction as follows:

1. Determine the actual bias (eps*), e.g., using a series of simulation studies under different flux scenarios.
2. Check $\mathbf{x}_k$ while evaluating the ^{13}C-labeling system at $\boldsymbol{\Theta}_k$. If isotopomer or EMU model is applied, the criterion of $\mathbf{x}_j = \{\mathbf{x}_j \in \Re^{n=\dim(\mathbf{x}_j)} | \Sigma_{i=1}^{n} \mathbf{x}_j(i) = 1 \wedge \forall i: \ 0 \leq \mathbf{x}_j(i) \leq 1\}$ has to be satisfied for each jth component, where $\mathbf{x}_j \in \mathbf{x}_k$, $j = 1, 2, 3, \ldots p$, and p the number of components (metabolites or EMUs) in the model.
3. Detect any $\mathbf{x}_j(i) \approx -\mathrm{eps}^*$, and set those to be zeros. Subsequently, detect $1 - \Sigma_{i=1}^{n} \mathbf{x}_j(i) \approx \pm \mathrm{eps}^*$, add or subtract eps* from the largest value of $\mathbf{x}_j$. This operation can be conducted while evaluating each level of the cascade system such as (Eq. 9) if EMU is applied.

The same correction procedure can be employed when generating substrate ^{13}C-labeling values ($\mathbf{x}_0$) and random numbers of $\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}(\mathbf{x}_{\mathrm{C}})$ as well as during mass isotopomer data processing discussed in Subheading 3.2.

A similar computational round-off error can also occur when generating fluxes by $\boldsymbol{\nu}_{\text{depend}} = \mathbf{n}(\boldsymbol{\Theta}_k)$ during optimization runs, which results in $\mathbf{S}\cdot\boldsymbol{\nu} = \mathbf{0} \pm \mathbf{eps}^*$. In this case, one may set up a bias model such that $\mathbf{S}\cdot(\boldsymbol{\nu} - \mathbf{b}) = \mathbf{0}$ and minimize $(\mathbf{S}(\boldsymbol{\nu} - \mathbf{b}))^{\mathrm{T}}\cdot(\mathbf{S}(\boldsymbol{\nu} - \mathbf{b}))$ with respect to $\mathbf{b}$ subject to $(\boldsymbol{\nu} - \mathbf{b}) \geq \mathbf{0}$ and $-\mathbf{eps}^* \leq \mathbf{b} \leq \mathbf{eps}^*$.

4. Notes

Here, a few mathematical tips are provided that are useful for computational modeling of mass isotopomer data processing and numerical ^{13}CMFA implementation.

1. *Substrate ^{13}C-Labeling*

   ^{13}C-sustances purchased from chemical manufacturers are typically defined by its ^{13}C-position(s) and atom-% purity. For instance, [1,6-$^{13}C_2$]glucose with 99 atom-% purity means that C1 and C6 carbons are 99% ^{13}C-labeled and other positions have natural ^{13}C-abundance (1.07%). Based on this fact, one can generate carbon isotopomer distributions. For instance, the abundance of an isotopomer for [1,6-$^{13}C_2$] glucose ^{12}C–^{12}C–^{12}C–^{12}C–^{12}C–^{12}C equals $0.01 \times (1 - 0.0107)^4 \times 0.01$, ^{13}C–^{12}C–^{12}C–^{12}C–^{12}C–^{12}C $0.99\times(1 - 0.0107)^4 \times 0.01$, ^{13}C–^{13}C–^{12}C–^{12}C–^{12}C–^{12}C $0.99 \times 0.0107 \times (1 - 0.0107)^3 \times 0.01$, ^{13}C–^{12}C–^{12}C–^{12}C–^{12}C–^{13}C, $0.99 \times (1 - 0.0107)^4 \times 0.99$, ^{13}C–^{13}C–^{13}C–^{13}C–^{13}C–^{13}C $0.99 \times 0.0107^4 \times 0.99$, etc. For [U–$^{13}C_6$] glucose with 99 atom-%, the abundance ^{13}C–^{13}C–^{13}C–^{13}C–^{13}C–^{13}C is not 0.99 but $0.99^6 = 0.94$.

   It is also trivial to prepare ^{13}C-labeling values for any substrate EMUs. For instance, the MID of C1–C2 EMU for the above [1,6-$^{13}C_2$] glucose can be obtained by the polynomial convolution of the MID of C1 $(0.01, 0.99)^{\mathrm{T}}$ with that of C2 $(1 - 0.0107,\ 0.0107)^{\mathrm{T}}$, resulting in $(0.0099, 0.9795, 0.0106)^{\mathrm{T}}$. Now, C1–C2–C3 can be obtained by the convolution of the MID of C1–C2–C3 EMU with that of C3, i.e., convolution of $(0.0099, 0.9795, 0.0106)^{\mathrm{T}}$ with $(1 - 0.0107, 0.0107)^{\mathrm{T}}$.

2. *Polynomial Convolution and Mass Isotopomer Correction Matrix*

   The *k*th entry of a vector resulting from the polynomial convolution of two vectors **a•b** is defined such that

$$\mathbf{a} \bullet \mathbf{b}(k) = \sum_j \mathbf{a}_j \cdot \mathbf{b}_{k+1-j}, \tag{33}$$

where the index j comprises all integers satisfying $1 \leq j \leq \dim(\mathbf{a})$ and $1 \leq k + 1 - j \leq \dim(\mathbf{b})$. Thus, the dimension of $\mathbf{a}\cdot\mathbf{b}$ equals $\dim(\mathbf{a}) + \dim(\mathbf{b}) - 1$.

Using polynomial convolution, the isotopic mass shift effect due to the naturally occurring isotopes in the non-skeleton moiety $C_\alpha H_\beta O_\delta N_\gamma$ can be computed, i.e., computation of the MID given for $C_\alpha H_\beta O_\delta N_\gamma$. Note that the carbons in the non-skeleton moiety is supposed to be from a chemical derivatization agent for GC/MS. For convenience, let the vector $\mathbf{a}_X$ the isotope abundance of an element X and $\boldsymbol{\rho}_{X_pY_q}$ the MID of an X_pY_q-moiety. For instance, we start with the non-skeleton carbon whose fractional isotopic abundance $\mathbf{a}_C = (a, b)^T$ corresponds to ^{12}C and ^{13}C. One can initiate the procedure by computing the MID for C_2 moiety as follows:

$$\rho_{C_2} = \mathbf{a}_C \bullet \mathbf{a}_C = \begin{pmatrix} {}^{m+0}\rho_C \\ {}^{m+1}\rho_C \\ {}^{m+2}\rho_C \end{pmatrix} = \begin{pmatrix} a^2 \\ 2ab \\ b^2 \end{pmatrix}. \tag{34}$$

As shown, the entries of the above vector equal the binomial series for two possibilities. Further, the convolution of $\boldsymbol{\rho}_{C2}$ with $\mathbf{a}_C$ gives the MID of a C_3-moiety, $\boldsymbol{\rho}_{C3}$. Hence, by repeating the above $(\alpha - 1)$-times, we get the vector $\boldsymbol{\rho}_{C\alpha} = ({}^{m+0}\rho_C, {}^{m+1}\rho_C, \ldots, {}^{m+\alpha}\rho_C)^T$ that represents the MID of C_α-moiety. The same procedure can now be applied to get $\boldsymbol{\rho}_{C\alpha H\beta}$ for $C_\alpha H_\beta$-moiety. The loop restarts with computing the convolution of $\boldsymbol{\rho}_{C\alpha}$ with the fractional isotopic abundance vector of hydrogen, $\mathbf{a}_H = (a, b)^T$ of which entries correspond to ^{1}H and ^{2}H.

$$\begin{pmatrix} {}^{m+0}\rho_H \\ {}^{m+1}\rho_H \\ \vdots \\ {}^{m+\alpha}\rho_H \end{pmatrix} \bullet \mathbf{a}_H = \begin{pmatrix} {}^{m+0}\rho_H \cdot a \\ {}^{m+0}\rho_H \cdot b + {}^{m+1}\rho_H \cdot a \\ \vdots \\ {}^{m+\alpha}\rho_H \cdot b \end{pmatrix} \tag{35}$$

Repeating this calculation β-times, we get $\boldsymbol{\rho}_{C\alpha H\beta}$. Analogously, using $\boldsymbol{\rho}_{C\alpha H\beta}$ and $\mathbf{a}_O$ for oxygen and further $\mathbf{a}_N$ for nitrogen, we get the complete MID given for the non-skeleton moiety, i.e., $\boldsymbol{\rho}_{C\alpha H\beta O\delta N\gamma} = ({}^{m+0}\rho_{CHON}, {}^{m+1}\rho_{CHON}, \ldots, {}^{m+\alpha+\beta+2\delta+\gamma}\rho_{CHON})^T$.

For the compound with the non-skeleton moiety $C_\alpha H_\beta O_\delta N_\gamma$, the MID of its skeleton ($\mathbf{MID}^{\mathbf{C}}_{\mathbf{SIM}}$) and the corresponding measurement ($\mathbf{MID}_{\mathbf{MS}}$) satisfy the following relationship:

$$\mathbf{MID}_{\mathbf{MS}} = \begin{pmatrix} {}^{m+0}\rho_{CHON} \\ {}^{m+1}\rho_{CHON} \\ \vdots \\ {}^{m+\alpha+\beta+2\delta+\gamma}\rho_{CHON} \end{pmatrix} \bullet \mathbf{MID}^{\mathbf{SIM}}_{\mathbf{C}}. \tag{36}$$

The above convolution can also be formulated as a matrix–vector product, and the matrix is often referred as the correction matrix:

$$\mathbf{MID_{MS}} = \begin{pmatrix} {}^{m+0}\rho_{\mathrm{CHON}} & 0 & \cdots & 0 \\ {}^{m+1}\rho_{\mathrm{CHON}} & {}^{m+0}\rho_{\mathrm{CHON}} & \ddots & 0 \\ {}^{m+2}\rho_{\mathrm{CHON}} & {}^{m+1}\rho_{\mathrm{CHON}} & \ddots & \vdots \\ \vdots & \vdots & \ddots & {}^{m+0}\rho_{\mathrm{CHON}} \\ \vdots & \vdots & \ddots & \vdots \\ {}^{m+\alpha+\beta+2\delta+\gamma}\rho_{\mathrm{CHON}} & {}^{m+\alpha+\beta+2\delta+\gamma-1}\rho_{\mathrm{CHON}} & \cdots & {}^{m+\alpha+\beta+2\delta+\gamma-n}\rho_{\mathrm{CHON}} \end{pmatrix} \cdot \mathbf{MID}_{\mathbf{C}}^{\mathbf{SIM}}. \tag{37}$$

3. *Statistical Modeling of MID Analysis*

In Subheading 3.2, we addressed how to estimate $\mathbf{MID}_{\mathbf{SIM}}^{\mathbf{C}}$ and its statistical qualities from normalized MS signals in conjunction with a constrained NLSP. A generalized nonlinear regression model such as

$$\eta = \mathbf{F}(\Theta) + \varepsilon \tag{38}$$

can be expressed as a minimization problem such that

$$\min_{\Theta} (\eta - \mathbf{F}(\Theta))^{\mathrm{T}} \cdot \Sigma \cdot (\eta - \mathbf{F}(\Theta)), \tag{39}$$

where the measurement error ε is typically assumed to be normally distributed with expectation of $\varepsilon \in \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. If the measurement error is a priori unknown or not available, the typical approach is to solve the minimization problem without covariance and then estimate the measurement variance at the solution (37, 63):

$$\sigma^2 = \frac{(\eta - \mathbf{F}(\Theta))^{\mathrm{T}} \cdot (\eta - \mathbf{F}(\Theta))}{\dim(\eta) - \dim(\Theta)} \tag{40}$$

The above error approximation works fairly well for any unbiased systems generating independent signals with random noise.

In the case of the MID analysis which relies on normalized MS signals, the above approach fails because the normalization introduces mathematical dependencies between entries in $\boldsymbol{\eta}$. Thus, the data normalization needs to be taken into account in terms of the Gaussian error propagation, as shown in (Eq. 20). Hence, the objective function in (Eq. 16) with the covariance given for the normalized MID data, which is

$$\min_{\mathbf{x}_{\mathrm{C}}} (\tilde{\mathbf{y}} - \mathbf{F}(\mathbf{x}_{\mathrm{C}}))^{\mathrm{T}} \cdot \Sigma_{\tilde{\mathbf{y}}}{}^{-1} \cdot (\tilde{\mathbf{y}} - \mathbf{F}(\mathbf{x}_{\mathrm{C}})) \tag{41}$$

can be transformed for the covariance given for the MS signals based on the Gaussian error propagation given in (Eq. 20):

$$\min_{\mathbf{x}_C} \left[\left(\frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{y}} \right)^{-1} \cdot (\tilde{\mathbf{y}} - \mathbf{F}(\mathbf{x}_C)) \right]^T \cdot \Sigma_y{}^{-1} \cdot \left[\left(\frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{y}} \right)^{-1} \cdot (\tilde{\mathbf{y}} - \mathbf{F}(\mathbf{x}_C)) \right]. \tag{42}$$

As a result, we get the variance estimate of (Eq. 17), analogouslyto (Eq. 39) and (Eq. 40). For a non-constrained problem such as (Eq. 39), the covariance given for the parameters $\mathbf{\Sigma}_{\mathbf{\Theta}}$ would equal the inverse of $\partial \mathbf{F}/\partial \mathbf{\Theta}^T \cdot \mathbf{\Sigma}^{-1} \cdot \partial \mathbf{F}/\partial \mathbf{\Theta}$ by linearization at the solution (57, 64). However, for the problem (Eq. 16), the model's Jacobian matrix $\partial \mathbf{F}_C/\partial \mathbf{x}_C$ is actually rank-deficient due to the normalization. The rank-deficiency is fixed by including the equality constraint in (Eq. 16), that is, the sum of $\mathbf{x}_C$ equals 1. To do this, we defined three vectors of $\mathbf{x}_C = (x_1, x_2, \ldots, x_{n-1}, x_n)^T$, $\mathbf{z}_C = (x_1, x_2, \ldots, x_{n-1})^T$ and $\mathbf{c} = (0, 0, \ldots 0, 1)^T$ and introduced a matrix $\mathbf{T}_{\sum}$ that is a linear mapping of $\mathbf{x}_C = \mathbf{T}_{\sum} \cdot \mathbf{z}_C + \mathbf{c}$ or, equivalently, $\mathbf{T}_{\sum} = \partial \mathbf{x}_C/\partial \mathbf{z}_C$. Hence, $\partial \mathbf{F}_C/\partial \mathbf{x}_C \cdot \mathbf{T}_{\sum}$ givens $\partial \mathbf{F}_C/\partial \mathbf{z}_C$ by the chain rule, which has the full rank with respect to $\mathbf{z}_C$. Now, the inverse of $\partial \mathbf{F}_C/\partial \mathbf{z}_C{}^T \cdot \mathbf{\Sigma}_{\tilde{\mathbf{y}}-z}{}^{-1} \cdot \partial \mathbf{F}_C/\partial \mathbf{z}_C$ can be computed, which gives the covariance given for $\mathbf{z}_C$ and this corresponds to $n - 1$ variables of $\mathbf{x}_C$. To get the covariance for all n variables of $\mathbf{x}_C$, we additionally need to rely on the Gaussian error propagation from $\mathbf{z}_C$ to $\mathbf{x}_C$, similarly as shown in (Eq. 20). Hence, the covariance for the parameter estimate $\mathbf{x}_C$ can be obtained analytically by computing (Eq. 21).

4. *Gradient, Hessian, and Optimal Solution*

To solve an NLSP using a gradient-based local optimization algorithms such as sequential quadratic problem (SQP, see Note 5) and trust-region-reflective algorithm, it is often computationally beneficial to provide analytical gradient and Hessian. At each kth iteration during NLSP, the model function $\mathbf{F}$ is evaluated at the kth iterate $\mathbf{\Theta}_k$ to get the values corresponding to the measurement $\boldsymbol{\eta}$. If the model function is continuous and differentiable, the gradient ∇f can be computed analytically by:

$$\nabla f(\Theta_k) = \frac{\partial f(\Theta_k)}{\partial \Theta}^T = \frac{\partial \mathbf{F}(\Theta_k)}{\partial \Theta}^T \cdot \Sigma_\eta{}^{-1} \cdot (\mathbf{F}(\Theta_k) - \eta). \tag{43}$$

For an NLSP, the Hessian matrix $\mathbf{H}$, the second-order partial derivatives of f which specifies the curvature of the search surface can be simplified as follows:

$$\mathbf{H}_k = \nabla^2 f(\Theta_k) = \frac{\partial \mathbf{F}(\Theta_k)}{\partial \Theta}^T \cdot \Sigma_\eta{}^{-1} \cdot \frac{\partial \mathbf{F}(\Theta_k)}{\partial \Theta}^T. \tag{44}$$

Rigorously, there is an additional term in the above equation, which requires the second-order derivatives of $\mathbf{F}$, but this

term can be neglected because of near-linearity of the model in the vicinity the solution or small residuals. (36) (37).

Using the Taylor first-order expansion, the model equation $\mathbf{F}$ can be linearized at the kth iterate $\mathbf{\Theta}_k$ as follows:

$$\mathbf{F}(\Theta_{k+1}) = \mathbf{F}(\Theta_k) + \frac{\partial \mathbf{F}(\Theta_k)}{\partial \Theta} \cdot (\Theta_{k+1} - \Theta_k) \tag{45}$$

To solve an optimization problem, we basically seek a certain point satisfying the first-order necessary optimality condition, that is, $\nabla f = \mathbf{0}$. Hence, by combining (Eq. 43) with (Eq. 45) and solving $\nabla f = \mathbf{0}$ for $\mathbf{\Theta}_{k+1}$, we get an idea how to compute the next iterate from the current $\mathbf{\Theta}_k$.

$$\Theta_{k+1} = \Theta_k + \left(\frac{\partial \mathbf{F}(\Theta_k)}{\partial \Theta}^{\mathrm{T}} \cdot {\Sigma_\eta}^{-1} \cdot \frac{\partial \mathbf{F}(\Theta_k)}{\partial \Theta} \right)^{-1} \cdot \frac{\partial \mathbf{F}(\Theta_k)}{\partial \Theta}^{\mathrm{T}} \cdot {\Sigma_\eta}^{-1} \cdot (\eta - \mathbf{F}(\Theta)) \tag{46}$$

To accurately compute the iteration steps, the invertiability of the model's Jacobian $\partial \mathbf{F} / \partial \mathbf{\Theta}$ is of importance. The invertiability is model-dependent and also on the ^{13}C-labeling positions of tracer substrates. Therefore, the Jacobian can be applied as the objective to be maximized when designing ^{13}C-labeling experiments.

5. *Sequential Quadratic Programming*

In practice, the constrained nonlinear minimization problem such as (Eq. 13) is often formulated as the Lagrangian function, that is, a linear combination of the objective function and the constraints, i.e.,

$$L(\Theta, \lambda) = f(\Theta) - {\lambda_{\mathrm{E}}}^{\mathrm{T}} \cdot C_{\mathrm{E}}(\Theta) - {\lambda_{\mathrm{I}}}^{\mathrm{T}} \cdot C_{\mathrm{I}}(\Theta). \tag{47}$$

Here, $\lambda = [\lambda_{\mathrm{E}}, \lambda_{\mathrm{I}}]$ is a set of the Lagrangian multipliers for p equality constraints $C_{\mathrm{E}} = \mathbf{0}$, $\lambda_{\mathrm{E}} = (\lambda_1, \lambda_2, \ldots, \lambda_p)^{\mathrm{T}}$, and for q inequality constraints $C_{\mathrm{I}} \geq \mathbf{0}$, $\lambda_{\mathrm{I}} = (\lambda_1, \lambda_2, \ldots, \lambda_q)^{\mathrm{T}}$. By the first-order necessary optimality condition, $\mathbf{\Theta}^*$ is a local minimizer if there exists a vector $\lambda^* = ({\lambda_{\mathrm{E}}}^*, {\lambda_{\mathrm{I}}}^*)$ that satisfies the so-called Karush–Kuhn–Tucker (KKT) condition (36), i.e.,

$$\mathbf{D}_L = \nabla L(\Theta, \lambda) = \begin{pmatrix} \nabla f(\Theta^*) - \nabla C(\Theta^*) \cdot \lambda^* \\ -\nabla C(\Theta^*) \end{pmatrix} = \mathbf{0}. \tag{48}$$

Here, $C(\mathbf{\Theta})$ denotes the equality and active inequality constraints, and inactive inequality constraints are not included in the above equation because $\boldsymbol{\lambda}^* \, \mathrm{I} = \mathbf{0}$ for those inactive species of the inequality constraints ($C_{\mathrm{I}} > \mathbf{0}$). If the objective and constraints are convex functions, then (Eq. 48) is both necessary and sufficient for a global optimum.

The frequently applied algorithm associated with the Lagrangian formulation is the sequential quadratic programming (36) and its framework is forming a quadratic programming (QP) subproblem at the current iterate $\boldsymbol{\Theta}_k$ and $\boldsymbol{\lambda}_k$ to get a new iterate $\boldsymbol{\Theta}_{k+1}$ and $\boldsymbol{\lambda}_{k+1}$. The underlying idea is that the KTT condition can be linearly approximated using the Taylor first-order expansion such that

$$\nabla L(\Theta_{k+1}, \lambda_{k+1}) = \nabla L(\Theta_k, \lambda_k) + \begin{pmatrix} \frac{\partial \nabla L(\Theta_k, \lambda_k)}{\partial \Theta_k} & \frac{\partial \nabla L(\Theta_k, \lambda_k)}{\partial \lambda_k} \end{pmatrix} \cdot \begin{pmatrix} \Delta\Theta_k \\ \Delta\lambda_k \end{pmatrix}, \tag{49}$$

where

$$\begin{pmatrix} \frac{\partial \nabla L(\Theta_k, \lambda_k)}{\partial \Theta_k} & \frac{\partial \nabla L(\Theta_k, \lambda_k)}{\partial \lambda_k} \end{pmatrix} = \begin{pmatrix} \frac{\partial \nabla f(\Theta_k)}{\partial \Theta} - \frac{\partial \nabla C(\Theta_k)}{\partial \Theta} \cdot \lambda_k & -\nabla C(\Theta_k) \\ -\nabla C(\Theta_k)^T & \mathbf{0} \end{pmatrix} \tag{50}$$

By solving (Eq. 49) for the search direction of $\Delta\boldsymbol{\Theta}_k = \boldsymbol{\Theta}_{k+1} - \boldsymbol{\Theta}_k$ and $\Delta\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k$, one can compute the next iterate for $\boldsymbol{\Theta}$ and $\boldsymbol{\lambda}$ as follows:

$$\begin{pmatrix} \Delta\Theta_k \\ \Delta\lambda_k \end{pmatrix} = \begin{pmatrix} \frac{\partial \nabla f(\Theta_k)}{\partial \Theta} - \frac{\partial \nabla C(\Theta_k)}{\partial \Theta} \cdot \lambda_k & -\nabla C(\Theta_k) \\ -\nabla C(\Theta_k)^T & \mathbf{0} \end{pmatrix}^{-1} \cdot \begin{pmatrix} -\nabla f(\Theta_k) + \nabla C(\Theta_k) \cdot \lambda_k \\ C(\Theta_k) \end{pmatrix}. \tag{51}$$

The $(1, 1)$th entry in the inversed matrix is the Hessian of the Lagrangian function are typically computed as a positive definite quasi-Newton approximation of the Hessian, e.g., using BFGS ((Broyden–Fletcher–Goldfarb–Shanno) method (36).

6. *Parameter Compactification*

To increase optimization efficiency, the independent fluxes belonging to Θ can be compactified using a single rule such that (24):

$$\varphi_i = \frac{v_i}{\alpha + v_i} \text{ with } 0 \leq v_i < \infty \wedge \alpha > 0 \Rightarrow \varphi_i \in [0,\ 1). \tag{52}$$

The above compactified flux variables φ, the $[0, 1)$-fluxes, are bijective mapping of $[0, \infty)$ domain onto $[0, 1)$ range: if $v_i \to 0$, then $\varphi_i \to 0$ and if $v_i \to \infty$, then $\varphi_i \to 1$ for a certain real positive number of the parameter scaling constant α. These $[0, 1)$-fluxes can potentially elevate the output sensitivity and, thus, the convergence speed. The output sensitivities of a carbon flux system with respect to $\boldsymbol{\Theta}$ and $[0, 1)$-fluxes are $\partial \mathbf{x}_{\mathrm{mdv}}/\partial\boldsymbol{\Theta}$ and $\partial \mathbf{x}_{\mathrm{mdv}}/\partial\boldsymbol{\varphi}$ respectively, where the latter equals

$(\partial \mathbf{x}_{\mathrm{mdv}}/\partial \boldsymbol{\Theta})\cdot(\partial \boldsymbol{\Theta}/\partial \boldsymbol{\varphi})$ by the chain rule. Differentiating v_i with respect to φ_i results in

$$\frac{\mathrm{d}v_{\mathrm{i}}}{\mathrm{d}\varphi_{\mathrm{i}}} = \frac{\alpha}{(1-\varphi_i)^2}. \tag{53}$$

If $\alpha > (1-\varphi_i)^2$ holds and, thus, $\mathrm{d}v_i/\mathrm{d}\varphi_i > 1$, we get increased sensitivity by $\partial \mathbf{x}_{\mathrm{mdv}}/\partial \boldsymbol{\varphi} = (\partial \mathbf{x}_{\mathrm{mdv}}/\partial \boldsymbol{\Theta})\cdot(\partial \boldsymbol{\Theta}/\partial \boldsymbol{\varphi})$ compared to $\partial \mathbf{x}_{\mathrm{mdv}}/\partial \boldsymbol{\Theta}$. In particular, a higher sensitivity can always be obtained by setting $\alpha \geq 1$ due to the finite values of fluxes, that is, $0 \leq v_i < \infty$ or $0 \leq \varphi_i < 1$. Hence, setting the parameter scaling constant $\alpha \geq 1$ is more preferable for numerical optimization than $\alpha > 0$. Moreover, the mapping such as (Eq. 52) has proven to decrease the curvature of the ^{13}C-labeling system in the parameter space and is advantageous for model linearization (65). One may adjust α at the beginning of the numerical process of (Eq. 13) and at every failed optimization trial (see Note 7) such that the condition number of the partial derivatives in (Eq. 32) is maximized. By doing so, the matrix inversion in (Eq. 46) or (Eq. 51) and the corresponding computation of search step become more accurate.

7. *Optimization Termination Criteria*
Termination tolerance placed on parameters or objective function value may not clearly be defined *a priori*. Especially, they depend on the measurement noise associated with experimental data as well as output sensitivities for an NLSP. Thus, termination tolerance is set rather empirically, or one may rather apply the chi-square criterion. For instance, at a local optimum of $\boldsymbol{\Theta}^*$ that satisfies the constraints given in (Eq. 13), one can check if the objective function value meets the following criterion.

$$(\eta - \mathbf{F}(\Theta^*))^{\mathrm{T}} \cdot \Sigma_\eta^{-1} \cdot (\eta - \mathbf{F}(\Theta^*)) \leq \chi^2_{\dim(\eta)-\dim(\Theta)}(1-\alpha) \tag{54}$$

Here, $\chi^2{}_{\dim(\boldsymbol{\eta})\,-\,\dim(\boldsymbol{\Theta})}(1-\alpha)$ denotes the χ^2 critical value at a degree of freedom of $\dim(\boldsymbol{\eta}) - \dim(\boldsymbol{\Theta})$ and a significant level of α. This so-called goodness-of-fit test can be applied as the ultimate termination criterion, yet one still need to define proper tolerance values to prevent immature termination at a suboptimal point or excessive iteration.

Herein, a method was suggested that dynamically adjust tolerance values by repeatedly restarting optimization trials (24). The method was proven to improve solution quality as depicted in Fig. 7.

In particular, the optimization consists of a series of suboptimization trials, starting from a large tolerance value, e.g., 10^2 and properly decreasing the tolerance at each *k*th trial. This

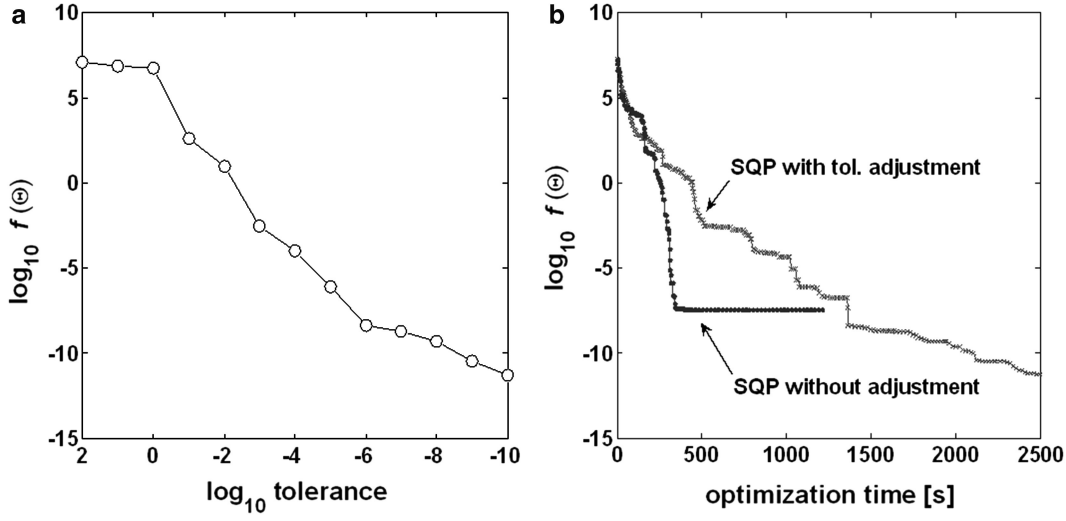

Fig. 7. Decrease of the objective function at each termination of SQP suboptimization using tolerance adjustment (**a**) and its comparison with SQP carried out at a constant tolerance during optimization (**b**). (reproduced from ref. 24).

dynamic adjustment was found to always hold $f(\mathbf{\Theta}^*_k) < f(\mathbf{\Theta}^*_{k-1})$ when starting the kth trial from the $(k-1)$th local optimum $\mathbf{\Theta}^*_{k-1}$ with a properly decreased tolerance. If the $(k-1)$th trial fails, the use of a feasible starting point $\mathbf{\Theta}^\circ$ recorded for the smallest function value up to the current trial with an increased tolerance value helps obtain $f(\mathbf{\Theta}^*_k) < f(\mathbf{\Theta}^\circ)$. For this case, one can additionally adjust the scaling factor α if the parameter compactification is applied. When starting the kth suboptimization trial from $\mathbf{\Theta}^\circ$ that is isolated from the previous $(k-n)$th successful trial with α_{k-n}, the $[0, 1)$-fluxes have to be rescaled in accordance with the new scaling constant α_k. Since $\nu = \alpha_{k-n}\varphi_{k-n}/(1-\varphi_{k-n})$ from (Eq. 52), substituting ν in $\varphi_k = \nu/(\alpha_k + \nu)$ results in:

$$\varphi_k = \frac{\alpha_{k-n}\varphi_{k-n}/(1-\varphi_{k-n})}{\alpha_k + \alpha_{k-n}\varphi_{k-n}/(1-\varphi_{k-n})}. \tag{55}$$

References

1. Yim H, Haselbeck R, Niu W, Pujol-Baxley C, Burgard A, Boldt J, Khandurina J, Trawick JD, Osterhout RE, Stephen R, Estadilla J, Teisan S, Schreyer HB, Andrae S, Yang TH, Lee SY, Burk MJ, Van Dien S (2011) Metabolic engineering of *Escherichia coli* for direct production of 1,4-butanediol. Nat Chem Biol 7:445–452
2. Wiechert W (2001) ^{13}C metabolic flux analysis. Metab Eng 3:195–206
3. Zupke C, Stephanopoulos G (1994) Modeling of isotope distributions and intracellular fluxes in metabolic networks using atom mapping matrixes. Biotechnol Prog 10:489–498
4. Ravikirthi P, Suthers PF, Maranas CD (2011) Construction of an *E. coli* genome-scale atom mapping model for MFA calculations. Biotechnol Bioeng 108:1372–1382

5. Hellerstein MK, Neese RA (1999) Mass isotopomer distribution analysis at eight years: theoretical, analytic, and experimental considerations. Am J Physiol 276: E1146–E1170
6. Schmidt K, Carlsen M, Nielsen J, Villadsen J (1997) Modeling isotopomer distributions in biochemical networks using isotopomer mapping matrixes. Biotechnol Bioeng 55:831–840
7. Christensen B, Nielsen J (1999) Isotopomer analysis using GC-MS. Metab Eng 1:282–290
8. Choi J, Antoniewicz MR (2010) Tandem mass spectrometry: a novel approach for metabolic flux analysis. Metab Eng 13:225–233
9. Rosman KJR, Taylor PDP (1998) Isotopic compositions of the elements 1997. J Phys Chem Ref Data 27(6):1275–1287
10. Pingitore F, Tang Y, Kruppa GH, Keasling JD (2007) Analysis of amino acid isotopomers using FT-ICR MS. Anal Chem 79:2483–2490
11. Sonntag K, Schwinde J, de Graaf A, Marx A, Eikmanns B, Wiechert W, Sahm H (1995) 13C NMR studies of the fluxes in the central metabolism of *Corynebacterium glutamicum* during growth and overproduction of amino acids in batch cultures. Appl Microbiol Biotechnol 44:489–495
12. Yang TH, Bolten CJ, Coppi MV, Sun J, Heinzle E (2009) Numerical bias estimation for mass spectrometric mass isotopomer analysis. Anal Biochem 388:192–203
13. Crawford JM, Blum JJ (1983) Quantitative analysis of flux along the gluconeogenic, glycolytic and pentose phosphate pathways under reducing conditions in hepatocytes isolated from fed rats. Biochem J 212:585–598
14. Baranyai JM, Blum JJ (1989) Quantitative analysis of intermediary metabolism in rat hepatocytes incubated in the presence and absence of ethanol with a substrate mixture including ketoleucine. Biochem J 258:121–140
15. Rognstad R, Katz J (1972) Gluconeogenesis in the kidney cortex. Quantitative estimation of carbon flow. J Biol Chem 247:6047–6054
16. Walsh K, Koshland DE Jr (1984) Determination of flux through the branch point of two metabolic cycles. The tricarboxylic acid cycle and the glyoxylate shunt. J Biol Chem 259:9646–9654
17. Kelleher JK (1985) Analysis of tricarboxylic acid cycle using [14C]citrate specific activity ratios. Am J Physiol 248:E252–E260
18. Katz J (1985) Determination of gluconeogenesis in vivo with 14C-labeled substrates. Am J Physiol 248:R391–R399
19. Goebel R, Berman M, Foster D (1982) Mathematical model for the distribution of isotopic carbon atoms through the tricarboxylic acid cycle. Fed Proc 41:96–103
20. Christensen B, Gombert AK, Nielsen J (2002) Analysis of flux estimates based on (13) C-labelling experiments. Eur J Biochem 269:2795–2800
21. Wiechert W, Mollney M, Isermann N, Wurzel M, de Graaf AA (1999) Bidirectional reaction steps in metabolic networks: III. Explicit solution and analysis of isotopomer labeling systems. Biotechnol Bioeng 66:69–85
22. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2007) Elementary metabolite units (EMU): a novel framework for modeling isotopic distributions. Metab Eng 9:68–86
23. Srour O, Young JD, Eldar YC (2011) Fluxomers: a new approach for 13C metabolic flux analysis. BMC Syst Biol 5:129
24. Yang TH, Frick O, Heinzle E (2008) Hybrid optimization for ^{13}C metabolic flux analysis using systems parametrized by compactification. BMC Syst Biol 2:29
25. Mollney M, Wiechert W, Kownatzki D, de Graaf AA (1999) Bidirectional reaction steps in metabolic networks: IV. Optimal design of isotopomer labeling experiments. Biotechnol Bioeng 66:86–103
26. Young JD, Walther JL, Antoniewicz MR, Yoo H, Stephanopoulos G (2008) An elementary metabolite unit (EMU) based method of isotopically nonstationary flux analysis. Biotechnol Bioeng 99:686–699
27. Noh K, Gronke K, Luo B, Takors R, Oldiges M, Wiechert W (2007) Metabolic flux analysis at ultra short time scale: isotopically nonstationary 13C labeling experiments. J Biotechnol 129:249–267
28. Wiechert W, Noh K (2005) From stationary to instationary metabolic flux analysis. Adv Biochem Eng Biotechnol 92:145–172
29. Young JD, Shastri AA, Stephanopoulos G, Morgan JA (2011) Mapping photoautotrophic metabolism with isotopically nonstationary (13)C flux analysis. Metab Eng 13:656–665
30. Bolten CJ, Kiefer P, Letisse F, Portais JC, Wittmann C (2007) Sampling for metabolome analysis of microorganisms. Anal Chem 79:3843–3849
31. Wahl SA, Noh K, Wiechert W (2008) 13C labeling experiments at metabolic nonstationary conditions: an exploratory study. BMC Bioinformatics [electronic resource] 9:152
32. Fischer E, Sauer U (2003) Metabolic flux profiling of *Escherichia coli* mutants in central

carbon metabolism using GC-MS. Eur J Biochem 270:880–891

33. Zamboni N, Fischer E, Sauer U (2005) FiatFlux—a software for metabolic flux analysis from ^{13}C-glucose experiments. BMC Bioinformatics [electronic resource] 6:209
34. Rantanen A, Rousu J, Jouhten P, Zamboni N, Maaheimo H, Ukkonen E (2008) An analytic and systematic framework for estimating metabolic flux ratios from 13C tracer experiments. BMC Bioinformatics [electronic resource] 9:266
35. Floudas CA, Pardalos PM (1992) Recent advances in global optimization. Princeton University Press, Princeton, NJ
36. Nocedal J, Wright SJ (1999) Numerical optimization. Springer, New York
37. Press WH (1992) Numerical recipes in C: the art of scientific computing, 2nd edn. Cambridge University Press, Cambridge; New York
38. Schmidt K, Nielsen J, Villadsen J (1999) Quantitative analysis of metabolic fluxes in *Escherichia coli*, using two-dimensional NMR spectroscopy and complete isotopomer models. J Biotechnol 71:175–189
39. Brackin P, Colton SC (2002) Using genetic algorithms to set target values for engineering characteristics in the house of quality. J Comput Inf Sci Eng 2:106–114
40. Kelner V, Capitanescu F, Léonard O, Wehenkel L (2008) An hybrid optimization technique coupling an evolutionary and a local search algorithm. J Comput Appl Math 215 (2):448–456
41. Lambert TW, Hittle DC (2000) Optimization of autonomous village electrification systems by simulated annealing. Sol Energy 68:121–132
42. Mendes P, Kell D (1998) Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics (Oxford, England) 14:869–883
43. Moles CG, Mendes P, Banga JR (2003) Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res 13:2467–2474
44. Xu P (2003) A hybrid global optimization method: the multi-dimensional case. J Comput Appl Math 155:423–446
45. Long CE, Polisetty PK, Gatzke EP (2006) Nonlinear model predictive control using deterministic global optimization. J Process Contr 16:635–643
46. Nash SG, Sofer A (1996) Linear and nonlinear programming. McGraw-Hill, New York
47. Hill MC, Osterby O (2003) Determining extreme parameter correlation in ground water models. Ground Water 41:420–430
48. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2006) Determination of confidence intervals of metabolic fluxes estimated from stable isotope measurements. Metab Eng 8:324–337
49. Wittmann C (2007) Fluxome analysis using GC-MS. Microb Cell Fact 6:6
50. Krömer JO, Fritz M, Heinzle E, Wittmann C (2005) In vivo quantification of intracellular amino acids and intermediates of the methionine pathway in Corynebacterium glutamicum. Anal Biochem 340:171–173
51. Yang TH, Heinzle E, Wittmann C (2005) Theoretical aspects of ^{13}C metabolic flux analysis with sole quantification of carbon dioxide labeling. Comput Biol Chem 29:121–133
52. Yang TH, Wittmann C, Heinzle E (2006) Respirometric ^{13}C flux analysis—Part II: in vivo flux estimation of lysine-producing *Corynebacterium glutamicum*. Metab Eng 8:432–446
53. Rabinowitz JD, Kimball E (2007) Acidic acetonitrile for cellular metabolome extraction from *Escherichia coli*. Anal Chem 79:6167–6173
54. Canelas AB, ten Pierick A, Ras C, Seifar RM, van Dam JC, van Gulik WM, Heijnen JJ (2009) Quantitative evaluation of intracellular metabolite extraction techniques for yeast metabolomics. Anal Chem 81:7379–7389
55. Antoniewicz MR, Kraynie DF, Laffend LA, Gonzalez-Lergier J, Kelleher JK, Stephanopoulos G (2007) Metabolic flux analysis in a nonstationary system: fed-batch fermentation of a high yielding strain of E. coli producing 1,3-propanediol. Metab Eng 9(3):277–292
56. Lee WN, Bergner EA, Guo ZK (1992) Mass isotopomer pattern and precursor-product relationship. Biol Mass Spectrom 21:114–122
57. Wahl SA, Dauner M, Wiechert W (2004) New tools for mass isotopomer data evaluation in ^{13}C flux analysis: mass isotope correction, data consistency checking, and precursor relationships. Biotechnol Bioeng 85:259–268
58. Fernandez CA, Des Rosiers C, Previs SF, David F, Brunengraber H (1996) Correction of ^{13}C mass isotopomer distributions for natural stable isotope abundance. J Mass Spectrom 31:255–262
59. van Winden WA, Wittmann C, Heinzle E, Heijnen JJ (2002) Correcting mass isotopomer distributions for naturally occurring isotopes. Biotechnol Bioeng 80:477–479
60. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2007) Accurate assessment of amino acid

mass isotopomer distributions for metabolic flux analysis. Anal Chem 79:7554–7559

61. Mahadevan R, Schilling CH (2003) The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. Metab Eng 5:264–276
62. Yang TH, Coppi MV, Lovley DR, Sun J (2010) Metabolic response of Geobacter sulfurreducens towards electron donor/acceptor variation. Microb Cell Fact 9:90
63. Massart DL (1997) Handbook of chemometrics and qualimetrics. Elsevier, Amsterdam; New York
64. Arnold SF (1990) Mathematical statistics. Prentice-Hall, Englewood Cliffs, NJ
65. Wiechert W, Siefke C, de Graaf A, Marx A (1997) Bidirectional reaction steps in metabolic networks: II. Flux estimation and statistical analysis. Biotechnol Bioeng 55 (1):118–135

Chapter 16

Nuclear Magnetic Resonance Methods for Metabolic Fluxomics

Shilpa Nargund, Max E. Joffe, Dennis Tran, Vitali Tugarinov, and Ganesh Sriram

Abstract

Fluxomics, through its core methodology of metabolic flux analysis (MFA), enables quantification of carbon traffic through cellular biochemical pathways. Isotope labeling experiments aid MFA by providing information on intracellular fluxes, especially through parallel and cyclic pathways. Nuclear magnetic resonance (NMR) and mass spectrometry (MS) are two complementary methods to measure abundances of isotopomers generated in these experiments. 2-D [^{13}C, ^{1}H] heteronuclear correlation NMR spectra can detect ^{13}C isotopes coupled to protons and thus noninvasively separate molecules and atoms with a specific isotopic content from a mixture of molecular species. Furthermore, the fine structures of the peaks in these spectra can reveal scalar couplings between chemically bonded carbon atoms in the sample, from which isotopomer abundances can be quantified. This chapter introduces methods for NMR sample preparation and spectral acquisition of 2-D [^{13}C, ^{1}H] correlation maps, followed by a detailed presentation of methods to process the spectra and quantify isotopomer abundances. We explain the use of the software NMRViewJ for spectral visualization and processing, as well as MATLAB scripts developed by us for peak extraction, deconvolution of overlapping peaklets, and isotopomer abundance quantification. Finally, we discuss the applications of NMR-derived isotopomer data toward quantitatively understanding metabolic pathways.

Key words: Systems metabolic engineering, Flux analysis, NMR, Isotope labeling

1. Introduction

Fluxomics is the branch of systems biology dedicated to measuring metabolic fluxes in cells (1). Through its core methodology of metabolic flux analysis (MFA), fluxomics enables visualization of the impact of environmental or genetic perturbations on carbon traffic in cellular metabolic pathways. It also permits the identification of rigid and flexible metabolic nodes that may be less or more

Shilpa Nargund and Max E. Joffe contributed equally to this work.

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_16,

responsive to engineering (2). Furthermore, this method can pinpoint carbon redirection through apparently futile cycles, unexpected or even previously unknown pathways, and thus explain why a metabolic engineering attempt was a success or a failure. Orthogonally, metabolomics can measure metabolite concentrations and thereby complement fluxomics (3, 4).

MFA is performed by implementing mass balances on steady-state metabolic models (e.g., (5)). This frequently results in an underdetermined system of linear equations, the solution of which requires extracellular or intracellular measurements. Commonly measured extracellular fluxes include the rates of carbon source uptake, product secretion, or biomass component synthesis. Intracellular measurements may include in vivo flux measurements by fluorescence resonance energy transfer (FRET) (1, 6) or the isotope labeling patterns resulting from the metabolic processing of an isotopic mixture of carbon source(s). Although FRET is direct, it can currently be used to measure only one or a few fluxes. Alternatively, isotope labeling patterns indirectly provide powerful information on several dozen intracellular fluxes in a metabolic network. In particular, they enable the determination of fluxes through two intracellular pathways with shared start and end points (7, 8) (also see Subheadings 3.4.1 and 3.4.2) as well as through intra- and intercompartmental metabolic cycles (8). Isotope labeling patterns can be detected by two techniques: nuclear magnetic resonance (NMR) spectroscopy, which distinguishes isotopes by their magnetic properties (9–13), and mass spectrometry (MS), which distinguishes isotopes by their masses (e.g., (14, 15)). Although NMR is less sensitive than MS, it provides unique labeling information complementary to that accessible from MS. For instance, NMR can directly measure ^{13}C atom enrichments and carbon–carbon scalar coupling. Additionally, NMR can distinguish between isotopes of different elements (e.g., ^{13}C and ^{15}N), which is difficult to perform on an MS without very high mass resolution (16).

