Adv Biochem Engin/Biotechnol (2012) 127: 27–70
DOI: 10.1007/10_2011_116

Published Online: 23 December 2011

Transcriptome Data Analysis for Cell Culture Processes

Marlene Castro-Melchor, Huong Le and Wei-Shou Hu

Abstract In the past decade, DNA microarrays have fundamentally changed the way we study complex biological systems. By measuring the expression levels of thousands of transcripts, the paradigm of studying organisms has shifted from focusing on the local phenomena of a few genes to surveying the whole genome. DNA microarrays are used in a variety of ways, from simple comparisons between two samples to more intricate time-series studies. With the large number of genes being studied, the dimensionality of the problem is inevitably high. The analysis of microarray data thus requires specific approaches. In the case of time-series microarray studies, data analysis is further complicated by the correlation between successive time points in a series.

In this review, we survey the methodologies used in the analysis of static and time-series microarray data, covering data pre-processing, identification of differentially expressed genes, profile pattern recognition, pathway analysis, and network reconstruction. When available, examples of their use in mammalian cell cultures are presented.

Keywords Alignment · Clustering · Differential analysis · DNA microarrays · Gene expression · Mammalian cells · Network reconstruction · Pathway analysis · Transcriptome · Time-series

Contents

(Authors Marlene Castro-Melchon, Huong Le and Wei-Shou Hu are contributed equally to this work).

M. Castro-Melchor · H. Le · W.-S. Hu (✉)
Department of Chemical Engineering and Materials Science,
University of Minnesota, 421 Washington Avenue SE,
Minneapolis, MN 55455-0132, USA
e-mail: wshu@umn.edu

1 Introduction

In the past decade, genome science has drastically changed our approaches to studying biosciences and broadened our ability to harness the potential of industrial organisms for technological applications. Importantly, genome-wide gene expression profiling using DNA microarrays has become widely employed in biotechnological research. Through DNA microarrays, we are able to look at the dynamics at the transcript level of the entire set of genes in order to explore the intricate relationships among the biochemical reactions, the signaling and regulation, the physiological events in the cells, and the global gene expression. In the next few years, we anticipate a greatly expanded reach of transcriptome analysis in cell culture research due to the dramatic advances in sequencing technology. Until recently, the application of transcriptome analysis in cell culture bioprocess has been rather limited because the genome sequence information available for the most commonly used cells, Chinese Hamster Ovary (CHO) cells, is not extensive. With the cost of DNA sequencing drastically reduced compared to even three years ago and the readily accessible sequencing services, one can expect that genome sequences for reference species will become available in the very near future. Furthermore, we can also expect that sequencing the genome of individual cell lines will become commonplace in a few years. Therefore, the affordability of high-throughput sequencing technology will push DNA microarrays to the forefront of cell culture bioprocess characterization, along with many routinely used quantitative tools such as HPLC and ELISA. However, unlike the conventional variables typically measured in a cell cultivation process, transcriptome data is unique in its high dimensionality: each time point of measurement yields up to

tens of thousands of transcript level data. In some ways, the examination of the data is like looking for patterns in a starry sky; the comparison of different datasets is as if comparing the skies in different seasons or on different days.

In this review, we summarize commonly used microarray platforms and experimental designs, and review methods used in differential expression analysis, profile pattern recognition, pathway analysis, and network reconstruction. In each section, an overview of the basic methodology is provided, followed by a sequence of specific modifications and associated software. Finally, several examples are presented in which the methodology has been successfully applied. When available, examples using antibody-producing recombinant cell lines are emphasized.

2 Platform Overview

Several microarray platforms are currently available, each of them offering certain advantages. As new platforms are introduced, a reduction in cost and an increase in flexibility have been observed. Microarray platforms are generally classified into two-dye or single-dye, referring to the number of fluorescently labeled samples applied to each chip.

2.1 Two-Dye Microarrays

Two-dye microarrays were first used by Schena et al. [1] to measure the expression level of 45 *Arabidopsis* genes, and were soon followed by studies at the genome-wide level in yeast [2]. Two-dye cDNA arrays are prepared by immobilizing long (>500 nucleotides [nt]) cDNA probes prepared by PCR amplification onto a glass slide. cDNA microarrays allow the direct comparison of genes in two samples, each labeled with a different fluorescent dye. The native intensities of the two dyes are indicative of the transcript levels in each sample. The probes can be designed against the genome sequence of the organism to minimize the segments which may cause cross-hybridization with transcripts from other genes. However, for mammalian cell applications, the large number of probes renders this approach very costly, as specific primers have to be designed for the amplification of specific segments of a sequence. Thus universal primers that amplify the entire cDNA region of an expressed sequence tag clone are more frequently used. However, they are prone to non-specific hybridization, especially for alternatively spliced transcripts. cDNA microarrays also suffer from imprecise control of the amount of DNA immobilized on the surface, making it difficult to compare the levels of different genes in the same sample.

With the much reduced cost in oligonucleotide synthesis, cDNA microarrays are now used less frequently. In the past few years, many synthetic oligo-DNA-based

microarrays have evolved to be suitable for use as either single-dye or two-dye arrays. One such platform that can be used as either single-dye or two-dye is that by Agilent [3]. Similar to cDNA microarrays, short (~60 nt) oligonucleotides synthesized in situ are printed onto a glass surface. As little as individual slides are available for unique custom designs. Multiplexing, that is, the availability of testing multiple samples in a single slide, is also available in Agilent's microarrays.

2.2 Single-Dye Microarrays

In contrast to two-dye arrays, single-dye arrays are designed to provide "absolute" measurement of the relative transcript level of each gene within a sample. With a "relative" measurement in two-dye arrays, a multiple sample comparison is cumbersome, requiring either a myriad of pairings of samples or the use of a common reference. With an absolute measurement and one sample for each array, even meta-analysis using hundreds of microarrays can be performed.