NMR can noninvasively detect magnetically responsive atomic nuclei to provide information on molecular structure and isotopic composition (17, 18). External magnetic fields distribute NMR-active atomic nuclei, such as ^{1}H and ^{13}C, between "spin-up" and "spin-down" states. The nuclei resonate between these states at a characteristic frequency that depends on frequency of the applied magnetic field and the chemical nature of the nucleus. Nuclei found in different chemical environments, such as different functional groups, exhibit very small but measurable frequency differences. This difference is known as an isotropic chemical shift (δ) and is expressed in parts per million (ppm) of the applied field (17). Consequently, a one-dimensional (1-D) NMR ^{13}C spectrum of a mixture of compounds will separate carbon nuclei according to their δ_{13C} values, whereas a ^{1}H spectrum will separate them according to δ_{1H} values (of their associated protons). Similarly, a two-dimensional

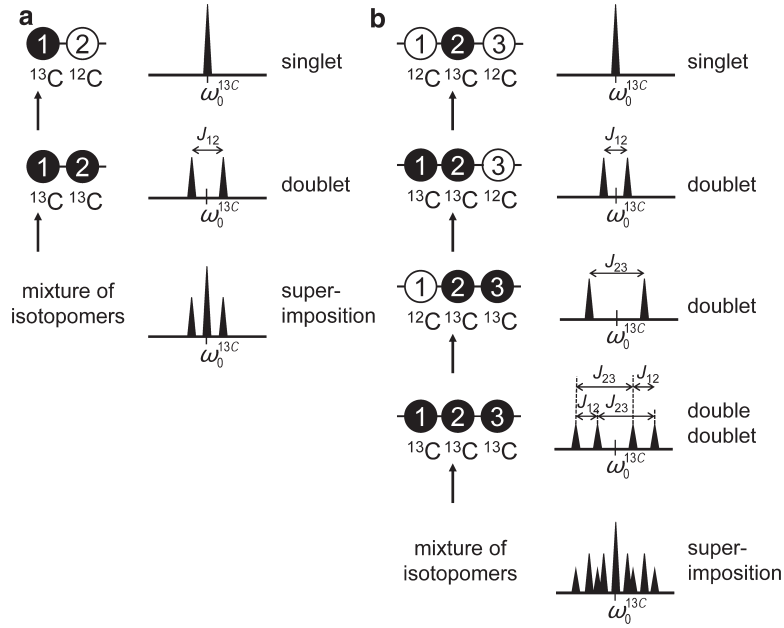

Fig. 1. Fine structures (peaklets) observable on a 1-D ^{13}C slice of a 2-D [^{13}C, ^{1}H] correlation map: (**a**) a terminal ^{13}C atom with one neighboring carbon atom displays a singlet (^{13}C–^{12}C) and a doublet (^{13}C–^{13}C); (**b**) a central ^{13}C atom with two neighboring carbon atoms displays a singlet (^{12}C–^{13}C–^{12}C), a doublet (^{13}C–^{13}C–^{12}C or ^{12}C–^{13}C–^{13}C), or a double doublet (^{13}C–^{13}C–^{13}C). ^{13}C atoms are depicted as *black circles* and ^{12}C atoms as *white circles*. The detected (terminal or central) carbon atom, indicated with an *arrow*, has a central ^{13}C frequency ω_0^{13C}. The two peaklets of the doublet are separated by the carbon–carbon scalar coupling constant (J_{CC}) between the corresponding atoms; J_{12} is the value of this constant for the coupling between ^{13}C atoms 1 and 2. The four peaklets of a double doublet are separated by the sum or difference of the two J_{CC} values between the detected carbon and its two neighbors.

(2-D) [^{13}C, ^{1}H] heteronuclear single-quantum correlation (HSQC) map will contain cross peaks corresponding to the chemical shifts of ^{13}C nuclei coupled to one or more protons in the indirect (F1) dimension and the corresponding chemical shifts of directly bonded ^{1}H nuclei in the directly acquired (F2) dimension.

Information on isotope labeling patterns is accessible from the fine structure of cross peaks in [^{13}C, ^{1}H] correlation maps or homonuclear 2-D [^{1}H, ^{1}H] correlation (COSY) and total correlation spectra (TOCSY) (17) maps. Figure 1 depicts the fine structures (multiplets) observable on a 1-D ^{13}C slice of a [^{13}C, ^{1}H] correlation map, each type of multiplet representing a specific isotopomer or a sum of isotopomers. A terminal ^{13}C atom connected to only one other carbon atom displays up to two types of multiplets depending on the isotopic state of the neighboring carbon atom. A singlet, consisting of one peaklet, represents a $\boxed{^{13}C}$–^{12}C population; a doublet, consisting of two peaklets, represents a $\boxed{^{13}C}$–^{13}C population (the box indicates the detected terminal ^{13}C

atom; Fig. 1a). The two peaklets in a doublet are separated by a frequency known as the carbon–carbon scalar coupling constant (J_{CC}), a property of the atoms represented in the doublet that is independent of the frequency of the external magnetic field (e.g., J_{12} in Fig. 1a). A central ^{13}C atom with two neighboring carbon atoms displays up to three types of multiplets—a singlet consisting of one peaklet (^{12}C–$\boxed{^{13}\text{C}}$–^{12}C), two doublets consisting of two peaklets each (^{13}C–$\boxed{^{13}\text{C}}$–^{12}C or ^{12}C–$\boxed{^{13}\text{C}}$–^{13}C), or a double doublet consisting of four peaklets (^{13}C–$\boxed{^{13}\text{C}}$–^{13}C) (the box indicates the detected central ^{13}C atom; Fig. 1b). Pairs of peaklets in a double doublet are separated by the sum or the difference of the J_{CC} values between the detected ^{13}C atom and its two neighbors (Fig. 1b).

Often, the doublets and double doublets may not be symmetrically distributed around the central frequency of the peak (ω_0^{13C}). This is caused by isotope effects between the carbon atoms, represented by a frequency T_{CC}. Given the value of the central frequency ω_0^{13C} of the detected central carbon atom as well as the two J_{CC} and the two T_{CC} between it and its two neighboring carbon atoms, the frequencies ω of the different peaklets can be expressed as (19):

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \omega_0^{13C} \\ J_{12}/2 \\ J_{23}/2 \\ T_{12}/2 \\ T_{23}/2 \end{bmatrix} = \begin{bmatrix} \omega_s^{13C} \\ \omega_{d1,1}^{13C} \\ \omega_{d1,2}^{13C} \\ \omega_{d2,1}^{13C} \\ \omega_{d2,2}^{13C} \\ \omega_{dd,1}^{13C} \\ \omega_{dd,2}^{13C} \\ \omega_{dd,3}^{13C} \\ \omega_{dd,4}^{13C} \end{bmatrix}, \qquad (1)$$

where d1 and d2 represent the doublets and dd represents the double doublet. Typical J_{CC} values for one-bond couplings are 35 Hz for a C_α–C_β bond in an amino acid and 60 Hz for a C_α–COOH bond. T_{CC} values for one-bond couplings are typically in the range −5 to 5 Hz.

Additional types of multiplets are possible. For example, if the couplings J_{12} and J_{23} in Fig. 1b were equal, the two inner peaklets of the double doublet would collapse into one peaklet of twice the height. The resulting multiplet is called a triplet. Furthermore, the two doublets would merge, due to which the fine structure would have five peaklets instead of nine (the merger of the inner double doublet peaklets as well as the merger of the doublets each reduces the number of peaklets by two). Furthermore, long-range coupling

between carbon atoms that are not connected by a covalent bond is also possible, although J_{CC} for such couplings is generally smaller than for one-bond couplings. If tangible, the long-range couplings can yield valuable information on isotopomer abundances of long molecules (e.g., levulinic acid, discussed in (20)).

This chapter briefly introduces methods for sample preparation and spectral acquisition, followed by a detailed protocol for spectral processing and quantification of isotopomer abundances.

2. Materials

2.1. Sample Preparation

1. Dialysis cassette, e.g. Slide-A-Lyzer (Thermo Fisher Scientific, Waltham, MA).
2. 6 N constant-boiling HCl (Thermo Fisher Scientific).
3. Hydrolysis tubes, 20 mL capacity (Thermo Fisher Scientific).
4. Acid evaporator, e.g. RapidVap (Labconco, Kansas City, MO).
5. 0.22 μm Spin-X centrifuge tube filters (Corning Life Sciences, Tewksbury, MA).
6. Lyophilizer, e.g. MicroModulyo (Thermo Fisher Scientific).
7. NMR tubes (Thermo Fisher Scientific).
8. D_2O (Sigma-Aldrich, St. Louis, MO).
9. A pH meter equipped with an electrode that can fit into an NMR tube.

2.2. Spectral Acquisition

1. An NMR spectrometer of frequency 500 MHz or higher (e.g., Bruker BioSpin, Billerica, MA), equipped with probes for ^{1}H and ^{13}C.
2. Bruker Xwinnmr™ software (Bruker BioSpin) for spectral acquisition and initial processing, installed on a Unix computer.

2.3. Spectral Processing

1. NMRViewJ software (One Moon Scientific, Inc.; available free of charge at http://www.onemoonscientific.com).
2. A Pentium or higher computer.
3. Microsoft Excel (Microsoft Corporation, Redmond, WA).
4. MATLAB (MathWorks, Natick, MA).
5. MATLAB modules findPix.m, findNoise.m, loadData.m, autoPeak.m, fPeakSim.m, fPeakChi2.m, fPeakPlot.m, fPlotChi2.m, fSimAreas.m, PeakAnalyzerZn.m, and PeakAnneal.m for extracting, deconvoluting, and quantifying peaks. A package called NMRisotopomer that contains these modules is available free of charge at http://openwetware.org/wiki/Sriram_Lab.

3. Methods

3.1. Sample Preparation

1. Extract protein from the cell or tissue sample of interest. For obtaining a good spectral resolution, the sample should contain at least 5 mg of ^{13}C-labeled protein. see Note 1.
2. If protein is extracted by contacting with phosphate buffer saline or any other method that introduces salts into the extract, a dialysis step is necessary to eliminate the salts and reduce the ionic noise generated by conductive species. see Note 2.
3. Vacuum-hydrolyze the protein in 6 N constant-boiling HCl at 140 °C (or higher) for 4 h. Evaporate the acid on an acid evaporator. Reconstitute the residue in water, filter it, and lyophilize it. see Note 3.
4. Reconstitute the lyophilized amino acid mixture in D_2O. Verify that its pH is around 1.

3.2. 2-D [^{13}C, ^{1}H] HSQC Spectral Acquisition

The following is an overview of HSQC spectral acquisition; details are beyond the scope of this chapter, and their description in the context of flux analysis can be found in, for example, Szyperski (10). Specific software commands and instrumental adjustments for some of these steps will depend on the spectrometer used.

1. Lock the sample on deuterium signal. Set the desired sample temperature (we use 298 K). When the sample attains this temperature, use gradient shimming to homogenize the magnetic field. Tune and match the ^{1}H and ^{13}C channels of the NMR probe.
2. To measure isotopomer abundances, acquire 2-D [^{13}C, ^{1}H] HSQC spectra by employing a modified version of the insensitive nuclei enhanced by polarization transfer (INEPT) pulse scheme of Bodenhausen and Ruben (21). Typical acquisition parameters used in our laboratory are as follows: ^{13}C (F1) resonance frequency, 150 MHz; ^{1}H (F2) resonance frequency, 600 MHz; spectral width along ^{13}C (F1) dimension, 40 ppm or 6,039 Hz (this can be minimized with peak aliasing); spectral width along ^{1}H (F2) dimension, 13 ppm or 8,371 Hz; number of complex data points, 4,096 (^{13}C) × 1,024 (^{1}H); and number of scans, 4. see Note 4 for tips on using J-scaling to deconvolute overlapping peaklets.
3. To quantify ^{13}C enrichment, acquire a 2-D [^{1}H, ^{1}H] TOCSY spectrum with DIPSI-2 (22) isotropic mixing (mixing time ~80 ms) in the presence of scalar couplings.
4. Small variations in experimental conditions (e.g., pH, temperature, and chemical shift referencing) may cause spectra from different laboratories to be slightly different from

each other. Assignment of cross peaks on the HSQC and TOCSY spectra may require recording of 2-D [^{1}H, ^{1}H] and (optionally) 3-D [^{13}C, ^{1}H, ^{1}H] TOCSY spectra (23) of a 100% ^{13}C-labeled sample.

3.3. Spectral Processing

NMRViewJ is an open-source program useful for visualizing and quantitatively analyzing 2-D NMR spectra. It is written in tool command language (tcl) and has a programmable user interface. Its features include multiple views of one or more spectra, an unlimited number of windows and data files, as well as extraction of any plane in a higher dimensional spectrum.

3.3.1. Loading and Viewing a Spectrum

1. Open NMRViewJ (see Fig. 2 for a typical screenshot). A command window and a navigation bar appear. On the navigation bar click on the Datasets > Open and Draw Dataset menu item and select the appropriate NMR spectral file (Bruker-generated spectral files are named 2rr and stored in a nested directory corresponding to the spectrum).
2. The spectrum opens in a new window. To identify negative and positive peaks, right click and open the Attributes panel. Under the File tab, you can select colors to represent positive and negative peaks. You can also adjust the resolution level of the spectrum by using the scroll bar on the right.
3. Two sets of crosshairs (generally in different colors) are visible on the spectrum. To zoom in, enclose the desired zoom area within the crosshairs, and click on the magnifying glass icon at the top of the window. To pan out, click on the icon with two arrows heading out. To go back to a previous view, click on the back/front arrow.

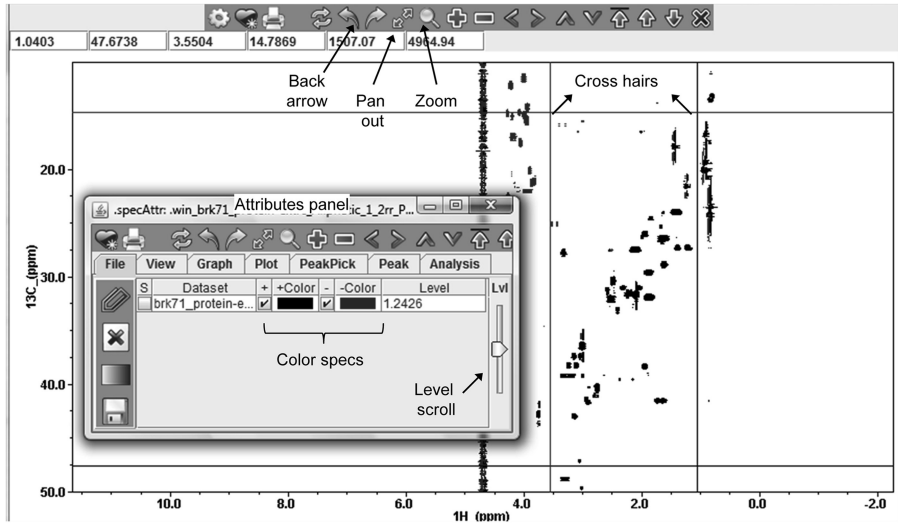

Fig. 2. NMRViewJ screenshot: (*background*) spectrum and (*foreground*) Attributes Panel.

3.3.2. Extraction of 1-D Slices Along ^{13}C Dimension

1-D slices of 2-D [^{13}C, ^{1}H] spectra along the ^{13}C dimension contain information on carbon isotopomers. The following steps are necessary to extract a slice along the ^{13}C dimension at a particular ^{1}H frequency, i.e., to obtain a list of intensities that describe the peak of interest:

1. Zoom into the peak of interest and center a vertical crosshair on it. This crosshair represents the ^{1}H frequency along which the 2-D spectrum will be sectioned. Right click and select Extract > Replace > Y. A new window appears, displaying a plot of intensity versus δ_{13C} for the extracted slice. If you have chosen a "negative" peak, i.e., a peak that was aliased such that its intensities are negative, you may have to adjust the parameter XOffset. To do this right click and select Attributes, then select the Plot tab, and increase XOffset.
2. The selected cross section may contain several peaks with the same 1H chemical shift, although these peaks will generally not overlap each other. Zoom into the peak of interest, and determine the chemical shift range of the peak in ppm. Next, convert this range to a frequency in Hz for use in the extraction command. For this conversion you will need to use the spectral width along the ^{13}C dimension in ppm and frequency (40 ppm and 6,039 Hz in the specifications listed above). The MATLAB script findPix.m in the NMRisotopomer package performs this calculation, given the ^{13}C spectral width. The syntax for executing this script is

   ```
   findPix(lower bound ppm, higher bound ppm).
   ```

   see Note 5 for tips on initial quality control of the extracted peak.
3. To extract peak intensities, execute the following script in the tcl command window of NMRViewJ:
   for {set i lower bound Hz} {$i < higher bound Hz} {incr i} {puts [v_getval work0 $i]}.

   A specific instance of this command
   for {set i 3891} {$i < 3993} {incr i} {puts [v_getval work0 $i]}
   will extract peaks lying between 3,891 and 3,993 Hz. On completion, this extraction procedure will display peak heights in the command window; the heights can then be copied into a text file.

3.3.3. Peak Deconvolution and Estimation of Areas Under Peaklets

Areas under multiplets are proportional to abundances of the corresponding isotopomers (Fig. 1). However, different multiplets could overlap, resulting in a complicated spectrum (Fig. 1b). To resolve this, we employ a curve-fitting algorithm that uses a mathematical model of the shapes of the fine structures in the spectrum and adjusts (optimizes) the parameters in the model until it accurately simulates the observed peak (see Fig. 3 for an illustrative fit).

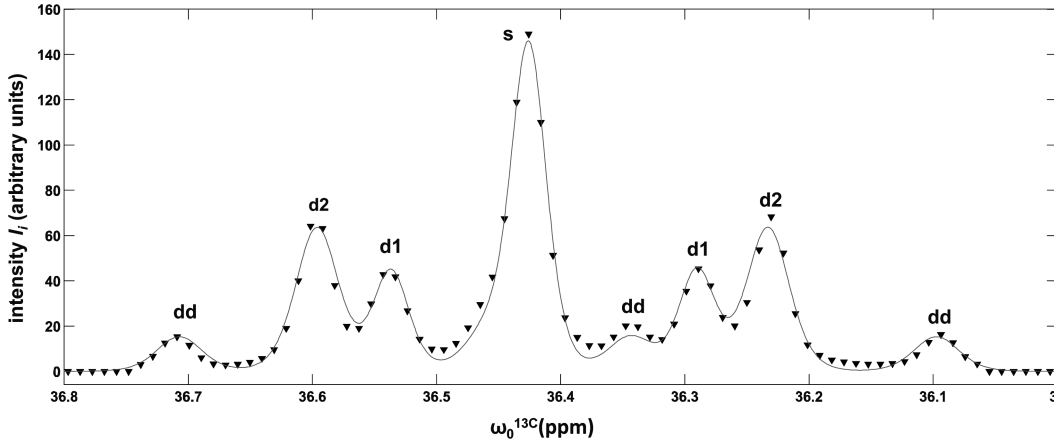

Fig. 3. 1-D ^{13}C slice of Asp-β peak from a 2-D [^{13}C, ^{1}H] correlation map of protein hydrolysate from a plant cell suspension. The full [^{13}C, ^{1}H] map contains ~55 such peaks. The *points* represent the experimentally observed peak; the *line* represents a simulation of the peak fitted using the NMRisotopomer package as explained in this chapter. Multiplets are abbreviated as follows: *s* singlet, *d1* and *d2* doublets, *dd* double doublet. One peaklet of the double doublet is obscured due to overlap with the taller singlet.

After obtaining an acceptable fit, this algorithm integrates the areas under the simulated multiplets.

Our spectral model deploys the Pearson VII waveform (19) to approximate NMR fine structures:

$$I_i(\omega^{13C}) = \frac{h_i}{\left[\frac{1}{p}\left(\frac{\omega^{13C} - \omega_{\max}{}^{13C}{}_i}{w_i}\right)^2 + 1\right]^p}$$

where I_i is the intensity of the peaklet i as a function of the ^{13}C frequency ω^{13C}, $\omega_{\max}{}^{13C}$ is the central frequency of the peaklet, h_i is the height of the peaklet, w_i is the width of the peaklet, and p is a number that determines the shape of the waveform. When $p = 1$, the waveform is identical to the Gaussian function, whereas when $p \to \infty$, the function equals the Cauchy–Lorentz distribution (19). Our curve-fitting algorithm incorporates a simulated annealing global optimization algorithm (24) that minimizes a χ^2 metric to efficiently find parameter values to fit complex, overlapping multiplets. To use the curve-fitting algorithm, proceed as follows:

1. It is helpful to first estimate the frequencies of the various peaklets. For this, (1) read the central peak frequency ω_0^{13C} (the frequency of the singlet), and (2) estimate the J_{CC} by measuring the separation between the peaklets in the doublets or the double doublet. Then proceed as follows.
2. Copy the peak heights extracted in Subheading 3.3.2 to a CSV file (default file name: peak.csv), with one height on each line. The intensities of aliased peaks may have negative values. To

process such peaks, replace the intensities in the CSV file with their absolute values.

3. Populate an Excel file (default file name: peakindex.xls) with the following information: (1) path and file name of the CSV file containing the peak extract, (2) path and file name of the file in which the results are to be stored, (3) spectral frequency (in Hz) corresponding to the start and end of the peak of interest, (4) ranges for the central peak frequency ω_0^{13C} and the J_{CC} estimated as described in step 1, (5) ranges for T_{CC}, typically −7 to 7 Hz, (6) the number of peaklets (3 for the example in Fig. 1a and 9 in Fig. 1b), (7) a starting value for the height of each type of multiplet, (8) a starting value for the linewidth of each type of multiplet (an initial value of ~0.015 works very well), and (9) an estimate for the Pearson curve parameter p, which determines the shape of the Pearson correlation curve that is fitted to the data. Values of $p > 2$ are good fits for our peaklets; however, peaklet shapes may vary. Table 1 lists illustrative values for the above parameters.
4. Ensure that Excel file of step 4 is in the same directory as the autoPeak.m script or in the MATLAB path.
5. Run autoPeak.m. The curve fitting procedure begins, and its progress is visible on the MATLAB command window. On completion, this procedure populates an Excel file (default name: Results.xls) in the folder specified in step 3 with the results of the peak fitting and the areas under the different multiplets. Additionally, this routine will generate two figures depicting the goodness of fit. Figure 3 illustrates a sample result of this peak-fitting algorithm. see Notes 6 and 7.

In the curve-fitting algorithm, the MATLAB script PeakAnneal.m performs the global optimization, calling fPeakSim.m to simulate peaks for a set of parameter values and fPeakChi2.m to calculate the χ^2 metric that determines how accurately the current set of parameter values simulates the observed peak. After the optimization is complete, fSimAreas.m returns the areas under each individual multiplet. Additionally, fPeakPlot.m and fPlotChi2.m plot the observed and simulated peaks.

3.4. Current and Future Applications of NMR-Based Fluxomics

Intensities of multiplets on ^{13}C NMR spectra relate to isotope labeling information in the form of isotopomers, bondomers, and elementary metabolite units (EMUs). Isotopomers (isotope isomers) are isomers of a compound that differ in the isotopic state (^{13}C or ^{12}C) of their carbon atoms. A molecule with n carbon atoms can have a maximum of 2^n isotopomers. Analytical methods including NMR typically do not provide information on all of these 2^n isotopomers, especially in the case of molecules with five or more carbon atoms. Instead, they provide the abundances of certain

Table 1
Illustrative parameters input to the curve-fitting algorithm, which quantifies intensities of overlapping multiplets

Peak	Asp-β
Number of peaklets	9
Starting chemical shift (ppm)	36.0
Ending chemical shift (ppm)	36.8
Central frequency, ω_0^{13C} (ppm)	36.42
J_{12} (Hz)	37.5
J_{23} (Hz)	55.0
T_{12} (Hz)	−2.0
T_{23} (Hz)	−2.0
Pearson curve parameter (usually >2)	2.0
Estimated height of singlet	150
Estimated height of doublet d1	45
Estimated height of doublet d2	70
Estimated height of double doublet	18
Linewidth of singlet (ppm)	0.015
Linewidth of doublet 1(ppm)	0.015
Linewidth of doublet 2 (ppm)	0.015
Linewidth of double doublet (ppm)	0.015

Estimated peak heights are in arbitrary units. Figure 3 shows the outcome of peak fitting using these parameter values

linear combinations of isotopomers. Additionally, since the NMR analysis of each carbon atom in a molecule provides isotopic nature of the carbon atoms neighboring it, there is overlap and redundancy in the information obtained from multiple carbon atoms of a given molecule. In experiments involving a single carbon source with more than one carbon atom, bondomers (25, 26) are defined as molecules whose carbon–carbon bonds are either intact (the atoms connected by the bond were not separated between carbon source and metabolite) or biosynthetic (the atoms connected by the bond were biosynthetically connected during metabolism). The more recent concept of EMU can also be correlated to isotopomer measurements available from NMR fine structures (27).

3.4.1. Metabolic Information Contained in Nonrandom Labeling Patterns

The abundances of isotopomers, bondomers, or EMUs in an isotope labeling experiment relate to metabolic flux patterns in the investigated cell or tissue. Bondomers and EMUs are particularly relevant as they provide more direct metabolic information. For example, bondomers directly relate to the multiplets observed in the fine structure of an NMR spectrum. In an experiment in which cells are fed a mixture of ~5–10 % U-^{13}C glucose diluted with naturally abundant glucose, a doublet mostly represents an intact bond, whereas a singlet mostly represents a biosynthetic bond. Statistical formulae to exactly translate multiplet intensities to bond integrities are available in Szyperski (9). Furthermore, information on bondomer abundances is especially useful in distinguishing between metabolic pathways featuring different carbon skeletal rearrangements, such as linear pathways versus pathways featuring significant bond breakage and reassembly. One illustration of this is provided by glycolysis (a linear pathway) and the pentose phosphate pathway (a pathway involving significant bond breakage). A 3-carbon glycolytic intermediate such as glyceraldehyde-3-phosphate or pyruvate synthesized solely through the glycolysis will only contain intact bonds. Conversely, if the same molecule were synthesized through the pentose phosphate pathway, it will contain many biosynthetic bonds reflecting the extensive bond breakage and reassembly in the non-oxidative section of this pathway.

For another illustration, consider the synthesis of the succinate, a potential platform chemical (28), by engineered *Escherichia coli* cells. San and coworkers engineered *E. coli* to synthesize succinate from glucose through two separate but concurrently operated metabolic pathways (29) (Fig. 4). In one pathway, succinate is synthesized from oxaloacetate formed anaplerotically by condensation of phosphoenolpyruvate and CO_2 (Fig. 4b). Assuming that flux through the pentose phosphate pathway is low (and thereby that the bond integrities of the supplied glucose are preserved in phosphoenolpyruvate), succinate synthesized through the pathway in Fig. 4b carries intact bonds between C-1 and C-2, C-2 and C-3 and a biosynthetic bond between C-3 and C-4 (or equivalently, intact bonds between C-2 and C-3, C-3 and C-4 and a biosynthetic bond between C-1 and C-2 due to the symmetry of the succinate molecule). A second pathway uses the glyoxylate cycle to synthesize succinate with an intact bond between C-1 and C-2 and/or C-3 and C-4, depending on whether (1) isocitrate or (2) acetyl CoA and glyoxylate are the reactants. The equivalent carbon atoms C-2 and C-3 of succinate are detectable as a single peak on a [^{13}C, ^{1}H] correlation map; their overlap is due to the symmetry of the molecule. In a labeling experiment employing ~5–10 % U-^{13}C glucose diluted with naturally abundant glucose as the sole carbon source, the C-2/C-3 atom of succinate synthesized through pathway in Fig. 4b will exhibit a doublet arising from intact and biosynthetic bonds on two sides of the atom and a double doublet arising from intact bonds on both sides of the atom. In contrast, the same atom of succinate synthesized through the pathway in Fig. 4c will exhibit

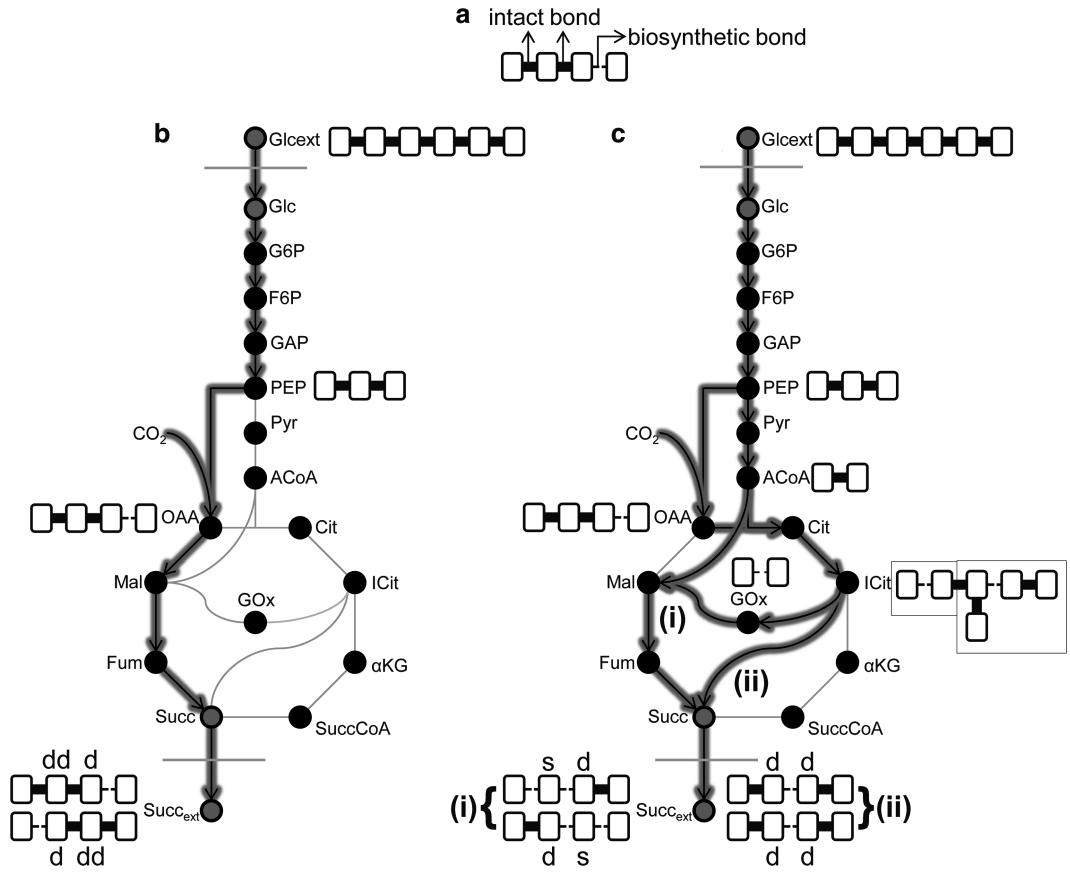

Fig. 4. NMR-derived isotopomers can distinguish two pathways for succinate biosynthesis. (**a**) Notation for intact and biosynthetic bonds. see text for definitions. Engineered *E. coli* cells can synthesize succinate from glucose through the two separate metabolic pathways depicted in (**b**) and (**c**) (29). Pathway (**b**) synthesizes succinate with intact bonds between {C-1 and C-2, C-2 and C-3} and {C-2 and C-3, C-3 and C-4}. Pathway (**c**) synthesizes succinate either with an intact bond between {C-1 and C-2} and/or {C-3 and C-4}, depending on whether route (*i*) or (*ii*) in this pathway is used. The equivalent carbon atoms C-2 and C-3 of succinate are detectable on a [^{13}C, ^{1}H] correlation map; they overlap due to the symmetry of the molecule. The fine structure of a succinate molecule synthesized through pathway (**b**) will exhibit a doublet/double doublet mixture; that of a succinate molecule synthesized through pathway (**c**) will exhibit a singlet/doublet mixture. This diagram depicts biochemical reactions participating in a pathway with *solid lines* in *dark color* (*red* in the online version of this chapter) and reactions not participating with *light*, *gray lines*. *ACoA* acetyl CoA, *Cit* citrate, *F6P* fructose-6-phosphate, *Fum* fumarate, *G6P* glucose-6-phosphate, *GAP* glyceraldehyde-3-phosphate, *Glc* glucose, *Glc$_{ext}$* extracellular glucose, *GOx* glyoxylate, *ICit* isocitrate, *αKG* α-ketoglutarate, *Mal* malate, *OAA* oxaloacetate, *PEP* phosphoenolpyruvate, *Pyr* pyruvate, *Succ* succinate, *SuccCoA* succinyl CoA, and *Succ$_{ext}$* extracellular succinate.

a singlet and/or a doublet. Thus, inspection of the fine structure of succinate will enable understanding which pathways contributed to its biosynthetic history.

3.4.2. Estimating Metabolic Fluxes from Labeling Information

Labeling information not only provides information on the pathways that differ in carbon skeletal rearrangements, but on the relative contributions of these pathways toward the synthesis of a compound of interest. For example, in both the glycolysis–pentose phosphate pathway example and the succinate biosynthesis example

described above, it is straightforward to write material balances on the isotopomers that can permit evaluation of flux through the competing pathways on the basis of measured NMR multiplet intensities or bondomer abundances. A full illustration of this concept is provided in (8). In this regard, it is important to remember that the reversible reactions and cyclic carbon flow can cause one pathway to use the labeling patterns (isotopomers, bondomers, or EMUs) generated by another pathway as reactants, resulting in a much larger number of labeling patterns than would occur in the absence of reversibilities and cycles. In such cases, comprehensive isotopomer (30), bondomer (25, 26), or EMU (27) balancing will have to be utilized for flux evaluation.

4. Notes

1. To obtain a concentrated sample, directly hydrolyze the cell pellet instead of a protein extract. A cell pellet hydrolysate will contain significant residue; eliminate it using 0.22 μm Spin-X centrifuge tube filters.
2. For acceptable NMR probe tuning, it is necessary that the sample be relatively free of ions. Ions introduced through HCl can be removed by evaporation and lyophilization; however, ions introduced through salty protein extraction reagents will need to be dialyzed out of the protein. To dialyze, use a Slide-A-Lyzer cassette and follow the manufacturer's instructions.
3. It is essential to eliminate all acid (see Note 2) and water from the sample, as the final sample has to be reconstituted in D_2O. To accomplish this, some optimization of the lyophilization time may be necessary.
4. Overlapping multiplet peaklets in 1-D slices of [^{13}C, ^{1}H] correlation maps can be deconvoluted by acquiring spectra that are *J*-scaled along the F1 dimension using pulse sequences described in, e.g., ref. (31). *J*-scaling increases multiplet separation by an even integral factor *J* and eliminates multiplet overlap. *J*-scaling factors of 6 or 8 are useful; however, the increased peaklet separation comes at the cost of resolution. We find it more efficient to acquire a non-*J*-scaled spectrum and use the computational peak deconvolution (curve fitting) methodology described in this chapter.
5. An important step in quality control of the extracted peak is ensuring that the extract contains a single peak and not two or more overlapping peaks. For this, manually inspect the peak extract in NMRViewJ to ensure that only near-zero or negative values occur at the end of the data set or in between the peaklets. see Fig. 3 for a depiction of a "clean" peak.

6. If the curve-fitting algorithm does not return an accurate fit, examine whether it has converged on extreme values of parameters listed in the "initial guess, limits, and stepsizes for the parameter vector, 'par'" block of PeakAnneal.m.
7. The parameters of the simulated annealing global optimization procedure used in the curve-fitting algorithm may require some tuning to ensure fast and reproducible convergence to a global optimum (i.e., an accurate fit of the observed peak). The "annealing parameters" section of PeakAnneal.m contains a set of parameter values that works for our spectral extracts. Typically, adjusting the parameters in the following directions results in greater accuracy, albeit at the cost of speed: increasing the value of the initial temperature (T_0), decreasing the value of the stop temperature (T_stop), and bringing the values of the temperature reduction factor (T_reduce) and the stepsize reduction factor (eps_reduce) closer to 1.

5. Author Contributions

S.N. contributed critically to the sections on sample preparation, spectral acquisition, and spectral processing. M.E.J. developed and wrote the current versions of the MATLAB scripts in the NMRisotopomer package and contributed to the spectral processing section. D.T. contributed to the section on metabolic interpretation of isotopomer abundances. V.T. provided critical guidance in spectral acquisition and contributed to the relevant sections. G.S. conceived the chapter, wrote significant sections, and prepared the final version.

Acknowledgments

This work was partially funded by the National Science Foundation (award number IOS 0922650) as well as Department of Chemical and Biomolecular Engineering, University of Maryland, and A. James Clark School of Engineering, University of Maryland (faculty startup grant to GS).

References

1. Wiechert W, Schweissgut O, Takanaga H, Frommer WB (2007) Fluxomics: mass spectrometry versus quantitative imaging. Curr Opin Plant Biol 10:323–330
2. Stephanopoulos G, Vallino J (1991) Network rigidity and metabolic engineering in metabolite overproduction. Science 252:1675–1681

3. Kim HK, Choi YH, Verpoorte R (2011) NMR-based plant metabolomics: where do we stand, where do we go? Trends Biotechnol 29:267–275
4. Kruger NJ, Troncoso-Ponce MA, Ratcliffe RG (2008) 1 H NMR metabolite fingerprinting and metabolomic analysis of perchloric acid extracts from plant tissues. Nat Protoco 3:1001–1012
5. Varma A, Palsson BO (1994) Metabolic flux balancing: basic concepts, scientific and practical use. Nat Biotechnol 12:994–998
6. Okumoto S, Jones A, Frommer WB (2012) Quantitative imaging with fluorescent biosensors: advanced tools for spatiotemporal analysis of biodynamics in cells. Annu Rev Plant Biol 63:663–706
7. Wiechert W (2001) 13C metabolic flux analysis. Metab Eng 3:195–206
8. Sriram G, Fulton DB, Shanks JV (2007) Flux quantification in central carbon metabolism of Catharanthus roseus hairy roots by 13C labeling and comprehensive bondomer balancing. Phytochemistry 68:2243–2257
9. Szyperski T (1995) Biosynthetically directed fractional 13C-labeling of proteinogenic amino acids. An efficient analytical tool to investigate intermediary metabolism. Eur J Biochem 232:433–448
10. Szyperski T (1998) 13C-NMR, MS and metabolic flux balancing in biotechnology research. Q Rev Biophys 31:41–106
11. Giraudeau P, Massou S, Robin Y, Cahoreau E, Portais J-C, Akoka S (2011) Ultrafast quantitative 2D NMR: an efficient tool for the measurement of specific isotopic enrichments in complex biological mixtures. Anal Chem 83:3112–3119
12. Sriram G, Fulton DB, Iyer VV, Peterson JM, Zhou R, Westgate ME, Spalding MH, Shanks JV (2004) Quantification of compartmented metabolic fluxes in developing soybean embryos by employing biosynthetically directed fractional 13C labeling, two-dimensional [13C, 1 H] nuclear magnetic resonance, and comprehensive isotopomer balancing. Plant Physiol 136:3043–3057
13. Iyer VV, Sriram G, Fulton DB, Zhou R, Westgate ME, Shanks JV (2008) Metabolic flux maps comparing the effect of temperature on protein and oil biosynthesis in developing soybean cotyledons. Plant Cell Environ 31:506–517
14. Yuan Y, Hoon Yang T, Heinzle E (2010) (13)C metabolic flux analysis for larger-scale cultivation using gas chromatography-combustion-isotope ratio mass spectrometry. Metab Eng 12:392–400
15. Klapa MI, Aon JC, Stephanopoulos G (2003) Ion-trap mass spectrometry used in combination with gas chromatography for high-resolution metabolic flux determination. Biotechniques 34:832–836, 838, 840 passim
16. Blank L, Desphande R, Schmid A, Hayen H (2012) Analysis of carbon and nitrogen co-metabolism in yeast by ultrahigh-resolution mass spectrometry applying 13C- and 15 N-labeled substrates simultaneously. Anal Bioanal Chem 403:2291–2305
17. Harris RK (1983) Nuclear magnetic resonance spectroscopy: a physiochemical view. Pitman Press, London
18. Grant DM, Harris RK, Ernst RR (1996) Encyclopedia of nuclear magnetic resonance. Angewandte Chemie Int Edit 35:2679
19. van Winden W, Schipper D, Verheijen P, Heijnen J (2001) Innovations in generation and analysis of 2D [13C, 1 H] COSY NMR spectra for metabolic flux analysis purposes. Metab Eng 3:322–343
20. Sriram G, Iyer VV, Fulton DB, Shanks JV (2007) Identification of hexose hydrolysis products in metabolic flux analytes: a case study of levulinic acid in plant protein hydrolysate. Metab Eng 9:442–451
21. Bodenhausen G, Ruben DJ (1980) Natural abundance nitrogen-15 NMR by enhanced heteronuclear spectroscopy. Chem Phys Lett 1:30
22. Rucker SP, Shaka AJ (1989) Broadband homonuclear cross polarization in 2D N.M.R. using DIPSI-2. Mol Phys 68:509–517
23. Griesinger C, Otting G, Wuethrich K, Ernst RR (1988) Clean TOCSY for proton spin system identification in macromolecules. J Am Chem Soc 110:7870–7872
24. Locatelli M (2002) Simulated annealing algorithms for continuous global optimization, Handbook of global optimization. Kluwer Academic Publisher, The Netherlands
25. van Winden WA, Heijnen JJ, Verheijen PJT (2002) Cumulative bondomers: a new concept in flux analysis from 2D [13C, 1 H] COSY NMR data. Biotechnol Bioeng 80:731–745
26. Sriram G, Shanks JV (2004) Improvements in metabolic flux analysis using carbon bond labeling experiments: bondomer balancing and Boolean function mapping. Metab Eng 6:116–132
27. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2007) Elementary metabolite units (EMU): a novel framework for modeling isotopic distributions. Metab Eng 9:68–86
28. Nikolau BJ, Perera MA, Brachova L, Shanks B (2008) Platform biochemicals for a biorenewable chemical industry. Plant J 54:536–545

29. Sanchez AM, Bennett GN, San K-Y (2005) Novel pathway engineering design of the anaerobic central metabolic pathway in Escherichia coli to increase succinate yield and productivity. Metab Eng 7:229–239
30. Wiechert W, Möllney M, Isermann N, Wurzel M, de Graaf AA (1999) Bidirectional reaction steps in metabolic networks: III. Explicit solution and analysis of isotopomer labeling systems. Biotechnol Bioeng 66:69–85
31. Willker W, Flögel U, Leibfritz D et al (1997) Ultra-high-resolved HSQC spectra of multiple-13C-labeled biofluids. J Magn Reson 125:216–219

Chapter 17

Using Multiple Tracers for ^{13}C Metabolic Flux Analysis

Maciek R. Antoniewicz

Abstract

^{13}C-Metabolic flux analysis (^{13}C-MFA) is a powerful technique for quantifying intracellular metabolic fluxes in living cells. These in vivo fluxes provide important information on the physiology of cells in culture, which can be used for metabolic engineering purposes and serve as inputs for systems biology modeling. The ^{13}C-MFA technique consists of several steps: (1) selecting appropriate tracers for a given system of interest, (2) performing isotopic labeling experiments, (3) measuring isotopic labeling distributions in metabolic products, (4) estimating metabolic fluxes using least-squares regression, and (5) evaluating the goodness of fit and computing confidence intervals for estimated fluxes. In this chapter, we provide guidelines for performing ^{13}C-MFA studies using multiple isotopic tracers, a technique that is especially useful for elucidating fluxes in complex biological systems where multiple carbon sources are present. Here, as an example, we describe key steps and decision points for designing ^{13}C-MFA studies for microbes grown on mixtures of glucose and xylose. The general concepts described in this chapter are applicable to many other biological systems. For example, the same procedures can be applied to design ^{13}C-MFA studies in mammalian cells, which are generally grown in complex media containing multiple substrates such as glucose and amino acids.

Key words: Fluxomics, Metabolism, Tracer experiment, Metabolic engineering, Metabolic network model, Systems biology, Metabolic flux, Biological modeling, Experiment design, Isotopomer

1. Introduction

Knowledge of intracellular metabolic fluxes provides important insights into cellular physiology that can be applied to engineer the metabolic, regulatory, and phenotypic characteristics of organisms (1, 2). In the past two decades, rigorous techniques have been developed to estimate intracellular metabolic fluxes in increasingly complex systems (3, 4). Currently, the most advanced technique for flux elucidation is ^{13}C-based metabolic flux analysis (^{13}C-MFA). Application of ^{13}C-MFA in metabolic studies requires that a number of computational, experimental, and analytical steps are successfully completed in sequence (5). Although the amount of effort

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_17,

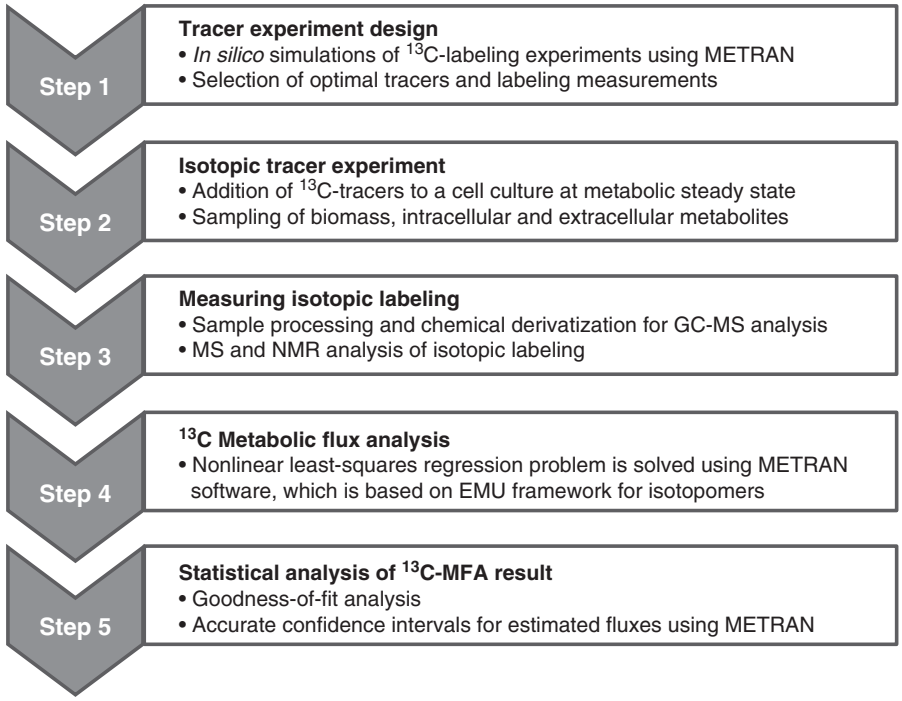

Fig. 1. Five main stages of ^{13}C metabolic flux analysis (^{13}C-MFA).

that is required from investigators to complete these studies is significant, a well-executed ^{13}C-MFA study provides quantitative information on cellular physiology that more than justifies the investment of time and resources; at present, there is no other technique that provides the same level of detail on in vivo metabolism. This chapter provides step-by-step guidelines for investigators to perform successful ^{13}C-MFA studies, with special emphasis on using multiple isotopic tracers for flux elucidation. Important steps and decision points in the design of labeling experiments are identified and discussed.

^{13}C-MFA studies can be broken down into five main stages (Fig. 1). The first stage is the tracer experiment design stage, which consists largely of in silico simulations based on prior knowledge regarding the metabolism of the organism and preliminary growth data (6). The second stage is the actual ^{13}C-tracer experiment, in which tracers are introduced at selected time points during the culture, followed by a labeling period and biomass sampling. In the third, analytical stage, the samples are processed and analyzed using techniques such mass spectrometry to quantify isotopic labeling distributions (7). Fourth, metabolic fluxes are estimated using programs such as METRAN (see Note 1), which solves a nonlinear regression problem where the goal is to find a set of fluxes that provides the best fit between the measured and simulated labeling

data (3). Finally, in the fifth stage, detailed statistical analysis is performed to determine the goodness of fit and quantify accurate confidence intervals for the estimated fluxes (8). Recent advances in computational methods for ^{13}C-MFA, based on the Elementary Metabolite Units (EMU) framework (3, 9, 10), and the availability of dedicated software packages for ^{13}C-MFA, such as the METRAN software (11), have greatly simplified the design and analysis of tracer experiments. The ^{13}C-MFA technique is now routinely used by many research labs around the world for metabolic flux studies.

In this chapter, we discuss the design and analysis of ^{13}C-labeling studies to elucidate fluxes in microbial systems where the cells are grown on mixtures of glucose and xylose, the two predominant sugars in cellulosic biomass. Flux studies of mixed sugar fermentations are becoming increasingly important in the metabolic engineering field, as many researchers are interested in developing microbial strains that can produce next-generation biofuels from renewable cellulosic resources (12). The use of multiple carbon sources in these cultures presents both new challenges and new opportunities for ^{13}C-MFA. While there are many studies that discuss how to best design and analyze ^{13}C-tracer experiments when only a single-labeled substrate is used (13, 14), very few studies provide specific guidelines for designing tracer studies using multiple isotopic tracers (6, 10). This is the main objective of this chapter. Several software packages for ^{13}C-MFA can be used for designing and analyzing multiple-tracer studies. Here, we will use the METRAN software for ^{13}C-MFA, developed by Dr. Antoniewicz, which offers several advanced features for optimal tracer experiment design. In the remainder of this chapter, we will provide a detailed discussion regarding the challenges in designing multiple-tracer studies and step-by-step guidelines for successfully implementing this approach for ^{13}C-MFA.

2. Materials

The following software packages and preliminary information are needed for tracer experiment design:

1. MATLAB 6.8 (or higher) installed on PC, including Optimization Toolbox and Statistics Toolbox (Mathworks Inc., Natick, MA).
2. METRAN software for ^{13}C-MFA (see Note 1).
3. A metabolic network model for the organism, consisting of stoichiometry and carbon-atom transitions for the reactions in central carbon metabolism and amino acid metabolism (see Note 2).

4. Growth profiles for the organism, preferably on defined growth medium with glucose and xylose as the only carbon sources (see Note 3).
5. Metabolic flux maps for the organism from literature (optional).

3. Methods

3.1. Importance of Tracer Experiment Design

Tracer experiment design is the first integral part of any ^{13}C-MFA study (10). Unfortunately, this step is often not given enough coverage in the literature, which is why this chapter devotes significant attention to this aspect of ^{13}C-MFA. Tracer experiment design goes beyond the actual selection of isotopic tracers. Other important considerations at this stage, which will be covered in this chapter, include the following: which measurements should be collected; when should the tracers be introduced during a culture; how should the tracers be introduced (e.g., continuous feeding vs. bolus addition); and how long should the labeling period be (15).

3.2. Selection of Isotopic Tracers

First, we discuss considerations related to the selection of appropriate ^{13}C-tracers for a given biological system, metabolic network model, and a given set of isotopic measurements, e.g., MS or NMR measurements. It is important to note that flux resolution depends on both the tracer selection and the choice of isotopic measurements, and ideally, the choice of labeling should be addressed in conjunction, rather than in isolation, of the set of measurements (10). However, for practical reasons we will first cover tracer selection, i.e., assuming that isotopic measurements have been selected already. Investigators often select isotopic tracers by convention, i.e., use the same tracers as others have used before; by following common heuristics, for example, it is often assumed that a mixture of [1-^{13}C]glucose and [U-^{13}C]glucose tracers will improve flux results compared to either tracer alone; and by considering the cost of available tracers. For glucose there are 64 ($=2^6$) possible ^{13}C-tracers, and for xylose there are 32 ($=2^5$) possible ^{13}C-tracers, assuming that each carbon atom can be either labeled (i.e., ^{13}C) or unlabeled (i.e., ^{12}C). Of these 96 possible tracers, only a handful of tracers are commercially available at prices that are practical for in vivo flux studies (i.e., less than \$3,000/g, see Note 4). Table 1 provides a partial list of commercially available glucose and xylose tracers. In addition to these tracers, custom synthesis of isotopic tracers with different labeling patterns is also a viable option, although this typically increases the price of tracers beyond \$3,000/g (see Note 5).

The choice of which isotopic tracers to apply in a study is not a trivial one, and different optimal tracers may exist for different organisms and culture conditions (6, 16). Recently, several rational

Table 1
Partial list of available glucose and xylose tracers for ^{13}C-MFA

Glucose tracers	Approx. price ($/g)	Xylose tracers	Approx. price ($/g)
[1-^{13}C]glucose	$100	[1-^{13}C]xylose	$700
[U-^{13}C]glucose	$150	[U-^{13}C]xylose	$900
[2-^{13}C]glucose	$500	[2-^{13}C]xylose	$1,000
[1,2-^{13}C]glucose	$700	[3-^{13}C]xylose	$1,500
[6-^{13}C]glucose	$1,000	[5-^{13}C]xylose	$1,900
[3-^{13}C]glucose	$1,200		
[4-^{13}C]glucose	$1,600		

and trial-and-error approaches have been proposed to guide the tracer selection process (6, 10, 13, 16). For simplicity, here we will use a traditional trial-and-error approach. The goal of optimal tracer selection is to maximize the resolution of metabolic fluxes of interest (see Note 6). It is likely that there will not be one set of tracers that maximizes the resolution of all fluxes in a given network model. Therefore, at this stage it is important to set priorities regarding which metabolic fluxes are of most interest for a given study. The strategy for isotopic tracer selection is as follows. First, isotopic labeling measurements are simulated in silico for an assumed metabolic network model and for several flux maps that may be available from literature, or some initial guesses of expected flux maps. Empirically, it was found that tracer selection is not strongly dependent on the actual values of fluxes used for in silico simulations; thus, at this stage, a detailed knowledge of pathway fluxes is not critical (see Note 7). In general, it is best to avoid setting fluxes to zero in these in silico simulations to ensure that metabolic fluxes can be optimally resolved under any flux condition. Data from in silico isotopic tracer simulations are then used to estimate fluxes by ^{13}C-MFA using METRAN, followed by calculation of confidence intervals for the estimated fluxes, also using METRAN (8). Tracer sets that produce the smallest confidence intervals, or at least, confidence intervals that are sufficiently small for the fluxes of interest are considered optimal tracer choices.

3.3. Selection of Isotopic Tracers Using METRAN

The METRAN software for ^{13}C-MFA provides all of the necessary capabilities for in silico simulations, flux estimation, and statistical analysis for optimal tracer selection. Figure 2 shows a screenshot of the METRAN software. After loading a predefined metabolic network model into METRAN (see Note 1), the user specifies which isotopic measurements and extracellular rates will be simulated, and

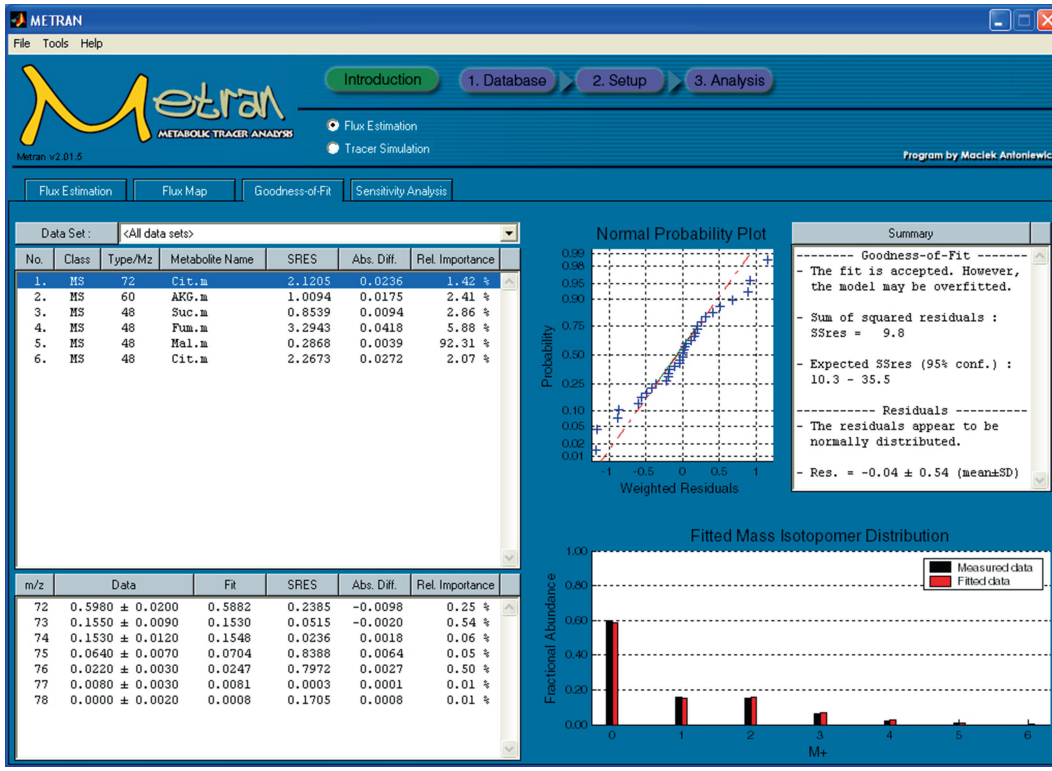

Fig. 2. Screenshot of METRAN software for ^{13}C-MFA.

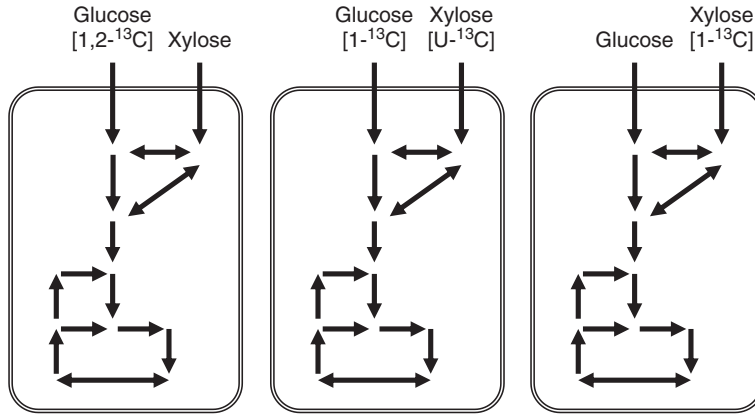

Fig. 3. Examples of tracer experiment designs for ^{13}C-MFA.

predefines different tracer schemes that will be evaluated in silico. Figure 3 illustrates several examples of tracer schemes that can be considered for ^{13}C-MFA study of mixed glucose/xylose fermentation. At the two extremes, the investigator may consider using a single pure glucose tracer (e.g., 100 % [1,2-^{13}C]glucose) or a single pure xylose tracer (e.g., 100 % [1-^{13}C]xylose). In addition,

mixtures of ^{13}C-tracers of the same substrate can also be considered (e.g., 25 % [1-^{13}C]glucose + 75 % [U-^{13}C]glucose, or 50 % [U-^{13}C] xylose + 50 % natural xylose); and finally, multiple-labeled substrates can be applied together, either as pure tracers (e.g., 100 % [1-^{13}C] glucose + 100 % [U-^{13}C]xylose), or mixtures of one or both. It should be clear that the number of possible tracer schemes that can be considered can be quite large, especially when mixtures of tracers are evaluated. The investigator must not only decide which two (or more) tracers to use, but also at what ratio the tracers should be mixed. We recommend evaluating at least three ratios of tracers for mixtures at the initial screening stage: 25/75, 50/50, and 75/25. This ratio can be fine-tuned later, once the list of possible tracers is narrowed down to a few candidates after the first round of in silico simulations and ^{13}C-MFA evaluations. This iterative trial-and-error method to tracer selection should, in most cases, produce several feasible tracer sets that yield satisfactory resolution of fluxes of interest for the given network model (see Note 6). If this is not the case, then it may be necessary to consider alternative, more involved approaches for optimal tracer selection. One such method is the rational EMU-basis vector methodology developed by Crown and Antoniewicz (6, 10). Discussion of this and other alternative approaches is beyond the scope of this chapter.

3.4. Selection of Measurements for Isotopic Labeling Studies

Next, we discuss the choice of isotopic labeling measurements for ^{13}C-MFA. At the moment, there are two main techniques for measuring isotopic labeling of molecules: NMR (17) and mass spectrometry coupled with either liquid (LC-MS) (18) or gas (GC-MS) chromatography (7, 19). The specific measurement technique that is used in flux studies is often limited by the available resources at the institutions of the investigators. The NMR technique provides two types of data: ^{13}C-labeling of specific carbon atoms, the so-called fractional enrichments, and more detailed labeling information for specific carbon atoms through fine spectra (20). A disadvantage of the NMR technique is that it requires a fairly large amount of sample, long analysis times, and expensive equipment. As a result, NMR is currently mainly used for ^{13}C analyses of proteinogenic amino acids. Generally, NMR cannot be used for intracellular metabolite measurements, with the possible exception of glutamate that may be present at high concentrations in some cells. Mass spectrometry, on the other hand, is widely accessible, less costly, and provides sensitive detection of molecular enrichment. MS methods are sensitive enough to detect ^{13}C-labeling in many intracellular metabolites. In particular, the GC-MS method has been extensively used for ^{13}C-MFA applications (21, 22). In GC-MS, a sample of extracted metabolites is first chemically derivatized (i.e., to clean up the sample and make the molecules more volatile), followed by fractionation by GC and ionization by electron impact (or chemical) ionization (see Note 8).

The resulting ions are then characterized based on their mass-to-charge ratio (m/z). A typical mass spectrum contains many peaks corresponding to different fragments of the detected compound. Using the MS method, the relative amount of each mass isotopomer is measured, providing the so-called full spectrum mass isotopomer distribution (MID) of each fragment. A recent advance in the field of isotopomer measurements is the use of tandem MS for isotopic analyses (23, 24). As an example, Choi and Antoniewicz developed analytical methods for measuring ^{13}C-labeling of molecules with GC-MS/MS and computational algorithms for incorporating tandem MS data into flux calculations (23, 25). It was demonstrated that tandem MS provides inherent advantages for both isotopic labeling measurements and for ^{13}C-MFA analyses compared to traditional full-scan MS and NMR. Given that the cost of tandem MS machines is rapidly decreasing, it is expected that tandem MS-based flux studies will become more widely used in the near future.

3.5. Introduction of Tracers in Cell Cultures

The next decision in tracer experiment design is to consider when the isotopic tracers should be introduced during a cell culture. For continuous cultures, i.e., so-called chemostats, tracers are introduced simply by switching the feed medium from an unlabeled feed (i.e., where all carbon sources are unlabeled) to a ^{13}C-labeled feed. It is important that the labeled feed is chemically identical to the unlabeled feed (i.e., same metabolite concentrations) to ensure that the cell culture is minimally perturbed and cellular metabolism remains at metabolic steady state. The timing of tracer introduction for batch and fed-batch cultures requires additional considerations. The easiest case is when tracers are added to the culture medium at the beginning of the experiment, which allows flux analysis of the initial growth phase (26). However, if the study requires measuring fluxes at other growth stages during a culture, then tracers must be introduced at later time points (21, 22). In general, tracers should be introduced as a bolus addition (i.e., as opposed to continuous addition of tracers) to ensure that substrate labeling in the medium is constant during the labeling period. Another important consideration is how much of each tracer should be added. There is a fine balance between adding too much tracer and adding too little. On the one hand, one must avoid adding too much tracer since a sudden increase in substrate concentration can induce metabolic shifts. On the other hand, if too little is added, then the effective labeling of substrates in the medium will be low, which has negative consequences for flux resolution in ^{13}C-MFA (21). As an example, consider a case where the cells are grown on a mixture of glucose and xylose, and after 20 h of culture glucose concentration has decreased to 20 mM and xylose concentration has decreased to 10 mM. Addition of a bolus of 10 mM 100 % [1-^{13}C]glucose and 10 mM 100 % [U-^{13}C]xylose will produce an effective glucose labeling of 33 % [1-^{13}C]glucose (i.e., 67 % natural glucose) and

xylose labeling of 50 % [U-^{13}C]xylose (i.e., 50 % unlabeled xylose). It is important to reevaluate if metabolic fluxes can still be sufficiently resolved for these reduced levels of ^{13}C-labeling, using the methods described in Subheading 3.3. As an example, in some systems it may be possible to resolve fluxes using 100 % [1-^{13}C] glucose + 100 % [U-^{13}C]xylose, but not with 33 % [1-^{13}C]glucose + 50 % [U-^{13}C]xylose. If the reduced labeling does not allow sufficient flux resolution, then there are several options to possibly remedy this problem: (1) other glucose and xylose tracers can be evaluated that may allow fluxes to be determined, even at lower effective ^{13}C-labeling; (2) more glucose and xylose tracers can be added to increase the effective ^{13}C-labeling of substrates in the medium; (3) tracers can be added at later time points during a culture, i.e., when the remaining concentrations of glucose and xylose are lower, which in effect produces higher percent ^{13}C-labeling of the substrates with the same addition of tracers; and finally (4) lower initial glucose and xylose concentrations can be considered so that the substrate concentrations will be lower throughout the culture, and the addition of the same amount of tracers will produce higher percent ^{13}C-labeling.

3.6. Length of Labeling Period

The final consideration in tracer experiment design is to determine the optimal length of the labeling period. A key requirement of ^{13}C-MFA is that cellular metabolism remains at metabolic steady state during the labeling period. In other words, it is assumed that intracellular metabolic fluxes are constant during ^{13}C-labeling incorporation. For chemostat cultures it is generally assumed that metabolism remains at metabolic steady state at all times, and therefore, there is no limit on the maximum length for the labeling period. In practice, however, the labeling period is limited by the cost of tracers. In batch and fed-batch cultures, the maximum labeling period is determined by the growth characteristics of the cells (21). For a given culture, the investigator must identify distinct growth phases, where at each growth phase, the metabolism can be assumed to be relatively time-invariant (Fig. 4). An easy method to identify different growth stages of a culture is to plot the

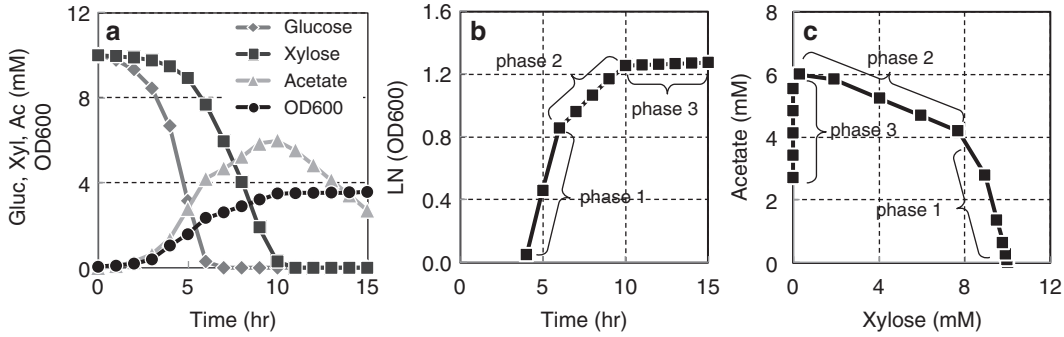

Fig. 4. Identification of distinct metabolic growth phases.

natural logarithm of the biomass concentration as a function of time. Figure 4b shows an example growth curve. In this case, three distinct growth phases are easily identified. Another method to identify different growth stages is to plot concentrations of substrates and products against each other (21, 27). For a distinct growth phase of a culture, it is expected that extracellular concentrations will change at the same relative rate, which produces straight lines in these plots. Figure 4c shows an example extracellular metabolite plot, from which the same three distinct metabolic phases can be identified.

In general, the length of the labeling period should be chosen such that sufficient time is allowed for ^{13}C-atoms to be incorporated into the measured metabolite pools (21). For intracellular metabolite measurements of ^{13}C-labeling, the labeling time can be relatively short due to rapid turnover of intracellular pools, e.g., on the order of minutes for microbial cells (28, 29), and up to several hours for mammalian cells (21). However, if proteinogenic amino acids are used for isotopic measurements, then significantly longer labeling times are required (22). For chemostat cultures, it is recommended that the labeling period is longer than one residence time. For batch and fed-batch cultures, the optimal labeling period depends strongly on when the tracers are introduced and how much biomass is present before the addition of tracers. If tracers are added at the beginning of a culture, i.e., when the initial biomass concentration is low, then the labeling period can be fairly short, on the order of a few hours. However, if the tracers are introduced at later times during a culture, i.e., when the biomass concentration is already high, then significantly longer labeling times may be needed (22). In general, at least 30 % of the sampled biomass should be produced during the labeling period. As an example, if tracers are initially introduced when the biomass concentration is 1 g/L, then the labeling period should be long enough such that the biomass concentration increases to 1.5 g/L, i.e., an increase of 0.5 g/L (=33 % new biomass, 0.5/1.5 = 0.33). For flux calculations, it is necessary to account for the presence of the preexisting unlabeled biomass (22). The METRAN software for ^{13}C-MFA automatically accounts for this effect using the concept of G-values (22). A G-value is defined as the mol-fraction of a measured metabolite pool produced during the labeling period, while (1-G) is the mol-fraction of preexisting (i.e., naturally labeled) metabolite pool. As an example, when METRAN estimates a G-value of 0.45, then this suggests that 45 % of the biomass was produced during the labeling period and 55 % of biomass was preexisting unlabeled biomass.

4. Notes

1. The METRAN software for ^{13}C-MFA is freely available for academic use. Requests for obtaining a copy of the METRAN software should be directed to the corresponding author at mranton@udel.edu. An example metabolic network model for *E. coli* and an example data set consisting of GC-MS mass isotopomers from an actual ^{13}C-labeling experiment are distributed with METRAN software.
2. ^{13}C-Metabolic flux analysis studies are performed using metabolic network models that contain only a subset of all known biochemical reactions for an organism. Most studies use metabolic network models that consist of central carbon metabolism (i.e., glycolysis, pentose phosphate pathway, TCA cycle, anaplerotic pathways, gluconeogenesis), a set of lumped amino acid pathways, and a lumped biomass reaction. In general, it is acceptable to lump linear metabolic pathways for the purposes of ^{13}C-MFA. It is important that all reactions in the metabolic network model are carbon-balanced and electron-balanced. Reversible reactions should be modeled as two separate reactions in the opposite directions, i.e., forward direction and backward direction. The use of genome-scale models is not practical for ^{13}C-MFA.
3. It is highly recommended that fully defined media are used for ^{13}C-MFA studies. The addition of yeast extract and other poorly characterized medium components (e.g., protein digests, serum) is not recommended, for two reasons. First, these medium components can significantly dilute ^{13}C-labeling, which has negative consequences for flux observability. Second, additional "dilution fluxes" must be included in the metabolic network models to account for dilution of isotopic labeling in order to obtain statistically acceptable fits of the data. In general, confidence intervals of estimated fluxes will be larger for models with the dilution fluxes compared to models without. In some cases, the addition of dilution fluxes can effectively produce a non-observable system.
4. The cost of tracers in ^{13}C-MFA studies is significant, and often limits the size of the cell culture experiment. As an example, a 1-L batch experiment with 10 g/L initial glucose concentration will cost at least \$1,000, i.e., even if "cheap" glucose tracers, such as [1-^{13}C]glucose and [U-^{13}C]glucose (~\$100/g), are used. The same experiment at 10 mL scale will cost only \$10. Therefore, miniaturization of cell culture experiments is critical for routine ^{13}C-MFA studies. In our lab, we generally perform tracer experiments with an effective culture volume of 5–10 mL.

It is important to validate that growth of cells at the smaller scales is representative of growth at larger scale fermentations.

5. The cost of custom synthesis of tracers depends both on the specific labeling pattern of the tracer and the amount of tracer that is purchased. In general, the more elaborate the labeling pattern is, the more the tracer will cost. As an example, [2,4,6-^{13}C]glucose is more difficult to synthesize, and it will in general be more expensive than, e.g., [2,3-^{13}C]glucose. Ordering more than 1 g of tracers will significantly reduce the price per gram.
6. Metabolic models used for ^{13}C-MFA studies typically have about 10–30 free fluxes. Most of the free fluxes are related to reaction reversibilities, and only a handful of the free fluxes represent net free fluxes in the model, which are of most interest in flux studies. A typical model will have between 3 and 8 net free fluxes. Thus, even though many reaction reversibilities will not be observable with ^{13}C-tracers, it is still possible to estimate most, if not all, of the net free fluxes with high precision using carefully selected tracers.
7. We recommend using between two and five different flux maps for in silico simulations and tracer evaluations. Examples of metabolic flux maps can be obtained from literature. In addition, we recommend constraining fluxes in these maps with measured values of external rates, such as substrate uptake rates, product accumulation rates, and cell growth rate, which are easily obtained from preliminary cell cultures without isotopic tracers.
8. Chemical derivatization is an integral part of GC-MS analyses of biological samples since most metabolites are not volatile enough for GC separation. Specialized derivatization methods have been developed for specific metabolite classes, e.g., sugars, amino acids, fatty acids, and organic acids. The use of these specialized derivatization methods offers the advantage of cleaning up the sample from potential contaminating molecules. This improves both the chromatographic separation, and also produces cleaner mass spectra with fewer co-eluting compared to LC-MS analyses.

References

1. Stephanopoulos G (1999) Metabolic fluxes and metabolic engineering. Metab Eng 1:1–11
2. Moxley JF, Jewett MC, Antoniewicz MR et al (2009) Linking high-resolution metabolic flux phenotypes and transcriptional regulation in yeast modulated by the global regulator Gcn4p. Proc Natl Acad Sci U S A 106:6477–6482
3. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2007) Elementary metabolite units (EMU): a novel framework for modeling isotopic distributions. Metab Eng 9:68–86
4. Leighty RW, Antoniewicz MR (2011) Dynamic metabolic flux analysis (DMFA): a framework for determining fluxes at metabolic non-steady state. Metab Eng 13:745–755

5. Niklas J, Schneider K, Heinzle E (2010) Metabolic flux analysis in eukaryotes. Curr Opin Biotechnol 21:63–69
6. Crown SB, Ahn WS, Antoniewicz MR (2012) Rational design of 13C-labeling experiments for metabolic flux analysis in mammalian cells. BMC Syst Biol 6:43
7. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2007) Accurate assessment of amino acid mass isotopomer distributions for metabolic flux analysis. Anal Chem 79:7554–7559
8. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2006) Determination of confidence intervals of metabolic fluxes estimated from stable isotope measurements. Metab Eng 8:324–337
9. Young JD, Walther JL, Antoniewicz MR et al (2008) An elementary metabolite unit (EMU) based method of isotopically nonstationary flux analysis. Biotechnol Bioeng 99:686–699
10. Crown SB, Antoniewicz MR (2012) Selection of tracers for (13)C-Metabolic Flux Analysis using Elementary Metabolite Units (EMU) basis vector methodology. Metab Eng 14:150–161
11. Yoo H, Stephanopoulos G, Kelleher JK (2004) Quantifying carbon sources for de novo lipogenesis in wild-type and IRS-1 knockout brown adipocytes. J Lipid Res 45:1324–1332
12. Stephanopoulos G (2008) Metabolic engineering: enabling technology for biofuels production. Metab Eng 10:293–294
13. Metallo CM, Walther JL, Stephanopoulos G (2009) Evaluation of 13C isotopic tracers for metabolic flux analysis in mammalian cells. J Biotechnol 144:167–174
14. Wittmann C, Heinzle E (2001) Modeling and experimental design for metabolic flux analysis of lysine-producing Corynebacteria by mass spectrometry. Metab Eng 3:173–191
15. van Winden WA, Heijnen JJ, Verheijen PJ et al (2001) A priori analysis of metabolic flux identifiability from (13)C-labeling data. Biotechnol Bioeng 74:505–516
16. Walther JL, Metallo CM, Zhang J et al (2012) Optimization of (13)C isotopic tracers for metabolic flux analysis in mammalian cells. Metab Eng 14:162–171
17. Szyperski T, Glaser RW, Hochuli M et al (1999) Bioreaction network topology and metabolic flux ratio analysis by biosynthetic fractional 13C labeling and two-dimensional NMR spectroscopy. Metab Eng 1:189–197
18. Iwatani S, Van Dien S, Shimbo K et al (2007) Determination of metabolic flux changes during fed-batch cultivation from measurements of intracellular amino acids by LC-MS/MS. J Biotechnol 128:93–111
19. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2011) Measuring deuterium enrichment of glucose hydrogen atoms by gas chromatography/mass spectrometry. Anal Chem 83:3211–3216
20. van Winden W, Schipper D, Verheijen P et al (2001) Innovations in generation and analysis of 2D [(13)C, (1)H] COSY NMR spectra for metabolic flux analysis purposes. Metab Eng 3:322–343
21. Ahn WS, Antoniewicz MR (2011) Metabolic flux analysis of CHO cells at growth and non-growth phases using isotopic tracers and mass spectrometry. Metab Eng 13:598–609
22. Antoniewicz MR, Kraynie DF, Laffend LA et al (2007) Metabolic flux analysis in a nonstationary system: fed-batch fermentation of a high yielding strain of E. coli producing 1,3-propanediol. Metab Eng 9:277–292
23. Choi J, Antoniewicz MR (2011) Tandem mass spectrometry: a novel approach for metabolic flux analysis. Metab Eng 13:225–233
24. Jeffrey FM, Roach JS, Storey CJ et al (2002) 13C isotopomer analysis of glutamate by tandem mass spectrometry. Anal Biochem 300:192–205
25. Choi J, Grossbach MT, Antoniewicz MR (2012) Measuring complete isotopomer distribution of aspartate using gas chromatography/tandem mass spectrometry. Anal Chem 84:4628–4632
26. Ahn WS, Antoniewicz MR (2012) Towards dynamic metabolic flux analysis in CHO cell cultures. Biotechnol J 7:61–74
27. Deshpande R, Yang TH, Heinzle E (2009) Towards a metabolic and isotopic steady state in CHO batch cultures for reliable isotope-based metabolic profiling. Biotechnol J 4:247–263
28. Costenoble R, Muller D, Barl T et al (2007) 13C-Labeled metabolic flux analysis of a fed-batch culture of elutriated Saccharomyces cerevisiae. FEMS Yeast Res 7:511–526
29. Crown SB, Indurthi DC, Ahn WS et al (2011) Resolving the TCA cycle and pentose-phosphate pathway of Clostridium acetobutylicum ATCC 824: isotopomer analysis, in vitro activities and expression analysis. Biotechnol J 6:300–305

Chapter 18

Isotopically Nonstationary ^{13}C Metabolic Flux Analysis

Lara J. Jazmin and Jamey D. Young

Abstract

^{13}C metabolic flux analysis (MFA) is a powerful approach for quantifying cell physiology based upon a combination of extracellular flux measurements and intracellular isotope labeling measurements. In this chapter, we present the method of isotopically nonstationary ^{13}C MFA (INST-MFA), which is applicable to systems that are at metabolic steady state, but are sampled during the transient period prior to achieving isotopic steady state following the introduction of a ^{13}C tracer. We describe protocols for performing the necessary isotope labeling experiments, for quenching and extraction of intracellular metabolites, for mass spectrometry (MS) analysis of metabolite labeling, and for computational flux estimation using INST-MFA. By combining several recently developed experimental and computational techniques, INST-MFA provides an important new platform for mapping carbon fluxes that is especially applicable to animal cell cultures, autotrophic organisms, industrial bioprocesses, high-throughput experiments, and other systems that are not amenable to steady-state ^{13}C MFA experiments.

Key words: Metabolic flux analysis, Isotopically nonstationary, Isotopomer analysis, Mass spectrometry, Elementary metabolite unit, Isotopomer modeling

1. Introduction

The ability to quantitatively map intracellular carbon fluxes using isotope tracers and metabolic flux analysis (MFA) is critical for identifying pathway bottlenecks and elucidating network regulation in biological systems, especially those that have been engineered to alter their native metabolic capacities (1, 2). Typically, MFA relies on the assumption of both metabolic and isotopic steady state. Achieving this situation experimentally involves (1) equilibrating the system in a stable metabolic state, (2) introducing an isotopically labeled substrate without perturbing the metabolic steady state, (3) allowing the system to establish a new isotopic steady state that is dictated by the underlying metabolic fluxes, and (4) measuring isotopic labeling in the fully equilibrated system

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_18,

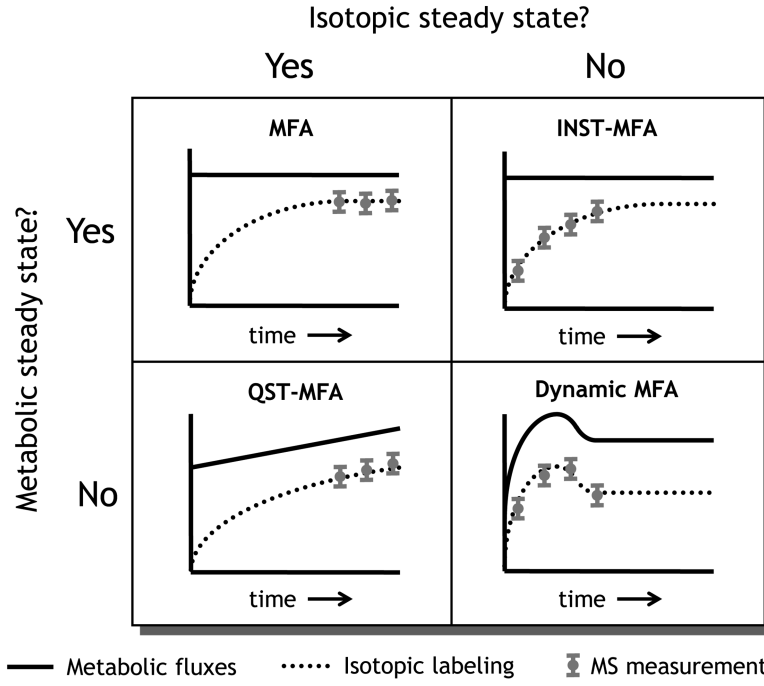

Fig. 1. Overview of different MFA methodologies. The relative speed of metabolic and isotopic dynamics will influence the type of MFA study performed. The *upper-left panel* shows the typical MFA setup under both metabolic and isotopic steady state. The *upper-right panel* shows INST-MFA at metabolic steady state, but not isotopic steady state. The *bottom-left panel* shows quasi-stationary MFA (QST-MFA) at isotopic quasi-steady state, but not metabolic steady state. The *bottom-right panel* shows Dynamic MFA, which is at neither metabolic nor isotopic steady state.

(Fig. 1, upper-left panel). Depending on the relative speed of metabolic versus isotopic dynamics, however, other experimental scenarios can be envisioned. If the isotopic labeling responds quickly to metabolic perturbations, quasi-stationary MFA (QST-MFA; Fig. 1, lower-left panel) can be applied to obtain a series of instantaneous snapshots that describes the variation in metabolic fluxes over time (3, 4). Because of the quasi-steady-state assumption on isotopic labeling, the isotopomer balances remain algebraic in nature, and the computational treatment applied to each time slice is essentially identical to that of steady-state MFA. Conversely, if labeling occurs slowly but metabolism is maintained in a fixed state, isotopically nonstationary MFA (INST-MFA; Fig. 1, upper-right panel) can be applied to determine fluxes from transient isotope labeling measurements (5). This requires repeated solution of differential balance equations that describe the time-dependent labeling of intermediate metabolites, while iteratively adjusting the flux parameters in those equations to match the experimental measurements. Finally, when measurements are obtained under both metabolically and isotopically nonstationary conditions, a fully

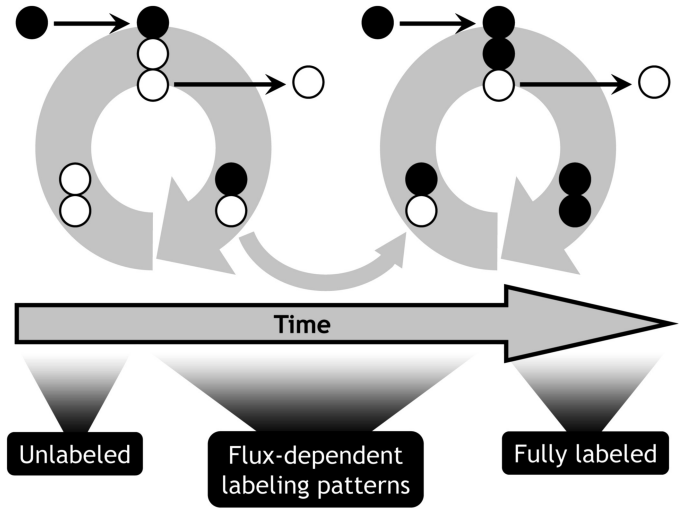

Fig. 2. Example of carbon labeling in an autotrophic system. Following the introduction of a labeled tracer to the system, intracellular metabolites become gradually labeled over time. Once steady-state labeling is achieved, all metabolites are uniformly ^{13}C labeled irrespective of fluxes and intracellular pool sizes. Labeling patterns observed during the isotopically transient period, however, can be computationally analyzed to determine fluxes.

dynamic modeling approach is required to estimate time-dependent fluxes (Fig. 1, lower right). This scenario has been referred to as Dynamic MFA (6–8), and it is an area of ongoing research for which appropriate methodologies and software tools are currently under development.

In this chapter, we present up-to-date protocols for performing INST-MFA under conditions of metabolic steady state, which has now matured to the point where optimized methodologies and software tools are rapidly emerging. INST-MFA holds a number of unique advantages over approaches that rely solely upon steady or quasi-steady isotopomer measurements.

1. ^{13}C INST-MFA can be applied to estimate fluxes in autotrophic or methylotrophic systems, which consume only single-carbon substrates (9, 10). This task is impossible with stationary ^{13}C MFA due to the fact that all carbon atoms in the system are derived from the same source and therefore will become uniformly labeled at steady state regardless of the flux distribution (Fig. 2). However, the transient labeling patterns that emerge following a step change from unlabeled to ^{13}C-labeled substrates can be analyzed by INST-MFA to determine fluxes with precision.

2. INST-MFA is ideally suited to systems that label slowly due to the presence of large intermediate pools or pathway bottlenecks. This approach not only avoids the additional time and cost of feeding isotope tracers over extended periods (e.g., see Zhao et al. (11)) but may become absolutely necessary in cases where the system cannot be held in a fixed metabolic state long enough to allow isotopic labeling to fully equilibrate. As a result, INST-MFA is expected to become an indispensible tool for extending MFA approaches to studies of mammalian systems (12–14), industrial bioprocesses (5, 15), and other scenarios where attaining a strict isotopic steady state may be impractical.
3. INST-MFA provides increased measurement sensitivity to system parameters. A prime example is the observation that nonstationary labeling measurements are sensitive to metabolite pool sizes, whereas steady-state measurements are not (16, 17). This enables INST-MFA to estimate not only fluxes but intracellular metabolite concentrations as well, which represents a potential framework for integrating metabolomic analysis with MFA. Several studies have also noted that nonstationary measurements often exhibit increased sensitivity to fluxes, especially to certain exchange fluxes (16, 18). Therefore, collecting transient isotopic measurements over a range of time points can improve the precision of flux estimates through application of INST-MFA.

Despite its advantages, the increased complexity of INST-MFA introduces additional difficulties at both the computational and experimental levels. However, these challenges have been largely addressed through recent technical advances.

1. The solution of large-scale ordinary differential equation (ODE) models poses a substantial challenge to efficiently simulating transient isotope labeling experiments. The application of EMU decomposition to INST-MFA has greatly reduced this computational burden and has enabled determination of fluxes and accurate confidence intervals in biologically relevant networks (19, 20).
2. Introducing isotopically nonstationary measurements adds further complexity to experimental design. In addition to the design parameters that must be considered in the steady-state case, INST-MFA requires selection of sampling time points and metabolite concentration measurements. These new dimensions make the search for an optimal experiment design even more difficult and time-consuming. Several computational tools have been developed to efficiently traverse this design space, including parameterized sampling and a posteriori ranking of measurement time points (16, 21).

3. The labeling of intracellular metabolites in organisms with rapid metabolisms exhibits very short isotopic transients on the seconds time scale. Therefore, rapid sampling and quenching must be applied to obtain meaningful data. The field of metabolomics has witnessed considerable progress in this area, and some of these measurement techniques have already been successfully adapted for INST-MFA studies in *Escherichia coli* (1, 18).

Overall, INST-MFA holds great potential for future applications. INST-MFA experiments are already performed in a fraction of the time required for stationary MFA. If downstream sample processing and data analysis can be streamlined and automated, INST-MFA could soon become the basis for high-throughput MFA approaches (18, 22). It is also likely that INST-MFA will become the preferred approach for studies of plants, algae, and animal cell cultures, where labeling is slow and lack of long-term phenotypic stability can restrict the maximum duration of isotope tracer experiments. In this contribution, we present the necessary experimental techniques and computational procedures for performing INST-MFA, with the aim of making this approach accessible to a broader range of investigators within the metabolic engineering and metabolic physiology communities. We focus on those aspects of the analysis that are unique to INST-MFA and refer the reader to other literature on isotopomer measurement techniques and related methods that are common to both INST-MFA and steady-state MFA approaches. We also restrict ourselves to isotope labeling experiments performed on cultured cells, since this is the most common experimental system that has been used in INST-MFA studies to date.

2. Materials

2.1. ^{13}C Labeling Experiment

1. Cell culture at metabolic steady state (see Note 1).
2. Isotopically labeled substrates (see Note 2).
3. Syringes, valves, tubing, and associated equipment for introducing tracers and rapidly removing samples at precise time intervals.

2.2. Quenching and Metabolite Extraction

This protocol is appropriate for quenching and extraction of microbial cells based on a modified Folch extraction method (23). Refer to other references for quenching and extraction of plant cells (24) or mammalian cells (25).

1. Chloroform.
2. Methanol.

3. DI water.
4. Vortexer.
5. Benchtop centrifuge (capable of at least 5,000 rpm).
6. Centrifuge tubes (50 mL and 15 mL).

2.3. Extracellular Uptake and Excretion Flux Measurements

1. Cell culture at metabolic steady state (see Note 3).
2. Syringes, valves, tubing, and associated equipment for removing samples.
3. Analytical instruments (and associated reagents) for measuring extracellular metabolite concentrations, such as GC–MS, LC–MS, HPLC, biochemical analyzer, and microplate reader.

2.4. Mass Spectrometry Analysis

1. GC–MS and/or LC–MS.
2. Derivatization agents, vials, heating blocks, and nitrogen evaporator for GC–MS sample preparation (see Note 4).
3. Vials, columns, gases, buffers, solvents, and other consumables for GC–MS or LC–MS.

2.5. MS Data Processing

1. Computer equipped with either (1) freeware MS analysis software (see Note 5) or (2) commercial software for searching and integrating mass spectra.
2. Mass spectral library for compound verification, such as the NIST/EPA/NIH Mass Spectral Database (26), Golm Metabolome Database (27), FiehnLib (28), METLIN (29), HMDB (30), or MassBank (31).

2.6. Isotopically Nonstationary Metabolic Flux Analysis

1. Computer equipped with research code or publically available software capable of performing isotopically nonstationary metabolic flux analysis (INST-MFA), such as Isotopomer Network Compartmental Analysis (INCA; http://www.vanderbilt.edu/younglab), which runs through the computing environment of MATLAB.

3. Methods

3.1. ^{13}C Labeling Experiment

The ^{13}C labeling experiment should be initialized once a desired cell density has been attained, and the system is at metabolic steady state. A typical biomass sample size required for MS quantification is in the range of 1–10 mg of dry cell weight (9). Therefore, the volume and target cell density of the culture should be chosen so that repeated samples can be efficiently collected and processed without depleting the culture or significantly impacting its phenotypic state. It is suggested that cells be grown to the mid- to late

exponential growth phase before introducing the tracer to batch cultures, as this will provide for maximal cell densities and experimental repeatability. Alternatively, the experiment could be performed in a chemostat operating at an established steady state. Once the tracer has been introduced to the system, repeated samples should be withdrawn and rapidly quenched so that the labeling of intracellular metabolites can be accurately assessed at multiple time points during the transient labeling period.

1. *Introduce tracer to the system*. Labeled substrates can be dissolved in media and rapidly fed via syringe injection to a batch culture or by switching feed reservoirs to a chemostat culture. Autotrophic cultures that rely on gassed CO_2 can be connected to a $^{13}CO_2$-enriched gas feed. The introduction of tracer should not alter the chemical composition of the culture environment in a way that disturbs its metabolic steady state.
2. *Remove samples at multiple time points (~5–15) prior to reaching isotopic steady state*. Samples can be manually withdrawn using a syringe and needle at 20 s intervals, which is adequate for INST-MFA experiments with animal, plant, and slowly growing microbial cells. However, automated sampling techniques have been developed for applications to *E. coli* and other microbes that exhibit extremely fast isotopic transients in central metabolism (5). Samples should be collected more frequently near the beginning of the tracer experiment, as the isotopic labeling will be changing most rapidly during this initial time period. For example, Wiechert et al. (16) have recommended using an approach where the length of each time interval increases exponentially (e.g., 1, 2, 4, 8, and 16) following an initial period where uniformly spaced samples are collected at the maximum rate.