An example of a single-dye array is that by Affymetrix, Inc. [4], which uses a photolithographic process for printing probes. Gene expression is interrogated by probe sets, which consist of eight to eleven probe pairs. Each probe pair consists of two 25-mers, one being a perfect match, the other containing a mismatch at the 13th base pair. The photolithographic process, however, requires the creation of a set of masks for each array design (essentially four masks for each base position, thus each 25-mer will require 100 masks). The cost of generating a new set of masks limits the frequency of modifying or updating probes.

Probes on another single-dye platform, commercialized by Roche NimbleGen, Inc., are synthesized by photo-mediated chemistry using a proprietary Maskless Array Synthesizer [5]. The use of digital mirrors creates "virtual masks", allowing for flexible designs that can be easily modified. With their ability to control the area of probe to be very small, a very large number (in the millions) of probes can be placed on a single slide. This presents an advantage for large genomes, such as those of mammalian species. For smaller genomes or with a subset of genes, array multiplexing, i.e., using a single array for multiple samples, can be implemented. Furthermore, without the use of a mask, the cost of production is reduced. Making an array for only a small number of samples and frequent updating of probe design thus becomes affordable.

2.3 Other Platforms and Technologies

In addition to the glass-slide chip-based microarrays, other types of arrays have been developed. One such technology is Illumina's BeadArray, which uses three-micron silica beads that self-assemble in microwells with uniform spacing. In this

capture technology, each bead is covered with thousands of copies of specific oligonucleotides.

With the rapid advances in DNA sequencing technologies and the decrease in sequencing cost, transcriptomes can now be analyzed by direct cDNA sequencing. In RNA-Seq, a population of RNA is converted to a cDNA library, which is then fragmented and sequenced using high-throughput technologies [6]. The abundance level of a particular sequence fragment is indicative of the abundance level of the transcript from which it is derived. Unlike DNA microarrays which can be used only to probe the expression of genes represented on the arrays, RNA-Seq detects all RNA species, including novel RNAs and alternative transcripts. It can also identify transcript boundaries, and has a much wider dynamic range, over several orders of magnitude (>8,000 fold), as there is no saturation of highly expressed transcripts.

3 Static Studies vs Time-Series Studies

Although microarrays can be used to probe transcript profiles of a large array of genes in a cell sample, most applications involve the comparison of different cell samples, either the same cell line under different conditions or different cell lines. In other words, most studies involve two or more cell samples. The use of DNA microarrays in the study of cell culture processes can be categorized into static or dynamic (time-series) types according to how samples are taken and compared.

Static studies compare two samples to identify differences in gene expression between them. The samples may be different cells or tissues, such as when comparing cell lines of different levels of antibody production [7]. In other cases, different process variables or culture conditions might be under study. The following studies using NS0 cells include examples of the use of microarrays to assess the effect of cell density [8], to study cell proliferation in protein-free media [9], and to analyze the effect of hypoxic stress [10].

Cell culture process is intrinsically a time-evolving event, entailing various stages of culture, from early exponential and exponential phases followed by a transition to stationary phase. In most cases, the environmental conditions change over time, either due to the culture's self-evolution or due to process-imposed culture condition alterations such as temperature or pH shift. The gene expression profiles thus inevitably change with culture time. Static studies offer rich information on the difference in gene expression between two conditions or two cell populations but only as a snap-shot frozen at a point of a long process.

Time-series studies sample over different time points along the duration of the culture and aim at capturing the trends in gene expression changes originated by regulatory events and fluctuations in environmental conditions. Furthermore, the temporal information held in time-series microarray data also enables one to infer causality in gene regulatory networks. An aim of time-series data analysis is thus to identify genes which have different dynamic behaviors over time in the same

sample or to identify the same genes whose transcripts follow different time trends under different treatments [11]. Time-series studies are particularly relevant for cellular processes exhibiting periodic behaviors, such as cell cycle and circadian clock, as well as other intrinsically dynamic processes such as development and differentiation. Although this type of studies is abundant in yeast, *C. elegans*, and stem cells, fewer examples have been demonstrated in antibody-producing mammalian cells. In one example, gene expression time-profiles were compared between fed-batch processes yielding high and low titers using the same CHO cell line [12]. Dynamic regulation of transcription in a Human Embryonic Kidney (HEK) cell line in protein-free batch and fed-batch cultures was also unraveled [13]. In addition, time-series transcriptome data was explored to elucidate cellular mechanisms leading to an increase in productivity in CHO cells under sodium butyrate treatment and temperature shift [14].

In DNA microarray studies of mammalian cell cultures, the number of differentially expressed genes and the degree of their differential expression are often lower than typical changes observed in other systems such as in developmental processes or in microbial cultures [12]. Using a fold-change cut-off of 1.4–2.0, and a p-value cut-off of 0.05–0.1, it is common to identify much less than 10% of the genes as significant. This number often decreases sharply when the fold-change cutoff is raised above 2.0. For example, in studying the productivity of antibody-producing cell lines, a relative small number of genes are consistently different between high- and low-productivity clones [7]. These modest changes in gene expression thus require careful experimental design and subsequent data analysis. This situation contrasts with most cases found in bacteria undergoing changes in nutritional or other environmental conditions, and stem cells under directed differentiation in which often a large number of genes change their expression and many show large differences in gene expression levels.