3.2. Quenching and Metabolite Extraction

1. After withdrawing each sample from the ^{13}C-labeled culture, initiate the quench by immediately spraying 1 volume of cell culture (containing 1–10 mg cell dry weight) into a 50 mL centrifuge tube containing 2 volumes of a 60/40 methanol/water solution at −40°C (see Note 6).
2. Separate cells from the quenching medium by centrifuging at 5,000 rpm for 5 min in a benchtop centrifuge precooled to the lowest temperature setting (e.g., −10°C). Aspirate the quench solution from the cell pellet.
3. Resuspend cells in 4 mL chloroform (−20°C).
4. Add 2 mL methanol (−20°C).
5. Vortex tubes for 30 min in cold room.
6. Add 1.5 mL ice-cold water.
7. Vortex tubes for additional 5 min.

8. Transfer to 15 mL centrifuge tubes.
9. Centrifuge at 5,000 rpm for 20 min at lowest temperature setting.
10. Collect aqueous (upper) phase in a new 15 mL tube or two microcentrifuge tubes.
11. Collect organic (lower) phase in a new 15 mL tube or two microcentrifuge tubes.
12. Add internal standards if quantification of metabolite concentrations is desired.
13. Evaporate all extracts to dryness using nitrogen at room temperature.
14. Store samples at −80°C.

3.3. Extracellular Uptake and Excretion Measurements

Extracellular uptake and excretion measurements are necessary to define the absolute scale of the intracellular fluxes and to constrain external fluxes that cross the system boundary. Regression analysis has been previously applied to determine metabolic fluxes from extracellular time courses of substrate depletion or product accumulation (32–35). However, if these measurements are unavailable, all fluxes can be normalized to a fixed "reference" flux (e.g., the net CO_2 uptake rate in autotrophic systems or the glucose uptake rate in heterotrophic systems). We have recently developed a program called ETA for performing the extracellular time-course analysis (see Note 7). It should be noted that these measurements typically require a separate unlabeled culture and a longer experimental time course than the ^{13}C labeling experiment.

3.4. Mass Spectrometry Analysis

The pathways of interest will dictate the types of metabolites and MS analysis to be performed. Generally, amino acids, organic acids, fatty acids, and sugars can be analyzed using GC–MS following chemical derivatization. Sugar phosphates and acyl-CoA molecules, on the other hand, are typically analyzed via LC–MS or LC–MS/MS, to avoid thermal degradation of these nonvolatile analytes. GC–MS analysis of a wide range of metabolites is most readily achieved by (1) methoximation to prevent keto-enol tautomerization followed by (2) conversion to trimethylsilyl (TMS) or *tert*-butyldimethylsilyl (TBMDS) derivatives (36). GC–MS analyses of derivatized amino acids, organic acids, and sugars are generally analyzed on nonpolar columns, while fatty acids are analyzed on polar columns. GC–MS analysis is typically performed in electron ionization (EI) mode to generate multiple fragment ions of the target analytes. LC–MS/MS analysis of sugar phosphates and acyl-CoA molecules can be accomplished using an ion-pairing gradient LC–MS/MS method with a nonpolar column and a solution of tributylamine + acetic acid as eluent A and methanol as eluent B (37). The acquisition of labeling and concentration data can be performed using negative electrospray ionization in multiple

reaction monitoring (MRM) mode. Initial suggestions for chromatographic parameters for GC–MS or LC–MS/MS are described by Roessner et al. (38) and Luo et al. (39), respectively. These parameters will typically need to be further optimized depending on the target analytes of interest and the complexity of the sample matrix.

3.5. MS Data Processing

Analysis of MS data requires (1) identification of chromatographic peaks and fragment ions associated with target analytes of interest, (2) integration of ion chromatograms over time to quantify relative abundance of specific isotope peaks, and (3) assessment of measurement standard errors. In many cases, it is also desirable to "correct" the raw mass isotopomer distributions (MIDs) to account for the presence of naturally occurring stable isotopes. Corrected mass isotopomer data provides a more intuitive picture of the labeling that is attributable to the introduction of a tracer compound and is generally the preferred method for presenting data from an isotope labeling experiment. However, some INST-MFA software, such as INCA, is capable of performing these corrections internally, and therefore, it is only necessary to input the raw, uncorrected MIDs.

1. *Identify the chromatographic peaks associated with the analytes of interest.* This is based on both the retention time (RT) and the MS fingerprint of the peak. Automated searching of mass spectral databases can facilitate the identification of compounds within complex mixtures. Furthermore, when pure standards of the target analytes are commercially available, these can be run separately or spiked into extract samples to confirm the identity of uncertain peaks.
2. *Identify ions to be used for mass isotopomer analysis and determine their molecular composition*. The best GC–EI-MS fragment ions are highly abundant ions with masses greater than 150 Da, since these are less likely to be contaminated by interfering fragment ions of similar mass. Determining the elemental composition of these ionic species is facilitated by references that list common fragmentation patterns and molecular rearrangements obtained for particular classes of compounds and derivatization groups (24, 40). The precursor ions formed in negative-ESI mode from LC–MS/MS analysis typically result from simple proton extraction. Since current application of LC–MS/MS to mass isotopomer analysis only makes use of product ions that are formed without breaking the carbon backbone of their precursor ions, the product ion spectra reflect the MIDs of the intact precursor ions when ^{13}C is used as tracer (41).

3. *Integrate the mass isotopomer peaks using either custom or commercial software*. In order to maximize the accuracy of mass isotopomer data, it is necessary to integrate each ion chromatogram over its full peak width and the exact same time window of integration. It is important for these parameters to be determined consistently for all mass isotopomers of a given fragment ion so that errors in the MID will not occur (42). This also involves integrating all single ion traces over all scans of the peak, including masses up to 3 Da heavier than the fully labeled fragment ion. For example, to quantify the MID of a fragment with monoisotopic mass 200 m/z and up to three-labeled carbons, extract and independently integrate the ion traces of 200, 201, 202,. . ., 206. Normalize the integrated areas such that the sum of all mass isotopomers for a given fragment ion is 1 (i.e., 100 mol %).
4. *Correct mass isotopomer distributions (MIDs) for natural isotope abundance (optional)*. The method of Fernandez et al. (43) can be applied to perform the correction.
5. *Calculate the mean and standard error of MIDs for each metabolite at each time point*. In order to perform statistical analysis of best-fit flux solutions, MFA software requires the user to input standard errors of each mass isotopomer measurement. Typically, the precision (i.e., repeatability) of these measurements is superior to their absolute accuracy (42). Inaccurate MIDs can occur due to interference from overlapping fragment ions or gas-phase proton exchanges that contaminate the mass spectrum of the target ions (44). Therefore, it is important to assess both precision and accuracy using standards of known isotope labeling. At minimum, it is necessary to run samples from naturally labeled cell extracts and compare the experimentally determined MIDs to theoretically predicted values. The approach of Fernandez et al. (43) can be used to predict MIDs of unlabeled samples based on reported values of elemental isotope abundance (45). A more thorough error assessment would also involve analyzing mixtures of labeled standards to quantify the uncertainty in measuring MIDs that differ from natural labeling (e.g., see Antoniewicz et al. (42)). In general, fragment ions used for MFA should be accurate to within 1.5 mol % (and preferably 0.8 mol %) of the predicted value (36).

3.6. Isotopically Nonstationary Metabolic Flux Analysis

A flow chart of a typical INST-MFA process is shown in Fig. 3. INST-MFA is concerned with solving an "inverse problem" where fluxes and pool sizes are estimated from measured labeling patterns and extracellular rates through the means of an iterative least-squares fitting procedure. At each iteration, a "forward problem" is solved where an isotopomer model is used to simulate labeling

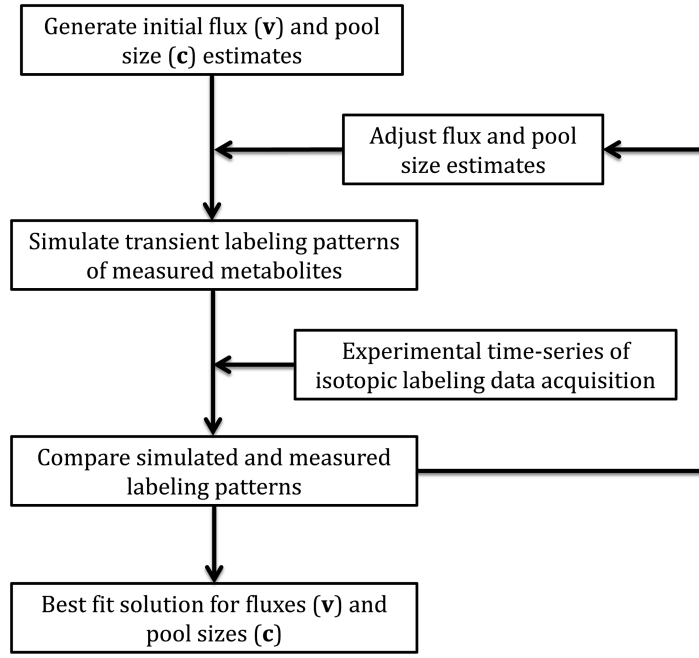

Fig. 3. Flowchart showing the overall schematic of ^{13}C INST-MFA. Following the labeling experiment and MS analysis of the measured metabolites, computational analysis of the dynamic changes in isotope labeling patterns can be used to estimate metabolic pathway fluxes and pool sizes. This involves solving an inverse problem whereby the vectors of flux (v) and pool size (c) parameters are iteratively adjusted until the mismatch between simulated and experimentally measured data sets is minimized.

measurements for a given metabolic network and a given set of parameter estimates. The discrepancy between the simulated and measured labeling patterns is then assessed, and the parameter estimates are updated to achieve an improving fit. Once convergence to the best-fit solution is obtained, the procedure terminates, and the optimal flux and pool size estimates are returned.

3.6.1. Build an Isotopomer Model for INST-MFA

In order to perform INST-MFA, it is necessary to reconstruct a metabolic network from biochemical literature and the annotated genome of the organism of interest. This network must prescribe both (1) the stoichiometry of all enzymatic reactions under consideration and (2) atom transitions for each reaction (see Note 8). Reactions must also be classified as either reversible or irreversible.

1. *Construct a stoichiometric model including all substrates, products, and intermediate metabolites.* When constructing a model, it is important to strive for parsimony in describing the available experimental measurements. The model must be sophisticated enough to reconcile all available experimental measurements while simultaneously avoiding unnecessary complexity and redundancy that leads to overfitting of

parameters. Fortunately, there are statistical tests to assess goodness-of-fit and to detect loss of precision due to overfitting (presented in Subheading 3.6.5). Overly sophisticated models can be reduced by (1) combining linear pathways into a single reaction, (2) combining isoenzymes or parallel pathways that catalyze identical conversions, and (3) omitting irrelevant pathways based on biological knowledge, such as repression of pathways under certain conditions (36). Additionally, if the cells are growing at a significant rate, all fluxes toward biomass production can be lumped into a single biosynthetic reaction that summarizes the withdrawal of all necessary growth precursors. Cofactors that contribute to energy balancing (e.g., ATP) or redox balancing (e.g., NADH or NADPH) are usually omitted from the model to ensure that these difficult-to-quantify balances do not unduly bias the resulting flux estimates (36).

Construction of a stoichiometric model can be further complicated by (1) compartmentalization of metabolites, (2) reaction reversibility, and (3) tracer dilution from unlabeled sources. First, as a result of subcellular compartmentalization in eukaryotes, the same biochemical reactions can occur simultaneously in different organelles, giving rise to multiple distinct metabolic pools that must be treated as separate nodes in the isotopomer model. Transport of metabolites between different compartments also needs to be defined in the model (e.g., exchange of pyruvate between the cytosol and mitochondria). Because each metabolite measurement obtained by MS analysis represents an aggregation of these different metabolic pools, pseudoreactions can be introduced into the model to represent the contribution from each compartment (see Note 9). However, this also introduces additional parameters into the model that must be determined from the isotopomer measurements. Second, reaction reversibility is another crucial consideration, since exchange fluxes (defined as the minimum of the forward and reverse reaction rates) affect metabolite labeling patterns in addition to net fluxes (defined as the difference between forward and reverse reaction rates). While all enzymes are reversible to some extent, many can be classified as practically unidirectional as a result of thermodynamic and kinetic considerations (e.g., pyruvate kinase in glycolysis). Third, enrichment of the tracer can also be diluted by unlabeled sources, such as CO_2 present in air, unlabeled carbon sources in complex culture media, or even breakdown of macromolecular biomass components. The inclusion of these unlabeled sources in the model can be critical to obtaining a statistically acceptable description of actual experimental data sets.

2. *Classify each metabolite as balanced or unbalanced.* Balanced metabolites are intermediate nodes for which the total incoming flux is constrained to balance the total outgoing flux. Unbalanced metabolites can refer to any carbon sources or sinks within the stoichiometric network, such as glucose or biomass, respectively. Additionally, unbalanced metabolites can also arise at intermediate nodes that exchange rapidly with the extracellular environment, as is often the case for CO_2. It is important to distinguish between balanced and unbalanced metabolites, since unbalanced metabolites do not impose stoichiometric constraints on the network and their labeling is typically considered to be fixed and externally specified.
3. *Specify the atom transitions for each reaction in the stoichiometric model.* It is necessary to include the atom transitions for each reaction present in the metabolic network so that the fate of each atom can be traced from substrate to product. Generally, the atom mapping for a particular enzyme is conserved between species and can be extracted from existing MFA models or articles found in biochemical literature. Furthermore, the model must account for the scrambling that occurs due to symmetric metabolites or chemically equivalent groups of atoms (20).
4. *Specify the tracer substrates and their positional ^{13}C labeling.* This information is necessary to define the labeled inputs to the isotopomer model.

3.6.2. Solve the Forward Problem to Simulate Labeling Measurements

In INST-MFA, the isotopomer balances are described by a system of ordinary differential equations, which is significantly more expensive to solve than the algebraic systems that describe steady-state labeling. Due to this additional difficulty, algorithms for solving the forward problem of INST-MFA need to be carefully designed so that computational expense does not become prohibitive. The most efficient approach involves first decomposing the isotopomer network into Elementary Metabolite Units (EMUs) (19, 20). By only solving for the isotopomer distributions of EMUs that contribute to the available measurements, this approach minimizes the number of ODEs that need to be integrated and thereby enables the forward problem to be solved thousands of times faster than previous methods. This, in turn, increases the efficiency of the inverse problem of INST-MFA because each iteration of the parameter estimation procedure can be completed in minimal time.

An EMU is defined as a distinct subset of a metabolite's atoms and can exist in a variety of mass states depending on its isotopic composition. In its lowest mass state, an EMU is referred to as M0, while an EMU that contains one additional atomic mass unit

(e.g., as a result of a ^{13}C atom in place of ^{12}C atom) is referred to as M1, with higher mass states described accordingly. An MID is a vector that contains the fractional abundance of each mass state of an EMU. To solve the forward problem of simulating metabolite labeling in INST-MFA, the isotopomer network is first systematically searched to enumerate all EMUs that contribute to measurable MS fragment ions. Then, these EMUs are grouped into mutually dependent blocks using a Dulmage–Mendelsohn decomposition (46, 47) (see Note 10). Therefore, by definition, all EMUs within a particular block have the same number of atoms and must be solved simultaneously and not sequentially.

The decoupled blocks can be arranged into a cascaded system of ODEs with the following form:

$$\mathbf{C}_n \cdot \frac{\mathrm{d}\mathbf{X}_n}{\mathrm{d}t} = \mathbf{A}_n \cdot \mathbf{X}_n + \mathbf{B}_n \cdot \mathbf{Y}_n \tag{1}$$

Level n of the cascade represents the network of EMUs within the nth block. The rows of the state matrix $\mathbf{X}_n$ correspond to MIDs of EMUs within the nth block. The input matrix $\mathbf{Y}_n$ is analogous but with rows that are MIDs of EMUs that are previously calculated inputs to the nth block (or MIDs of source EMUs that are unbalanced). The concentration matrix $\mathbf{C}_n$ is a diagonal matrix whose elements are pool sizes corresponding to EMUs represented in $\mathbf{X}_n$. The system matrices $\mathbf{A}_n$ and $\mathbf{B}_n$ describe the network as follows:

$$\mathbf{A}_n(i,j) = \begin{cases} -\text{sum of fluxes consuming, } i\text{th EMU in } \mathbf{X}_n & i = j \\ \text{flux to } i\text{th EMU in } \mathbf{X}_n \text{ from } j\text{th EMU in } \mathbf{X}_n & i \neq j \end{cases} \tag{2}$$

$$\mathbf{B}_n(i,j) = \{\text{flux to } i\text{th EMU in } \mathbf{X}_n \text{ from } j\text{th EMU in } \mathbf{Y}_n \tag{3}$$

1. *Simulate the time course of isotope labeling.* Given initial estimates of all fluxes and pool sizes, Eq. 1 can be constructed and then integrated. This can be accomplished using standard ODE numerical solvers or specialized algorithms that take advantage of the linear structure of this dynamical system, as described by Young et al. (19).
2. *Analyze the simulation results.* Solving the forward problem enables calculation of isotopomer distributions for each metabolite of interest, based on the initial flux and pool size estimates. The simulated MIDs can be plotted versus time and compared to the measured data. Fig. 4 shows an example of the labeling dynamics of several metabolites in an autotrophic system using ^{13}C-labeled bicarbonate as the tracer. The relative abundances of unlabeled mass isotopomers (M0) dropped at the start of the labeling period and were replaced by M1, M2, and higher mass isotopomers following the introduction of tracer. Additionally,

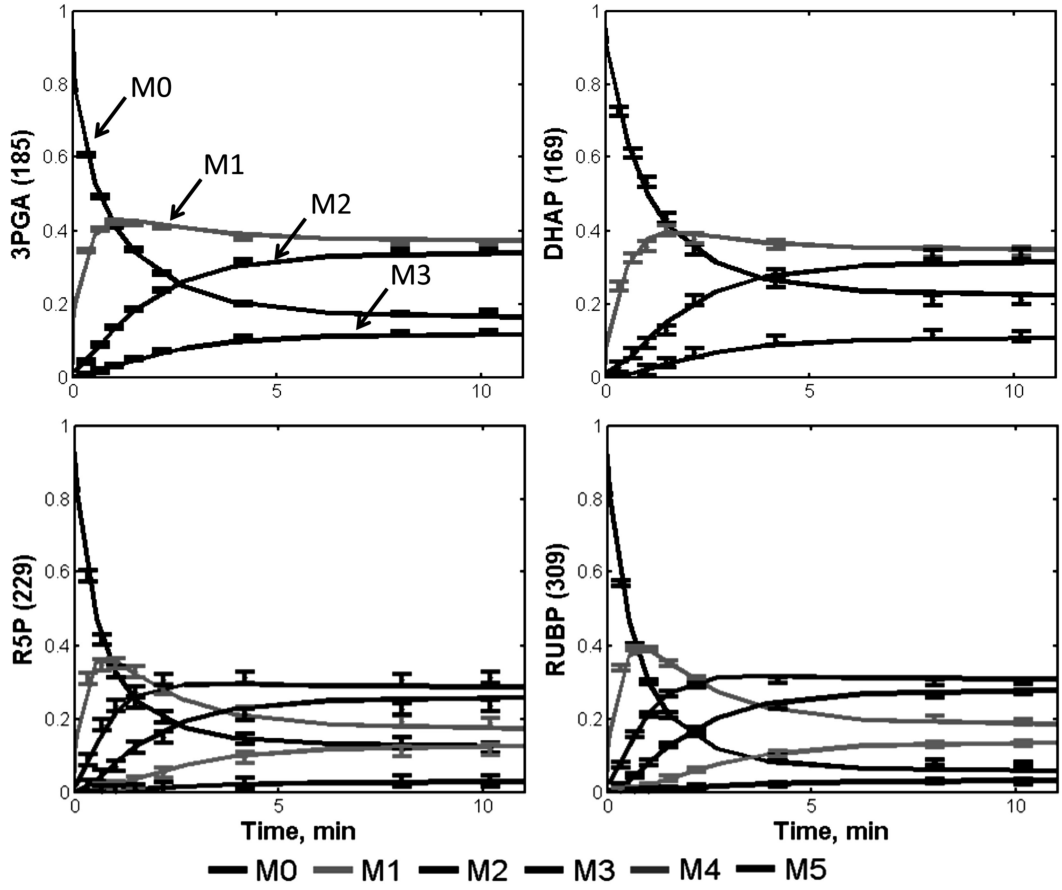

Fig. 4. Experimentally measured labeling trajectories of central metabolic intermediates (data points) and INST-MFA model fits (*solid lines*) from an autotrophic INST-MFA study. The *error bars* represent standard measurement errors. Ions shown are for 3-phosphoglycerate (3PGA), dihydroxyacetone phosphate (DHAP), ribose-5-phosphate (R5P), and ribulose-1,5-bisphosphate (RUBP). Nominal masses of M0 mass isotopomers are shown in parentheses (Adapted from Young et al. (10)).

it is also informative to plot the average enrichments of various MS fragment ions as shown in Fig. 5. The average ^{13}C enrichment is calculated using the following expression:

$$\frac{1}{N}\sum_{i=1}^{N} Mi \times i \tag{4}$$

where N is the number of carbon atoms in the metabolite and Mi is the fractional abundance of the ith mass isotopomer.

3.6.3. Sensitivity Calculation Estimation of both the unknown fluxes and pool sizes using INST-MFA is accomplished by finding a best-fit solution to the inverse problem. Efficient solution of this problem typically relies on optimization algorithms that choose the search direction based on the gradient of the least-squares objective function (see Eq. 6) with respect to all adjustable parameters. The most accurate and least

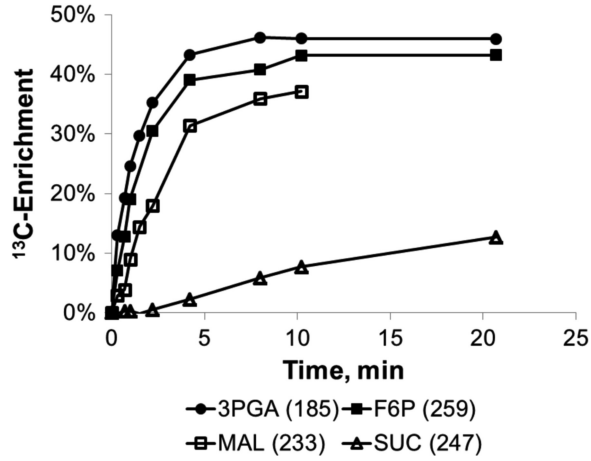

Fig. 5. Average ^{13}C enrichments of selected ion fragments from an autotrophic INST-MFA study. The labeling trajectory is shown for 3-phosphoglycerate (3PGA), fructose-6-phosphate (F6P), malate (MAL), and succinate (SUC) over the course of 10 min (Adapted from Young et al. (10)).

expensive way to obtain the required gradient information is to integrate a system of sensitivity equations whose solution describes how the calculated MIDs vary in response to changes in the model parameters. Implicit differentiation of Eq. 1 yields the following sensitivity equation:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathbf{X}_n}{\partial \mathbf{p}} &= \mathbf{C}_n^{-1}\cdot\mathbf{A}_n\cdot\frac{\partial \mathbf{X}_n}{\partial \mathbf{p}} + \frac{\partial\left(\mathbf{C}_n^{-1}\cdot\mathbf{A}_n\right)}{\partial \mathbf{p}}\cdot\mathbf{X}_n \\ &\quad + \mathbf{C}_n^{-1}\cdot\mathbf{B}_n\cdot\frac{\partial \mathbf{Y}_n}{\partial \mathbf{p}} + \frac{\partial\left(\mathbf{C}_n^{-1}\cdot\mathbf{B}_n\right)}{\partial \mathbf{p}}\cdot\mathbf{Y}_n\end{aligned} \tag{5}$$

where $\mathbf{p}$ is the vector of adjustable flux and pool size parameters. This system of equations can be solved in tandem with those of Eq. 1, and the time-dependent sensitivities can be used to evaluate the objective function gradient during each iteration of the INST-MFA inverse problem. Furthermore, if approximate values of the parameters are available prior to performing the labeling experiment, calculation of measurement sensitivities can provide useful information pertaining to parameter identifiability and experimental design.

3.6.4. Experimental Design

While solving the forward problem is an important step in the determination of fluxes using INST-MFA, it can also inform the experimental design. The precision with which a particular flux or pool size can be estimated, if at all, is solely determined by the sensitivity of the available measurements to the flux in question, which is a function of (1) the isotopic tracer applied, (2) the structure of the metabolic network, (3) the intracellular flux distribution, (4) the timing of the measurements, and (5) the

metabolites that are measured. Since (2) and (3) are not under the control of the experimenters, the key elements of experimental design entail choosing appropriate combinations of (1), (4), and (5) to identify the fluxes of interest. For the most part, the prevailing philosophy has been to measure as many metabolites as possible that are relevant to the pathways of interest. Therefore, the focus of experimental design has been on choosing the labeled substrate(s) and sampling strategy that will maximize the precision of flux estimates based on the available isotopic measurements. There is a wide literature on optimal design of ^{13}C labeling experiments, and the extension of these concepts to INST-MFA experiments has been presented by Wiechert and colleagues (16, 21).

3.6.5. Solving the Inverse Problem to Determine Flux and Pool Size Parameters

Fluxes and pool sizes are estimated by minimizing the difference between measured and simulated data according to the following equation (16, 20):

$$\min_{\mathbf{u},\mathbf{c}} \phi = [\mathbf{m}(\mathbf{u},\mathbf{c},\mathrm{t}) - \hat{\mathbf{m}}(t)]^T \cdot \sum_{m}^{-1} \cdot [\mathbf{m}(\mathbf{u},\mathbf{c},\mathrm{t}) - \hat{\mathbf{m}}(t)]$$
$$\text{s.t. } \mathbf{N}\cdot\mathbf{u} \geq 0,\ \ \mathbf{c} \geq 0$$

where ϕ is the objective function to be minimized, $\mathbf{u}$ is a vector of free fluxes, $\mathbf{c}$ is a vector of metabolite concentrations, t is time, $\mathbf{m}(\mathbf{u},\mathbf{c},t)$ is a vector of simulated measurements, $\hat{\mathbf{m}}(t)$ is a vector of observed measurements, $\mathbf{\Sigma_m}$ is the measurement covariance matrix, and $\mathbf{N}$ is the nullspace of the stoichiometric matrix. A reduced gradient method can be implemented to handle the linear constraints of this problem within a Levenberg–Marquardt nonlinear least-squares solver (48, 49). Alternatively, gradient-free optimization approaches have been applied by Noh et al. (16).

1. *Perform the flux estimation analysis by minimizing the difference between the measured and simulated measurements.* The flux estimation is performed by calculating the solution to Eq. 6. To ensure a global solution is obtained, it is advisable to repeat the parameter estimation from multiple initial guesses when using a gradient-based local optimization search. Alternatively, stochastic global optimization algorithms based on genetic programming or simulated annealing can be applied to ensure broad coverage of the parameter space.
2. *Assess the overall fit of the flux estimation.* Testing the goodness-of-fit will determine whether the optimal solution is statistically acceptable based on the minimized sum of squared errors (SSE). At convergence, the minimized variance-weighted SSE is a stochastic variable drawn from a chi-square distribution with n-p degrees of freedom (DOF), where n is the number of independent measurements and p is the number of estimated parameters. The SSE that is calculated should therefore be in the interval

$[\chi^2_{\frac{\alpha}{2}} \quad \chi^2_{1-\frac{\alpha}{2}}]$, where α is a chosen threshold value corresponding to the desired confidence level (e.g., 0.05 for 95 % confidence or 0.01 for 99 % confidence). The model fit is accepted when the SSE falls within the limits of the expected chi-square range (50). Additionally, the distribution of residuals should be assessed for normality. The standard deviation-weighted residuals should be normally distributed with a mean of zero and standard deviation of one. One approach that can be used to evaluate the hypothesis that the residuals are normally distributed is the Lilliefors test (51). Various plots can also be constructed to assess normality of the residuals.

3. *Assess the goodness-of-fit of each measurement.* In addition to checking the overall distribution of the residuals, it is often informative to plot the simulated and measured MIDs of each MS fragment ion. Furthermore, one should check the residuals between any measured extracellular fluxes and the estimates derived from INST-MFA. This provides a visual assessment of which measurements are mostly responsible for the lack of fit. If a poor fit is obtained, further investigation needs to be performed to identify the source of disagreement between experimental measurements and the isotopomer model. There are three possible causes for a poor fit that should be evaluated: (1) there are gross errors associated with the measurements, (2) there is an inappropriate weighting of the residuals, or (3) there is a mistake or omission in the metabolic reaction network. One should proceed by process of elimination to determine which of these is the root cause of a poor fit and then take corrective steps.
4. *Identify measurements that contribute significantly to the precision of estimated fluxes.* The fractional contribution of each measurement to the local variance of each flux can be calculated as described in Antoniewicz et al. (50). The higher the contribution value, the more important the measurement is for determining a particular flux. Fluxes that depend on only one measurement are very sensitive to errors in that one measurement. It is therefore desirable that more than one measurement significantly contributes to the estimation of each flux.

3.6.6. Calculate Parameter Uncertainties

Once an optimal solution has been obtained, nonlinear confidence intervals on the fitted parameters should be computed using robust, global methods instead of relying solely upon local standard errors. The local standard errors can be easily obtained from the parameter covariance matrix at the optimal solution; however, they do not accurately reflect changing sensitivities at points removed from the optimal solution. Furthermore, the calculation of the covariance matrix becomes ill conditioned when the Hessian of ϕ with respect to the fitted parameters is close to singular.

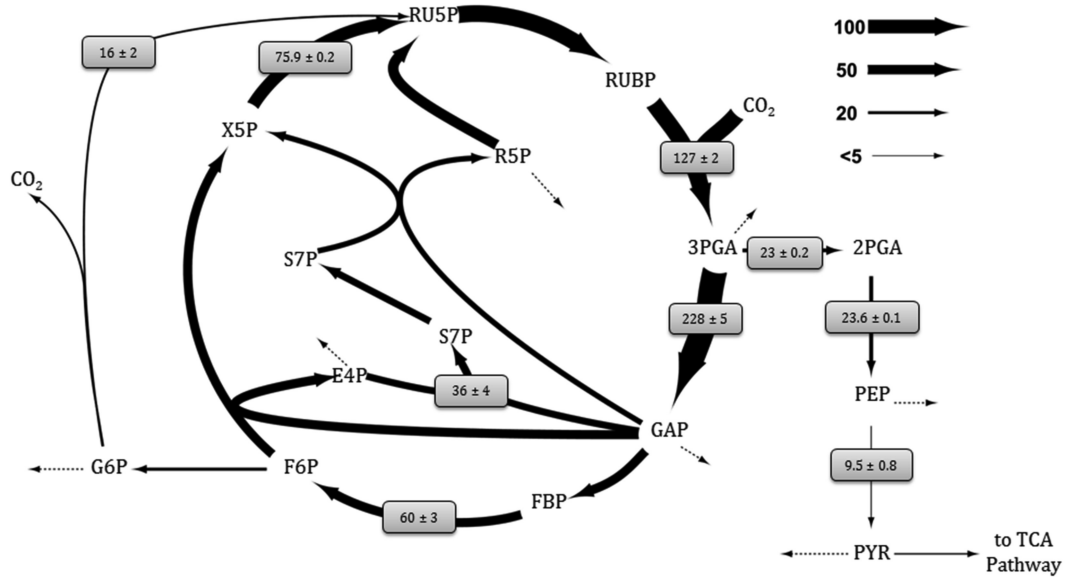

Fig. 6. Example of a flux map constructed for an INST-MFA study determined under photoautotrophic growth conditions. This flux map shows the estimated fluxes associated with glycolysis and the Calvin cycle for a *Synechocystis* INST-MFA study. Net fluxes are shown normalized to a net CO_2 uptake rate of 100. Values are represented as $M \pm SE$, where M is the median of the 95 % flux confidence interval and SE is the estimated standard error of M. Arrow thickness is scaled proportional to net flux. *Dotted arrows* indicate fluxes to biomass formation (Adapted from Young et al. (10)).

1. *Calculate the 95 % confidence intervals using either continuation methods or Monte Carlo analysis.* Parameter continuation can be performed to calculate accurate upper and lower bounds on the 95 % confidence interval for each flux or pool size parameter (50). This determines the sensitivity of the minimized SSE to varying a single parameter away from its optimal value while allowing the remaining parameters to adjust in order to minimize $\Delta\phi$. Large confidence intervals indicate that the flux cannot be estimated precisely. On the other hand, small confidence intervals indicate that the flux is well determined. Monte Carlo simulation can also be used to calculate the 95 % confidence intervals. This method is typically more expensive than the parameter continuation approach but is expected to yield similar results.

3.6.7. Report the Flux Values and Flux Uncertainties

Once an acceptable fit to the experimental measurements has been achieved and confidence intervals have been computed for all parameters, the results are best summarized visually in the form of a flux map. Fig. 6 shows an example of a flux map for the Calvin cycle and glycolytic pathway of *Synechocystis* sp. PCC6803 determined under photoautotrophic growth conditions using INST-MFA (10). Several software tools have been recently developed, which aid in the construction of these maps (see Note 11).

4. Notes

1. These cell cultures can be grown in flasks or bioreactors with working volumes ranging from 50 mL to 1 L, depending on the sampling volume and number of samples to be taken, as discussed in Subheading 3.6.4, "Experimental Design." Variables that should be controlled or monitored include temperature, pH, dissolved oxygen, and nutrient concentrations. Additionally, light intensity should be controlled for photoautotrophic cell cultures.
2. Until recently, most ^{13}C labeling experiments have been performed using labeled glucose tracers (e.g., [U-$^{13}C_6$]glucose, [1-^{13}C]glucose, [1,2-$^{13}C_2$]glucose, or mixtures thereof). However, INST-MFA studies of photoautotrophic systems, such as plants and cyanobacteria, by definition must rely on labeling solely from $^{13}CO_2$ or labeled bicarbonate. The choice of tracer(s) should be made to provide the maximum amount of information from the labeling dynamics while minimizing changes to the chemical composition of the medium; this is discussed further in Subheading 3.6.4, "Experimental Design."
3. The cell culture used for measuring extracellular uptake and excretion flux measurements should be different from the one used in the ^{13}C labeling experiment. This is due to the fact that this cell culture does not require isotope labeling, and the time course for this experiment will usually be longer than the ^{13}C labeling experiment.
4. Derivitization agents such as methoxyamine (MOX), trimethylsilane (TMS), or *tert*-butyl dimethylsilane (TBDMS) are typical for GC–MS analysis. The MOX reaction protects ketone and aldehyde functional groups and thereby prevents the formation of multiple TMS or TBDMS derivatives. This step is unnecessary if no ketone or aldehyde functional groups are present in the analytes of interest. TMS and TBDMS derivatives produce several characteristic fragment ions that facilitate identification (40). Huege et al. (24) provide a list of several GC–EI-MS ion fragments of TMS derivatives that have been used for isotopomer analysis. Ahn and Antoniewicz (52) provide a similar list for TBDMS derivatized metabolites.
5. Available GC–MS freeware include AMDIS (http://chemdata.nist.gov/mass-spc/amdis/) and Wsearch32 (http://www.wsearch.com.au/wsearch32/wsearch32.htm). Two popular freeware programs for LC–MS/MS data analysis are MZmine and XCMS, the latter of which runs in the R statistical programming environment. Both programs require the user to convert raw data files into a nonproprietary format such as

mzXML, NetCDF, or mzData. Conversion to mzXML format can be accomplished using one of several instrument-specific software tools developed and maintained by the Seattle Proteome Center (http://tools.proteomecenter.org/software.php).

6. A −40°C bath can be achieved by creating a slurry of 4.5 M calcium chloride chilled in a −80°C freezer for 3–4 h prior to the start of the quench.
7. Extracellular Timecourse Analysis (ETA) is a software package that has been coded in MATLAB (http://mfa.vueinnovations.com). It can be used to estimate the specific growth rate as well as the cell-specific uptake and excretion rates of extracellular metabolites based upon time-course concentration measurements.
8. Networks used for heterotrophic MFA typically include glycolysis, pentose phosphate pathway, amino acid metabolism, TCA cycle, and various amphibolic pathways that interact with the TCA cycle. This backbone of central metabolic pathways may be further augmented by additional reactions of interest. Some helpful online databases include KEGG (Kyoto Encyclopedia of Genes and Genomes; http://www.genome.jp/kegg/), BioCyc (http://biocyc.org/), metaTIGER http://www.bioinformatics.leeds.ac.uk/metatiger/), ENZYME (http://enzyme.expasy.org/), and BRENDA (http://www.brenda-enzymes.info/).
9. One way to model pseudoreactions of compartmental mixing in INCA is as follows:

$$0^{*}\text{G6P.plastid (abcdef)} \rightarrow \text{G6P.pseudo (abcdef)}$$

$$0^{*}\text{G6P.cytosol (abcdef)} \rightarrow \text{G6P.pseudo (abcdef)}$$

$$\text{G6P.pseudo} \rightarrow \text{Sink (flux fixed to 100)}$$

   These equations signify glucose-6-phosphate (G6P) coming from two different compartments, the plastid and cytosol. The letters in parentheses are the carbon atoms associated with G6P. The "0" in front of the first two reactions indicates that no carbon is actually withdrawn from the network, even though the carbon labeling is preserved in the G6P pseudometabolite (i.e., this essentially creates the G6P pseudometabolite without siphoning carbon away from the "real" metabolic network). The third reaction has a fixed flux set to an arbitrary value of 100 so that the fluxes estimated for the first two reactions represent the relative percentage contributions from the two compartments. More pseudoreactions may be added as more compartments are included in complex networks.
10. Blocks are defined by sets of EMUs whose MIDs are mutually dependent within the context of the EMU reaction network.

The EMUs are arranged into blocks where the EMU reaction network is regarded as a directed graph, where the nodes represent EMUs and edges represent EMU reactions. An $N \times N$ adjacency matrix is constructed for the directed graph, where N is the total number of EMUs. A nonzero entry $a(i, j)$ of the adjacency matrix indicates the dependence of the ith EMU's MID on the jth EMU's MID. A Dulmage–Mendelsohn decomposition is performed on the adjacency matrix, returning an upper block triangular matrix from which the diagonal blocks are extracted. Blocks can be arranged so that each is a self-contained subproblem that depends on the outputs of previously solved blocks, creating a cascaded system.

11. Several tools have been recently developed for flux visualization in the context of metabolic networks, such as FluxMap (53), FluxViz (54), faBINA (55), Omix (56), BioCyc Omics Viewer (57), Reactome Skypainter (58), Pathway Projector (59), MetaFluxNet (60), and OptFlux (61).

Acknowledgements

This work was supported by NSF EF-1219603. LJJ was supported by a GAANN fellowship from the US Department of Education under grant number P200A090323.

References

1. Wiechert W (2001) 13C metabolic flux analysis. Metab Eng 3:195–206
2. Sauer U (2006) Metabolic networks in motion: 13C-based flux analysis. Mol Syst Biol 2:62
3. Antoniewicz MR, Kraynie DF, Laffend LA et al (2007) Metabolic flux analysis in a nonstationary system: fed-batch fermentation of a high yielding strain of *E. coli* producing 1,3-propanediol. Metab Eng 9:277–292
4. Noguchi Y, Young JD, Aleman JO et al (2009) Effect of anaplerotic fluxes and amino acid availability on hepatic lipoapoptosis. J Biol Chem 284:33425–33436
5. Nöh K, Grönke K, Luo B et al (2007) Metabolic flux analysis at ultra short time scale: isotopically non-stationary 13C labeling experiments. J Biotechnol 129:249–267
6. Leighty RW, Antoniewicz MR (2011) Dynamic metabolic flux analysis (DMFA): a framework for determining fluxes at metabolic non-steady state. Metab Eng 13:745–755
7. Lequeux G, Beauprez J, Maertens J et al (2010) Dynamic metabolic flux analysis demonstrated on cultures where the limiting substrate is changed from carbon to nitrogen and vice versa. J Biomed Biotechnol 2010:1–19
8. Wahl SA, Nöh K, Wiechert W (2008) 13C labeling experiments at metabolic nonstationary conditions: an exploratory study. BMC Bioinforma 9:152
9. Shastri AA, Morgan JA (2007) A transient isotopic labeling methodology for 13C metabolic flux analysis of photoautotrophic microorganisms. Phytochemistry 68:2302–2312
10. Young JD, Shastri AA, Stephanopoulos G et al (2011) Mapping photoautotrophic metabolism with isotopically nonstationary (13)C flux analysis. Metab Eng 13:656–665
11. Zhao Z, Kuijvenhoven K, Ras C et al (2008) Isotopic non-stationary 13C gluconate tracer method for accurate determination of the pentose phosphate pathway split-ratio in *Penicillium chrysogenum*. Metab Eng 10:178–186

12. Maier K, Hofmann U, Bauer A et al (2009) Quantification of statin effects on hepatic cholesterol synthesis by transient (13)C-flux analysis. Metab Eng 11:292–309
13. Maier K, Hofmann U, Reuss M et al (2008) Identification of metabolic fluxes in hepatic cells from transient 13C-labeling experiments: part II. Flux estimation. Biotechnol Bioeng 100:355–370
14. Munger J, Bennett BD, Parikh A et al (2008) Systems-level metabolic flux profiling identifies fatty acid synthesis as a target for antiviral therapy. Nat Biotechnol 26:1179–1186
15. Iwatani S, Yamada Y, Usuda Y (2008) Metabolic flux analysis in biotechnology processes. Biotechnol Lett 30:791–799
16. Nöh K, Wahl A, Wiechert W (2006) Computational tools for isotopically instationary 13C labeling experiments under metabolic steady state conditions. Metab Eng 8:554–577
17. Wiechert W, Nöh K (2005) From stationary to instationary metabolic flux analysis. Adv Biochem Eng Biotechnol 92:145–172
18. Schaub J, Mauch K, Reuss M (2008) Metabolic flux analysis in *Escherichia coli* by integrating isotopic dynamic and isotopic stationary 13C labeling data. Biotechnol Bioeng 99:1170–1185
19. Young JD, Walther JL, Antoniewicz MR et al (2008) An elementary metabolite unit (EMU) based method of isotopically nonstationary flux analysis. Biotechnol Bioeng 99:686–699
20. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2007) Elementary metabolite units (EMU): a novel framework for modeling isotopic distributions. Metab Eng 9:68–86
21. Noh K, Wiechert W (2006) Experimental design principles for isotopically instationary C labeling experiments. Biotechnol Bioeng 94:234–251
22. Sauer U (2004) High-throughput phenomics: experimental methods for mapping fluxomes. Curr Opin Biotechnol 15:58–63
23. Folch J, Lees M, Stanley GH (1957) A simple method for the isolation and purification of total lipides from animal tissues. J Biol Chem 226:497–509
24. Huege J, Sulpice R, Gibon Y et al (2007) GC-EI-TOF-MS analysis of in vivo carbon-partitioning into soluble metabolite pools of higher plants by monitoring isotope dilution after $13CO_2$ labelling. Phytochemistry 68:2258–2272
25. Yoo H, Antoniewicz MR, Stephanopoulos G et al (2008) Quantifying reductive carboxylation flux of glutamine to lipid in a brown adipocyte cell line. J Biol Chem 283:20621–20627
26. Ausloos P, Clifton CL, Lias SG et al (1999) The critical evaluation of a comprehensive mass spectral library. J Am Soc Mass Spectrom 10:287–299
27. Kopka J, Schauer N, Krueger S et al (2005) GMD@CSB.DB: the Golm Metabolome Database. Bioinformatics 21:1635–1638
28. Kind T, Wohlgemuth G, Lee DY et al (2009) FiehnLib: mass spectral and retention index libraries for metabolomics based on quadrupole and time-of-flight gas chromatography/mass spectrometry. Anal Chem 81:10038–10048
29. Smith CA, O'Maille G, Want EJ et al (2005) METLIN: a metabolite mass spectral database. Ther Drug Monit 27:747–751
30. Wishart DS, Tzur D, Knox C et al (2007) HMDB: the Human Metabolome Database. Nucleic Acids Res 35:D521–D526
31. Horai H, Arita M, Kanaya S et al (2010) MassBank: a public repository for sharing mass spectral data for life sciences. J Mass Spectrom 45:703–714
32. Glacken MW, Adema E, Sinskey AJ (1988) Mathematical descriptions of hybridoma culture kinetics: I. Initial metabolic rates. Biotechnol Bioeng 32:491–506
33. Goudar CT (2012) Computer programs for modeling mammalian cell batch and fed-batch cultures using logistic equations. Cytotechnology 64(4):465–475
34. Kim J-w, Tchernyshyov I, Semenza GL et al (2006) HIF-1-mediated expression of pyruvate dehydrogenase kinase: a metabolic switch required for cellular adaptation to hypoxia. Cell Metab 3:177–185
35. Zupke C, Sinskey AJ, Stephanopoulos G (1995) Intracellular flux analysis applied to the effect of dissolved oxygen on hybridomas. Appl Microbiol Biotechnol 44:27–36
36. Zamboni N, Fendt S-M, Rühl M et al (2009) (13)C-based metabolic flux analysis. Nat Protoc 4:878–892
37. Oldiges M, Kunze M, Degenring D et al (2004) Stimulation, monitoring, and analysis of pathway dynamics by metabolic profiling in the aromatic amino acid pathway. Biotechnol Prog 20:1623–1633
38. Roessner U, Wagner C, Kopka J et al (2000) Technical advance: simultaneous analysis of metabolites in potato tuber by gas chromatography–mass spectrometry. Plant J 23:131–142
39. Luo B, Groenke K, Takors R et al (2007) Simultaneous determination of multiple intracellular metabolites in glycolysis, pentose phosphate pathway and tricarboxylic acid cycle by

liquid chromatography-mass spectrometry. J Chromatogr A 1147:153–164
40. Kitson FG, Larsen BS, McEwen CN (1996) Gas chromatography and mass spectrometry: a practical guide. Academic, San Diego
41. Kiefer P, Nicolas C, Letisse F et al (2007) Determination of carbon labeling distribution of intracellular metabolites from single fragment ions by ion chromatography tandem mass spectrometry. Anal Biochem 360:182–188
42. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2007) Accurate assessment of amino acid mass isotopomer distributions for metabolic flux analysis. Anal Chem 79:7554–7559
43. Fernandez CA, Des Rosiers C, Previs SF et al (1996) Correction of 13C mass isotopomer distributions for natural stable isotope abundance. J Mass Spectrom 31:255–262
44. Allen DK, Shachar-Hill Y, Ohlrogge JB (2007) Compartment-specific labeling information in 13C metabolic flux analysis of plants. Phytochemistry 68:2197–2210
45. Coplen TB, Bohlke JK, De Bievre P et al (2002) Isotope-abundance variations of selected elements: (IUPAC technical report). Pure Appl Chem 74:1987–2017
46. Dulmage AL, Mendelsohn NS (1958) Coverings of bipartite graphs. Canad J Math 10:517–534
47. Pothen A, Fan CJ (1990) Computing the block triangular form of a sparse matrix. ACM Trans Math Softw 16:303–324
48. Gill PE, Murray W, Wright MH (1981) Practical optimization. Academic, London
49. Madsen K, Nielsen HB, Tigleff O (2004). Methods for non-linear least squares problems. http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3215.
50. Antoniewicz MR, Kelleher JK, Stephanopoulos G (2006) Determination of confidence intervals of metabolic fluxes estimated from stable isotope measurements. Metab Eng 8:324–337
51. Conover WJ (1999) Practical nonparametric statistics. Wiley, New York
52. Ahn WS, Antoniewicz MR (2011) Metabolic flux analysis of CHO cells at growth and non-growth phases using isotopic tracers and mass spectrometry. Metab Eng 13:598–609
53. Rohn H, Hartmann A, Junker A et al (2012) FluxMap: a VANTED Add-on for the visual exploration of flux distributions in biological networks. BMC Syst Biol 6:33
54. König M, Holzhütter H-G (2010) Fluxviz—cytoscape plug-in for visualization of flux distributions in networks. Genome informatics International conference on genome informatics 24:96–103
55. Hoppe A, Hoffmann S, Gerasch A et al (2011) FASIMU: flexible software for flux-balance computation series in large metabolic networks. BMC Bioinformatics 12:28
56. Droste P, Miebach S, Niedenführ S et al (2011) Visualizing multi-omics data in metabolic networks with the software Omix: a case study. Biosystems 105:154–161
57. Paley SM, Karp PD (2006) The pathway tools cellular overview diagram and omics viewer. Nucleic Acids Res 34:3771–3778
58. Matthews L, Gopinath G, Gillespie M et al (2009) Reactome knowledgebase of human biological pathways and processes. Nucleic Acids Res 37:D619–D622
59. Kono N, Arakawa K, Ogawa R et al (2009) Pathway projector: web-based zoomable pathway browser using KEGG atlas and Google Maps API. PLoS One 4:e7710
60. Lee SY, Lee D-Y, Hong SH et al (2003) MetaFluxNet, a program package for metabolic pathway construction and analysis, and its use in large-scale metabolic flux analysis of *Escherichia coli*. Genome informatics International conference on genome informatics 14:23–33
61. Rocha I, Maia P, Evangelista P et al (2010) OptFlux: an open-source software platform for in silico metabolic engineering. BMC Syst Biol 4:45

Chapter 19

Sample Preparation and Biostatistics for Integrated Genomics Approaches

Hein Stam, Michiel Akeroyd, Hilly Menke, Renger H. Jellema, Fredoen Valianpour, Wilbert H.M. Heijne, Maurien M.A. Olsthoorn, Sabine Metzelaar, Viktor M. Boer, Carlos M.F.M. Ribeiro, Philippe Gaudin, and Cees M.J. Sagt

Abstract

Genomics is based on the ability to determine the transcriptome, proteome, and metabolome of a cell. These technologies only have added value when they are integrated and based on robust and reproducible workflows. This chapter describes the experimental design, sampling, sample pretreatment, data evaluation, integration, and interpretation. The actual generation of the data is not covered in this chapter since it is highly depended on available equipment and infrastructure.