4 Experimental Design

Microarray and RNA-Seq studies can provide a wealth of information. However, even with the decrease in cost in the past few years, they are still not bargain-price. The number of conditions to be tested and the number of samples for each condition have to be planned. When single-dye microarray platforms are used, there is no limitation to which comparisons among multiple samples are made. For two-dye arrays, however, the experimental design is crucial. With two-dye microarray platforms, one aims to measure the ratio of each gene's transcript level between two samples. When only two samples are involved, direct comparison is obtained using a single chip. With three samples, loop designs in which three arrays are used to obtain direct pair-wise comparison of the three sample pairs (1–2, 2–3, 3–1) can be applied [15]. An alternative, often referred to as reference design, is to hybridize each of the three samples to a common reference, and obtain indirect comparisons for each sample pair in the experiment. An often-used

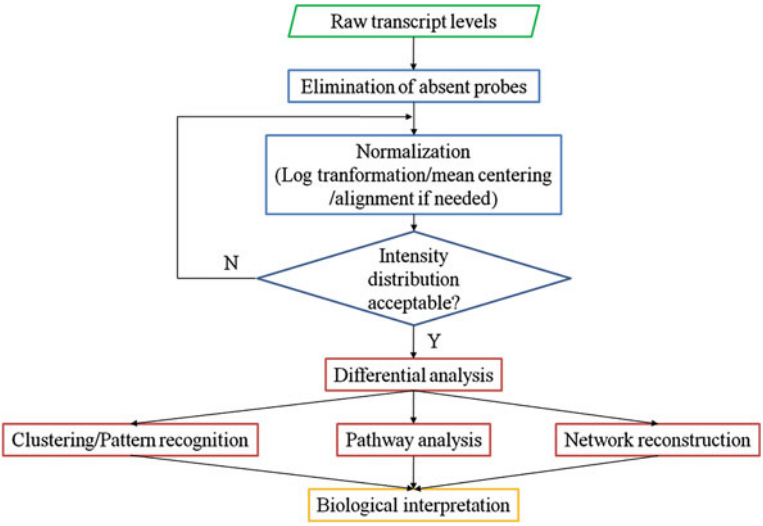


Fig. 1 General analysis process of microarray data. Raw expression data are often filtered to eliminate absent probes. The filtered data is subsequently normalized and processed using different methods if necessary. Genes exhibiting differential expression are identified using statistical tools. In addition to clustering, further analysis can be performed in a pathway/network context to finally interpret the biological meaning of differential expression

reference is a pool of RNA, either from all samples, to ensure that the transcripts of all genes on the array are present, or from sample(s) external to the experiment. An internal reference, for instance the first sample, can also be used to directly compare some of the pairs (1–2 and 1–3) and infer comparisons for the others (in this case 2–3). The amount of available reference sample might limit the number of arrays that can be done.

Time-series microarray studies present additional experimental design challenges. Frequently, the comparison is not only among data from different time points within the same treatment but also among series under different treatments. The number of samples to be collected and their distribution in time will define the ability of the experiment to capture the gene expression dynamics. The sample collection frequency should be high enough to capture the dynamics of genes with periodic behaviors or propensity for sudden changes in expression. This, however, might result in a very large number of samples, which is not always feasible due to cost or the amount of work involved [16]. If critical changes are suspected between the time points originally analyzed, additional microarrays can be performed. This is possible if samples were collected at intermediate time points. Another possibility is to fill these gaps using quantitative PCR measurements of transcripts of the target genes.

The general steps in analyzing gene expression data from microarrays are shown as a flowchart in Fig. 1. First, raw data is filtered to eliminate absent probes

using intensity and/or detection *p*-value cutoffs. Filtered data is further normalized to generate a baseline for comparison across samples. Time alignment, log transformation, and scaling can be performed if necessary. Once the data has been properly processed, genes exhibiting differential expression can be identified using multiple statistical approaches. These significant genes are often further analyzed in a pathway/network context or by using clustering tools to infer the biological meanings of differential expression.

5 Data Pre-Processing

5.1 Normalization, Transformation, and Scaling

Gene expression levels measured using DNA microarrays are subject to a number of systematic biases, and hence should be globally adjusted (or normalized) to attain a common basis for all the microarrays to be compared. These variations in gene expression measures are often the result of differences in starting amounts of RNA, labeling, hybridization, and scanning efficiency [17]. Normalization is thus a necessary step regardless of the platform, or whether the experiment involves static or time-series samples. Different normalization methods (based on different sets of assumptions) often give different quantifications. Most normalization methods assume that the microarray contains a large and random set of genes. Furthermore, the number of differentially expressed genes is considered to be relatively small compared to the total number of genes present on the array. As a result, this differential expression does not affect the overall distribution of gene expression levels in each sample.

Linear and quantile normalization are most commonly used in microarray data processing. Linear normalization is often applied when gene expression measures in all arrays have similar distributions but different median values. Given the assumption that equal amounts of RNA are used in each sample, a normalization factor is calculated as the ratio of the median gene expression levels in two samples [17]. All gene expression measures are subsequently scaled using this factor such that these two samples have the same median gene expression level after normalization. A target median value can also be defined to linearly scale multiple samples. Linear normalization is thus conceptually simple, yet applicable to most cases in which the assumptions stated above are satisfied. However, possible lack of linearity between fluorescence intensity and the amount of DNA or RNA hybridized could introduce errors when linear normalization is applied.

Quantile normalization, on the other hand, assumes that all samples have the same gene expression level distribution [18]. Gene expression measures are adjusted such that each sample follows the same distribution, which is assumed to be the average distribution of all samples. This normalization method is frequently used to correct the gene expression level distribution in single-dye and two-dye

arrays when genomic DNA is used in one channel. Sometimes, a drastic change in cell physiology may occur, causing a major shift in gene expression profiles. In such cases, the use of quantile normalization might not be appropriate. For example, as stem cells differentiate or cells enter different phases of growth, their transcriptional responses or cellular RNA composition may change drastically. Large variations in cellular RNA composition among samples violate the assumption that all samples have the same gene expression level distribution.

It is important to note that, in most experimental protocols, the amount of total RNA (in the case of prokaryotic samples) or poly(A)-tailed transcripts (in the case of eukaryotic samples) applied to each array is kept equal, and thus normalization methods only adjust the data to equal quantities of RNA. However, the RNA content per cell does not always remains constant under different conditions. Fast-growing cells have far more RNA than cells in the stationary phase, and thus total RNA content per cell varies. It is therefore important to know whether differential expression calls are based on per cell or per unit amount of RNA.