The enormous amount of data generated by these technologies are integrated and interpreted inorder to generate leads for strain and process improvement. Biostatistics are becoming very important for the whole work flow therefore, some general recommendations how to set up experimental design and how to use biostatistics in enhancing the quality of the data and the selection of biological relevant leads for strain engineering and target identification are described.

Key words: Systems biology, Metabolic engineering, Genomics, Transcriptomics, Proteomics, Metabolomics, Biostatistics

1. Introduction

Applied genomics is gathering pace due to the increasing availability of genome sequence information and technological advances in the various omics fields, resulting in decreased costs and increased quality and speed. This enables novel ways of improving strains and processes and for deciphering complex cellular pathways. Crucial for effective use of the genomics toolbox is an integrative approach based on good experimental design, reliable procedures, and advanced biostatistics.

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_19,

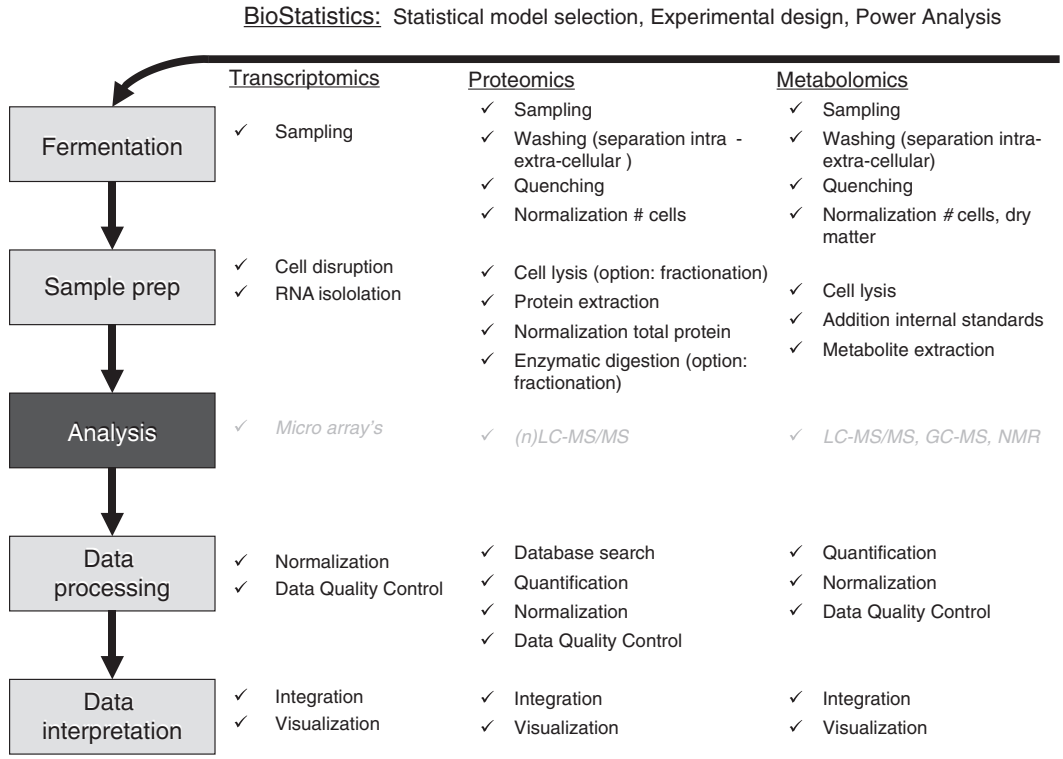

Fig. 1. Overview of integrated genomics approach described in this chapter. Sample preparation and biostatistical approaches for experimental setup and data processing are described in this chapter. Analysis of the samples is outside the scope of this chapter since it is highly dependent on available infrastructure of the laboratory.

In order to obtain high-quality data, it is important to build upon proven procedures for reproducible cultivation, sampling, sample pretreatment, labeling, measurements, and pre- and post-processing of data and biostatistics linked to a well-defined biological question. This chapter describes robust approaches which have shown to generate high-quality data. The actual analysis of the samples is outside the scope of this chapter since it is highly dependent on the available infrastructure in the laboratory, see Fig. 1.

The selection of leads out of the large datasets which are typically generated using genomics is challenging. The systems biology approach aims to integrate the data generated with various omics techniques into in-depth knowledge of the status of the cell. This knowledge has to be translated in biological relevant leads or hypotheses which should be followed up by experiments in the wet lab [1]. Since the throughput of strain construction is by far not on the same level as data generation, an effective selection is the key for successful strain improvement.

Biological regulation is much more complex than the central dogma of DNA–RNA–protein. It is this variety of regulatory

mechanisms which calls for a multidisciplinary approach for effective strain and process improvement. The different levels of regulation, on protein level and mRNA level, make straightforward genome modification not always successful in strain optimization. Therefore, the selection of biological relevant data remains a combination of profound knowledge on cellular architecture, physiology, and classical disciplines like biochemistry and microscopy.

This chapter outlines an effective approach for lead selection including biostatistics and experimental protocols.

2. Materials

Prepare all solutions using ultrapure water and analytical grade reagents. Prepare and store all reagents at room temperature, unless indicated otherwise.

2.1. RNA Isolation

To avoid degradation of RNA, all materials must be RNase-free:

2.1.1. RNA Isolation from Yeast and Fungi

Trizol.

Chloroform.

RNeasy Maxi Kit from QIAGEN.

RNase-free MQ.

96 % ethanol.

DEPC-treated water.

Ethanol, absolute.

3 M sodium acetate pH 5.2.

Liquid N_2, dry ice can be an alternative.

Phenol (optional).

RNeasy Midi kit (Qiagen).

2.2. Proteomics

Buffer A: 0.1 % FA in water (cat # 232441 Biosolve).

Buffer B: 0.1 % FA in AcN (cat # 019341 Biosolve).

DL-dithiothreitol (DTT, Sigma Aldrich D0632).

Iodoacetamide (IAA, Sigma Aldrich I1149).

Trypsin (sequencing grade cat # 11047841001, Roche Applied Sciences, Penzberg, Germany, or equivalent).

Ammonium bicarbonate (NH_4HCO_3, Sigma Aldrich 09830).

Bovine serum albumin (BSA, Sigma Aldrich A9418).

Protein determination: Qubit protein assay (cat # Q33212, Invitrogen).

Lysis tubes VK05 (cat # 03961-1-004, Bertin, Montigny-le-Bretonneux, France).

Centrifuge: 5417R (Eppendorf, Hamburg, Germany) or equivalent.

Columns: Zorbax SB-C_{18} 2.1 × 50 mm, Poroshell 300 SB-C_3 2.1 × 12.5 mm, (Agilent, Santa Clara, CA, USA).

LC–MS/MS: Accela-LTQ-Velos (Thermo Scientific, Waltham, MA, USA) or equivalent.

Sequest search engine: Sorcerer 2 (Sagen, San Diego, CA, USA).

Precellys (Bertin, Montigny-le-Bretonneux, France).

Reagents to prepare:

100 mM NH_4HCO_3 (pH 7.8).

500 mM DTT, in 100 mM NH_4HCO_3.

550 mM IAA, in 100 mM NH_4HCO_3, should be prepared freshly before adding.

0.1 mg/ml BSA in 100 mM NH_4HCO_3.

0.25 mg/ml trypsin, in 0.01 N HCl pH 3.

Visualization software: Spotfire (TIBCO, Palo Alto, CA, USA).

2.3. Metabolomics

Acetic acid (Merck, 1.00063.1011).

Tributylamine (Sigma, 471313).

Acetonitril LiChrosolve (Merck, 1.00030.2500).

Water (Fluka, 39253-1L-R).

Methanol (Merck, 1.06007.2500).

Ethanol (VWR, 83804.360).

Centrifuge: Multifuge 4KR (Heraeus, Hanau, Germany).

Or equivalent.

References, standards, and controls

FBP (fructose bis phosphate), Fluka, 79470.

G1P (glucose-1-phosphate), Sigma, G9380.

G6P (glucose-6-phosphate), Sigma, G6526.

F6P (fructose-6-phosphate), Sigma, F1502.

2PG (2-phosphoglyceric acid), Fluka, 79470.

PEP (phospho enol pyruvic acid), Fluka, 860077.

AMP (adenosine mono phosphate), Sigma, A1752.

ADP (adenosine di phosphate), Sigma, 01905.

ATP (adenosine tri phosphate), Sigma, A7699.

NAD (nicotinamide adenine dinucleotide), Sigma, N7132.

NADH (reduced nicotinamide adenine dinucleotide), Sigma, N4505.

NADP (nicotinamide adenine dinucleotide phosphate), Sigma, 3205.

NADPH (reduced nicotinamide adenine dinucleotide phosphate), Sigma, N5130.

FAD (flavine adenine dinucleotide), Sigma, F6625.

Ppi (pyrophosphate), Sigma, P8010.

Citric acid, Sigma, C0759.

Isocitric acid, Sigma, I1252.

OAA (oxalo acetic acid), Sigma, O4126.

Pyruvic acid, Sigma, P2256.

Malic acid, Sigma, M0875.

Succinic acid, Fluka, 14080.

Fumaric acid, Sigma, F8509.

AKG (alpha keto gluterate), Sigma, K1750.

Labeled yeast ($U^{13}C$) extract as internal standard.

Or equivalent.

Reagents to prepare:

Mobile phase A: 10 mM tributylamine (TBA) + 15 mM acetic acid (HAc) in MilliQ water: MeOH 95:5 v/v. To 50 ml MeOH, first add, while stirring, 870 µl HAc followed by 2.42 ml TBA. When dissolved, add 950 ml water.

Mobile phase B: 10 mM tributylamine (TBA) + 15 mM acetic acid (HAc) in acetonitrile (ACN). To 1,000 ml ACN, first add, while stirring, 870 µl HAc followed by 2.42 ml TBA.

Needle flush: MilliQ water/acetonitrile/FA 95/5/0.1 v/v/v.

3. Methods

3.1. Transcriptomics Sample Preparation for Filamentous Fungi and Yeast

1. Grow the selected strain using the desired conditions, e.g., conditions for specific mRNA induction.
2. Directly freeze the fermentation broth in liquid nitrogen. Store the frozen material at −80 °C. For most of the standard media, this procedure works fine. If very complex media or high concentrations of specific chemicals or extremely low or high pH are used, it is better to filter the mycelium. For sampling on large scale, see Note 1.
3. Chill pestle and mortar with liquid nitrogen. Grind the biomass under liquid nitrogen; transfer the grinded sample with a chilled spoon to a 50 ml Greiner tube till the 5 ml mark. Immediately proceed with extraction procedure described in **step 4** or store the grinded biomass at −80 °C. Remaining

grinded biomass can be stored in an extra 50 ml Greiner tube at −80 °C.

4. Take a 50 ml Greiner tube containing 5 ml grinded biomass from −80 °C and immediately add 15 ml TRIzol reagent.
5. Quickly resuspend the sample completely in TRIzol by vortexing for at least 1 min until a clear solution is obtained. Incubate at room temperature for 5 min.
6. Spin the cell debris down for 10 min at 3,000 × *g* at 4 °C.
7. Transfer the supernatant including the organic phase to a 50 ml Phase Lock Heavy tube.
8. Add 0.2 volume of chloroform (=3 ml for 15 ml of TRIzol) and mix by vortexing for at least 1 min.
9. Centrifuge for 45 min at 4,000 × *g* at 4 °C.
10. Transfer the upper aqueous phase (about 60 % of the volume of TRIzol reagent used) to an RNase-free 50 ml Greiner tube.
11. Add an equal volume of RNase-free 70 % ethanol and mix.
12. Apply 4 ml of the mix to an RNeasy Midi column.
13. Centrifuge for 5 min at 1,500 × *g* and 20 °C. Discard the flow-through.
14. Apply the rest of the mix to the column.
15. Centrifuge for 5 min at 1,500 × *g* at 20 °C. Discard the flow-through.
16. If necessary, repeat **steps 16** and **17** till all the mixture is applied to the column.
17. Wash the column with 4 ml RW1 buffer. Centrifuge for 5 min at 1,500 × *g* at 20 °C. Discard the flow-through.
18. Wash the column with 2.5 ml RPE buffer. Centrifuge for 1 min at 1,500 × *g* at 20 °C. Discard the flow-through.
19. Wash the column with 2.5 ml RPE buffer. Centrifuge for 2 min at 1,500 × *g* at 20 °C. Discard the flow-through.
20. To dry the column, centrifuge 8 min, 1,500 × *g* at 20 °C. Discard the flow-through and replace the waste tube with a clean collection tube.
21. To elute the RNA, add 0.5 ml RNase-free water to the column and incubate 1 min at room temperature.
22. Centrifuge for 5 min at 1,500 × *g* at 20 °C.
23. Repeat steps 22 and 23.
24. Distribute the eluate over 2 2 ml Eppendorf test tubes. Store the RNA as an alcohol precipitate by addition of 1/10 volume of 3 M NaAc and 3 volumes of 96 % ethanol. Store the RNA at least overnight at −20 °C.

25. Pellet the RNA by centrifuging for 15 min 11,000 × *g* at 4 °C.
26. Discard the supernatant, air-dry the pellet, and dissolve it in 100 µl RNase-free MQ.

Determine the amount of RNA on the spectrophotometer (see Note 2). Dilute the RNA sample (normally a dilution of 50 to 100× can be measured). Store the RNA samples at −80 °C.

3.2. Proteomics Sample Preparation for Yeast

1. Find the best solution to obtain a homogeneous representative sample and a workable volume. Generally, ~100 µg total protein is sufficient for protein analysis. Make duplicate samples for OD determination or cell counts (see Note 3).
2. Form cell pellets by centrifugation (10 min 800 × *g* at 4 °C) and wash cell pellets with twice the volume physiological salt solution at 4 °C to remove extracellular proteins.
3. Determine the OD_{600nm} of the washed cell pellets from the duplicate samples by suspension in physiological salt solution at 4 °C. Normalize samples to $OD_{600} = 1$ in MeOH. Take 1 ml cell suspension to start with approximately equal numbers of cells. Transfer samples to lysis tubes (see Note 4).
4. Lyse the cells using the Precellys, 2× 20 s at 6,000 rpm with 10 s pause. Cool samples on ice as much as possible (see Note 5).
5. Determine total protein concentration with the QUBIT according to the protocol supplied by the vendor. Take a volume corresponding to 100 µg of total protein and spike in 1 µg BSA as an internal control.
6. Perform "Bligh and Dyer" extraction [2]. Carefully remove the top and lower layers and continue with the protein precipitate (see Note 6).
7. Solubilize the precipitate as much as possible in 50 µl 50 mM NaOH, dilute with 400 µl mM NH_4HCO_3 take 5–10 µl to perform a protein determination with QUBIT according to the protocol supplied by the vendor. Normalize samples to a concentration of 100 µg/ml in 100 mM NH_4HCO_3 (see Note 7).
8. Add DTT to a final concentration of 5 mM and incubate at room temperature for 30 min. Add freshly prepared IAA to a final concentration of 5.5 mM and incubate at room temperature for 30 min in the dark. Add 20 µl 0.25 mg/ml trypsin pH 3 and incubate at 37 °C overnight. Add 4 µl 0.25 mg/ml trypsin pH 3 and incubate at 37 °C for 3 h to ensure complete digestion (see Note 8). Acidify the digests to pH 3 with formic acid (FA). Generate an extra sample by a 1:1 mix of all samples (see Note 9).

9. The focus of this chapter is on sample preparation for genomics approaches (see Note 10); therefore, the analysis will not be discussed in detail. Briefly, protein digests should be analyzed with U-HPLC–MS/MS system capable of data-dependent MS/MS (LTQ or equivalent). Use a randomized injection sequence to minimize sample-specific memory effects and "MS in time effects." Analyze samples in triplicate with U-HPLC–MS/MS, using a 180 min data-dependent LC–MS/MS run, 0–150 min 5–30 % buffer B, 150–170 min 30–40 % buffer B, 170–175 min 80 % buffer B, 175–180 min 5 % buffer B. Separate peptides by 25 µl injection on a C_{18} column using a guard column at 50 °C and a flow rate of 0.4 ml/min. Measure MS data 300–2,000 *m/z* and perform MS/MS on the 2+ and 3+ ions. Enable dynamic exclusion to optimize the number of identified proteins. Quality of the raw MS data should be checked by an MS expert. This method is sufficient for analysis of the top ~1,000 proteins (see Note 11).
10. Search the data in the *Saccharomyces cerevisiae* (or other relevant) fasta database including a decoy database [3] using a SEQUEST database search engine or equivalent. Chose filter criteria such that the false positives are below 1 %.
11. Quantify the proteins using APEX [4] based on spectral counting and machine learning. Required are the .prot.xml files, the used fasta database, and an accession file. Use the mix sample to generate the accession file by selecting the accessions of all proteins identified with protein probability (PP) = 1. The APEX quantification consists of three stages:
    (a) Build a training file (.arrf) to determine the important physical and chemical properties of peptides to be identified in your experimental setup.
    (b) Generate the predicted counts (Oi values) for each peptide in the fasta database.
    (c) Calculate APEX scores for the individual samples, using the .prot.xml files. Combine the data in an excel file.

    It is possible to perform quantification based on the isotope-labeled standards using MSQuant, MAXQuant, or equivalent or quantify in MS/MS based on the iTRAQ reporter ratios. Determine absolute quantities of the proteins for which an AQUA peptide or QConCat standard was added by determining the areal ratio of the internal standard peptide to its natural counterpart.
12. Import the data in Spotfire or equivalent. Inspect if the protein abundances correlate between the triplicate injections. Check if the internal control is present in a 1:1 ratio. Average the triplicates and check the RSDs of the APEX scores and if the protein

abundances correlate between samples. Normalize the APEX scores based on the scores of household proteins and BSA.

13. The interesting leads can be confirmed using a targeted (SRM) method.

3.3. Metabolomics Sample Preparation for Yeast

1. At least 1 day before the quenching, the reception plates are filled with 3.2 ml quenching solution (in this case methanol) per well and precooled to at least −40 °C in a freezer.
2. The samples are grown (biomass phase and production phase) according to an optimized procedure depending on the organism used in a sterilized fritted plate with sterile 4 mm beads. OD should be above 1 and the volume 400 µl or 600 µl when one needs to measure the OD.
3. Quench the samples by placing the fritted cultivation plate over the cold reception plate in the vacuum manifold. Quench 2 wells of 400 µl culture (recommended OD 0.8–1.5) into one well of reception plate, mix briefly, and cool for approximately 1 min in an ice-cold ethanol bath supplied with dry ice or in a precooled chiller set at −40 °C.
4. Centrifuge 4 min at 2,700 × *g* at −9 °C.
5. Discard the supernatant.
6. If necessary, wash the cells by adding approximately 3 ml precooled methanol (keep the methanol at all times in an ethanol bath supplied with dry ice or a chiller set at −40 °C) to each well.
7. Shake and centrifuge 4 min at 2,700 × *g* at −9 °C.
8. Discard the supernatant and repeat the steps a second time (see Note 12).
9. To the pellet, add exactly 100 µl of internal standard and 500 µl of boiling water. Vortex and cook 10 min at 100 °C. A 48-square-well silicon lid should be used to prevent evaporation.
10. Remove the lid and place a new one.
11. Cool down 1 min at 0 °C (water bath supplied with ice).
12. Centrifuge 5 min 1,000 × *g* at 0 °C.
13. The supernatant can be used for analysis.
14. Prepare stock solutions as follows: weigh 4 mg for all metabolites to be analyzed listed in Subheading 2, except fumaric acid, and dissolve in MilliQ water to obtain a concentration of 4 mg/ml. For fumaric acid, make a solution of 1 mg/ml. Mix 100 µl from each solution (fumaric acid 400 µl) to end up with a mixture of the 23 components with a concentration of 100 µg/ml = (A). Dilute this mixture further 10 times with MilliQ water to obtain a concentration of 10 µg/ml = (B).

15. Use solution (A) and (B) to prepare the calibration curve.
16. As internal standard, preferably use $U^{13}C$-labeled yeast extract. This is a broth extract whereby the yeast is grown using labeled $U^{13}C$-glucose only. In this way, all components formed should be $U^{13}C$ labeled.
17. Analyze the samples using the best available method (e.g., LC–MS, GC–MS, NMR) for answering the biological question. The analysis itself is outside the scope of this book chapter.
18. Design and perform statistical evaluation of the data.

3.4. Biostatistics

Biostatistics covers the steps from experimental design to raw data processing and statistical assessment to facilitate interpretation. The handling of raw data from large-scale functional genomics experiments requires powerful data processing equipment and algorithms. Dedicated software packages exist that help to quantify signals, reduce noise, and correct for aberrancies introduced by the technical setup. The steps in preprocessing of raw data to clean data and the options for data mining and interpretation are described in a step-wise manner below, but several steps may be performed in an interchangeable order.

In the field of applied genomics, an overwhelming amount of data is generated, and statistics is essential to obtain objective answers. A general approach for statistical assessment, applicable for all omics techniques, is shown below. Some references to technique-specific approaches are introduced later in this section.

3.4.1. Experimental Design

In order to extract biological information from genomics experiments, an excellent experimental design is crucial. The complexity of biological systems requires the correct number of biological and analytical replicates. Each part of the experiment should be considered, being sampling, sample pretreatment, as well as analysis, to account for the possible sources of variation that may arise, technically as well as biologically.

The first step in performing proper statistics is having a clear problem definition. Fishing expeditions often fail due to lack of a problem definition. Trying to answer multiple questions within the same experiment has a high risk of failure and often results in a (too) large experiment. It is preferred to validate the individual parts of a large experiment prior to performing measurements on the (often expensive) samples. By testing the influence of possible sources of variation in validation studies, the experimenter can focus in the study on those aspects (e.g., temperature or substrate composition) that need to be investigated.

Once a clear problem definition is given, a model needs to be chosen to process the obtained data. This can be either a simple *fold change* model but also more complex and informative multiway

ANOVA models or multivariate models can be chosen. The model selection is closely related to the problem definition that has been defined.

Information is required about the variability within the experiment and the expected effects due to the experimental settings. Earlier experiments and specifications presented by instrument or assay suppliers are most of the times the only source of information for those numbers. Also, it is often difficult to give an estimate what differences are of interest. Still, this information is required to perform the next step in the total process: power analysis. With power analysis and often also with simulations using assumptions about the magnitude of variances within the experiments, the statistical expert is able to estimate the chance of success to solve the problem at hand (often the question to be answered is "how many repeated experiments do we need to do to find a significant difference with a given magnitude").

3.4.2. Sample Description

The interpretation of "genomics" information from microbial fermentations is significantly facilitated by a good knowledge of the sample origin. It is therefore recommended to control as much as possible the environment conditions of the microorganism: lab-grade medium ingredients, pH, temperature, nutrients concentrations, metabolism-controlling fluxes, and specific growth rates. Furthermore, once a (pseudo) steady state has been reached and sampled, the pulse disturbance of the fermentation conditions can then be monitored.

The information that describes the experimental conditions should be secured in detail, preferably automatically stored in a Laboratory Information Management System (LIMS), and organized in a searchable manner.

3.4.3. Raw Data Processing

The raw data consist of digitalized images that can be very large in size and are often discarded after quantification and storage as numerical data files. MS data are stored completely and can be used later for reprocessing or later retrieval of details needed, for instance, to confirm presence of compounds. After collection of the data, systematic bias originating from various technical sources should be corrected. A signal-to-noise threshold should exclude weak and unreliable signals from further interpretation.

To bring data on an equal scale prior to data analysis, scaling or normalization is required. For scaling an arbitrary, signal is designated toward which the average intensity of all signals (gene, metabolite, protein) is scaled per sample, within a dataset, to the target signal you specified. This process enables you to compare multiple samples within a dataset. We advise to use the same target signal across all samples being compared. Scaling can be performed independently of the comparison analysis. Scaling of the data corrects for experiment-wide differences, for example, in the different

amounts of starting material or luminescence recording time (total signal intensity). Normalization is performed by dividing multiple sets of data by a common variable (e.g., optical density or dry weight of the sample) in order to diminish that variable's effect on the data, see Note 13. This allows for comparison of the underlying characteristics of the data: the data is brought to a common scale, independent of undesired sample differences like concentration. Normalization should not be performed if quantitative data is available and needs to be used, for example, to perform kinetic modeling. Normalization usually assumes that on average over the total pool of transcripts or proteins, no changes are found between samples.

3.4.4. Data Storing and Deposit

After raw data processing, secondary results files are obtained that should be stored along with the description of the method of data analysis.

For GeneChips or cDNA microarrays, data should be treated and stored according to internationally established MIAME requirements (Minimal Information About Microarray Experiments) [5]. Similar standards have been developed for other types of data within the genomics fields as well, among others by FGED (Functional Genomics Data Society). Data fulfilling these requirements should be deposited for publication at, for instance, ArrayExpress (EBI) or GEO (NCBI). The clean data files are the basis for the interpretation of the results, described in the next section.

3.4.5. Biological Interpretation and Hypothesis Generation

To assess relationships between molecules, the large datasets have to be structured and organized, also called data mining. Multivariate statistical techniques and the (automated) integration of measurements with preexisting knowledge can facilitate data mining.

Genomics data may be analyzed with univariate and/or multivariate statistical techniques. Univariate statistical methods (e.g., ANOVA, Student's t-tests) are efficient and typically used to determine significance of changes in a single parameter under study. Mathematical clustering algorithms like hierarchical clustering, K-means clustering, and self-organizing maps calculate a measure of similarity between levels of gene or proteins in profiles. Clustering creates subsets of molecules that behave similar to each other and enables to select molecules with biologically relevant characteristics. Clustering may be based on expression level, fold change, etc., such that trends of expression (e.g., during fermentation time or within a strain lineage) are identified. Within these clusters, functional classes may be over- or underrepresented. Functional classes can, for example, be biological processes, cellular localization, or protein complexes.

Unsupervised methods such as principal component analysis (PCA) determine intrinsic structure within datasets, without prior

knowledge. PCA can be used to calculate a measure of similarity between samples measurements within large datasets, such as LC–MS, GC–MS, NMR spectra, or DNA microarray measurements. Supervised methods such as partial least squares (PLS) and principal component discriminant analysis (PCDA) use additional information such as biochemical, histopathological, or clinical data to optimize the discrimination between samples. These methods allow to identify the molecules in the large datasets that most contribute to differences between the samples, for instance, in samples obtained from different growth experiments or varied nutrient sources.

If more than one dataset and multivariate techniques are used, data fusion [6] can help greatly in identifying cross-platform correlations. Examples [7] show that proper scaling of the underlying datasets is critical to find all relevant compounds, independent of the error structure in each individual dataset.

Structuring of large datasets can be facilitated with the use of preexisting knowledge about biological processes. The incorporation of methods for automated literature searching, text mining, and interpretation of data into statistical software speeds up this process.

By categorizing and visualizing the molecules according to biological processes, rather than finding changes in one enzyme, changes in entire biochemical pathways can be identified in response to applied conditions. A nowadays common technique is enrichment analysis like Gene Set Enrichment Analysis (GSEA) and Metabolite Set Enrichment Analysis (MSEA). Enrichment analysis techniques take the results of a statistical analysis and determine the chance that a set of, e.g., metabolites that are known to be related to each other in a distinct pathway are either down- or upregulated. When using a well-annotated organism, the statistical assessment of over- or underrepresented functional classes in a group of genes can be determined using a hypergeometric distribution algorithm. The p-values calculated using such a tool represent the probability that the number of genes associated with such a functional class could have been found by chance. Use of these algorithms can give direct leads for process or strain improvement.

4. Notes

1. At production scale (>100,000 l), there are issues to consider during sampling. The sample volume taken from an industrial fermentation should be designed in order to have representative samples. Sometimes the sampling pipes and port may still contain leftovers of broth from previous samples; therefore, it is

important to flush the pipes with a copious amount of broth before taking the sample. In addition, cells passing, e.g., the heat exchanger may induce expression of heat shock proteins within minutes. Depending on mixing times, the sample may contain a heterologous population, and the use of complex raw materials can influence reproducibility. In a factory environment, it is advisable to prepare well in advance since regulations are of a different level compared to (academic) laboratories.

2. The A_{260}/A_{280} is a measure for the purity of the RNA sample and must be greater than 1.8. To determine a reliable RNA concentration, the measured A_{260} must lie between 0.1 and 1. If the measured A_{260} is higher than 1, the sample should be diluted accordingly.
3. Mix metabolic heavy-labeled fermentation samples with their non-labeled counterparts [8]. Also, cell sorting can be applied at this stage to obtain more homogeneous cell populations.
4. Determine cell counts or weigh 0.5 mg lyophilized cells.
5. Alternatively, yeast protein extraction reagents from Pierce Protein Research Products can be used, a dismembrator or equivalent, to lyse the cells; the lysis method will influence the protein isolation.
6. Perform TCA precipitation or equivalent instead of extraction. This step greatly influences the protein isolation and therefore has a big impact on the end results.
7. Run an SDS PAGE and follow an in-gel digestion protocol [9]. Spike in AQUA peptides [10] with internal trypsin cleavage site or QConCat standard for absolute quantification.
8. Keep in mind that any proteolytic activity, other than step 8, will corrupt the results. Also, contamination of samples with other protein sources, e.g., keratins from skin and hair, should be avoided. Limit the number of pipette steps and work with calibrated pipettes. Work in low-protein-binding tubes as much as possible.
9. Other proteases in parallel to trypsin can be used to obtain better sequence coverage. Label samples with iTRAQ [11] for MS/MS quantification.
10. For proteomics and metabolomics, a targeted or untargeted approach can be chosen. The targeted approach concerns the identification and quantification of defined sets of proteins/metabolites, e.g., involved in a metabolically engineered pathway. The untargeted approach involves the analysis of all the metabolites/proteins at a certain time point under certain conditions. In this chapter, we focus on untargeted genomics approaches.

11. Longer gradients, other columns, or multidimensional separations can be applied to identify more peptide, or purify PTM containing peptides, for in-gel approaches as nano-LC–MS/MS or MALDI.
12. During the setup of experiments, one should check for leakage of the metabolites out of the cell by measuring those metabolites in the washing solution.
13. For the widely used Affymetrix GeneChips®, there are standardized algorithms in the dedicated software (e.g., MAS or GCOS or Expression Console, www.affymetrix.com). For proteomics and metabolomics, most of the normalization is performed by using internal standards. Preferred is to have one individual internal standard per compound of interest. If this is not feasible, groups of compounds can be assigned to single internal standards with similar behavior in experimental variations. Ideally, the concentrations of all compounds are determined from calibration lines obtained for each compound individually.

Acknowledgements

We sincerely thank Prof. Dr. Uwe Sauer and his group, especially Stefan Christen from ETH Zurich Institute of Molecular Systems Biology Zurich, Switzerland, for their time and help in starting the metabolomics work. Equally, we thank Prof. Dr. Joseph J. Heijnen and his group especially Dr. Lodewijk de Jonge from Department of Biotechnology, Faculty of Applied Sciences, Technical University of Delft, The Netherlands, for their contribution and collaboration on earlier works on metabolomics.

References

1. Jacobs DI, Olsthoorn MM, Maillet I, Akeroyd M, Breestraat S, Donkers S, van der Hoeven RA, van den Hondel CA, Kooistra R, Lapointe T, Menke H, Meulenberg R, Misset M, Müller WH, van Peij NN, Ram A, Rodriguez S, Roelofs MS, Roubos JA, van Tilborg MW, Verkleij AJ, Pel HJ, Stam H, Sagt CM (2009) Effective lead selection for improved protein production in Aspergillus niger based on integrated genomics. Fungal Genet Biol 46(Suppl 1): S141–S152
2. Bligh EG, Dyer WJ (1959) A rapid method of total lipid extraction and purification. Can J Biochem Physiol 37(8):911–917
3. Gupta N, Bandeira N, Keich U, Pevzner PA (2011) Target-decoy approach and false discovery rate: when things may go wrong. J Am Soc Mass Spectrom 22(7):1111–1120
4. Lu P, Vogel C, Wang R, Yao X, Marcotte EM (2007) Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. Nat Biotechnol 25(1):117–124
5. Brazma A, Hingamp P, Quackenbush J, Sherlock G, Spellman P, Stoeckert C, Aach J, Ansorge W, Ball CA, Causton HC, Gaasterland T, Glenisson P, Holstege FCP, Kim IF, Markowitz V, Matese JC, Parkinson H, Robinson A,

Sarkans U, Schulze-Kremer S, Stewart J, Taylor R, Vilo J, Vingron M (2001) Minimum information about a microarray experiment (MIAME) - toward standards for microarray data. Nat Genet 29(4):365–371

6. Smilde AK, Van Der Werf MJ, Bijlsma S, Van Der Werff-Van Der Vat BJC, Jellema RH (2005) Fusion of mass spectrometry-based metabolomics data. Anal Chem 77 (20):6729–6736
7. Doeswijk TG, Hageman JA, Westerhuis JA, Tikunov Y, Bovy A, van Eeuwijk FA (2011) Canonical correlation analysis of multiple sensory directed metabolomics data blocks reveals corresponding parts between data blocks. Chemometr Intell Lab 107(2):371–376
8. Munday DC, Surtees R, Emmott E, Dove BK, Digard P, Barr JN, Whitehouse A, Matthews D, Hiscox JAT (2012) Using SILAC and quantitative proteomics to investigate the interactions between viral and host proteomes. Proteomics 12(4–5):666–672
9. Kang S-U, Fuchs K, Sieghart W, Pollak A, Csaszar E, Lubec G (2009) Gel-based mass spectrometric analysis of a strongly hydrophobic GABAA-receptor subunit containing four transmembrane domains. Nat Protoc 4 (7):1093–1102
10. Kettenbach AN, Rush J, Gerber SA (2011) Absolute quantification of protein and post-translational modification abundance with stable isotope-labeled synthetic peptides. Nat Protoc 6(2):175–186
11. Evans C, Noirel J, Ow SY, Salim M, Pereira-Medrano AG, Couto N, Pandhal J, Smith D, Pham TK, Karunakaran E, Zou X, Biggs CA, Wright PC (2012) An insight into iTRAQ: where do we stand now? Anal Bioanal Chem 2012(404):1011–1027

Part IV

Integrating Large Datasets for Modeling and Engineering Applications

Chapter 20

Targeted Metabolic Engineering Guided by Computational Analysis of Single-Nucleotide Polymorphisms (SNPs)

D.B.R.K. Gupta Udatha, Simon Rasmussen, Thomas Sicheritz-Pontén, and Gianni Panagiotou

Abstract

The non-synonymous SNPs, the so-called non-silent SNPs, which are single-nucleotide variations in the coding regions that give "birth" to amino acid mutations, are often involved in the modulation of protein function. Understanding the effect of individual amino acid mutations on a protein/enzyme function or stability is useful for altering its properties for a wide variety of engineering studies. Since measuring the effects of amino acid mutations experimentally is a laborious process, a variety of computational methods have been discussed here that aid to extract direct genotype to phenotype information.

Key words: Genome sequencing, Microbial cell factories, Amino acid substitution, Metabolic engineering, Protein stability, Single-nucleotide polymorphism

1. Introduction

The development of a bio-based economy calls for a wide variety of diverse compounds—which are important in chemical, food, pharmaceutical, and health care fields—to be produced by microorganisms grown on cheap sugar feedstocks (1, 2). The extreme diversity of the microbial biosynthetic potential provides the basis to commercialize the compounds without chemical modifications and fuels the efforts to discover new natural products. Since for billions of years nature has been continually carrying out and developing its own version of combinatorial chemistry, the synthesis of bioactive molecules by microbial fermentation is an appealing proposition (3, 4). On top of that, the prohibitively low yields and much higher costs associated with de novo chemical synthesis or extraction from natural hosts (mainly plants) of natural compounds makes in many cases microbes a source of great challenges but also great opportunities for biotechnology industries (5).

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_20,

To meet the demands of industrial production, it is desirable to overcome the natural bottlenecks of metabolic pathways, which serve as control points within a native organism and regulate resource utilization and production of metabolites (6, 7). The ultimate goal is to maintain a maximized carbon flux through a desired pathway and toward target metabolites regardless of variations in the environment (either intracellular or extracellular fluctuations) (8). In these efforts for maintaining functional stability and dynamic homeostasis in a given physiological state, it is required manipulation of components from all strata of the metabolic network (9). Stable maintenance or alteration of the metabolic landscape within a cellular system can be achieved by genetic engineering, enzyme engineering, and biochemical reaction engineering, techniques that can modulate key features (dynamic controllability and modular and hierarchical organization) of biological robustness within the host cell (10, 11).

The field of biology is currently in the midst of an explosion of information fuelled by the completion of genome sequencing for thousands of species as well as many individuals within species (12). The advent of next-generation sequencing technologies is given access to population genomic data and is rapidly changing the face of the genome annotation and analysis field (13, 14). Since it is unnecessary anymore to prioritize the genomes and biological samples to be sequenced, we divert efforts to transform our understanding of the amount, distribution, and functional significance of genetic variation in natural populations (15). The sequencing of multiple individuals of the same species that allows the identification of single-nucleotide polymorphisms (SNPs) is an essential step for determining the forces shaping sequence variation. Genome-wide investigation of the patterns of polymorphism in microbes and microbial communities has proven important to understand the relationship between genotype and phenotype and an essential tool to engineering the dynamic controllability of a biosynthesis pathway (16, 17).

Computational methods have allowed the analysis of polymorphisms in particular genes for pathway modulation and efficient operation. Due to their ecological, geographical, and genetic diversity, yeast represents an ideal model system to assess the relationship between genotype and phenotype (18). In such a study, Madsen et al. described the utilization of detected metabolic SNPs for constructing yeast mutants engineered to enhance carbon flux through the mevalonate pathway and accumulate high levels of β-amyrin (18). Their analysis serves as a foundation for comparative metabolic engineering SNP analysis, where in the future reference strains may be compared to their metabolically engineered derivatives that use directed evolution, in order to answer what changes have made a strain a preferred microbial cell factory.

Here, we discuss the entire process of decoding the genome from sequence to function, we present essential tools for analyzing multiple individuals, and we describe the challenges typically encountered when we attempt to offer a precise view of the evolution of the genotype–phenotype relationship.

2. Materials

Operating system: Mac/Linux.

Free software: Java runtime environment, FastQC, Fastx-Toolkit, cmpfastq, prinseq-lite, Quake, BWA, SAMtools, bcftools, vcfutils, Picard, BEDtools, VCFtools, and GATK.

All programs are assumed to be installed and in the search path.

Online tools

SIFT: http://sift.bii.a-star.edu.sg/

ERIS: http://eris.dokhlab.org

BONGO: http://www-cryst.bioc.cam.ac.uk/~tammy/Bongo

PMP: http://www.proteinmodelportal.org

I-TASSER: http://zhanglab.ccmb.med.umich.edu/I-TASSER/

PROSA-WEB: https://prosa.services.came.sbg.ac.at/prosa.php

SUPERPOSE: http://wishart.biology.ualberta.ca/SuperPose/

Q-SITE FINDER: http://www.modelling.leeds.ac.uk/qsitefinder/

3. Methods

3.1. Genome Sequencing and Applications

In the recent years, several high-throughput genome-sequencing technologies have become available each with their own strengths and weaknesses. Here, we will briefly discuss the applicability of the individual technologies for SNP detection of small- to medium-sized genomes and provide examples of best practice and simple command-line examples for SNP detection using Illumina data.

The next-generation sequencing (NGS) technologies can roughly be divided based on the properties of the data that they produce. The most striking difference are the lengths of the DNA reads and data quantities produced from the different systems where Illumina (19) and Solid-based systems (20) produce short reads (50 to 150 nt) but in very large quantities, and the Roche 454 (21) and the Pacific Biosciences (22) produce longer reads (500 to >1 kb) but at medium or low quantities, respectively. In between those technologies the Ion Torrent PGMs are able to produce

~200–250 nt reads at low to medium throughput but are promising higher throughputs with the release of the Proton System. A second important parameter is the error types and the error rates within the produced data and whether the errors are random or systematic. For SNP calling, substitution errors are the main problem; however, the effect of these can largely be overcome by sequencing the genome to high depths. Therefore, SNP calling is mainly performed using Illumina or Solid data, with the edge being on Illumina data due to the color-space property of Solid data which makes it a little more nonintuitive to use. Typical sequencing depth for SNP calling is 20–100× coverage, equal to 80–200 or 200–1,200 Mbases for a bacterial or *S. cerevisiae* genome, respectively. Typically the sequencing depth used will depend on practical considerations such as lane throughput, multiplexing strategies, and total number of strains to be analyzed. It is also viable to perform SNP calling on low-coverage populations; however, the main application for SNP identification for metabolic engineering should be high-coverage experiments (>20×).

For SNP calling, and NGS data analysis in general, it is preferable to create the DNA libraries as paired end libraries. Here, the DNA fragments are size selected during the library preparation steps so that the approximate lengths of the fragments are known. During sequencing, both the 5′ and 3′ ends of the DNA fragment are sequenced as two separate reads, and the spatial information can be used for aligning the reads to the reference genome. Using this approach, compared to standard fragment reads, it is possible to map the reads with greater precision hereby minimizing false-positive SNP calls and to map reads within and across genomic repeats (24).

An important matter when performing SNP calling is that the SNPs will be relative to the reference genome that was used. Therefore, it can be valuable to *de novo* assemble a reference genome if it is distant to published/available reference genomes, i.e., an interesting mutation may drown in reference-based mutations.

Example is given based on Illumina sequencing data. All commands are given in Subheading 4 for performing the analysis on a Mac/Linux computer.

3.1.1. Preprocessing of Data

To assess the quality of the sequence data, it is possible, using FastQC (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/), to create a report on several quality metrics and to identify overrepresented sequences. To increase precision for alignment and SNP calling, bad quality bases should be removed from the reads using the quality information reported with the read sequences (25). Typically for Illumina reads, the relative error rate is much higher at the 3′ compared to the 5′ of the read, and quality trimming is often performed by removing bases from the 3′. Reads with low mean quality and containing unknown bases (*N*)

should also be removed prior to alignment. Overrepresented sequences, which will often be adaptor or primer sequences used in the library preparation and sequencing, should be removed from the reads as well (see Note 1).

Several programs exist to trim and remove adaptor/primer sequences and low quality bases such as Fastx-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/), prinseq-lite, (26) and more. In the notes, an example is given using Fastx-Toolkit and prinseq-lite trimming for trimming reads from 3′ to quality of 20, read mean quality of 20, removing all sequences with at least one undetermined base (*N*) and minimum length after trimming to 35 nt. When working with paired end data, it is important to maintain the pairing of the two files as many programs expect read *n* in fileA to be paired with read *n* in fileB. For this, cmpfastq (http://compbio.brc.iop.kcl.ac.uk/software/cmpfastq.php) can be used on the trimmed data to create files where the pairs are in order (see Note 2).

Because Illumina substitution sequencing errors are random, erroneous reads can be corrected using an approach where *k*-mers from all reads are compared to each other. A sequencing error will show as a *k*-mer with very low occurrences and if high coverage is available, can be compared to true *k*-mers from reads covering the same position (see Note 3). For this, we will use Quake (25).