After normalization, the data is usually log-transformed. The variance, which is inherently large in microarray data, is reduced in log-transformed data. Normalized gene expression values can also be scaled to a mean or median value of zero. This is equivalent to centering the gene expression level distribution over zero (mean- or median-centering). Additionally, a standard deviation of one can be achieved using z-transformation. These data pre-processing steps can be performed using several software including Expressionist, GeneSpring, and R packages such as *affy*, *limma*, *beadarray* and *oligo*. Although data normalization, transformation, and scaling have become routine, these steps remain vital to all subsequent stages along the analysis pipeline of gene expression data.

5.2 Time Alignment

When comparing time-series experiments, it is important to control the starting cell population in different treatments to be identical, or at least as similar as possible. Under some conditions, variability is difficult to eliminate, resulting in somewhat different kinetic profiles even among biological replicate cultures. When applying microarrays to time-series studies, the aim is to identify the genes whose transcript dynamics change beyond the fluctuations in biological replicate cultures, and where the change can be attributed to experimental treatment. In assessing the similarity or difference between two cultures under different treatments, a direct comparison of time profiles is an obvious first approach. This is sound in the cases where the trends of growth and other growth-related variables (such as chemical profiles) are mostly identical. Often growth and other culture indicators reveal a difference, strongly hinting that the identical time points in two cultures may not correspond to identical "culture stage". In other words, the time frame of one culture has shifted from the reference time frame of the other culture. Direct comparison of time profiles of gene expression may give rise to many

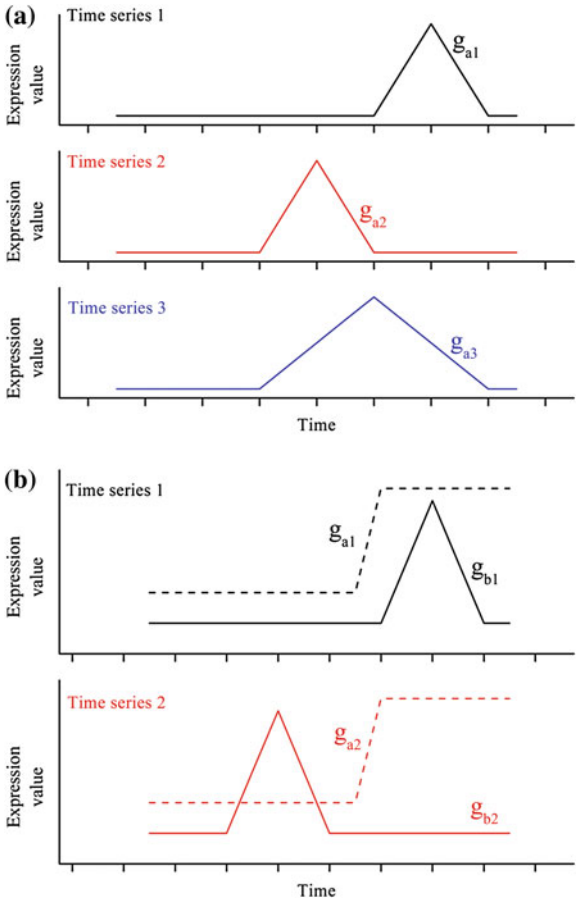


Fig. 2 Possible forms of time asynchronization in different series. **a** Expression profile of gene g_a in three different series. The expression profile in series 2, g_{a2}, shows a frame shift with respect to series 1, g_{a1}. The expression profile in series 3, g_{a3}, shows an expansion with respect to g_{a1}. These types of asynchronization can be adjusted globally. **b** Expression profiles of two genes, g_a and g_b, in two different series. Gene g_a displays the same expression profile in the two series. The peak observed in the expression profile of gene g_b in series 1 appears earlier in series 2. This time flip often requires local adjustment

falsely identified genes with different kinetic behaviors. Time alignment aims to identify potential time misalignments and correct them.

The change in time dynamics could be global, i.e., all the transcripts change their temporal profiles similarly. This change may also be segmented and local, i.e., only some sets of genes change coordinatedly apart from the rest of the genes or different sets of genes which change their dynamics differently. Such asynchronous behaviors need to be dealt with using some form of time alignment. Asynchronization between transcriptome time profiles appears in multiple forms, which can be largely divided into four types: frame shift, elastic compression or expansion, and time flip [19]. Frame shift occurs when one of the series experiences a lag phase with respect to the others. If the growth rate differs significantly between the series, their gene expression profiles may display elastic compression or expansion. Examples of frame shift and expansion are shown in Fig. 2a. These types of asynchronization are often adjusted globally. In addition, changes in a few subsets of genes can result in a flip in time order between different subsets of genes

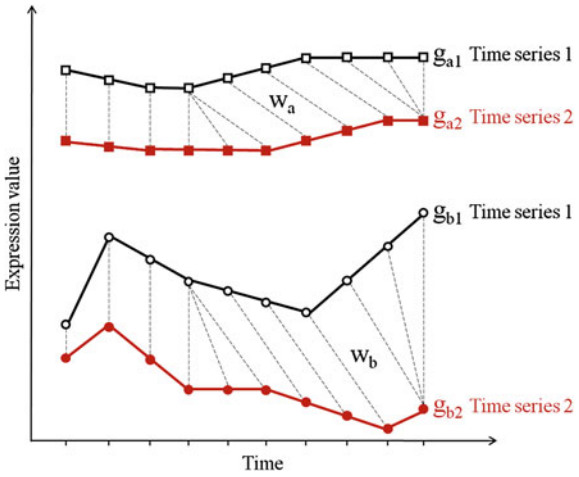


Fig. 3 Alignment of gene expression profiles using DTW. Two genes (g_a and g_b) are shown in two different series. For each gene, discrete time points in series 1 are mapped to those in series 2. The weighting factors (w_a and w_b) indicate the contribution of each gene to the final global adjustment. Gene g_a with a relatively flat profile is given a lower weight ($w_a < w_b$). The same alignment is imposed on both genes

(Fig. 2b). This time flip suggests the existence of multiple biological clocks controlling varied cellular processes in the experimental system and thus requires local alignment. As a result, when multiple treatments are being compared, gene expression data sets should be examined and, if necessary, properly aligned before subsequent analyses can be performed.

Conceptually, aligning time-series microarray data entails matching two patterns by locally compressing, expanding, or translating one with respect to the other such that their similar characteristics are aligned without altering the ordering of each sample. This can be performed on either the continuous representation of each series or the discrete values of gene expression. The alignment between time series can be achieved at a global level or at a local level to allow different subsets of genes to follow varied biological clocks.

An example of global alignment is the B-spline-based alignment method, which presents each gene expression profile as a spline curve of multiple low-degree polynomials [20]. To align different time series, one of the series is chosen as the reference, and the time points of the other series are mapped to the reference series by stretching and shifting the continuous representation of gene profiles. This method is particularly suited for long time series (e.g., $\geq$10 time points) [21]. The use of B-splines for alignment was demonstrated by aligning three yeast time series that begin in different phases and occur in different time scales [20].

A second example of global alignment, dynamic time warping (DTW), involves non-linear mapping between discrete time points of two series along the time dimension such that the distance between them is minimized [22]. In the case of transcriptome time series, the overall distance between the two series is computed as the weighted sum of distances contributed by all genes. The use of a weighting factor for each gene allows higher contribution to the overall distance measure to be given to genes with consistent expression profiles across two treatments, or to genes important to the biological activities being considered. In Fig. 3, the

algorithm is exemplified with two genes (g_a and g_b) in two different series. The weighting factors (w_a and w_b) indicate the contribution of each gene to the final adjustment. g_a with a less dynamic profile thus has a lower weighting factor ($w_a < w_b$). In addition to alignment of transcriptome data, DTW has also been used to synchronize offline and online data of batch processes [23, 24].

Global alignment algorithms assume that all genes share the same alignment, that is, that all genes were affected in the same manner. The existence of multiple biological clocks within the same cell, however, can result in sets of genes being affected independently. In other words, genes in one set correspond to genes that follow a particular biological clock, sharing the same alignment, but they need to be warped separately from the rest of the genes. Recently, Smyth et al. [25] have proposed an algorithm capable of identifying sets of genes that present similar alignments when aligned independently. The resulting sets include genes that follow similar warpings, even though their expression profiles might be very different.

6 Identification of Differentially Expressed Genes

After transcriptome data has been pre-processed, a number of statistical approaches can be used to identify differentially expressed genes. Commonly used analytical methods for static transcriptome data include *t*-tests, ANalysis Of VAriance (ANOVA), Significance Analysis of Microarray data (SAM), and LInear Models for MicroArray data (*limma*). These methods are not all directly applicable for dynamic studies involving a chronological set of samples collected over time since a change in the time order will result in a different statistical inference [26]. Recently several methods based on regression, ANOVA, and Bayesian models have been adapted to handle time-series microarray data. In addition, a distance calculation approach has also been proposed for identification of kinetically differentially expressed genes.

6.1 Statistical Analysis of Gene Expression Data

The estimates of gene expression levels provided by microarray data are generally prone to two types of errors—systematic and random errors. Systematic error resulting from several factors such as RNA concentration measurement or dye-labeling efficiency can give rise to a systematic bias in the expression level estimates of all genes on the same array. This bias is often corrected using one of the normalization methods presented in the previous section on data pre-processing. Random error in the measurement of gene expression levels arises from random fluctuations in other steps, for instance array scanning. Inferential statistics is used to ensure that the observed change in gene expression did not occur by random chance.

Inferential statistics is applied to microarray data by invoking a null hypothesis. The null hypothesis holds true when all samples have the same average expression value for the gene of interest. Conversely, if the gene is expressed at a different level in at least one sample, the alternative hypothesis becomes valid. In order to assess the validity of either hypothesis, a test statistic is often estimated as the ratio between the change in a gene's expression values among samples and the variability in those measurements. Furthermore, a p-value computed using this test statistic is compared to an acceptable significance level α. The smaller the p-value is compared to α, the stronger the evidence is against the null hypothesis, and in support of the gene being differentially expressed in at least one sample.

In a typical microarray experiment, tens of thousands of genes are tested simultaneously, and a large number of them are likely to be identified as differentially expressed. Even with a small p-value, such as 0.01 that is normally considered to be rather stringent, a significant number of those genes identified as differentially expressed might be by random chance. For example, if 1,000 genes out of 10,000 in total are identified as differentially expressed, each with a p-value <0.05, then 500 of these 1,000 genes might have been identified by chance. One way to control the potentially high error rate is to set each gene's p-value to an n-fold lower significance level, α/n, where n is the total number of genes. This is often referred to as the Bonferroni correction for the family-wise error rate—(FWER) [27]. However, this correction imposes an extremely stringent criterion. In the previous example, the p-value will have to be set at less than 0.000005. This would likely result in failure to identify the majority of genes that are indeed differentially expressed. An alternative is to control the number of false positives among the number of genes declared as differentially expressed rather than the total number of genes. This statistic, referred to as false discovery rate (FDR), is less stringent than the FWER and thus offers more power than the FWER to detect differential expression [28]. Therefore, in multiple hypothesis testing, FDR is often used in place of p-value.