3.1.2. Alignment

The volume of reads produced from NGS machines, which are in tens to hundreds of millions of reads, has lead to the development of numerous software for aligning short reads to reference genomes. Generally, they work by hashing either the reads or the reference genome or by using a fast and memory efficient Burrows–Wheeler Transformation (BTW) of the reference. The aligners using hashing algorithms are typically more sensitive compared to the BWT-based aligners but are slower and require more memory (Flicek and Birney, 2009; 19844229). For the purpose of SNP calling, we recommend using Burrows–Wheeler Aligner (BWA) (27) or Bowtie2 (28) for Illumina data and for Solid data to use SHRiMP2 (29). For 454 and Ion Torrent data, it is possible to use the BWA implementation of Smith–Waterman algorithm (BWA-SW). Generally, short read alignments are stored and processed in the SAM/BAM format (30). Examples are given for aligning single and paired end Illumina reads using BWA (see Note 4).

3.1.3. Alignment Processing

The raw alignment must be processed to create alignment files ready for SNP calling. First, the alignments are filtered using SAMtools (30) based on the mapping quality of each read and then sorted after chromosomal location. Then each alignment file should be merged to DNA library level for removal of potential PCR duplicates using Picard tools (http://picard.sourceforge.net/) and then merged to the final alignment for a particular sample. It may be advantageous to perform a sensitive realignment of reads that span indels in the

alignment and to recalibrate the quality scores using known variants for the particular organism (31). However, as known variants may not always be present, recalibration may not always be feasible and for smaller genomes with high read coverage, we have not seen much improvement using realignment. Genome coverage can then be determined using BEDtools (32) for the final alignment file. The extent of the coverage is naturally dependent on the distance to the reference genome used for alignment and the properties of the technology used to generate the read data (see Note 5).

3.1.4. Genotyping and SNP Calling

Calling the genotypes and identifying SNPs can be done using several programs, where we recommend using Bayesian probabilistic approaches, such as SAMtools and Bcftools (30), FreeBayes (https://github.com/ekg/freebayes), or Genome Analysis ToolKit (GATK) (33). These natively output the Variant Call Format (VCF) (34) that is the standard output format for variants and widely useable as input to other programs. A powerful feature of the Bayesian probabilistic approaches is that the uncertainties of the assigned genotypes are reported as posterior probabilities. The likelihoods are calculated incorporating information of the individual base qualities, the mapping qualities from the alignment, and the position in the read and allele frequencies. Therefore, each genotype and SNP call can be assessed in terms of whether it is likely to be a false- or true-positive call (35). For a given variant call, the posterior probability (Phred scaled) for a position to be a variant is reported together with the possible variant. Both SNPs, complex variants and indels, are reported; however, complex variants and indels are typically harder to call and should be used with care or called using designated programs (see Note 6).

3.1.5. SNP Filtering

The raw set of SNPs contains a large amount of false positives and should be filtered. One may use hard filtering that is based on setting thresholds for different parameters or soft filtering, which includes calibration of SNP calls and associated values for organisms where additional SNP data is available. Here, we describe hard filtering which generally works well for high-coverage genomes(>20X) (35).

Firstly, the genotype and SNP calls are filtered for technical biases such as strand and distances bias and whether it is close to an alignment gap. Additionally, SNPs should be filtered on minimum posterior probability that could typically be a minimum Phred score of 30 which is equal to a probability of 0.001 that the call is wrong. Second, a minimum depth threshold should be employed to remove SNP calls where to few observations are present, such as at least ten reads covering the SNP. This would imply at least ten reads on a haploid chromosome and on average five reads for each allele of a diploid chromosome. This threshold can be increased or decreased based on total coverage of the sample. Because mapping

errors around indels often lead to erroneous SNP calls, it can also be considered to remove SNPs that are within 5 nt of another variant. An efficient tool for applying these filters is VCFtools (34).

When genotyping and calling SNPs on haploid genomes, typically such as bacterial sex and mitochondrial chromosomes, all heterozygote calls on these should be removed. These arise from SNP callers that assume diploid chromosomes and errors in read mapping. When genotyping diploid chromosomes, one may also consider filtering based on allele frequency, say only accepting heterozygote SNP calls when the minor allele frequency is above 0.2 (see Note 7).

3.2. Computational Analysis of Non-Silent SNPs

Despite the fact that thousands of SNPs exist in a given group of population, only a small subset of variants can actually affect the phenotype. Non-silent SNPs (nsSNPs) that lead to amino acid changes in the proteins or enzymes are of major interest, since amino acid substitutions resulting from nsSNPs can enhance the properties of a protein such as stability or catalytic activity and are essential raw material of evolution (36). They are starting points for the adaptive evolution of new functions and often occur through pathways consisting of sequential beneficial mutations (37). Predicting the effects of the nsSNPs on the protein structure–stability–function facilitates the selection and understanding of the metabolic engineering targets. A variation within the catalytic domains of the protein would be more likely to affect the function, but the variations in the "noncritical" locations of proteins that may affect the folding and stability should also be given an equal importance. So, understanding the impact of each amino acid mutation caused by nsSNPs on protein's stability and function is valuable for selecting the metabolic engineering targets (38). Probing the effects of nsSNPs and respective amino acid mutations experimentally is a laborious process. Several computational methods that are explained below have been devised to predict the effects of nsSNPs using the amino acid sequences and 3D structures (18).

3.2.1. Sequence-Based Method for Analysis of nsSNPs

Sorting Intolerant From Tolerant (*SIFT*): The non-synonymous SNPs, the so-called non-silent SNPs, which are single-nucleotide variations in the coding regions that give "birth" to amino acid mutations, are often involved in the modulation of protein function. Identifying the SNPs that affect protein function provides information that is crucial for protein engineering researchers. The conserved residues in a protein family are expected to be functionally important, and even a conservative substitution at one of these residues may affect protein function (39). The SIFT algorithm relies solely on sequence to predict whether an amino acid substitution at a particular position in a protein will have a phenotypic effect. To predict the effect of an amino acid

substitution, SIFT considers the information about the position at which the change occurred and the type of amino acid change.

SIFT is a multistep procedure that, for a query sequence, (1) searches for similar sequences, (2) chooses closely related sequences that may share similar function, (3) obtain multiple alignment of these chosen sequences, and (4) calculates normalized probabilities for all possible substitutions at each position from the alignment. Substitutions at each position with normalized probabilities less than the chosen SIFT cutoff are predicted to be deleterious, and those that are greater than or equal to the SIFT cutoff are predicted to be tolerated. Therefore, the accuracy for predicting the phenotype that results from an amino acid substitution based on sequence alignment of protein family members has been assumed to be better than using a generalized substitution scoring matrix (40).

The SIFT tool is available online for free at http://sift.bii.a-star.edu.sg/. The underlying principle of the SIFT algorithm is that it generates alignments with a large number of homologous sequences and assigns a tolerance index score to each amino acid substitution ranging from 0 to 1 (41). The higher the tolerance index of a mutant is, the less functional impact the respective amino acid substitution is likely to have. Example 1 given below shows the results obtained from SIFT.

3.2.2. Structure-Based Method for Protein Stability Estimation

Amino acid substitutions resulting from SNPs can enhance the properties of a protein such as stability or catalytic activity and are essential raw material of evolution (36). They are starting points for the adaptive evolution of new functions and often occur through pathways consisting of sequential beneficial mutations (37). The effect of mutations on stability ($\Delta\Delta G$) of proteins has been explored by several researchers, and it has been shown that mutated proteins that are more stable than a particular threshold energy can fold properly and result in improved or changed function (18). *Eris* is a protein stability prediction server (42) that employs improved Medusa force field (43) for estimation of change in free energy difference ($\Delta\Delta G$) upon mutation. *Eris* features an all-atom force filed, a fast side-chain packing algorithm, and a backbone relaxation method for accurate protein stability predictions. The server is freely accessible at http://eris.dokhlab.org. Example 2 given below shows the results obtained from Eris.

3.2.3. Graph Theoretic Measures of Structural Effects in Proteins Caused by Individual nsSNPs

A single amino acid substitution encoded by a nsSNP may often not only give rise to rearrangement of amino acid side chains near the mutation site but also to a substantial local or global movement of polypeptide backbone. Interaction graphs of protein structures can be used to analyze the structural effects of single point mutations. Analysis of changes in residue–residue interactions caused by

nsSNPs can be probed using the *Bongo* server (*B*onds *ON G*raph). The server can be accessed at http://www-cryst.bioc.cam.ac.uk/~tammy/Bongo, or the protein structures can be submitted by e-mail to Tammy.Cheng@cancer.org.uk. A major advantage of *Bongo* is that it considers the long-distance structural impact of a point mutation. It uses graph theoretic measures to annotate nsSNPs and represent residue–residue interaction networks within proteins on graphs. *Bongo* calculates the overall impact (*I*) of a mutation according to the "key" residues affected by the mutation (44). Example 3 given below shows the results obtained from *Bongo*.

3.3. Examples

Example 1:
The tolerance index results from SIFT for four amino substitutions in Phosphomevalonate Kinase (PMK) from two different yeast strains, viz., *Saccharomyces cerevisiae* S288C and *S. cerevisiae* CEN.PK113-7D, are given here. It is important to analyze the results in both directions to identify whether the nsSNPs can be structurally tolerated by a particular variant. In step 1, the results were examined by considering the *S. cerevisiae* S288C as the "wild-type strain" and the CEN.PK113-7D as the "mutant strain" and vice versa in the second step.

Step 1

Amino acid position	R49	R75	R192	R247
PMK:S288C	G ↓	S ↓	A ↓	D ↓
PMK:CEN.PK113-7D	E	T	S	N
SIFT score	1	0.25	1	0.22
SIFT result	Tolerant	Tolerant	Tolerant	Tolerant

Step 2

Amino acid position	R49	R75	R192	R247
PMK:CEN.PK113-7D	E ↓	T ↓	S ↓	N ↓
PMK:S288C	G	S	A	D
SIFT score	0.11	0.81	0	0.72
SIFT result	Borderline	Tolerant	Intolerant	Tolerant

Example 2:
Protein stability calculations for Phosphomevalonate Kinase (PMK) from two different yeast strains, viz., *Saccharomyces cerevisiae* S288C and *S. cerevisiae* CEN.PK113-7D, are given here. It is important to analyze the results in both directions to identify whether the nsSNPs can be

structurally tolerated by a particular variant. The change in the protein stability (ΔΔG) induced by mutations calculated by the *Eris* server indicates that PMK from *S. cerevisiae* CEN.PK113-7D was probably more stable than that from S288C

			ΔΔG (kcal/mol)
PMK:S288C	G49E; S75T; A192S → D247N	PMK:CEN. PK113-7D	9.36
PMK:CEN. PK113-7D	E49G; T75S; S192A → N247D	PMK:S288C	−6.32

ΔΔG < 0: stabilizing mutations; ΔΔG > 0: destabilizing mutations
HFA1 (CT) = Carboxyl transferase domain of *HFA1* protein product

Example 3:
To understand the notation of "key" residues, let's consider the amino acid substitutions of Phosphomevalonate Kinase (PMK) from two different yeast strains, viz., *Saccharomyces cerevisiae* S288C and *S. cerevisiae* CEN.PK113-7D, that were discussed in earlier examples. Comparison of residue–residue interaction graphs (Fig. 1a–e) clearly shows that amino acid substitutions, viz., G49E, S75T, and D247N, have no change in local environment of interactions with other residues, whereas A129S amino acid substitution changes both local and global residue–residue interaction networks. Analysis of the effect of individual nsSNPs by *Bongo* shows that G49E, S75T, and D247N amino acid substitutions have an overall impact value within the threshold ($I < 1$), whereas A129S amino acid substitution shows an impact value greater than 1 ($I > 1$) and therefore may cause structural effects on the PMK. It should be noted that a protein can tolerate functionally beneficial but destabilizing substitutions only if it has previously acquired one or more stabilizing mutations (38).

4. Notes

Examples are shown for Illumina-paired end data; if only single end data is available, some steps can be skipped. Input data is assumed to be reads_1.fq and reads_2.fq, which are the two paired end files.

1. Data quality report.

```
mkdir fastqc_report
fastqc -o fastqc_report reads_1.fq reads_2.fq
firefox    fastqc_report/reads_1.fq_fastqc/fas-
tqc_report.html
firefox    fastqc_report/reads_2.fq_fastqc/fas-
tqc_report.html
```

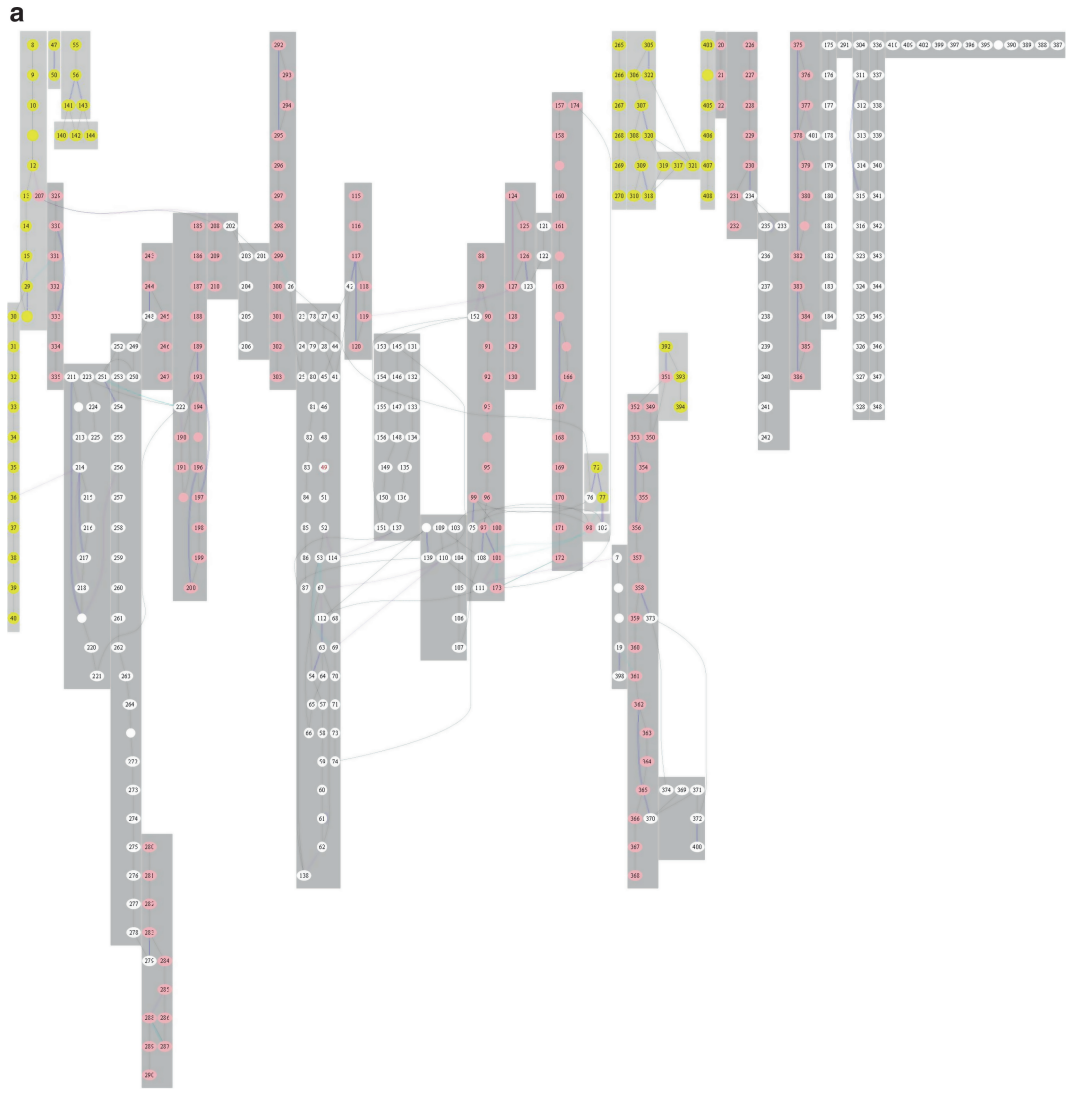

Fig. 1. Residue–residue interaction graphs generated using the *Bongo* server. Each *circle* in the graph is a vertex which represents a residue; each *black* line is an edge that connects two vertexes, in which case it represents a backbone that links two residues. Notation of secondary structures: the *pink* vertexes represent the residues in α-helices, the *yellow* vertexes represent the residues in β-strands, and the *white* vertexes represent the residues in loops. Notation of secondary structure segments: the *gray* patches indicate the segment of secondary structures (some patches seems to contain residue numbers that are not consecutive, but they actually contain multiple patches which are too close to each other and cannot be separated in the graph). Notation of residue–residue interactions: the *blue* lines represent hydrogen bonds, the *cyan* lines represent pi–pi interactions, the *purple* lines represent pi–cation interactions, the *green* lines represent hydrophobic interactions around residues that are considered to be involved in hydrophobic cores. (**a**) PMK *S. cerevisiae* S288C. (**b**) G49E mutant of PMK. (**c**) S75T mutant of PMK. (**d**) A192S mutant of PMK. (**e**) D247N mutant of PMK.

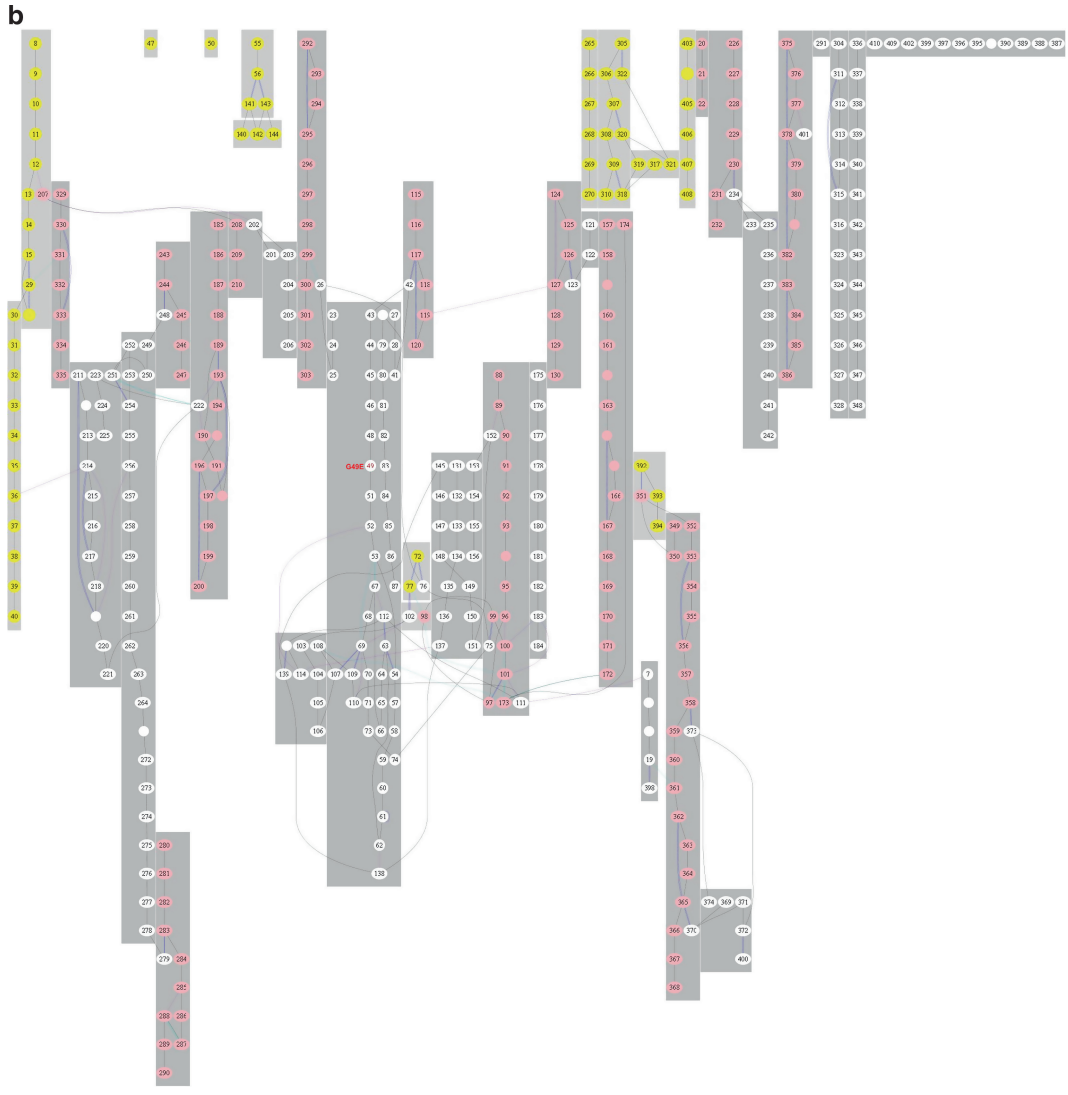

Fig. 1. (continued)

2. Trimming. Here, the "adaptor_seq" is identified from the fastqc_report. If no adaptor or primer sequences are overrepresented, the fastx_clipper step can be omitted.

```
fastx_clipper -a adaptor_seq -l 35 -i reads_1.fq -o
reads_1.clip.fq
prinseq-lite.pl -fastq reads_1.clip.fq -out_bad
null -out_good reads_1.clip.trim -trim_qual_-
right 20 -min_qual_mean 20 -min_len 35 -log
reads_1.clip.prinseq.log
fastx_clipper -a adaptor_seq -l 35 -i reads_2.fq -o
reads_2.clip.fq
```

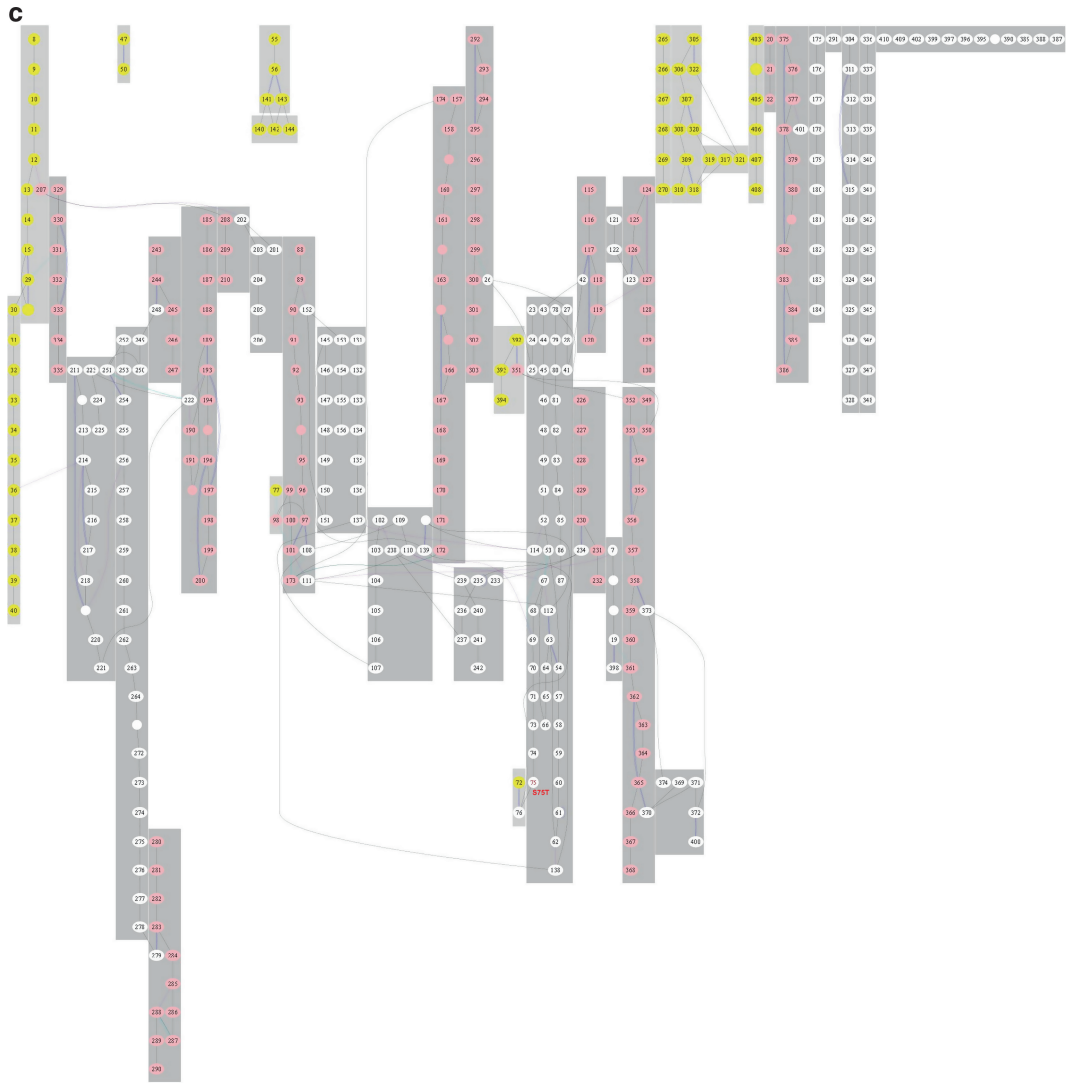

Fig. 1. (continued)

```
prinseq-lite.pl -fastq reads_2.clip.fq -out_bad
null -out_good reads_2.clip.trim -trim_qual_-
right 20 -min_qual_mean 20 -min_len 35 -log
reads_1.clip.prinseq.log
```

If paired end data, make sure each file is in the same order:

```
cmpfastq.pl reads_1.clip.trim.fastq reads_2.
clip.trim.fastq
```

Rename read files

```
mv reads_1.clip.trim.fastq-common.out reads_1.
trim.fastq
mv reads_2.clip.trim.fastq-common.out reads_2.
trim.fastq
```

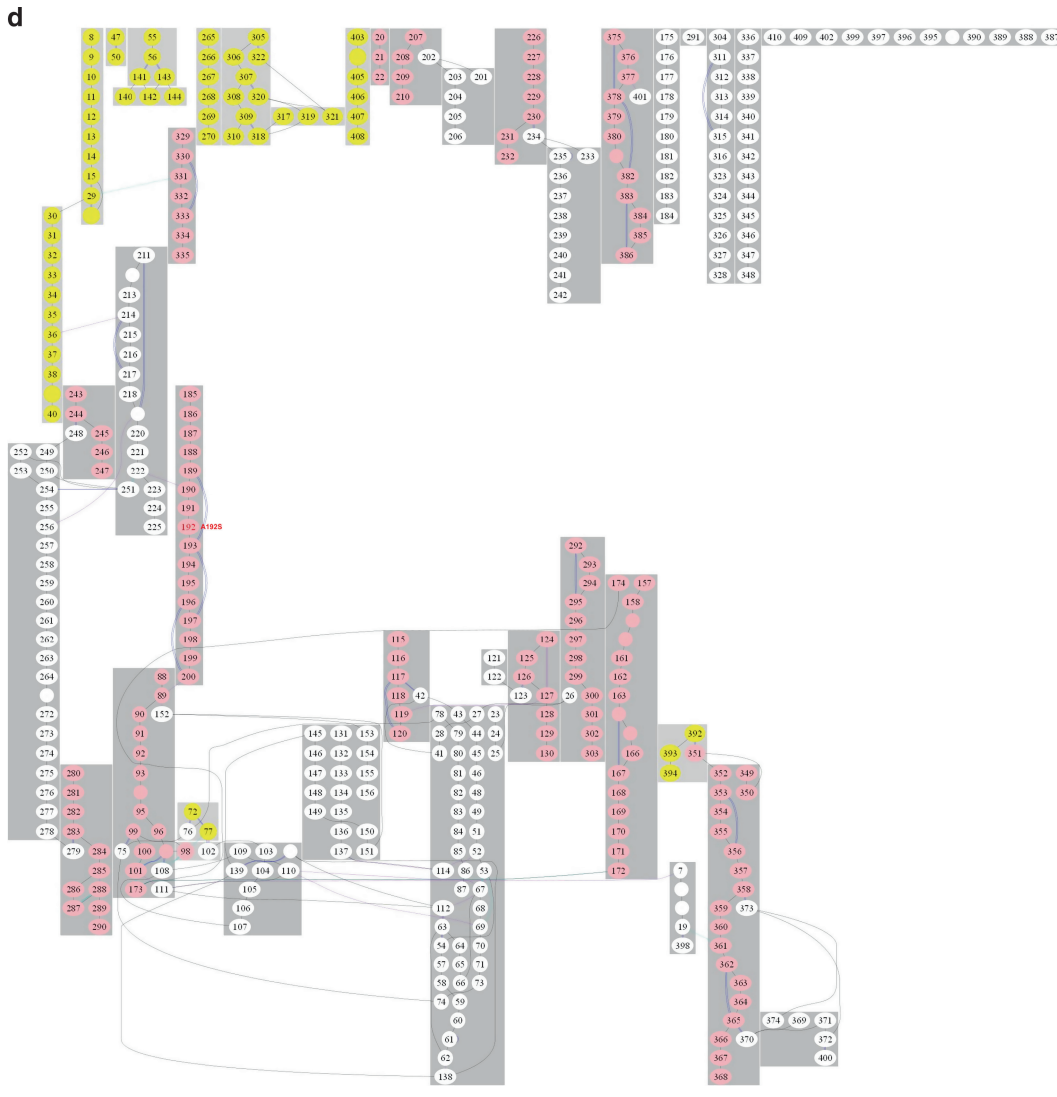

Fig. 1. (continued)

3. Error correction (reads from cmpfastq.pl).

   ```
   echo "reads_1.trim.fastq reads_2.trim.fastq" > quake.ls
   quake.py -k 15 -f quake.ls --headers
   ```

   If single end data is available, the command is

   ```
   quake.py -k 15 -r reads.fastq
   ```

4. Alignment using corrected reads from quake.
   Paired end read alignment of reads to reference genome (ref. fa). Note that we pipe (|) the output from bwa sampe to samtools to write the output as the binary format (BAM) instead of SAM. Reference genome indexing only needs to be done once.

Fig. 1. (continued)

```
bwa index ref.fa
bwa aln ref.fa reads_1.trim.cor.fastq > reads_1.
sai
bwa aln ref.fa reads_2.trim.cor.fastq > reads_2.
sai
bwa sampe ref.fa reads_1.sai reads_2.sai reads_1.
trim.cor.fastq reads_2.trim.cor.fastq | samtools
view -Sb - > aln.bam
```

Example of single end read alignment of reads (reads.fq) to reference genome (ref.fa).

```
bwa index ref.fa
bwa aln ref.fa reads.fq > reads.sai
bwa samse ref.fa reads.sai reads.fq | samtools view
-Sb - > reads.bam
```

5. Alignment processing.

   Filtering on mapping quality to minimum 30 and hereafter sorting the alignments.

   ```
   samtools view -u -q 30 aln.bam | samtools sort - aln.
   sort.q30
   ```

   If more than one lane has been run for the same DNA libraries, these should be merged into a single file, e.g., merging filtered and sorted "lane1" and "lane2" to "lib".

   ```
   samtools merge aln.lib.q30.bam aln.lane1.sort.
   q30.bam aln.lane2.sort.q30.bam
   ```

   Then remove duplicates from the alignments:

   ```
   java -jar MarkDuplicates.jar INPUT = aln.sort.
   q30.bam OUTPUT = aln.sort.q30.rmdup.bam METRICS_-
   FILE = aln.q30.rmdup.bam.log REMOVE_DUPLICATES =
   true ASSUME_SORTED = true TMP_DIR=/tmp/ VALIDA-
   TION_STRINGENCY = LENIENT
   ```

   If multiple libraries were available, they can at this point be merged to a single file:

   ```
   samtools merge aln.sample.q30.rmdup.bam aln.
   libsA.q30.rmdup.bam aln.libsB.q30.rmdup.bam...
   ```

   Realignment of reads—optional for small genomes.

   ```
   samtools index aln.sort.q30.rmdup.bam
   java -Xmx2g -jar GenomeAnalysisTK.jar -I aln.sort.
   q30.rmdup.bam -R ref.fa -T RealignerTargetCreator
   -o IndelRealigner.intervals
   java -Xmx2g -jar GenomeAnalysisTK.jar -I aln.sort.
   q30.rmdup.bam -R ref.fa -T IndelRealigner -targe-
   tIntervals IndelRealigner.intervals -o aln.sort.
   q30.rmdup.realn.bam
   ```

   It is then possible to determine how much of the genome is covered with reads. The last column in the output file is the fraction of the genome covered by a certain number of reads (column 2).

   ```
   samtools view -H aln.sort.q30.rmdup.bam | perl -ne
   'if ($_ = ~ m/^\@SQ/) { print $_ }' | perl -ne 'if
   ($_ = ~ m/SN:(.+)\s + LN:(\d+)/) { print $1, "\t",
   $2, "\n"}' > genome.txt
   genomeCoverageBed -ibam aln.q30.rmdup.bam -g
   genome.txt | grep "genome" > aln.coverage.tab
   ```

6. SNP calling.
   Genotyping and calling SNPs on processed file from alignment, using samtools and bcftools. The var.raw.bcf is the raw output of SNP calling and can be filtered for strand, distance, and alignment gap bias by vcfutils.pl.
   ```
   samtools mpileup -uf ref.fa aln.sort.q30.rmdup.bam | bcftools view -bvcg - > var.raw.bcf
   bcftools view var.raw.bcf | vcfutils.pl varFilter > var.flt.vcf
   ```
7. SNP filtering.
   The putative SNPs in var.flt.vcf should be filtered based on, e.g., minimum depth (d), minimum quality score (Q), and SNPs within a window of 5 bases (c). This will fill the "filter" column in the vcf-file, with "PASS" for position that passes all filters or with, for instance, "Qual" if they fail the quality threshold set.
   ```
   vcf-annotate --filter d=50/Q=50/c=2,5 var.flt.vcf > var.flt.1.vcf
   ```
   If working with a haploid genome, remove all heterozygote calls.
   ```
   grep -v "0/1" var.flt.1.vcf > var.flt.2.vcf
   ```
   Finally, filter out indels
   ```
   grep -v "INDEL" var.flt.2.vcf > var.flt.final.vcf
   ```
8. Structural analysis to probe the effect of nsSNPs at protein level requires 3D structures of all the protein variants. In the absence of any resolved X-ray or NMR structures, the three-dimensional atomic models can be obtained using Protein Model Portal (PMP) that provides a single interface to access more than 12.7 million comparative protein models across various protein structure databases and also provides interactive services for template selection, target template alignment, model building, and quality assessment (45). PMP is a module of the Protein Structure Initiative Knowledge Base (PSI KB) developed by the Protein Structure Bioinformatics group at the SIB—Swiss Institute of Bioinformatics and the Biozentrum—University of Basel. It is available at http://www.proteinmodelportal.org
9. Although the homology modeling approach provides reliable models, it can be applied only if the 3D structure of a similar sequence is already known. The LOMETS threading method that has been proposed (46, 47) provides a platform, in which, given an amino acid sequence and a set of structures/structural patterns, a structure will be computed into which the sequence is most likely to fold. In our study, we used ab initio *multiple threading alignment* approach (48, 49) based on I-TASSER predictor that makes an alignment computation between the

amino acids of the sequence and spatial positions of the 3D structure using scoring functions followed by LOMETS threading. The top five model structures were derived from I-TASSER simulations each having a *C*-score. *C*-score is a confidence score for estimating the quality of predicted models, and it is calculated based on the significance of threading template alignments and the convergence parameters of the structure assembly simulations; a high *C*-score signifies a model with a high confidence and vice versa. The server is freely accessible at http://zhanglab.ccmb.med.umich.edu/I-TASSER/

10. The overall model quality of structures obtained either from homology modeling or from multiple threading approach can be validated using ProSA-web Protein Structure Analysis tool (50). ProSA-web calculates the overall quality *z*-score for a specific input structure and relates to the scores computed from all experimental structures deposited in Protein Data Bank (PDB). The *z*-score is displayed on a plot so that low-resolution structures and approximate models obtained through homology modeling can be evaluated and compared against high-resolution structures. The server is freely accessible at https://prosa.services.came.sbg.ac.at/prosa.php
11. Once the 3D structures are ready, analysis of overall structural differences between the variants should be probed using *SuperPose*, a sophisticated structural superposition program that uniquely combines sequence alignment and difference distance (DD) matrix calculations to constrain its quaternion superposition algorithm (51). From a superposition of the structures of the protein variants, *SuperPose* can generate sequence alignments, structure alignments, PDB coordinates, RMSD statistics, Difference Distance Plots, and interactive images of the superimposed structures. This online tool is freely accessible at http://wishart.biology.ualberta.ca/SuperPose/
12. Finally, the binding pockets of protein variants can be scanned using Q-SiteFinder (52) to find out the protein–ligand binding site differences caused by coding nsSNPs. The special feature of Q-SiteFinder is that it uses interaction energy and a simple van der Walls probe to locate energetically favorable binding sites. By scanning binding pockets, not only the ligand binding sites of a given protein can be identified but also protein residues within a suitable range of the binding pocket are identified, which could be used for analysis of functional sites and comparison. Q-SiteFinder is freely accessible at http://www.modelling.leeds.ac.uk/qsitefinder/

Acknowledgements

The authors would like to thank the Danish Research Council for Production and Technology Sciences and the Swedish Research Council (Vetenskapsrådet Grant no: 2008-2955) for financial support.

References

1. Zhang YP, Zhu Y, Zhu Y, Li Y (2009) The importance of engineering physiological functionality into microbes. Trends Biotechnol 27:664–672
2. Aristidou A, Penttila M (2000) Metabolic engineering applications to renewable resource utilization. Curr Opin Biotechnol 11:187–198
3. Keasling JD (2010) Manufacturing molecules through metabolic engineering. Science 330:1355–1358
4. Holtz WJ, Keasling JD (2010) Engineering static and dynamic control of synthetic pathways. Cell 140:19–23
5. Medema MH, Breitling R, Bovenberg R, Takano E (2011) Exploiting plug-and-play synthetic biology for drug discovery and production in microorganisms. Nat Rev Microbiol 9:131–137
6. Kobayashi H, Kaern M, Araki M, Chung K, Gardner TS, Cantor CR, Collins JJ (2004) Programmable cells: interfacing natural and engineered gene networks. Proc Natl Acad Sci U S A 101:8414–8419
7. Alper H, Stephanopoulos G (2009) Engineering for biofuels: exploiting innate microbial capacity or importing biosynthetic potential? Nat Rev Microbiol 7:715–723
8. Tyo KE, Alper HS, Stephanopoulos GN (2007) Expanding the metabolic engineering toolbox: more options to engineer cells. Trends Biotechnol 25:132–137
9. Santos CNS, Stephanopoulos G (2008) Combinatorial engineering of microbes for optimizing cellular phenotype. Curr Opin Chem Biol 12:168–176
10. Stelling J, Sauer U, Szallasi Z, Doyle FJ, Doyle J (2004) Robustness of cellular functions. Cell 118:675–685
11. Kitano H (2007) Towards a theory of biological robustness. Mol Syst Biol 3:137
12. Nieduszynski CA, Liti G (2011) From sequence to function: insights from natural variation in budding yeasts. Biochim Biophys Acta 1810:959–966
13. Louis EJ (2011) Population genomics and speciation in yeasts. Fungal Biol Rev 25:136–142
14. Liti G, Schacherer J (2011) The rise of yeast population genomics. C R Biol 334:612–619
15. Klein-Marcuschamer D, Stephanopoulos G (2008) Assessing the potential of mutational strategies to elicit new phenotypes in industrial strains. Proc Natl Acad Sci U S A 105:2319–2324
16. Conrado RJ, Varner JD, DeLisa MP (2008) Engineering the spatial organization of metabolic enzymes: mimicking nature's synergy. Curr Opin Biotechnol 19:492–499
17. Henry CS, Overbeek R, Xia FF, Best AA, Glass E, Gilbert J, Larsen P, Edwards R, Disz T, Meyer F et al (2011) Connecting genotype to phenotype in the era of high-throughput sequencing. Bba-Gen Subjects 1810:967–977
18. Madsen KM, Udatha GDBRK, Semba S, Otero JM, Koetter P, Nielsen J, Ebizuka Y, Kushiro T, Panagiotou G (2011) Linking genotype and phenotype of Saccharomyces cerevisiae strains reveals metabolic engineering targets and leads to triterpene hyper-producers. PLoS One 6: e14763
19. Bentley DR, Balasubramanian S, Swerdlow HP, Smith GP, Milton J, Brown CG, Hall KP, Evers DJ, Barnes CL, Bignell HR et al (2008) Accurate whole human genome sequencing using reversible terminator chemistry. Nature 456:53–59
20. Shendure J, Porreca GJ, Reppas NB, Lin XX, McCutcheon JP, Rosenbaum AM, Wang MD, Zhang K, Mitra RD, Church GM (2005) Accurate multiplex polony sequencing of an evolved bacterial genome. Science 309:1728–1732
21. Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, Berka J, Braverman MS, Chen YJ, Chen ZT et al (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature 437:376–380
22. Eid J, Fehr A, Gray J, Luong K, Lyle J, Otto G, Peluso P, Rank D, Baybayan P, Bettman B et al (2009) Real-time dna sequencing from single polymerase molecules. Science 323:133–138
23. Rothberg JM, Hinz W, Rearick TM, Schultz J, Mileski W, Davey M, Leamon JH, Johnson K, Milgrew MJ, Edwards M et al (2011) An integrated semiconductor device enabling

non-optical genome sequencing. Nature 475:348–352

24. Li H, Homer N (2010) A survey of sequence alignment algorithms for next-generation sequencing. Brief Bioinform 11:473–483
25. Kelley DR, Schatz MC, Salzberg SL (2010) Quake: quality-aware detection and correction of sequencing errors. Genome Biol 11:R116
26. Schmieder R, Edwards R (2011) Quality control and preprocessing of metagenomic datasets. Bioinformatics 27:863–864
27. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25:1754–1760
28. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods 9:357–359
29. David M, Dzamba M, Lister D, Ilie L, Brudno M (2011) SHRiMP2: sensitive yet practical SHort Read Mapping. Bioinformatics 27:1011–1012
30. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R (2009) The Sequence Alignment/Map format and SAMtools. Bioinformatics 25:2078–2079
31. DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, Hartl C, Philippakis AA, del Angel G, Rivas MA, Hanna M et al (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Genet 43:491
32. Quinlan AR, Hall IM (2010) BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics 26:841–842
33. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M et al (2010) The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res 20:1297–1303
34. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST et al (2011) The variant call format and VCFtools. Bioinformatics 27:2156–2158
35. Nielsen R, Paul JS, Albrechtsen A, Song YS (2011) Genotype and SNP calling from next-generation sequencing data. Nat Rev Genet 12:443–451
36. Tokuriki N, Tawfik DS (2009) Stability effects of mutations and protein evolvability. Curr Opin Struct Biol 19:596–604
37. Bloom JD, Arnold FH (2009) In the light of directed evolution: pathways of adaptive protein evolution. Proc Natl Acad Sci U S A 106:9995–10000
38. Bloom JD, Glassman MJ (2009) Inferring stabilizing mutations from protein phylogenies: application to influenza hemagglutinin. PLoS Comput Biol 5:e1000349
39. Ng PC, Henikoff S (2001) Predicting deleterious amino acid substitutions. Genome Res 11:863–874
40. Ng PC, Henikoff S (2003) SIFT: predicting amino acid changes that affect protein function. Nucleic Acids Res 31:3812–3814
41. Xi T, Jones IM, Mohrenweiser HW (2004) Many amino acid substitution variants identified in DNA repair genes during human population screenings are predicted to impact protein function. Genomics 83:970–979
42. George Priya Doss C, Rao S (2009) Impact of single nucleotide polymorphisms in HBB gene causing haemoglobinopathies: in silico analysis. N Biotechnol 25:214–219
43. Yin S, Ding F, Dokholyan NV (2007) Modeling backbone flexibility improves protein stability estimation. Structure 15:1567–1576
44. Cheng TM, Lu YE, Vendruscolo M, Lio P, Blundell TL (2008) Prediction by graph theoretic measures of structural effects in proteins arising from non-synonymous single nucleotide polymorphisms. PLoS Comput Biol 4: e1000135
45. Arnold K, Kiefer F, Kopp J, Battey JN, Podvinec M, Westbrook JD, Berman HM, Bordoli L, Schwede T (2009) The protein model portal. J Struct Funct Genomics 10:1–8
46. Bowie JU, Luthy R, Eisenberg D (1991) A method to identify protein sequences that fold into a known three-dimensional structure. Science 253:164–170
47. Wu S, Zhang Y (2007) LOMETS: a local meta-threading-server for protein structure prediction. Nucleic Acids Res 35:3375–3382
48. Roy A, Kucukural A, Zhang Y (2010) I-TASSER: a unified platform for automated protein structure and function prediction. Nat Protoc 5:725–738
49. Zhang Y, Skolnick J (2004) SPICKER: a clustering approach to identify near-native protein folds. J Comput Chem 25:865–871
50. Wiederstein M, Sippl MJ (2007) ProSA-web: interactive web service for the recognition of errors in three-dimensional structures of proteins. Nucleic Acids Res 35:W407–W410
51. Maiti R, Van Domselaar GH, Zhang H, Wishart DS (2004) SuperPose: a simple server for sophisticated structural superposition. Nucleic Acids Res 32:W590–W594
52. Laurie AT, Jackson RM (2005) Q-SiteFinder: an energy-based method for the prediction of protein-ligand binding sites. Bioinformatics 21:1908–1916

Chapter 21

Linking RNA Measurements and Proteomics with Genome-Scale Models

Christopher M. Gowen and Stephen S. Fong

Abstract

Genome-scale metabolic models (GMMs) have been recognized as being powerful tools for capturing system-wide metabolic phenomena and connecting those phenomena to underlying genetic and regulatory changes. By formalizing and codifying the relationship between the levels of gene expression, protein concentration, and reaction flux, metabolic models are able to translate changes in gene expression to their effects on the metabolic network. A number of methods are then available to interpret how those changes are manifest in the metabolic flux distribution. In addition to discussing how gene expression datasets can be interpreted in the context of a metabolic model, this chapter discusses two of the most common methods for analyzing the resulting metabolic network.

The chapter begins by demonstrating how a typical microarray dataset can be processed for incorporation into a GMM of the yeast *Saccharomyces cerevisiae*. Once the expression states of the reactions in the model are available, the method of directly trimming the metabolic model by removing or constraining reactions with low expression states is demonstrated. This is the simplest and most direct approach to interpret gene expression states, but it is prone to overvaluing the effects of down regulation and it can propagate false negative errors. We therefore also include a more advanced method that uses a mixed-integer linear programming optimization to find a flux distribution that maximizes agreement with global gene expression states. Sample MATLAB code for use with the COBRA toolbox is provided for all methods used.