6.1.1 Statistical Analysis of Static Gene Expression Data

A variety of methods are available for hypothesis testing. A t-test is often used when only two samples are compared for differential gene expression. When three or more samples are involved, ANOVA is recommended to avoid performing multiple t-tests, which will most likely result in an increased false-positive rate. Both methods assume the expression levels of a gene in different samples follow a normal distribution. When this assumption does not hold true, non-parametric tests including the Wilcoxon rank-sum test and permutation-based test are often the methods of choice.

t-Test

t-Tests are considered the simplest statistical methods to identify differentially expressed genes. A t-statistic is calculated as the ratio between the difference in

gene expression levels of two samples and the pooled variance. Furthermore, a degree of freedom is calculated from the sample sizes—with more penalties if the two samples have unequal variances (Welch's *t*-test), and no penalties if the assumption of equal variances holds true (Student's *t*-test). A *p*-value, which can be obtained using the *t*-statistic and the degree of freedom, is compared to a pre-defined significance level α to detect differential expression. *t*-Tests can be easily performed in Microsoft Excel, several R packages, and a variety of software including Spotfire and Expressionist.

Gene expression responses during metabolic shift in a hybridoma cell culture have been investigated using the Student's *t*-test on cDNA microarray data [29]. 123 probes were identified as changing their expression levels (fold-change $\geq$1.4 and *p*-value $\leq$0.1) when the cells shifted to a lactate consumption state. Another example involves the survey of global gene expression changes in a recombinant antibody-producing CHO cell line and a mouse hybridoma cell line under sodium butyrate treatment [30]. Using a fold-change cutoff of 1.4 and a *p*-value cutoff of 0.05, most transcripts were found to be expressed at similar levels in both cell lines, indicating that the transcriptional responses under exposure to sodium butyrate are rather conserved.

Analysis of Variance

When more than two samples are involved, single-factor ANOVA is often used. The overall variance in gene expression among different samples is partitioned into separate sources of variations. The total variation, as evaluated by sum of squares (SS_{Total}), arises from two sources—the actual differential expression among these samples ($SS_{Treatment}$) and the random error (SS_{Error}). The means sum of squares (MS) for treatment and error can be estimated by dividing each SS by the corresponding degree of freedom. The quotient of these two MSs is taken as the *F*-statistic, which further provides a *p*-value for inference of differential expression.

When the experiment involves several factors (or variables; in ANOVA they are referred to as "factors"), and one wishes to segregate the effects of those factors, multiple-factor ANOVA is used. Based on the same working principles described above, multiple-factor ANOVA also partitions the total variation into different sources—the actual effect of each experimental factor, their interactions, and the random error. A *p*-value for each term can be derived similarly, and whether these factors significantly affect the change in gene expression can thus be concluded. Both single-factor and multiple-factor ANOVA can be performed easily using Microsoft Excel, as well as several R packages.

Variation in gene expression within and between two populations of the genus *Fundulus* was uncovered using ANOVA on $\log_2$-normalized microarray data on 907 genes [31]. 161 genes were differentially expressed among individuals within a population, whereas only 15 genes differed between populations, suggesting that substantial natural variation exists in gene expression. A linear ANOVA model was also fitted to the expression levels of more than 3,000 genes expressed during

embryonic development of six *Drosophila* species [32]. More than 80% of genes best fit to models incorporating stabilizing selection, and maximal similarity is observed during mid-embryogenesis rather than early or late stages of development. This result thus supports the developmental hourglass model, and the theory that natural selection acts to conserve gene expression patterns during the phylotypic period.

Significance Analysis of Microarray (SAM)

Similar to *t*-tests, SAM also calculates a "relative difference" (d), which resembles the ratio between difference in average gene expression values and the pooled variance in two treatments for each gene [33]. The expression levels in all replicated samples of these two treatments are then permuted, and an average "relative difference" over these permutations (d_E) is estimated. For the majority of genes, which are assumed not to be differentially expressed, the average difference obtained from permutation (d_E) is largely the same as the observed one (d). If the discrepancy between d_E and d exceeds a threshold, the gene is considered differentially expressed. In order to calculate the FDR for each gene, two horizontal cutoffs are defined—one as the smallest observed difference of up-regulated genes, and the other as the least negative of down-regulated genes. The average number of genes with d_E exceeding these cutoffs in all permutations can be considered as the number of false positives, and is used to assess FDR. A convenient Microsoft Excel add-in for SAM is available, and the packages *siggenes* and *samr* in R are also publicly accessible.

The advantage of SAM over other statistical methods was demonstrated when examining the transcriptional responses of human lymphoblastoid cells under irradiation [33]. 34 genes were identified as significant at an FDR of 12% using SAM compared to more than 60% using other methods. In another example, SAM was used to identify about 400 genes contributing to the impaired differentiation capacity of murine neural stem cells (NSCs) defective in p53 and PTEN genes [34]. The majority of genes involved in cell cycle regulation were also found to be significantly down-regulated when HeLa cells were transfected with siRNA against PHF8, an H4K20me1 demethylase [35].

Linear Models of MicroArray data

In this approach, a linear hierarchical model with arbitrary coefficients and contrasts across multiple samples for each gene is developed [36, 37]. Furthermore, marginal distributions of the observed statistics are used to estimate the hyperparameters under consistent and closed forms. In addition, the ordinary *t*-statistic can be replaced by a moderated one, which implicitly results in shrinkage of all gene-wise variances into a common value. This moderate *t*-statistic follows a *t*-distribution with augmented degrees of freedom, and thus can be extended for multiple-sample comparisons by using the corresponding *F*-statistics. The R package *limma* is publicly available.

Transcriptional responses upon restoration of p53 in adenocarcinomas were revealed using *limma* [38]. p53-restored samples were shown to cluster with adenomas rather than carcinomas, suggesting that adenocarcinoma cells can be specifically removed from the tumors. *limma* was also used to compare gene expression signatures between cultured thymic epithelial cells (TECs) and multipotent hair follicle (HF) stem cells [39]. 119 genes were identified as being differentially expressed between these two samples with a fold-change cut-off greater than four and a p-value less than 0.001.