Key words: Genome-scale metabolic models, Transcriptomics, Proteomics, COBRA toolbox, Systems biology, Constraint-based modeling, Flux balance analysis

1. Introduction

Incredible technological advances in sequencing and quantification of nucleic acids and proteins have resulted in huge amounts of data, but physiological interpretation of these datasets—particularly as they relate globally to metabolism—remains a challenge. Genome-scale metabolic models (GMMs) provide a useful framework

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_21,

for the integration of "omics" scale datasets by forging connections between gene expression changes and their systems-level effects on metabolism (1). A number of computational methods have been developed in the past decade to integrate transcriptomic, proteomic, and metabolomic datasets with GMMs to facilitate interpretation and discovery. This chapter discusses two of the most straightforward methods in detail, as well as methods for processing the datasets for use in these methods. The most direct method for incorporation of this data is the application of Boolean constraints to the reaction list where relevant enzyme mRNA or protein is not detected (2). This method trims the solution space, thereby improving predictions using methods like FBA and FVA. In 2008, Shlomi et al. developed a mixed-integer linear programming (MILP)-based method to solve a flux distribution that maximizes agreement with global gene expression states (3). This method was developed for the prediction of metabolic fluxes in human tissues using microarray data, but it is generally usable for any system with expression data at either the mRNA or protein level (4). This method also has the benefit of being independent of a growth rate objective function, so it is particularly appropriate during nonoptimal growth conditions or in multicellular organisms.

More recently, Yizhak et al. have extended this approach using quadratic programming (QP)-based optimization to directly incorporate quantitative proteomic and metabolomics data with a GMM (5). Such methods are beyond the scope of this chapter, but it illustrates the versatility of genome-scale metabolic reconstructions for synthesizing and interpreting diverse and large-scale datasets.

In this chapter, we address in detail both the direct Boolean solution-space trimming approach (referred to here as the Akesson method), as well as the MILP-based data integration method (referred to here as the Shlomi method). Additionally, we begin with detailing the steps necessary to appropriately process expression datasets for these two methods. It should be noted that these approaches are flexible, and an understanding of the governing concepts and limitations will allow the reader to adapt them to suit their purposes and to avoid misinterpretation of the results. There is no single software tool or set of tools that is "best" for every researcher. For the examples here, we have tried to choose software tools that are well-documented, flexible, readily available, cross-platform (Linux/Unix, Windows, Mac OS X) and generally accessible to the nonexpert. Additional notes discuss alternative choices and considerations in further detail.

2. Materials

2.1. "Model-Management" and Analysis Software

A number of software tools are available in different languages for managing and analyzing metabolic models. In this protocol, we use the popular Constraint-Based Reconstruction and Analysis (COBRA) toolbox for MATLAB (see Note 1 for installation and setup tips).

- MATLAB 2012a.
- SBML Toolbox (http://sbml.org/Software/SBMLToolbox).
- COBRA toolbox (http://opencobra.sourceforge.net).

2.2. Optimization Software

Several commercial and open-source packages exist for solving linear programming (LP) problems (see Note 2 for popular examples). For the methods used here, the reader will need to have a solver installed for solving both simple LP and mixed-integer linear programming (MILP) problems. In our analysis, we used IBM ILOG CPLEX 12.1 for MATLAB.

2.3. Electronic Resources

Datasets, metabolic models, and Matlab code used in this protocol can be found online by visiting extras.springer.com and searching for this book's ISBN number.

- iMM904.mat.
- log2_expression.mat.
- process_mRNA_microarray.m.
- Akesson_method.m.
- Shlomi_method.m.
- shlomiAltOpt.m.
- mapGeneCalls.m.

3. Methods

3.1. Processing "Omics" Datasets for Constraints-Based Modeling

Both of the methods described later in this chapter are based on a discrete approximation of gene expression data, classifying each gene as having "high," "low," or "undetermined" gene expression levels. In addition to greatly reducing the complexity of the problem, this simplification allows the methods to more easily be generalized to many types of both transcriptomic and proteomic experiments, including both RNA microarrays and RNA deep sequencing (RNAseq). Due to the diversity of the methods and conditions available for these experiments, complete coverage of the topic is not possible here, but some common principles apply.

RNA microarray experiments and mass spectrometry-based proteomics experiments are inherently comparative in nature, in contrast to RNAseq experiments, which directly count mRNA molecules and can provide an absolute measure of concentration (see Note 3). As a result, decision strategies for calling genes' expression as "high" or "low" will differ depending on how the experiments were structured. A general workflow is outlined below.

1. Rudimentary bioinformatics analysis of the raw data is performed to filter data points for signal quality and statistical significance. The output should be relative expression intensities for every gene measured, as determined by signal intensity (RNA microarrays), sequence read density (RNAseq), or relative peak intensity (MS-based proteomics).
2. All genes that are not detected—either because the signal intensity falls below background noise, no reads are detected, or no peaks are detected—can conservatively be labeled as having "low" expression.
3. In the case where comparative datasets exist for a wide range of conditions, it may be possible to establish baseline signal levels for each gene with relatively high confidence. In those cases, it is a matter of statistical analysis to determine genes which have significantly high expression in a given growth condition. In many cases, however, sufficient data points are not available, so the definition of threshold values for high and low expression value is less straightforward. See the example below and Note 4 for more discussion.

3.1.1. Preparing Expression Data

For the demonstration in this protocol, analyses are based on the open-access microarray results from (6). In that study, the yeast *Saccharomyces cerevisiae* was grown aerobically in a glucose-limited chemostat culture until steady state was reached, at which point an addition of glucose created a pulse of glucose to obtain a concentration of 20 g/L. Affymetrix yeast2 microarray chips were used to monitor the time course of gene expression until the culture returned to steady state. The processed microarray data are available in the supplementary documentation for that article, and the $\log_2$-transformed expression levels are provided for the reader's convenience in "log2_expression.mat," available online. The model used here is the *S. cerevisiae* iMM904 model (7), provided in Matlab format as "iMM904.mat."

3.1.2. Selecting Appropriate Expression Thresholds

To begin, clear the Matlab workspace and load the expression data into memory:

```
>>clear
>>load log2_expression.mat
```

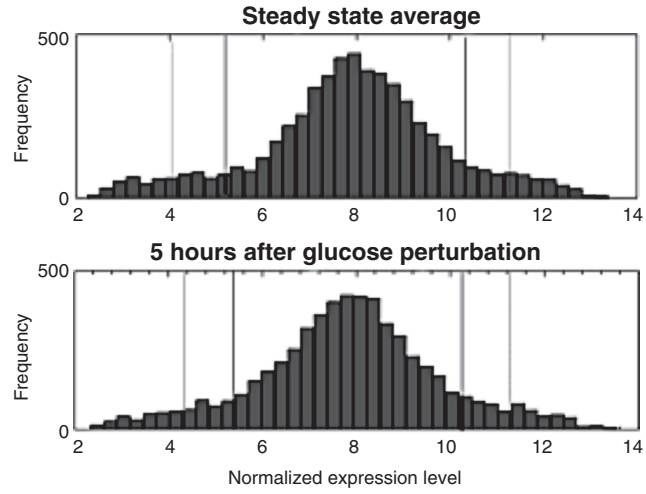

Fig. 1. Gene expression density of all transcripts at steady state and 5 h after glucose addition. The cutoff lines for the *top* and *bottom* 5th (*dotted lines*) and 10th (*solid lines*) percentiles are also shown.

The expression data will now be in memory as a double matrix with the name "log2." The dataset IDs and transcript IDs are cell arrays named "sample_IDs" and "transcript_IDs," respectively. In this protocol, we compare flux distributions at steady state (column 3) and 5 h after glucose pulse (column 14). For the sake of simplicity here, we select high and low expression cutoffs based on the overall distribution of expression levels at each sample point, rather than comparing across many datasets to determine baseline expression levels. To assist in this decision, it is useful to plot the gene expression density as a histogram:

```
>>figure()
>>subplot(2,1,1)
>>hist(log2(:,3),40)
>>title(sample_IDs(3));
>>ylabel(frequency); xlabel(log2 expression level);
>>subplot(2,1,2)
>>hist(log2(:,3),40)
>>title(sample_IDs(3));
>>ylabel(frequency); xlabel(log2 expression level);
```

This will produce the histograms in Fig. 1.

```
>>percentiles = prctile(log2,[10 90]);
```

After examining different percentile cutoffs, we can see that the top and bottom 10th percentiles roughly coincide with the tails of

the distribution, so we can say with modest confidence that these transcripts have "high" and "low" expression levels. Next we create containers for the lists of high and low transcripts,

```
>>high_transcr = cell(1,length(sample_IDs));
>>low_transcr = cell(1,length(sample_IDs));
```

and we gather the transcripts that fall into the top and bottom 10th percentiles:

```
>>for i=1:length(sample_IDs)
  high_transcr{i} = find(log2(:,i)>percentiles(2,i));
  low_transcr{i} = find(log2(:,i)<percentiles(1,i));
end
```

This will collect the locations of high and low transcripts, which we use to find the corresponding genes and reactions in the next sections.

3.2. Simple On/Off Solution Space Trimming (Akesson Method)

3.2.1. Loading the Genome-Scale Metabolic Model and Setting Boundary Conditions

Initiate the COBRA toolbox using the initCobraToolbox command according to the LP/MILP solver(s) you have installed. We then begin by loading the iMM904 model into memory:

```
>>load iMM904.mat
```

The model will now exist as a structure named "model" which can be used and modified by the methods in the COBRA toolbox. Next, set the FBA objective function as maximizing biomass production and create separate models for metabolism at steady state (model_ss_wt) and at 5 h (model_5h_wt):

```
>>model =
   changeObjective(model,{'biomass_SC5_notrace'},1);
>>model_ss_wt = changeRxnBounds( model, ...
  {'EX_glc(e)' 'EX_o2(e)'}, [-0.9376 -1000], 'l');
>>model_5h_wt = changeRxnBounds( model, ...
  {'EX_glc(e)' 'EX_o2(e)'}, [-3.2752 -1000], 'l');
```

Note that in the two commands above, the exchange boundary constraints were changed to the measured uptake rates of glucose and unconstrained oxygen uptake rate. In this study, oxygen was saturating through high aeration and stirring, and glucose uptake was calculated using the equation

$$\frac{\mathrm{d}S}{\mathrm{d}t} = D \times (s_i - s_o) - \frac{r_{\mathrm{up}}}{MW \times x}$$

where S is the substrate concentration, D is the dilution rate of 0.1 1/h, S_i is the glucose concentration entering the chemostat (2.5 g/L), S_o is the concentration leaving the chemostat, r_{up} is the substrate uptake rate in mmol/g-DW/h, MW is the molecular

weight of glucose (0.18016 g/mmol), and x is the biomass concentration. Solving for r_{up} based on Fig. 3 from ref. 6 yields glucose uptake rates of 0.9376 and 3.2752 mmol/g-DW/h during steady state and 5 h after perturbation, respectively.

3.2.2. Map Lowly Expressed Genes to Their Corresponding Reactions

Next, we match the low expression transcripts found in the previous section to genes that are accounted for in the model:

```
>>low_transcr_names_ss = transcr_IDs(low_transcr{3});
>>low_genes_ss = []; low_genes_5h = [];
>>for i=1:length(low_transcr_names_ss)
   gene = low_transcr_names_ss(i);
   low_genes_ss = [low_genes_ss strmatch(gene,
    model.genes, 'exact')];
  end
>>low_transcr_names_5h = transcr_IDs(low_transcr{14});
>>for i=1:length(low_transcr_names_5h)
   gene = low_transcr_names_5h(i);
   low_genes_5h = [low_genes_5h strmatch(gene,
    model.genes, 'exact')];
  end
```

3.2.3. Place Flux Constraints on Lowly Expressed Reactions

Next, block the flux on reactions that are coded by low-expression genes. The last option in this function is the fraction of the maximum flux (±1,000 in this model) to which the constrained fluxes should be limited. In this case, we are constraining the fluxes to below .01 mmol/g-DW/h because completely blocking all low reactions results in an infeasible model.

```
>>[model_ss_reg,hasEffect_ss,lowRxns_ss,deletedGenes_s
    s] = ...
deleteModelGenes(model_ss_wt,model.genes(low_genes_ss)
    ,0.00001);
>>[model_5h_reg,hasEffect_5h,lowRxns_5h,deletedGenes_5
    h] = ...
deleteModelGenes(model_5h_wt,model.genes(low_genes_5h)
    ,0.00001);
```

3.2.4. Flux Balance Analysis of Constrained Models

At this point, trimmed models have been created for both time points based on gene expression data (model_ss_reg for the steady state condition and model_5h_reg for the 5 h timepoint). These models can now be analyzed using a wide range of constraint-based modeling methods, many of which are available in the COBRA toolbox, including flux balance analysis (FBA), flux variability

analysis (FVA), flux sampling, and so on. As a simple demonstration of one analysis, we can now perform simple FBA maximizing growth rate:

```
>>fba_ss_wt = optimizeCbModel(model_ss_wt);
>>fba_ss_reg = optimizeCbModel(model_ss_reg);
>>fba_5h_wt = optimizeCbModel(model_5h_wt);
>>fba_5h_reg = optimizeCbModel(model_5h_reg);
```

The structures created by each of these contain fields describing the solution of the FBA problem:

```
fba_ss_wt =
           x: [1577x1 double]
           f: 0.0862
           y: [1228x1 double]
           w: [1577x1 double]
        stat: 1
    origStat: 1
      solver: 'cplex'
        time: 0.2633
```

In this structure, "f" is the optimal objective value found (in this case, growth rate), "x" is the primal vector solution for flux in all reactions, and "y" is the dual solution vector describing the shadow cost for every metabolite. The flux vector can be printed to the screen or to a file using the command printFluxVector in the COBRA toolbox. Here we use it to print the flux vectors for all of the above FBA simulations to a tab-delimited text file:

```
>>printFluxVector(model, ...
   [fba_ss_wt.x fba_ss_reg.x fba_5h_wt.x
    fba_5h_reg.x], ...

   0,0,-1,'yeast_glc_fluxes.txt',[],1);
```

The above method is a very straightforward way to incorporate expression data into a metabolic model framework. This has the benefit that the resulting trimmed model can be used in a very wide range of modeling methods, but its direct nature makes it more susceptible to errors when gene expression does not correlate well with metabolic flux. The next section uses a more advanced optimization technique that seeks a "best fit" between expression data and metabolic flux.

3.3. MILP-Based Integration of Expression Data (Shlomi Method)

In 2008, Shlomi et al. developed a mixed-integer linear programming (MILP)-based method to solve a flux distribution that maximizes agreement with global gene expression states (3) by formulating the following MILP problem:

$$\max_{v,y^+,y^-} \left(\sum_{i \in R_H} \left(y_i^+ + y_i^- \right) + \sum_{i \in R_L} \left(y_i^+ \right) \right)$$

s. t.

$S \cdot v = 0$	(1)
$v_{\min} \le v \le v_{\max}$	(2)
$v_i + y_i^+ \left(v_{\min,i} - \varepsilon \right) \ge v_{\min,i}, \quad i \in R_H$	(3)
$v_i + y_i^- \left(v_{\max,i} + \varepsilon \right) \le v_{\max,i}, \quad i \in R_H$	(4)
$v_{\min,i} \left(1 - y_i^+ \right) \le v_i \le v_{\max,i} \left(1 - y_i^+ \right), \quad i \in R_L$	(5)
$v \in R^m$	(6)
$y_i^+, y_i^- \in [0,1]$	(7)

In this problem, binary variables y_i^+ are defined for all low reactions such that $y_i^+ = 1$ when the low reaction, i, has zero flux (constraint equation 5, above), and two binary variables y_i^+ and y_i^- are defined for all high reactions such that $y_i^+ = 1$ when flux is above a threshold ε in the forward direction (constraint equation 3) and $y_i^- = 1$ when flux is above a threshold ε in the reverse direction (constraint equation 4). The sum of all y values then represents a score for the binary agreement between the flux distribution and the expression values.

3.3.1. Define Parameters and Constrain Model Growth Rate to Match Observation

This method is currently not directly available in the COBRA toolbox, so it will be necessary to manually define the MILP problems we want to solve. After loading the iMM904 model as before, we define structures that will hold the components of the MILP problems according to the documentation for the cplexmilp command:

```
>>milp_ss = struct;
```

We also define a value for epsilon above which a reaction flux will be considered "high":

```
>>epsilon=0.05;
```

Because the Shlomi method is independent of a biomass objective function, we can constrain the biomass reaction flux to match the observed behavior, thereby greatly improving the predictive ability of the model:

```
>>model_ss = μ
   [-0.9376*1.1 -1000],'l');
>>model_ss = changeRxnBounds(model_ss,...
    {'biomass_SC5_notrace'},0.1*1.1,'u');
>>model_ss = changeRxnBounds(model_ss,...
    {'biomass_SC5_notrace'},0.1*0.9,'l');
```

In a chemostat at steady state, the growth rate is equivalent to the dilution rate (0.1 1/h), provided growth rate is much greater than the death rate. After the perturbation, the growth rate can be calculated using the following equation (see Note 5):

$$\frac{\mathrm{d}X}{\mathrm{d}t} = DX_0 + (\mu - k_d - D)X$$

where X is the biomass concentration in the chemostat in g-DW/L, D is the dilution rate, μ is the biomass growth rate, and k_d is the biomass death rate, each in 1/h.

3.3.2. Define Shlomi Optimization as an MILP Problem Structure

The MATLAB command cplexmilp solves an MILP problem of the standardized form

```
min      f*x
st.      Aineq*x <= bineq
         Aeq*x    = beq
         lb <= x <= ub
         x belongs to BICSN
```

so we must structure the above optimization problem accordingly. To begin, it is useful to define some counts that will clean up later code:

```
>>nMets = length(model.mets);
>>nLow_ss = length(lowRxns_ss);
>>nHigh_ss = length(highRxns_ss);
>>nRxns = length(model.rxns);
>>nVar_ss = nRxns + length(lowRxns_ss)+2*length
   (highRxns_ss);
>>ny_ss = nVar_ss - nRxns;
```

The stoichiometric constraints make up the entirety of the equality constraints, and these are already available in the COBRA model structure. It is only necessary to extend the columns of the matrix Aeq to include the additional binary variables:

```
>>milp_ss.Aeq = [model_ss.S zeros(nMets,ny_ss)];
>>milp_ss.beq = model_ss.b;
```

3.3.3. Add Inequality Constraints for Binary Agreement Variables y_i^+, y_i^-

The inequality matrix will be made up of the constraints provided by constraint equations 3, 4, and 5, noting that equation 5 must be split into two inequalities per low reaction:

```
>>milp_ss.Aineq = zeros(2*nLow_ss+2*nHigh_ss,nVar_ss);
>>milp_ss.bineq = zeros(2*nLow_ss+2*nHigh_ss,1);
```

First looping over all low reactions, we add the constraints in equation 5:

```
>>for l=1:nLow_ss
    rxnID_ = lowRxns_ss(l);
    v_min_ = model_ss.lb(l);
    v_max_ = model_ss.ub(l);
```

beginning with the right hand side inequality:

```
% v_l <= v_max_l*(1-yf_l), l in low reactions
% v_l + v_max_l*yf_l <= v_max_l
i = l;
milp_ss.Aineq(i,rxnID_) = 1;                    %v_l
milp_ss.Aineq(i,nRxns+l) = v_max_;
 %v_max_l*yf_l
milp_ss.bineq(i) = v_max_;
 %v_max_l
```

and continuing with the left hand side inequality:

```
% v_min_l*(1-yf_l) <= v_l, l in low reactions
% -v_l - v_min_l*yf_l <= -v_min_l
i = length(lowRxns_ss) + l;
milp_ss.Aineq(i,rxnID_) = -1;                   %-v_l
milp_ss.Aineq(i,nRxns+l) = -v_min_;             %-
 v_min_l*yf_l
milp_ss.bineq(i) = -v_min_;                     %-
 v_min_
end
```

Then, looping over the high reactions in the same way,

```
>>for h=1:nHigh_ss
    rxnID_ = highRxns_ss(h);
    v_min_ = model_ss.lb(rxnID_);
    v_max_ = model_ss.ub(rxnID_);
```

The constraints associated with equation 3 are added for each of the high reactions

```
% v_h + yf_h*(v_min_h - eps) >= v_min_h, h in high
 reactions
i = 2*nLow_ss + h;
milp_ss.Aineq(i,rxnID_) = -1;
milp_ss.Aineq(i,nRxns+nLow_ss+h) = -v_min_ +
 epsilon;
milp_ss.bineq(i) = -v_min_;
```

Likewise, constraint equation 4 is added for each high reaction:

```
% v_h + yr_h*(v_max_h + eps) <= v_max_h, h in high
 reactions
i = 2*nLow_ss + nHigh_ss + h;
milp_ss.Aineq(i,rxnID_) = 1;
milp_ss.Aineq(i,nRxns + nLow_ss + nHigh_ss + h) =
 ...
    v_max_ + epsilon;
milp_ss.bineq(i) = v_max_;
end
```

At this phase, it is extremely important to carefully plan and track the row and column indices for each constraint and variable during this phase so that mistakes aren't introduced into the problem.

3.3.4. Define the Objective Function and Boundary Vectors

The remaining components of the MILP problem are also introduced to the structure, including the objective f:

```
>>milp_ss.f = -1*[ zeros(nRxns,1); ones(ny_ss,1) ];
```

the lower and upper bounds:

```
>>milp_ss.lb = [model_ss.lb; zeros(ny_ss,1)];
>>milp_ss.ub = [model_ss.ub; ones(ny_ss,1)];
```

and the definition of the variable type, given as a vector string of length nVar where "C" designates a continuous variable, and "B" designates a binary variable:

```
>>ys_ss = shlomi_ss.x(nRxns+1:end);
>>shlomi_ss.actualY = zeros(nRxns,1);
>>for i=1:length(lowRxns_ss)
   shlomi_ss.actualY(lowRxns_ss(i)) = -ys_ss(i);
end
>>for i=1:length(highRxns_ss)
   shlomi_ss.actualY(highRxns_ss(i)) = ...
      ys_ss(nLow_ss+i) + ys_ss(nLow_ss+nHigh_ss+i);
end
>>shlomi_ss.predictedY = zeros(nRxns,1);
>>shlomi_ss.predictedY(highRxns_ss) = 1;
>>shlomi_ss.predictedY(lowRxns_ss) = -1;
```

3.3.5. Solve the MILP Problem Using CPLEX and Store Results

Finally, we can create structures to hold the solutions results and solve the problem:

```
>>shlomi_ss = struct;
>>[x,fval,exitflag,output] = cplexmilp(milp_ss);
>>shlomi_ss.x = x;
>>shlomi_ss.fval = fval;
>>shlomi_ss.exitflag = exitflag;
>>shlomi_ss.output = output;
```

In order to help interpret the results, it is useful to create vectors describing the expression state and the flux state of each reaction based on the binary section of the solution vector:

```
>>ys_ss = shlomi_ss.x(nRxns+1:end);
>>shlomi_ss.actualY = zeros(nRxns,1);
>>for i=1:length(lowRxns_ss)
   shlomi_ss.actualY(lowRxns_ss(i)) = -ys_ss(i);
end
>>for i=1:length(highRxns_ss)
   shlomi_ss.actualY(highRxns_ss(i))= ...
      ys_ss(nLow_ss+i) + ys_ss(nLow_ss+nHigh_ss+i);
end
>>shlomi_ss.predictedY = zeros(nRxns,1);
>>shlomi_ss.predictedY(highRxns_ss)=1;
>>shlomi_ss.predictedY(lowRxns_ss)=-1;
```

Now, for each reaction, the solution structure contains fields describing the expression state of the reaction (predictedY), the flux state of the reaction (actualY), and the flux through the reaction (x). The flux vector can now be explored and viewed using the same tools available in the COBRA toolbox, e.g.,

```
>> printFluxVector(model, shlomi_ss.x(1:1577),1,1)
EX_co2(e) 3.35728
EX_dttp(e) -1.00023e-09
EX_epist(e) 0.010135
EX_glc(e) -1.03136
EX_gua(e) 0.0361133
EX_h2o(e) 3.79926
EX_h(e) 0.613817
```

```
EX_hxan(e) 0.0762171
EX_met_L(e) 0.0432707
EX_nh4(e) -0.456026
EX_o2(e) -3.15815
EX_pi(e) 0.448207
EX_so4(e) -0.0502277
EX_thym(e) 0.0244519
EX_urea(e) 0.035537
EX_xan(e) 0.113272
biomass_SC5_notrace 0.09
```

Notice that the flux distribution here tends to be far more distributed when compared to that found using FBA, where a smaller set of optimal pathways is utilized more heavily.

Finally, it is important to point out that, although the solution found using cplexmilp is guaranteed to be optimal, it is not necessarily (and not likely) unique, as many different network configurations could reach the same agreement between expression state and flux state. To address this problem, Shlomi et al. also demonstrated an extension of the flux variability analysis (8) idea, adapted towards this MILP problem. As implemented here, this analysis determines for each highly or lowly expressed reaction whether that reaction is "always high," "always low," or variable, by comparing the maximum achievable agreement (fval) when forcing the binary reaction state variables to be high or low. For example, if the optimal agreement when a high reaction is forced to have a high flux equals x, and the optimal agreement when that reaction is forced to have a non-high flux equals y, then $x>y$ implies that the reaction is always high and $x<y$ implies that the reaction is never high, with a confidence $|x-y|$. In the case that $x=y$, no determination about the state of that reaction can be made. A similar analysis is done for all low reactions as well to determine if each is "always low," "never low," or undetermined. An implementation of this analysis is included in the supplementary files online.

The two methods presented in this chapter are relatively simple, robust ways to incorporate expression datasets into genome-scale models. Not surprisingly, the success of these approaches depends on the quality and depth of the datasets themselves, and robust statistical methods are invaluable for accurately calling reactions "high" or "low" for a given condition or tissue. Furthermore, the quality of predictions is likely to increase as degrees of separation are removed between the expression data and the corresponding metabolic reactions, for example by using proteomics datasets to quantify enzyme levels.

4. Notes

1. *COBRA toolbox setup and use*
   For this method, the reader will be expected to have the COBRA toolbox installed and setup with appropriate LP solvers. The toolbox itself is merely a collection of MATLAB m-files, so the only installation needed is to ensure that these files can be found in the MATLAB path, either by adding their directory locations manually or through the use of a startup script. Many of the tools do, however, rely on external libraries. Detailed instructions for installation can be found online through the OpenCOBRA Web site (opencobra.sourceforge.net).

   Once installed, detailed documentation can be found by opening the file cobra/docs/index.html with your browser. The toolbox is quite extensive, so it is a good idea to browse the available functions; there may already be a function for what you want to do! If you still run into problems, the COBRA Google Group (http://groups.google.com/group/cobra-toolbox) discussion boards are quite active and welcoming to new users.
2. *Linear programming solvers*
   COBRA toolbox has built-in support for the following solvers:
   - LP:
     - Lindo
     - GLPK
     - LP_solve
     - Tomlab CPLEX
     - Mosek
     - Gurobi
   - MILP:
     - Tomlab CPLEX
     - GLPK
     - Gurobi

   In addition to the above, CPLEX can be accessed directly without the use of Tomlab, but this is not currently supported by the COBRA toolbox. Among these, GLPK (Gnu Linear Programming Kit) is a popular starting point because it is free and open source; however, GLPK does suffer from poor performance in comparison to commercial solvers, and it can be unpredictable for MILP problems, being more likely to require manual tweaking of solver parameters. Tomlab/CPLEX and Gurobi both offer significantly improved speeds, but they can

be quite costly, particularly for nonacademic users. Currently, both offer some free licensing options for academic users.

3. *RNAseq and "absolute" quantification*

   RNAseq is generally believed to provide a far more accurate measure of absolute mRNA concentrations than microarray experiments, but the technique is not absent of pitfalls and biases that must be accounted for. Of specific concern are biases in transcript read density across transcript length and preference for or exclusion of specific sequences during both the library prep and sequencing itself. The impact of these biases will vary depending on the class of organism being studies, the sample preparation, and the sequencing technology used, and the researcher will have to address these concerns or factor them into the processing step discussed in section 3.1. Still, RNAseq has been found to better correlate with proteomics measurements than do microarrays (9).

4. *Effect of parameter selection on Shlomi method predictions*

   In a previous study (4), we examined the impact of the parameters gamma (gene expression "high" threshold), epsilon (reaction flux "high" threshold) on how well the Shlomi method can predict metabolic flux. We found that, in our experience, gamma did not significantly impact predictive quality over the range tested (top 25th percentile to top 80th percentile), while epsilon needed to be set above 0.01 mmol/g-DW/h, or roughly 3–5 % of the biomass flux. This finding speaks to the inherent flexibility of the Shlomi algorithm with respect to outliers in gene expression, since the algorithm responds better to global changes in gene expression.

5. *Selection of boundary conditions (exchange fluxes)*

   Careful selection of boundary conditions is one of the most important considerations in all constraint based models because it plays such a large role in dictating the solution space of the model. In the Shlomi method, adding an additional constraint on growth rate is appropriate here because we can be sure that the cells are not growing optimally since they are in carbon limited chemostat culture and because doing so further refines the precision of the solution space. The modeler should beware, however, that confining boundary conditions too strictly could result in infeasibility. In this case, for example, strictly confining both the growth rate and the carbon uptake rate to the measured values results in an infeasible set of constraints. This could be explained by a combination of factors, including errors that are inevitably introduced during measurements and calculation of uptake rates and the inflexibility of certain parameters in the model, including the ATP maintenance cost. To address this problem, one approach is to allow an error when setting these constraints.

References

1. Oberhardt MA, Palsson BØ, Papin JA (2009) Applications of genome-scale metabolic reconstructions. Mol Syst Biol 5:320
2. Akesson M, Förster J, Nielsen J (2004) Integration of gene expression data into genome-scale metabolic models. Metab Eng 6(4):285–293
3. Shlomi T, Cabili MN, Herrgard MJ, Palsson BØ, Ruppin E (2008) Network-based prediction of human tissue-specific metabolism. Nat Biotechnol 26(9):1003–1010
4. Gowen CM, Fong SS (2010) Genome-scale metabolic model integrated with RNAseq data to identify metabolic states of *Clostridium thermocellum*. Biotechnol J 5(7):759–767
5. Yizhak K, Benyamini T, Liebermeister W, Ruppin E, Shlomi T (2010) Integrating quantitative proteomics and metabolomics with a genome-scale metabolic network model. Bioinformatics 26(12):i255–i260
6. Dikicioglu D, Karabekmez E, Rash B, Pir P, Kirdar B, Oliver SG (2011) How yeast reprogrammes its transcriptional profile in response to different nutrient impulses. BMC Syst Biol 5(1):148
7. Mo ML, Palsson BO, Herrgård MJ (2009) Connecting extracellular metabolomic measurements to intracellular flux states in yeast. BMC Syst Biol 3:37
8. Mahadevan R (2003) The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. Metab Eng 5(4):264–276
9. Fu X et al (2009) Estimating accuracy of RNA-Seq and microarrays with proteomics. BMC Genomics 10:161

Chapter 22

Comparative Transcriptome Analysis for Metabolic Engineering

Shuobo Shi, Tao Chen, and Xueming Zhao

Abstract

Transcriptome profiling allows massively parallel analysis of the dynamic expression of all genes and captures the cell physiology and regulatory mechanism in a holistic manner. Compared to other "omic" techniques, transcriptome is more tractable and sensitive. Transcriptomics has profoundly promoted development and applications of metabolic engineering by analysis of cell metabolism at a system level. Our recent effort was performed on a comparative transcriptome profiling between a riboflavin-producing *Bacillus subtilis* strain RH33 and the wild-type strain *B. subtilis* 168 to rationally identify new targets for improving riboflavin production. This transcriptome analysis-guided method improved the riboflavin titer by 32 ± 3 %. Herein, we describe the detailed experimental protocols for predicting new engineering targets using comparative transcriptome analysis.

Key words: Transcriptome, Metabolic engineering, Microarray, "Omic" techniques, Riboflavin, Systems biology

1. Introduction

Since the introduction of metabolic engineering, a more rational strain improvement approach emerges for biotechnology (1). Metabolic engineering is the directed improvement of cellular properties through the modification of specific biochemical reactions and requires detailed analysis of the organism's metabolic and genetic properties. Progress in metabolic pathway engineering has mainly relied on extensive biochemistry literature and regulatory circuits (2–5). This procedure has limitations because the entire network is often neglected and the unknown biological processes that are important for the question studied are also overlooked.

Transcriptome profiling, which is one of the first "omic" approaches to be developed (6), allows massively parallel analysis of the dynamic expression of all genes and captures the cell physiology and regulatory mechanism in a holistic manner. With the technical

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_22,

advances in transcriptome characterization (7, 8), transcriptome analysis has become the most tractable and sensitive high-throughput approach. Usually, application of transcriptome analysis for metabolic engineering can be divided into three stages of strategies to improve organism: (1) analysis: identification of the differentially expressed genes in genome scale using microarray; (2) design: analysis of the changed genes to find new metabolic engineering targets; and (3) synthesis: improvement of cellular properties through the new identified targets. Usually one does not reach the desired phenotype in one round, and the cycle can then be repeated. The last 10 years have witnessed the transcriptome technique turned into a fundamental tool in biotechnology for identification of metabolic engineering targets that are difficult to be intuitively identified (9, 10). This trend will be expected to continue in the post-genomic era.

Riboflavin (vitamin B2), the precursor of the coenzymes flavin mononucleotide (FMN) and flavin adenine dinucleotide (FAD), is required for feed and food fortification purposes in humans and animals to maintain health. The gram-positive bacterium *Bacillus subtilis*, currently the most competitive riboflavin producer, has been developed using the "classical" strain development approach that relied on iterative cycles of random mutagenesis and selection (11, 12) and using the standard metabolic engineering approach that was carried out by a number of conceivable strategies (2–5). A major hurdle in this traditional exploitation is not enough knowledge available to build a global picture of cellular metabolism.

Here we took a strategy to increase riboflavin production in a riboflavin-producing *B. subtilis* strain based on comparative transcriptome analysis between a riboflavin high-producer RH33 and wild-type 168 (10). This system approach reveals gene expression levels at global scale and, combined with known regulatory and metabolic informations, enables rapid identification of hotspots in the metabolism for strain improvement using the three stages of strategies (i.e., analysis, design, and synthesis). The whole-genome gene expression analysis as illustrated here has identified new targets and has been capable of elevating riboflavin titer by 32 ± 3 % in shake flask. This system approach could be used as a general method for facilitating the identification of novel targets for strain improvement.

2. Materials

2.1. Strains

1. Wild-type *B. subtilis* 168.
2. A riboflavin-producing *B. subtilis* RH33, which was used as the parental strain in our study.
3. *Escherichia coli* Top10, which was used as host strain for constructing plasmids.

2.2. Growth Conditions

1. Luria Bertani broth (LB): 10 g/l NaCl, 10 g/l tryptone (casein peptone), and 5 g/l yeast extract.
2. LB plate: 10 g/l NaCl, 10 g/l tryptone (casein peptone), 5 g/l yeast extract, and 20 g/l agar.
3. Appropriate antibiotics (e.g., ampicillin, kanamycin, chloramphenicol, erythromycin, spectinomycin): after autoclaving add appropriate antibiotics to medium.
4. LBG medium: 10 g/l NaCl, 10 g/l tryptone (casein peptone), 5 g/l yeast extract, and 10 g/l glucose.
5. Minimal medium: 20 g/l glucose, 2 g/l $(NH_4)_2SO_4$, 13.1 g/l KH_2PO_4, 6 g/l K_2HPO_4, 1.2 g/l $NaC_6H_5O_7{\cdot}5.5H_2O$, and 10 mg/l $MgSO_4{\cdot}7H_2O$ and supplemented with tryptophan, phenylalanine, and tyrosine (25 mg/l each).
6. Spizizen's medium: 20 g/l $(NH_4)_2SO_4$, 140 g/l K_2HPO_4, 60 g/l KH_2PO_4, and 12 g/l sodium citrate.
7. Spizizen-I-medium: Spizizen's medium supplemented with 8 g/l glucose, 5 mg/l tryptophan, 0.4 g/l casein hydrolysate, 1 g/l yeast extract, and 0.2 g/l $MgSO_4$.
8. Spizizen-II-medium: Spizizen's medium supplemented with 8 g/l glucose, 0.2 g/l casein hydrolysate, and 1.6 g/l $MgSO_4$.

2.3. Biological and Chemicals Materials

1. Sterile distilled water.
2. DEPC (diethylpyrocarbonate)-treated water: Add 1 ml of 0.1 % DEPC to 1,000 ml distilled water; mix well and let set at room temperature for 4 h; and autoclave.
3. TAE buffer: 40 mM Tris-acetate, 1 mM EDTA (ethylenediaminetetraacetic acid).
4. Agarose gel: dissolve 1 g agarose in 100 ml TAE buffer by heating with microwave.
5. 10× FA (formaldehyde agarose) gel buffer: 400 mM MOPS (morpholinopropanesulfonic acid), 100 mM sodium acetate, and 10 mM EDTA.
6. Formaldehyde denaturing agarose gel: dissolve 1 g agarose in 72 ml DEPC-treated water, and add 10 ml 10× FA gel buffer and 18 ml 37 % formaldehyde.
7. DNA polymerase with its buffer and dNTP mixture (Fermentas, USA).
8. Appropriate restriction enzymes and T4 DNA ligase (Fermentas, USA).
9. Gel loading dye and 1 kb DNA ladder (Fermentas, USA).
10. PCR purification kit (Qiagen, Germany), plasmid extraction kit (Fermentas, USA), gel extraction kit (Fermentas, USA), and RiboPure™-Bacteria RNA Isolation Kit (Ambion, UK).

11. Design and synthesize oligonucleotides as primers for PCR from a commercial company (AuGCT Biotechnology, China), stored at −20 °C (see Note 1).
12. RNaseZap® (Ambion, USA).
13. SuperScript II reverse transcriptase (Invitrogen, USA).
14. Random primers (Invitrogen, USA).
15. Poly-A RNA Control Kit (Affymetrix, USA).
16. Superase-In RNase Inhibitor (Ambion, USA).
17. Deoxyribonuclease I (DNase I) (Amersham Biosciences, Sweden).
18. Terminal deoxynucleotidyl transferase (Promega, USA).
19. Bovine serum albumin (BSA) (Invitrogen, USA).
20. Herring sperm DNA (Promega, USA).
21. B2 control oligonucleotide (Affymetrix, USA).
22. GeneChip® DNA Labeling Reagent (Affymetrix, USA).
23. Wash buffer A: non-stringent wash buffer (Affymetrix, USA).
24. Wash buffer B: stringent wash buffer (Affymetrix, USA).
25. Streptavidin phycoerythrin (SAPE) stain solution (Affymetrix, USA).
26. Antibody solution (Affymetrix, USA).
27. Chemical competent *E. coli* Top10 (Invitrogen, USA).
28. Orotidine-5′-phosphate pyrophosphorylase (Sigma Chemical Co., USA).

2.4. Equipment and Software

1. iCycler Thermal Cycler (Bio-Rad).
2. Agarose gel running system.
3. Centrifuge.
4. Spectrophotometer.
5. UV transilluminator.
6. RNase-free tips.
7. *Bacillus subtilis* genome array.
8. Affymetrix Fluidics Station 450.
9. GeneChip Scanner 3000.
10. Hybridization Oven 640 (Affymetrix, USA).
11. GeneChip Operating Software (GCOS 1.4).
12. dChip software.

3. Methods

The method for "comparative transcriptome analysis for metabolic engineering" consists of three steps: (1) analysis: identification of the differentially expressed genes in genome scale using microarray; (2) design: analysis the changed genes and found new metabolic engineering targets; and (3) synthesis: improvement of cellular properties through the new identified target. Here we took this three-step strategy to increase riboflavin production in a riboflavin-producing *B. subtilis* strain as a case study of comparative transcriptome analysis for metabolic engineering.

3.1. Analysis: Identification of the Differentially Expressed Genes in Genome Scale Using Microarray

1. Preparation of *B. subtilis* strains to be analyzed. A riboflavin-producing *B. subtilis* RH33 and wild-type *B. subtilis* 168 were grown overnight (16–20 h) at 37 °C in LBG medium. The revived overnight culture was then inoculated into 50 ml of LBG medium and grown at 37 °C. The optical density at 600 nm of the culture was measured in spectrophotometer. Cell growth rate was calculated by log-linear regression analysis of OD_{600nm} versus time. Harvested cell when cell reaches the mid-exponential phase by centrifuging at 10,000 × *g* for 5 min at 4 °C and the pellet was stored at −80 °C until RNA isolation and analysis (see Note 2).
2. Extracted RNA from harvested cells using RiboPure™-Bacteria RNA Isolation Kit (see Note 3) according to the manual of producer. The concentration and purity of RNA is determined by measurement of the optical density (OD) at 260 and 280 nm. The samples were diluted with DEPC-treated water. An OD_{260} of 1 is equivalent to 40 μg RNA/ml. The ratio of OD_{260} to OD_{280} values shows RNA purity, and it should fall in the range of 1.8–2.1. Run 0.5 μg of each RNA sample on formaldehyde denaturing agarose gel at 5 V/cm, and visualize the gel on a UV transilluminator (see Note 4).
3. An aliquot of 10 μg *B. subtilis* total RNA was used to synthesize first-strand cDNA with random primers and SuperScript II reverse transcriptase (see Note 5). Firstly, prepare the 30 μl reaction mix for primer annealing that contained 10 μg total RNA, 750 ng random primers, 2 μl Poly-A RNA Control Kit; secondly, incubate the mix at the following temperatures: 70 °C for 10 min, and 25 °C for 10 min, then chill to 4 °C; thirdly, prepare the 60 μl reaction mix for cDNA synthesis that contained 30 μl reaction mix for primer annealing (from previous step), 12 μl 5× buffer, 6 μl 100 mM DTT, 3 μl 10 mM dNTP, 30 U Superase-In RNase Inhibitor, and 1,500 U Superscript II reverse transcriptase; fourthly, incubate the mix at the following temperatures: 25°C for 10 min, 37°C for 60 min,

42 °C for 60 min, and 70 °C for 10 min, then chill to 4 °C; fifthly, add 20 μl of 1 M NaOH to the mix and incubate at 65 °C for 30 min, followed by addition of 20 μl of 1 M HCl to neutralize; and sixthly, purify cDNA using PCR purification kit and quantify the purified cDNA product by 260 nm absorbance (an OD_{260} of 1 is equivalent to 33 μg single-stranded DNA/ml).

4. Fragment cDNA using the following reaction mix: add DNase I to purified cDNA at the ratio of 0.6 U DNase I·cDNA per μg, then incubate at 37 °C for 10 min and 98 °C for 10 min. The fragmented cDNA could be applied directly to next step for terminal labeling. Alternatively, the material can be stored at −20 °C for later use.
5. Label cDNA with biotin. The 50 μl labeling mix contained 10 μl 5× buffer, 0.3 mM GeneChip DNA Labeling Reagent, 2 μl terminal deoxynucleotidyl transferase, and 20 μl fragmentation cDNA product. Then the mix was incubated at 37 °C for 60 min and stopped by adding 2 μl of 0.5 M EDTA. The labeled cDNA can be stored at −20 °C for later use (see Note 6).
6. Hybridization of Affymetrix GeneChip probe arrays. Firstly, prepare 250 μl hybridization solution mix: 125 μl 2× hybridization buffer, 4.125 μl B2 control oligonucleotide (3 nM), 2.5 μl herring sperm DNA (10 mg/ml), 2.5 μl BSA (50 mg/ml), 30 μl fragmented and labeled cDNA, and 85.875 μl water; secondly, add 200 μl of hybridization solution mix to the probe array that has been equilibrated to room temperature before use; and thirdly, load probe arrays into hybridization oven preheated to 45 °C with the rotation at 60 rpm for 16 h.
7. After hybridization, on Affymetrix Fluidics Station 450, the microarray slides were washed with the appropriate volume of wash buffers A and B (see Note 7). Three staining solutions: 2× SAPE stain solution and antibody solution are loaded into fluidic station, and washing and staining of the GeneChip® expression probe array are performed.
8. After staining has been done, GeneChip® expression probe array could be scanned with the GeneChip Scanner 3000 (Affymetrix) and was analyzed using the default setting of GeneChip Operating Software (GCOS 1.4) (see Note 8). Then a LOWESS normalization method was performed to normalize the different arrays using dChip software. The differentially expressed genes were identified through overlapping gene analysis of biological duplicate experiments using a two-fold change as an empirical criterion.