6.1.2 Statistical Analysis of Dynamic Gene Expression Data

Time-series transcriptome data offer a great advantage when exploring transcription as a dynamic process, yet their analysis is more complicated than analyzing multiple samples unrelated in time. Transcriptional responses at a certain time point often carry information about cellular behaviors in previous stages. Thus samples within a series are mutually dependent, and should not be analyzed using traditional statistical approaches. Rather, methods taking this interdependency into consideration such as Extraction of Differential Gene Expression (EDGE), Microarray Significant Profiles (maSigPro), ANalysis Of Variance–Simultaneous Component Analysis (ANOVA-SCA), and multivariate Bayesian models are more suitable. The number of time points in each series, the number of series, and the availability of replicates will guide the selection of algorithm to use in data analysis. This analysis can become even more challenging if the sampling frequency is not uniform across multiple series.

Extraction of Differential Gene Expression (EDGE)

In EDGE, differential analysis is also approached as a hypothesis-testing problem. The null hypothesis is that a gene's expression does not change both over time within a single treatment and across multiple treatments [26, 40]. The expression profile of each gene is modeled using a p-dimensional basis, usually a pth-order polynomial, or a natural cubic spline function. The parameters of these functions are then estimated by minimizing the sum of squared errors (SSE) between the model-fitted expression values and the corresponding actual ones. The parameterization of gene expression profiles allows the hypothesis testing to be performed by comparison of the fitted parameters. As such, an F-statistic is calculated for each gene to reflect the relative difference in SSE of the model-fitted gene expression profiles under the null and the alternative hypotheses, respectively. This statistic is used alongside a null distribution generated using a resampling method to estimate a q-value, which accounts for the FDR incurred in multiple hypothesis testing [41].

The open-source software EDGE [42] has facilitated the use of this methodology in analyzing time-course gene expression data. Differential expression can

be surveyed along the time axis within each treatment or across multiple treatments. EDGE was used to define the transcriptomic signatures of aging in several tissues in *Drosophila melanogaster* [43]. In a mouse model, a complex transcriptional hierarchy comprising more than one thousand genes regulated during endocrine differentiation was also identified using EDGE [44].

Microarray Significant Profiles (maSigPro)

Microarray Significant Profiles, maSigPro [45], uses a two-step regression approach to identify differentially expressed genes in time-series microarray data. Single or multiple time series can be analyzed, with multiple time series being analyzed directly instead of performing multiple pair-wise analyses. This methodology not only detects kinetically differentially expressed genes, but also uncovers changes in gene expression trends. In the first step of gene selection, expression data is fitted using a global regression model which considers all experimental variables and their interactions. If there are n groups, $(n - 1)$ dummy variables are defined. Each dummy variable allows the distinction between each group and the reference group. Furthermore, an ANOVA table is generated for each gene. If the gene shows differences between any group and the reference group, the regression coefficients will be statistically significant as determined by an F-statistic and its associated p-value. In the second step of variable selection, the best model for each gene is obtained using a stepwise regression approach. The variables that best fit the data represent the time effects and their interactions with the dummy variables. For finding those genes with significant differences in group x with respect to the reference series, the genes with significant coefficient for the dummy variable $(x - 1)$ are selected.

The package maSigPro is available in R and includes several tools for result visualization. In addition, it is part of the oneChannelGUI package [46], which provides a graphical interface for the analysis of Affymetrix microarrays, and was included in the popular software Gene Expression Pattern Analysis Suite (GEPAS) [47]. An extension of maSigPro, maSigFun [48], is used to fit regression models for genes with the same functional class and for the functional assessment of time-course microarray data. maSigPro has also been implemented in Corra [49], an R package devoted to the analysis of LC–MS-based proteomics. maSigPro has been used to analyze data from intrinsically dynamic processes such as spatial differentiation in fungi [50], and plant development [51–53], as well as periodic responses such as the rhythmically expressed genes in mouse distal colon [54].

ANOVA-SCA

ANOVA-SCA (or ASCA for short) is considered a combination of a statistical method (ANalysis Of VAriance, ANOVA) and a dimensionality reduction approach (Simultaneous Component Analysis, SCA) [55–57]. ANOVA-SCA is

particularly useful when two or more quantitative variables are involved, such as time and dose. In the first step, an ANOVA model is applied for each gene expression measure to separate the variability caused by these two different variables. The model parameters obtained for all genes under each experimental condition are subsequently organized into a matrix form. The second step involves applying principal component analysis simultaneously on all matrices obtained under all experimental conditions. A number of constraints can be further imposed such that the resulting matrices are mutually independent. Such constraints on orthogonality enable the ASCA model parameters to be estimated independently by solving a simple least-squares optimization problem. Statistical significance of these experimental variables and their interactions can be further inferred using a permutation approach [58]. In particular, all experimental conditions are permutated to obtain a no-effect distribution, thus providing a baseline to conclude whether the observed effect is indeed significant.

One of the earliest applications of ASCA was for analyzing a metabolomics experiment in which the effects of time and vitamin C dose on the NMR spectra of guinea pig urine samples were delineated [55]. Individual variations caused by time and doxorubicin dose on metabolite mass spectrometry profiles were also uncovered using ASCA in a toxicology study on rats [59]. Given the intrinsic generalizability of ASCA, it is not surprising to find this approach extended into discovery of kinetically differentially expressed genes [60]. Two statistics—SPE (Squared Prediction Error) and leverage—were proposed to evaluate the goodness of fit of the ASCA model, and the degree of agreement with which a gene profile follows the main expression patterns, respectively. This adapted version of the original algorithm, *ASCA-genes*, has been implemented in the R language. Furthermore, *ASCA-fun* was devised to perform functional analysis on time-series microarray data [48]. In this method, genes ranked according to their correlation to the principal time components identified by ASCA were used to assess functional enrichment in the dataset following Gene Set Analysis (GSA) procedures.