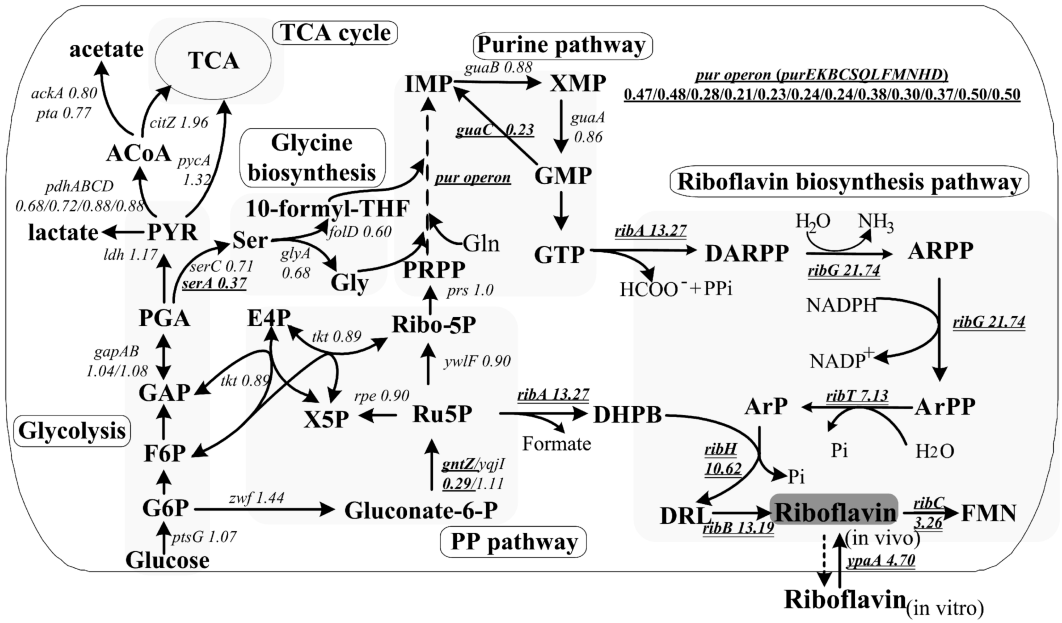

Fig. 1. Schematic overview of expression profiles of genes involved in relevant pathways of riboflavin production. The numbers are the ratios of the comparative expression levels in *B. subtilis* strain RH33 versus 168. *Bold letter underlined* indicates downregulation, *bold letter double underline* indicates upregulation, and *letter without underline* indicates without notable changes. *G6P* glucose-6-phosphate, *F6P* fructose-6-phosphate, *GAP* D-glyceraldehyde 3-phosphate, *PGA* 3-phosphoglycerate, *PYR* pyruvic acid, *AcoA* acetyl coenzyme A, *Ser* serine, *Gly* glycine, *10-formyl-THF* 10-formyl tetrahydrofolate, *gluconate-6-P* 6-phospho-D-gluconate, *Ru-5-P* ribulose-5- phosphate, *Ribo-5-P* ribose-5-phosphate, *PRPP* phosphoribosylpyrophosphate, *Gln* glutamine, *X5P* xylulose 5-phosphate, *E4P* D-erythrose 4-phosphate, *IMP* inosine monophosphate, *XMP* xanthosine monophosphate, *GMP* guanosine monophosphate, *GTP* guanosine triphosphate, *DARPP* 2,5-diamino-6-ribosylamino-4(3H)-pyrimidinone-5′-phosphate, *ARPP* 5-amino-6-(5′-phosphoribosylamino)uracil, *ArPP* 5-amino-6-(5′-phosphoribitylamino)uracil, *ArP* 4-(1-D-ribitylamino)-5-amino-2,6-dihydroxypyrimidine, *DHPB* 3,4-dihydroxy-2-butanone 4-phosphate, *DRL* 6,7-dimethyl-8-ribityl-lumazine, and *FAD* flavin adenine dinucleotide.

3.2. Design: Analysis of the Changed Genes and Found New Metabolic Engineering Targets

1. From the microarray results, it was found that 619 genes showed significant variations at the genome-wide transcriptional level between *B. subtilis* RH33 and *B. subtilis* 168, representing about 15 % of *B. subtilis* genome (see Note 9). The 619 differentially expressed genes fell into nearly all functional categories of *B. subtilis* (http://genolist.pasteur.fr/SubtiList/).
2. From the overview of expression profiles of genes involved in relevant pathways of riboflavin production, e.g., as shown in Fig. 1, it indicates some transcriptional regulation mechanism (s) underpinned the strain performance differences, directly or indirectly affecting riboflavin production in *B. subtilis* RH33 (see Note 10).
3. From Fig. 1, it was interesting to note that purine genes (*purEKBCSQLFMNHD*) together with glycine biosynthesis genes (*serA*, *serC*, *glyA*, *folD*) were all greatly downregulated in *B. subtilis* RH33. The downregulated purine and glycine biosynthesis genes would limit the supply of the precursors

for riboflavin overproduction. Nonetheless, it was found out that these downregulated purine, glycine, and other genes involved in purine metabolism (*guaC*, *pbuG*, *xpt-pbuX*, *yqh*, and *pbuO*) were all negatively regulated by global regulator PurR in *B. subtilis* (13). Hence, the downregulation of those genes may be due to the repression by PurR. According to the current model, the PRPP can bind PurR to prevent its binding to DNA and lead to derepression of PurR-regulated genes. Therefore, it can be deduced that the low expression of PurR-regulated genes might be caused by a small cellular PRPP pool.

4. Subsequently, metabolite analysis was carried out and tested this hypothesis. To test PRPP pool, immediately upon harvest, 5 ml of cell sample was added to 15 ml of quenching fluid containing 70 mM HEPES in 60 % (v/v) aqueous methanol. The samples were centrifuged to separate the quenching fluid, and the remaining cell pellets were resuspended in 35 % (v/v) perchloric acid. After one freeze–thaw cycle, the sample was neutralized by 5 M K_2CO_3. The precipitate was removed by another centrifugation, and resulting supernatants were stored at −80 °C for metabolite analysis.
5. Measurement of PRPP concentrations was carried out using the following method. Prepare the incubation mixture (1.0 ml) in a quartz cuvette that contained 0.02 ml of orotate (0.01 M), 0.02 ml of Tris buffer (1 M, pH 8), 0.02 ml of $MgCl_2$ (0.1 M), 0.2 ml of orotidine-5′-phosphate pyrophosphorylase (500 U/ml, ethanol fraction), and an aliquot from the extracted metabolites (previous step). Incubate the mix at room temperature for 20 min, and measure the decrease in optical density at 295 nm. The molar absorption coefficient for the density decrease is 3,950. The PRPP pool in *B. subtilis* RH33 was only 24 % that of *B. subtilis* 168 measured at the conditions as the microarray experiments. It was proposed that enhancing PRPP pool would derepress the expression of *pur* operon and glycine biosynthesis genes, thus increasing the precursors supply and leading to increased production of riboflavin in *B. subtilis* RH33.

3.3. Synthesis: Improvement of Cellular Properties Through the New Identified Target

The low concentration of PRPP may be the bottleneck for further increasing riboflavin production in *B. subtilis* RH33. The formation of PRPP is catalyzed by the enzyme PRPP synthetase (encoded by *prs* gene) and ribose-5-phosphate isomerase B (encoded by *ywlF* gene) from ribulose-5-phosphate (Fig. 1). We decided to co-overexpress the *prs* and *ywlF* genes simultaneously to increase PRPP in *B. subtilis* RH33. Plasmid that can co-overexpress the *prs* and *ywlF* genes simultaneously was constructed by the following procedures:

1. Polymerase chain reaction (PCR) was used to amplify structural genes of *prs* (1.2 kb) and *ywlF* (0.6 kb) with the following primers:

prs-up 5′-GGGGCCCGGGCCAGAGCGAGACAAGTAAA,*prs-down* 5′-GGGGGAGCTCGCTAGCTCCTATTACAAACAATACCCA, *ywlF*-up 5′-GGGGCCCGGGGCTAGCGGCTGCGCGGTCAATA, and *ywlF*-down 5′-GGGGGAGCTCGCGGCCGCTTGTTTCAATTCCGCTTGGTC, based on the published *B. subtilis* 168 genome sequence. The *prs* and *ywlF* PCR products were obtained by PCR according to the manual of DNA polymerase producer. PCR products were then purified with PCR purification kit. The PCR products were run on an agarose gel to estimate the concentration of PCR product.

2. The *prs* PCR product was ligated into pUC18 (obtained from BGSC, *Bacillus* Genetic Stock Center, http://www.bgsc.org/) using *Xma*I and *Sac*I restriction sites to construct pRPU10. The constitutively expressed P43 promoter was obtained from the vector pHPL10 (obtained from Professor Xueming Zhao, Department of Biochemical Engineering, School of Chemical Engineering and Technology, Tianjin University) after digested with *Bam*HI and *Xma*I. It was then ligated in the *Bam*HI and *Xma*I restriction sites of pRPU10 to get plasmid pRPU12. The *ywlF* PCR product was digested with *Nhe*I and *Sac*I and cloned into *Nhe*I-*Sac*I-sites of pRPU12 to construct pRPU14. To facilitate further research, a spectinomycin resistance gene (*spc*) from pSG1192 (obtained from BGSC, *Bacillus* Genetic Stock Center, http://www.bgsc.org/) was inserted at the *Sal*I site of pRPU14 to give plasmids pRPU15. The enzyme digestion system and DNA ligation system were followed as the instructions of producers (see Note 11). Ligation systems were transformed into chemical competent *E. coli* Top10 that researchers could purchase from a commercial company, following the transformation protocol supplied with the competent cells or competent cell kit. The plasmids were extracted from transformants according to the producer's manual and confirmed by enzyme digestion or sequencing.

3. The constructed plasmids were transformed into competent cells of *B. subtilis* RH33 using the following protocol for *B. subtilis*. Firstly, 5 ml of Spizizen-I-medium was inoculated with a colony of spores of the *B. subtilis* strain to be transformed and was grown overnight (16–20 h) at 37 °C; secondly, 0.5 ml of the overnight culture was then inoculated into 5 ml of Spizizen-I-medium and grown at 37 °C; thirdly, the optical density at 600 nm of the culture was measured after 2 h, and then every 15 min until the culture was found to be in late log phase; fourthly, 0.75 ml of late log phase culture was then inoculated into 5 ml of Spizizen-II-medium at 37 °C for 1.5 h; fifthly, mix 1 ml of this culture and 1–2 μg plasmid DNA at 37 °C for 1 h; sixthly, grow the cells at 37 °C for 2 h; and finally, small aliquots

of the culture of transformed cells were transferred to plates containing the desired antibiotics for selection.

4. *B. subtilis* strains were grown in shake flask with LBG medium. For riboflavin measurements, the samples were diluted with 0.05 M NaOH to the linear range of the spectrophotometer and centrifuged at 12,000 rpm for 2 min to remove the cells. Then the OD_{444nm} was immediately measured (see Note 12). It was shown that, when both *prs* and *ywlF* were overexpressed, the *B. subtilis* RH33-PY mutant exhibited a riboflavin titer of 32 ± 3 % higher than *B. subtilis* RH33. Thus, it indicated that the metabolic engineering strategy identified by transcriptome analysis successfully improved the trait of riboflavin overproduction in *B. subtilis* RH33.

4. Notes

1. 5′ end of PCR primers can be designed to contain the restriction sites that will be used for subcloning. To ensure the effectiveness of enzyme digestion, protective bases could be added to the 5′ end of the restriction sites. This permits more efficient digestion of the PCR product by restriction endonucleases.
2. In *B. subtilis* RH33, mid-exponential phase cell corresponded to the state of high riboflavin production. During mid-exponential phase, the specific growth rates were nearly constant, without discernible accumulation of byproducts such as acetate. Our past experiences showed that there was no significant difference in transcriptional profiles at different time points of mid-exponential phase.
3. Clean the lab bench and pipettors with an RNase decontamination solution before working with RNA. Use RNase-free tips to handle all the transfers. Change laboratory gloves frequently. All these suggestions would protect the RNA samples from nucleases that exist in environment, including your hands. Extracted RNA samples should be stored at −80 °C.
4. From the gel results, the 16S and 23S ribosomal RNA (rRNA) bands should be observed with sharp and intense bands. In good quality RNA sample, the intensity of the 23S rRNA band should be about twice that of the 16S rRNA band. If there is some smearing in the gel, it indicates DNA contamination may present in the sample. If the rRNA bands do not have a discernible lower edge, it means RNA degradation happened, and the RNA samples should be discarded. In this traditional denaturing gel electrophoresis, a large amount of RNA was required.

Now Agilent's 2100 Bioanalyzer provides a particularly effective method for evaluating total RNA integrity for those samples, of which only limited amounts of sample for extracted RNA are available to begin with.

5. High-quality total RNA is essential for the success of microarray experiments. RNases are found everywhere, including your own hands. Use RNase-free tips to handle all the transfers. Change laboratory gloves frequently.
6. To estimate the labeling efficiency, a gel-shift assay can be performed (Follow GeneChip® Expression Analysis Technical Manual (http://media.affymetrix.com/support/downloads/manuals/expression_analysis_technical_manual.pdf) of Affymetrix). In general, greater than 90 % of the fragments should be labeled and, therefore, shifted.
7. In the Fluidics Station dialog box on the workstation, select the correct experiment name from the drop-down experiment list. The probe array type will appear automatically. Follow the instructions from the user's guide for your GeneChip® Fluidics Station.
8. Make sure there is no air bubbles inside of the probe array. Look over the operator's manual to be sure to be familiar with the operation of the scanner before attempting to scan a probe array.
9. To validate transcriptome results, relative abundances of selective transcripts could be measured by quantitative reverse transcription PCR.
10. The differentially expressed genes also fell into other pathway or regulation function that may affect riboflavin production. Here we only took some direct-related pathway for riboflavin producing as an example. Clustering of coexpressed genes and their biological functions allows us to visualize data in a high-dimensional view and get a straightforward analysis for the transcriptome data. The most interesting genes are those that have been clustered together and show similar function.
11. To make successful cloning, researcher may need to optimize ligation and transformation efficiencies (14). One of the most important parameters in a ligation reaction is the ratio of insert to vector molecules, which is approximately 2:1.
12. Riboflavin is a light sensitive vitamin, and the measurement should be fast to avoid its degradation.

References

1. Bailey JE (1991) Toward a science of metabolic engineering. Science 252(5013):1668–1675
2. Zhu YB, Chen X, Chen T et al (2006) Overexpression of glucose dehydrogenase improves cell growth and riboflavin production in *Bacillus subtilis*. Biotechnol Lett 28 (20):1667–1672
3. Wang Z, Chen T, Ma X et al (2011) Enhancement of riboflavin production with *Bacillus subtilis* by expression and site-directed mutagenesis of *zwf* and *gnd* gene from *Corynebacterium glutamicum*. Bioresour Technol 102 (4):3934–3940
4. Shi S, Shen Z, Chen X et al (2009) Increased production of riboflavin by metabolic engineering of the purine pathway in *Bacillus subtilis*. Biochem Eng J 46:28–33
5. Zamboni N, Mouncey N, Hohmann HP et al (2003) Reducing maintenance metabolism by metabolic engineering of respiration improves riboflavin production by *Bacillus subtilis*. Metab Eng 5(49–55)
6. Schena M, Shalon D, Davis RW et al (1995) Quantitative monitoring of Gene-Expression patterns with a complementary-DNA microarray. Science 270(5235):467–470
7. Shi S, Chen T, Zhao X (2010) Transcriptome platforms and applications to metabolic engineering. Sheng Wu Gong Cheng Xue Bao 26 (9):1187
8. Ozsolak F, Platt AR, Jones DR et al (2009) Direct RNA sequencing. Nature 461 (7265):814–818
9. Jewett MC, Oliveira AP, Patil KR et al (2005) The role of high-throughput transcriptome analysis in metabolic engineering. Biotechnol Bioprocess Eng 10:385–399
10. Shi S, Chen T, Zhang Z et al (2009) Transcriptome analysis guided metabolic engineering of *Bacillus subtilis* for riboflavin production. Metab Eng 11(4–5):243–252
11. Perkins JB, Pero JG, Sloma A (1991) Riboflavin overproducing strains of bacteria. European patent 0,405,370
12. Stahmann KP, Revuelta JL, Seulberger H (2000) Three biotechnical processes using *Ashbya gossypii*, *Candida famata*, or *Bacillus subtilis* compete with chemical riboflavin production. Appl Microbiol Biotechnol 53:509–516
13. Saxild HH, Brunstedt K, Nielsen KI et al (2001) Definition of the *Bacillus subtilis* PurR Operator Using Genetic and Bioinformatic Tools and Expansion of the PurR Regulon with *glyA*, *guaC*, *pbuG*, *xpt-pbuX*, *yqhZ-folD*, and *pbuO*. J Bacteriol 183(21):6175–6183
14. Sambrook J, Russell DW (2001) Molecular cloning: a laboratory manual, 3rd edn. Cold Spring Harbor Laboratory Press, Cold Spring Harbor

Chapter 23

Merging Multiple Omics Datasets In Silico: Statistical Analyses and Data Interpretation

Kazuharu Arakawa and Masaru Tomita

Abstract

By the combinations of high-throughput analytical technologies in the fields of transcriptomics, proteomics, and metabolomics, we are now able to gain comprehensive and quantitative snapshots of the intracellular processes. Dynamic intracellular activities and their regulations can be elucidated by systematic observation of these multi-omics data. On the other hand, careful statistical analysis is necessary for such integration, since each of the omics layers as well as the specific analytical methodologies harbor different levels of noise and variations. Moreover, interpretation of such multitude of data requires an intuitive pathway context. Here we describe such statistical methods for the integration and comparison of multi-omics data, as well as the computational methods for pathway reconstruction, ID conversion, mapping, and visualization that play key roles for the efficient study of multi-omics information.

Key words: Multi-omics analysis, Pathway reconstruction, Data normalization, Pathway visualization, Transcriptome, Proteome, Metabolome

1. Introduction

Molecular biology is rapidly gearing up to become a data-driven science, with the advent of a series of high-throughput technologies to obtain comprehensive and quantitative snapshots of dynamic intracellular activities. Ignited by the introduction of DNA microarrays in the mid-1990s for transcriptomics (1), the so-called omics approaches have now widely spread throughout different layers of cell biology, including proteomics, metabolomics, fluxomics, and interactomics (2). Each of the omics layers is coupled with several analytical technologies that realize the comprehensive measurement. For example, shotgun proteomics utilizes fractionation of peptides using high-performance or nano-liquid chromatography (HPLC, nano-LC) followed by automated tandem mass spectrometry (LC–MS/MS) (3, 4), and metabolomics is likewise performed using time-of-flight mass spectrometry (TOF-MS) following

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5_23,

metabolite separation using LC, gas chromatography (GC), or capillary electrophoresis (5–7). For transcriptomics, high-throughput sequencing is actively adopted as the successor of cDNA or oligonucleotide arrays, often in combination with other experimental procedures such as the chromatin immunoprecipitation (ChIP) and bisulfite sequencing to elucidate genome-wide epigenetic and expression regulations (8–10). Highly sensitive measurements at the level of single cell are also being explored (11–13). A cell, however, is not merely a collection of these omics layers, but rather an integrated system requiring holistic integration of multiple omics information (14, 15) in the ultimate goal to understand and design such dynamical system (16, 17). Therefore, the seamless integration of omics analyses is a key process to gain insights on the complex biological systems (18–22), and robust computational methods for pathway reconstruction, statistical analyses, and visualization are essential for this purpose (23–27). In this chapter, an introduction to these computational methods in the integration and interpretation of multiple omics information is presented.

2. Materials

For the computational study merging multiple omics datasets described below, the following materials can be used:

1. Omics datasets from transcriptome, proteome, and metabolome, each normalized by respective methods suited for the target molecules and for the analytical technology, such as locally weighted scatterplot smoothing (LOWESS) method for microarrays (28), exponentially modified protein abundance index (emPAI) for proteomics (29), and fragment per kilobase per million mapped fragments (fpkm) or read per kilobase per million mapped reads (rpkm) for mRNA-Seq (30).
2. Amino acid sequences of the genes of target species in FASTA format (see Note 1 for non-model organisms for which genome sequence is not available). For bacterial species with complete genomes available in public databases, this can be retrieved by accessing the G-language Genome Analysis Environment REST Web Service (31) URL such as http://rest.g-language.org/NC_000913/*/translation where the RefSeq ID NC_000913 (which is for *Escherichia coli* K12W1655) can be modified to that of interest.
3. Software for pathway reconstruction, such as Kyoto Encyclopedia of Genes and Genomes Automatic Annotation System (KEGG KAAS) (32) (see Table 1 for other possibilities).

Table 1
List of pathway reconstruction software

GEM System (34)	Orthology search based on UniProt and KEGG
identiCS (35)	Orthology search based on TrEMBL
KEGG KAAS (32)	Orthology search based on KEGG KO
metaSHARK (36)	Motif search with PRIAM
Pandora (37)	Gene ontology, interactions
Pathologic (38)	Text mining of annotations
SEED RAST (39)	Similarity search using FIGfam

Table 2
List of pathway mapping software

	Integrated map	Multi-omics data	Time-series data	Based on	Online
Omics Viewer(40)	Yes	Yes	Animation	BioCyc	Yes
iPath(41)	Yes	Yes	No	KEGG	Yes
Pathway Projector(33)	Yes	Yes	Graph	KEGG	Yes
Reactome Skypainter (42)	Yes	Not metabolome	No	Reactome	Yes
VANTED(43)	No	Yes	Graph	KEGG/ plant	No

4. Software for statistical analysis, such as SPSS (SPSS Inc., Chicago, IL), or R-project (http://www.r-project.org/).
5. Software for pathway visualization, such as Pathway Projector (33) (see Table 2 for other possibilities).

3. Methods

3.1. Semantic Matching of Multiple Omics Layers

In multi-omics analysis, it is essential to keep track of the correspondence of molecular components across different omics layers, and with public pathway databases. The understanding of multi-omics data and the complex systematic interactions requires the context of pathways. Moreover, the use of pathway-related tools, such as those for data mapping, requires the matching of local gene

names to IDs used in the databases. For this purpose, here we describe the basic methods for pathway reconstruction and ID matching (see Note 2).

1. Run KEGG KAAS online at http://www.genome.jp/tools/kaas/, with "complete or draft genome" mode in bidirectional best hit (BBH) method by uploading the amino acid sequences in FASTA format (see Note 3).
2. When an e-mail notifying the completion of KAAS task arrives, access the results online and retrieve the KEGG Orthology (KO) ID mapping list, pathway maps, and KEGG BRITE hierarchy files from the "Download" menu. The BRITE hierarchy files contain the mapping information between KO, EC number, and pathways (see Note 4).
3. For metabolites, use KEGG compound ID (those starting with C followed by five-digit number; e.g., α-D-glucose is C00267) for ID mapping. When mapping and integration with reactome database are desired, the use of ChEBI is more suitable (44). When a molecular compound is not found in KEGG or ChEBI, search in PubChem, which is most comprehensive (45).

3.2. Normalization of Omics Data and Statistical Analyses

One of the key challenges in the data analysis of multi-omics research is in the normalization procedure. Each of the omics layers and analytical technologies has characteristic rate of noise and variation. Therefore, in order to perform statistical analysis among these different samples, a highly robust normalization approach is necessary, and here we describe the use of expression index (EI) and average expression index (AEI) for this purpose (20).

1. Define a reference sample, such as a measurement of wild-type cell, and calculate the median value for each of the measured component j as $v_{j,\ \mathrm{control}}$ among the technical or biological replicates.
2. For each component j in the sample i, calculate the normalized measurement value M_{ij} by taking the log of the ratio of measured value over the median of reference sample, as follows:

$$M_{ij} = \log_2(v_{ij}/v_{j,control}).$$

   This operation is basically equivalent to the M log intensity ratio of MA plot (46) for the normalization of DNA microarray data.
3. In order to analyze the variability of each measured values, first calculate the median absolute deviation (MAD) for each component j as the median of the absolute deviations from the data's median in a data series (certain type of molecules analyzed by certain methodology; e.g., mRNA by microarray), as follows:

$$MAD_j = median_j\{|M_{ij} - median(M_{ij})|\}.$$

MAD is a robust statistic, which can better tolerate the existence of a small number of outliers that are likely to be present in omics studies (see Note 5).

4. Based on M_{ij} and MAD_j, calculate the expression index EI_{ij} as follows:

$$EI_{ij} = \frac{M_{ij} - median_j}{MAD_j}.$$

EI_{ij} is therefore the normalized expression ratio of the component j in sample i, which minimizes the sensitivity of the average to outliers or to components that show large variations (see Note 6).

5. Next, take the average of all EI for n components in a sample i to calculate the average expression index AEI_i as follows:

$$AEI_i = \frac{\sum_{j=1}^{n} EI_{ij}}{n}.$$

AEI is the index of global change within a specific type of cellular component (e.g., mRNAs, proteins, or metabolites) measured by a specific type of analytical method (e.g., microarray, qRT-PCR, mRNA-Seq, nano-LC–MS/MS proteomics, or CE–MS).

6. In order to identify samples with significant variations, statistical analysis can be performed based on the AEI. Firstly, for each analytical method, test whether all measured samples had equal variances using the Levene's test (47). Based on the test result, perform one-way analysis of variance (one-way ANOVA) when variances are equal between groups (48), and Welch ANOVA when variances are not equal between groups (49), to detect significant differences among the AEIs. Lastly, perform post hoc Games–Howell test (50) (see Note 7) to identify the sample pairs with significant differences. Throughout these analyses, apply a unified level of significance, typically $p < 0.05$.

7. To specifically observe which of the components are actually varying within the samples shown to be significantly different by the statistical analyses using AEI, visualization of EI values using heat map clustering is useful (Fig. 1).

3.3. Pathway Mapping and Visualization

Omics data are inherently comprehensive for the type of molecules. Transcriptome analysis in principle measures the expression of all mRNAs, and metabolomics quantify all small molecules. Interpretation of such data is therefore a complex task, especially in multi-omics analysis where the interactions of different omics layers

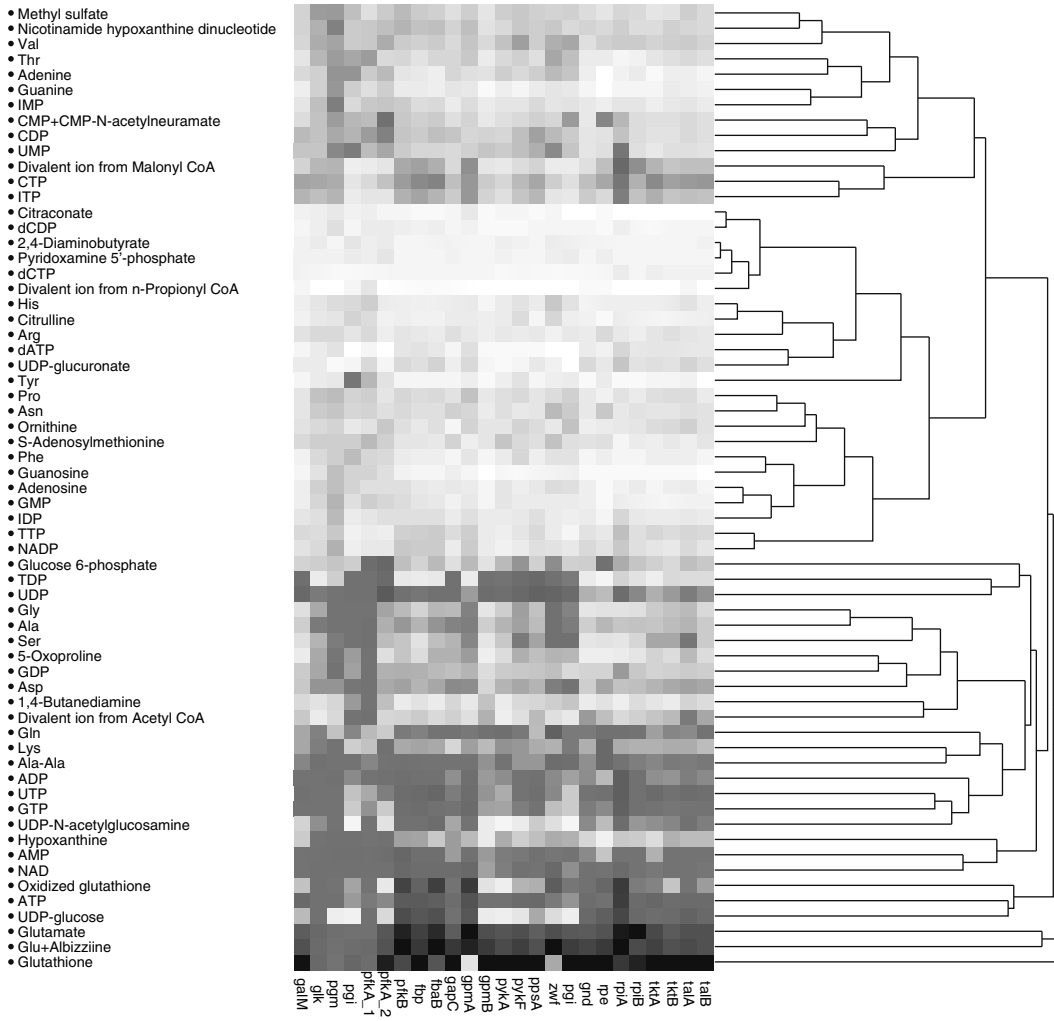

Fig. 1. Heat map clustering EI values of metabolomics measurements of *Escherichia coli* K12 knockouts (20). *Rows* represent the molecular components (metabolites), and the *columns* represent the knockout conditions. For example, the *left most column* represents a sample with *galM* gene knockout. Heat map values range from *white* (low EI) to *black* (high EI).

become the key to understanding the dynamics of intracellular activities. To this end, pathway mapping and visualization is an effective means for such tasks, for it provides a biochemical context in merging multiple layers of omics, and an intuitive way of presenting large amount of data. In this section, we describe the use of Pathway Projector for this purpose (see Note 8) (see Fig. 2).

1. Prepare the visualization setting by opening the "Tools" window of Pathway Projector (http://ws.g-language.org/g4/), and by inputting the commands such as the following in the "Data Mapping" section:

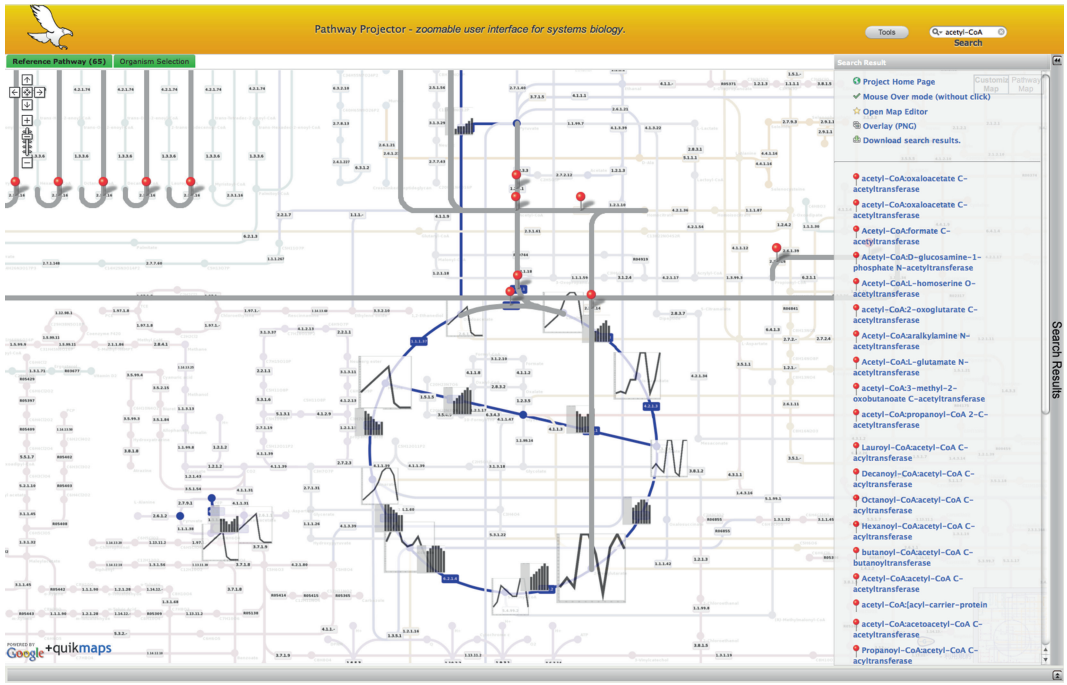

Fig. 2. Pathway mapping using Pathway Projector. Time-series metabolome data are visualized as *line graphs* over the compound nodes (*circle*) of TCA cycle, and proteome data are shown as *bar graphs*. By visualizing multi-omics data simultaneously in an integrated pathway map context, one can intuitively interpret the systematic interactions of intracellular networks.

Organism:eco

compound,#eeeeee

reaction,#eeeeee

where the first line defines the organism-specific pathway to map onto (eco, is the three-letter organism code for *Escherichia coli* K12W1655 in KEGG) and the second and third lines defined the default color of all compounds and reactions to be gray (#eeeeee). See also the documentations for mapping at http://www.g-language.org/PathwayProjector/annotation.html#mapping.

2. Prepare single point data in the syntax:

compound,color,size,label,label size

For example, to mark the node for α-D-Glucose in red with a circle of size 15, accompanied by a label saying "GLUCOSE" of size 13, can be written as:

C00267,#FF0000,15,GLUCOSE,13

Note to use the compound ID obtained in Subheading 3.1, **step 3**, and that the color code is an 8bit RGB code used in HTML. Likewise, edges can be marked by specifying KO, EC number, or KEGG reaction ID instead of the compound ID.

3. Prepare multipoint or time-series data in the syntax:

   >ID,type
   timepoint1, value1
   timepoint2, value2
   timepoint3, value3
   //
   For example, to draw a time-series line graph for the metabolome data of α-D-Glucose, with three time points 1,2,3, and respective amounts 10,20,30:
   >C00267,line
   1,10
   2,20
   3,30
   //
   where ">" marks the beginning of data block and "//" marks the end.
4. When all data input is prepared, press the "Generate Overlay Map" to start visualization.

4. Notes

1. With the advent of high-throughput sequencing technologies, it is becoming feasible to apply multi-omics analysis to non-model organisms without complete genomes. In such cases, the reference sequences can be obtained from de novo transcriptome assembly of mRNA-Seq data (51). For such data, KEGG KAAS can be run in EST mode from the assembled nucleotide sequences with single-directional best hit (SBH) assignment method to increase sensitivity, especially when genomes of closely related species are not available. However, note that the calculation of expression level using fpkm/rpkm using de novo transcriptome assembly can be erroneous when misassembly occurs.
2. The term "pathway reconstruction" is often used for two different goals. The first is an extension to the concept of functional annotation so that the goal is to compile a set of annotations for a given genome to achieve a pathway–genome database. The other use is for metabolic flux analysis using constraint-based modeling, and the goal is to create a stoichiometric model of intracellular reactions (52). Fluxome analysis based on ^{13}C metabolomics requires the latter model to calculate each flux (53). In this manuscript, we refer to the former use throughout.
3. Note that pathway reconstruction is essentially based on orthology detection. Pathway databases often assign an

enzyme class to an orthologous gene cluster, and thus, pathway is reconstructed by identifying orthologous clusters and subsequently retrieving the corresponding annotations. One of the most fundamental methodologies for orthology inference is the bidirectional best hit (or symmetrical/reciprocal best hits) in similarity searches (54, 55).

4. Enzyme Commission (EC) number maintained by the International Union of Biochemistry and Molecular Biology (IUBMB) is a standard enzyme classification nomenclature, where the four digits represent specific biochemical activities of the enzymes (56). For example, enzymes with the first digit (1) are oxidoreductases, (2) are transferases, (3) are hydrolases, (4) are lyases, (5) are isomerases, and (6) are ligases. While many databases utilize EC number as the primary term to describe enzymes, we strongly recommend the use of KO as the primary ID for omics studies, since (1) EC numbers are frequently updated by IUBMB, and some of them become updated, merged, or deprecated; (2) there are many intracellular reactions and interactions that have no EC number assigned (57); and (3) there are a number of EC numbers for which no corresponding gene is found (58).
5. MAD is a measure of statistical dispersion, and can be used likewise the sample variance or standard deviation as the estimator of scale. However, in the calculation of standard deviation, the distances from the mean are squared, and prescreening of outliers that can heavily influence it. MAD, on the other hand, is calculated based on the absolute deviation from the mean, and is more resilient to outliers (59).
6. Note that EI calculated from the median and MAD is analogous to z-score statistic calculated from the mean μ and standard deviation σ:

$$z = \frac{X - \mu}{\sigma}.$$

   Thus, EI can be considered a more robust alternative to z-score where large variations and outliers are assumed to be contained within the given data series.
7. When significant difference is observed by ANOVA, post hoc tests are required to identify the pairs of samples that exhibit significant differences, since such multiple testing increases the type-I error which must be corrected. A number of statistical tests are available for this purpose, each with different levels of robustness and sensitivity to errors, such as Fisher's Protected Least Significant Difference (PLSD), Tukey–Kramer, and Bonferroni/Dunn (60). Games–Howell test is a nonparametric test that is highly robust, and rather conservative of the methods.

8. As shown in Table 2, a number of pathway mapping and visualization tools are available. Note, however, that not many of them provide the capability to simultaneously map multiple omics data, and many of them do not distinguish individual genes or proteins in a protein complex. Since omics experiments are comprehensive for a given type of molecular species, it is also desirable that the pathways are not subdivided into small specific subgraphs; instead, an integrated pathway map showing the entire molecular network is more suited for omics study. Carefully consider these requirements when choosing a visualization tool.

Acknowledgements

This work was supported by funds from the Yamagata Prefectural Government and Tsuruoka City.

References

1. Stoughton RB (2005) Applications of DNA microarrays in biology. Annu Rev Biochem 74:53–82
2. Kandpal R, Saviola B, Felton J (2009) The era of 'omics unlimited. Biotechniques 46 (351–352):354–355
3. Becker CH, Bern M (2011) Recent developments in quantitative proteomics. Mutat Res 722:171–182
4. Ishihama Y (2005) Proteomic LC-MS systems using nanoscale liquid chromatography with tandem mass spectrometry. J Chromatogr A 1067:73–83
5. Patti GJ, Yanes O, Siuzdak G (2012) Innovation: metabolomics: the apogee of the omics trilogy. Nat Rev Mol Cell Biol 13:263–269
6. Ramautar R, Mayboroda OA, Somsen GW, de Jong GJ (2011) CE-MS for metabolomics: developments and applications in the period 2008-2010. Electrophoresis 32:52–65
7. Saito N, Ohashi Y, Soga T, Tomita M (2010) Unveiling cellular biochemical reactions via metabolomics-driven approaches. Curr Opin Microbiol 13:358–362
8. Gibbons JG, Janson EM, Hittinger CT, Johnston M, Abbot P, Rokas A (2009) Benchmarking next-generation transcriptome sequencing for functional and evolutionary genomics. Mol Biol Evol 26:2731–2744
9. Niedringhaus TP, Milanova D, Kerby MB, Snyder MP, Barron AE (2011) Landscape of next-generation sequencing technologies. Anal Chem 83:4327–4341
10. Werner T (2010) Next generation sequencing in functional genomics. Brief Bioinform 11:499–511
11. Citri A, Pang ZP, Sudhof TC, Wernig M, Malenka RC (2011) Comprehensive qPCR profiling of gene expression in single neuronal cells. Nat Protoc 7:118–127
12. Geiss GK, Bumgarner RE, Birditt B, Dahl T, Dowidar N, Dunaway DL, Fell HP, Ferree S, George RD, Grogan T et al (2008) Direct multiplexed measurement of gene expression with color-coded probe pairs. Nat Biotechnol 26:317–325
13. Vogel C, Marcotte EM (2012) Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. Nat Rev Genet 13:227–232
14. Kitano H (2002) Computational systems biology. Nature 420:206–210
15. Kitano H (2002) Systems biology: a brief overview. Science 295:1662–1664
16. Arita M, Robert M, Tomita M (2005) All systems go: launching cell simulation fueled by integrated experimental biology data. Curr Opin Biotechnol 16:344–349

17. Tomita M (2001) Towards computer aided design (CAD) of useful microorganisms. Bioinformatics 17:1091–1092
18. Buescher JM, Liebermeister W, Jules M, Uhr M, Muntel J, Botella E, Hessling B, Kleijn RJ, Le Chat L, Lecointe F et al (2012) Global network reorganization during dynamic adaptations of Bacillus subtilis metabolism. Science 335:1099–1103
19. Canelas AB, Harrison N, Fazio A, Zhang J, Pitkanen JP, van den Brink J, Bakker BM, Bogner L, Bouwman J, Castrillo JI et al (2010) Integrated multilaboratory systems biology reveals differences in protein metabolism between two reference yeast strains. Nat Commun 1:145
20. Ishii N, Nakahigashi K, Baba T, Robert M, Soga T, Kanai A, Hirasawa T, Naba M, Hirai K, Hoque A et al (2007) Multiple high-throughput analyses monitor the response of E. coli to perturbations. Science 316:593–597
21. Park SJ, Lee SY, Cho J, Kim TY, Lee JW, Park JH, Han MJ (2005) Global physiological understanding and metabolic engineering of microorganisms based on omics studies. Appl Microbiol Biotechnol 68:567–579
22. Moxley JF, Jewett MC, Antoniewicz MR, Villas-Boas SG, Alper H, Wheeler RT, Tong L, Hinnebusch AG, Ideker T, Nielsen J et al (2009) Linking high-resolution metabolic flux phenotypes and transcriptional regulation in yeast modulated by the global regulator Gcn4p. Proc Natl Acad Sci U S A 106:6477–6482
23. Gehlenborg N, O'Donoghue SI, Baliga NS, Goesmann A, Hibbs MA, Kitano H, Kohlbacher O, Neuweger H, Schneider R, Tenenbaum D et al (2010) Visualization of omics data for systems biology. Nat Methods 7: S56–S68
24. Zhang W, Li F, Nie L (2010) Integrating multiple 'omics' analysis for microbial biology: application and methodologies. Microbiology 156:287–301
25. Joyce AR, Palsson BO (2006) The model organism as a system: integrating 'omics' data sets. Nat Rev Mol Cell Biol 7:198–210
26. De Keersmaecker SC, Thijs IM, Vanderleyden J, Marchal K (2006) Integration of omics data: how well does it work for bacteria? Mol Microbiol 62:1239–1250
27. Steinfath M, Repsilber D, Scholz M, Walther D, Selbig J (2007) Integrated data analysis for genome-wide research. EXS 97:309–329
28. Cleveland WS (1979) Robust locally weighted regression and smoothing scatterplots. J Am Stat Assoc 74:829–836
29. Ishihama Y, Oda Y, Tabata T, Sato T, Nagasu T, Rappsilber J, Mann M (2005) Exponentially modified protein abundance index (emPAI) for estimation of absolute protein amount in proteomics by the number of sequenced peptides per protein. Mol Cell Proteomics 4:1265–1272
30. Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nat Methods 5:621–628
31. Arakawa K, Kido N, Oshita K, Tomita M (2010) G-language genome analysis environment with REST and SOAP web service interfaces. Nucleic Acids Res 38:W700–W705
32. Moriya Y, Itoh M, Okuda S, Yoshizawa AC, Kanehisa M (2007) KAAS: an automatic genome annotation and pathway reconstruction server. Nucleic Acids Res 35:W182–W185
33. Kono N, Arakawa K, Ogawa R, Kido N, Oshita K, Ikegami K, Tamaki S, Tomita M (2009) Pathway projector: web-based zoomable pathway browser using KEGG atlas and Google Maps API. PLoS One 4:e7710
34. Arakawa K, Yamada Y, Shinoda K, Nakayama Y, Tomita M (2006) GEM system: automatic prototyping of cell-wide metabolic pathway models from genomes. BMC Bioinformatics 7:168
35. Sun J, Zeng AP (2004) IdentiCS–identification of coding sequence and in silico reconstruction of the metabolic network directly from unannotated low-coverage bacterial genome sequence. BMC Bioinformatics 5:112
36. Hyland C, Pinney JW, McConkey GA, Westhead DR (2006) metaSHARK: a WWW platform for interactive exploration of metabolic networks. Nucleic Acids Res 34:W725–W728
37. Zhang KX, Ouellette BF (2009) Pandora, a pathway and network discovery approach based on common biological evidence. Bioinformatics 26:529–535
38. Karp PD, Paley SM, Krummenacker M, Latendresse M, Dale JM, Lee TJ, Kaipa P, Gilham F, Spaulding A, Popescu L et al (2010) Pathway tools version 13.0: integrated software for pathway/genome informatics and systems biology. Brief Bioinform 11:40–79
39. Aziz RK, Bartels D, Best AA, DeJongh M, Disz T, Edwards RA, Formsma K, Gerdes S, Glass EM, Kubal M et al (2008) The RAST server: rapid annotations using subsystems technology. BMC Genomics 9:75
40. Paley SM, Karp PD (2006) The pathway tools cellular overview diagram and omics Viewer. Nucleic Acids Res 34:3771–3778

41. Yamada T, Letunic I, Okuda S, Kanehisa M, Bork P (2011) iPath2.0: interactive pathway explorer. Nucleic Acids Res 39:W412–W415
42. Croft D, O'Kelly G, Wu G, Haw R, Gillespie M, Matthews L, Caudy M, Garapati P, Gopinath G, Jassal B et al (2011) Reactome: a database of reactions, pathways and biological processes. Nucleic Acids Res 39:D691–D697
43. Junker BH, Klukas C, Schreiber F (2006) VANTED: a system for advanced data analysis and visualization in the context of biological networks. BMC Bioinformatics 7:109
44. Degtyarenko K, de Matos P, Ennis M, Hastings J, Zbinden M, McNaught A, Alcantara R, Darsow M, Guedj M, Ashburner M (2008) ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Res 36:D344–D350
45. Wang Y, Xiao J, Suzek TO, Zhang J, Wang J, Bryant SH (2009) PubChem: a public information system for analyzing bioactivities of small molecules. Nucleic Acids Res 37: W623–W633
46. Dudoit S, Yang YH, Callow MJ, Speed TP (2002) Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Stat Sinica 12:111–139
47. Levene H (1960) Robust tests for the equality of variance. In: Olkin I (ed) Contributions to probability and statistics. Stanford University Press, Palo Alto, CA, pp 278–292
48. Bewick V, Cheek L, Ball J (2004) Statistics review 9: one-way analysis of variance. Crit Care 8:130–136
49. Welch BL (1951) On the comparison of several mean values: an alternative approach. Biometrika 38:330–336
50. Games PA, Howell JF (1976) Pairwise multiple comparison procedures with unequal N's and/or variances: a Monte Carlo study. J Educ Stat 1:113–125
51. Martin JA, Wang Z (2011) Next-generation transcriptome assembly. Nat Rev Genet 12:671–682
52. Baart GJ, Martens DE (2012) Genome-scale metabolic models: reconstruction and analysis. Methods Mol Biol 799:107–126
53. Toya Y, Kono N, Arakawa K, Tomita M (2011) Metabolic flux analysis and visualization. J Proteome Res 10:3313–3323
54. Koonin EV (2005) Orthologs, paralogs, and evolutionary genomics. Annu Rev Genet 39:309–338
55. Altenhoff AM, Dessimoz C (2012) Inferring orthology and paralogy. Methods Mol Biol 855:259–279
56. Tipton K, Boyce S (2000) History of the enzyme nomenclature system. Bioinformatics 16:34–40
57. Karp PD, Riley M, Paley SM, Pellegrini-Toole A (2002) The MetaCyc database. Nucleic Acids Res 30:59–61
58. Karp PD (2004) Call for an enzyme genomics initiative. Genome Biol 5:401
59. Yang YH, Dudoit S, Luu P, Lin DM, Peng V, Ngai J, Speed TP (2002) Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. Nucleic Acids Res 30:e15
60. Hilton A, Armstrong RA (2006) Statnote 6: Post-hoc ANOVA tests. Microbiologist 7:34–36

INDEX

Hal S. Alper (ed.), *Systems Metabolic Engineering: Methods and Protocols*, Methods in Molecular Biology, vol. 985,
DOI 10.1007/978-1-62703-299-5, © Springer Science+Business Media, LLC 2013

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

Y