Bayesian Approaches

A multivariate empirical Bayes model was applied to time-series microarray data by Tai and Speed [11]. The algorithm, implemented in the R package *timecourse*, however, requires replicates of the full time-series. This algorithm calculates multivariate versions of the log-odds, or *B*-statistic (*MB*-statistic), and the Hotelling statistic ($\tilde{T}^2$). When the numbers of replicates are the same for all genes, the *MB*-statistic is equivalent to the $\tilde{T}^2$-statistic. The algorithm can be used in one-treatment problems and multi-treatment problems. Although this method ranks the genes, it does not provide a significance cutoff.

A fully Bayesian approach for microarray analysis was implemented in clustering [61] and later for the analysis of time series [62]. This fully Bayesian approach can handle short series, non-uniform sampling and missing data and does take into consideration the temporal structure of the time series. Gene expression

profiles are modeled with Legendre or Fourier polynomials, and the coefficients and the degrees of these polynomials are estimated using a Bayesian approach. The differentially expressed genes identified in this Bayesian multiple-testing procedure are ranked, and their expression profiles are estimated. This estimation allows the visualization of each gene expression profile as a single smooth curve.

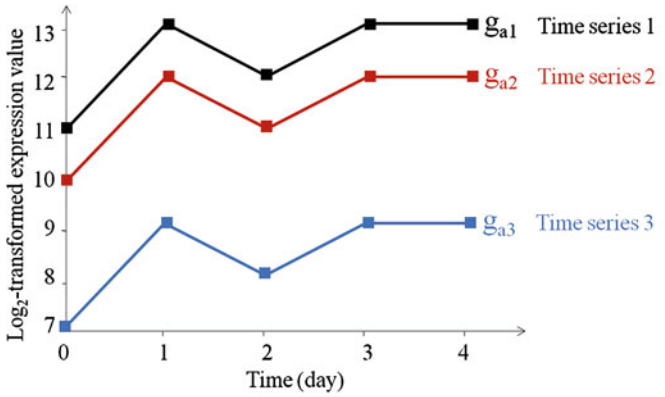
The fully Bayesian approach was demonstrated when analyzing the time series obtained by stimulating human breast cancer cells with estradiol after different time periods. The algorithm is implemented in the Bayesian user-friendly software for Analyzing Time Series (BATS) [63], a graphic user interface written in Matlab. The BATS package requires 5–6 time points and replicates are recommended but not required. At the moment, however, BATS can only handle one treatment time series. Its extension to multiple time series is under development.

6.2 Calculation of Distances Between Gene Expression Profiles

Just as in the calculation of the geometrical distance between any two vectors, a distance value can be computed to quantitatively describe the difference between two expression profiles of the same gene. By condensing all distance measures between the corresponding time points, the comparison of these two profiles is reduced into a single number. Two frequently used metrics are Euclidean distance and Pearson's correlation (Fig. 4). Euclidean distance, also known as $L-2$ *norm*, assesses the absolute difference between two time profiles. As a result, genes with the highest Euclidean distance between two treatments are often the ones with high expression levels, and are most likely to be identified as differentially expressed despite having similar expression trends in these treatments. Gene expression data can then be mean-centered or z-transformed to alleviate the dominance of these high-abundance transcripts. On the other hand, Pearson's correlation quantifies the overall similarity between the two trends regardless of the absolute values of gene expression. Small fluctuations in gene expression between low-abundance transcripts can thus be manifested as being markedly different since only expression trend is considered.

The choice of distance metric therefore depends on the question being asked. If the absolute values of expression measures are critical, the Euclidean metric is often preferred. Alternatively, the Pearson's correlation coefficient is a more suitable similarity measure if the overall trend of expression is pertinent to the analysis. A combination of both metrics is therefore recommended to integrate the differences in absolute expression magnitude and expression trend.

Following selection of a proper metric and distance calculation, a distribution of this representative difference can be plotted, and a threshold is often set to declare differential expression. Genes having distance measures between their expression profiles in two treatments above a certain threshold are considered to be differentially expressed. Manual inspection of gene expression profiles is often recommended to confirm the differential expression. In addition, if both treatments



Euclidean distance

$$Eucl(g_{a1}, g_{a2}) = \sqrt{\sum_{i=0}^{4}(g_{a1,i} - g_{a2,i})^2} = \sqrt{(11-10)^2 + (13-12)^2 + (12-11)^2 + (13-12)^2 + (13-12)^2} = 2.2$$

$$Eucl(g_{a1}, g_{a3}) = \sqrt{\sum_{i=0}^{4}(g_{a1,i} - g_{a3,i})^2} = \sqrt{(11-7)^2 + (13-9)^2 + (12-8)^2 + (13-9)^2 + (13-9)^2} = 8.9$$

Pearson correlation

$$Corr(g_{a1}, g_{a2}) = \frac{\sum_{i=0}^{4}(g_{a1,i} - \overline{g_{a1}})(g_{a2,i} - \overline{g_{a2}})}{\sqrt{\sum_{i=0}^{4}(g_{a1,i} - \overline{g_{a1}})^2 (g_{a2,i} - \overline{g_{a2}})^2}} = 1 \rightarrow \text{Pearson distance } (g_{a1}, g_{a2}) = 0$$

$$Corr(g_{a1}, g_{a3}) = \frac{\sum_{i=0}^{4}(g_{a1,i} - \overline{g_{a1}})(g_{a3,i} - \overline{g_{a3}})}{\sqrt{\sum_{i=0}^{4}(g_{a1,i} - \overline{g_{a1}})^2 (g_{a3,i} - \overline{g_{a3}})^2}} = 1 \rightarrow \text{Pearson distance } (g_{a1}, g_{a3}) = 0$$

Fig. 4 Calculation of distance between the expression profiles of a gene in two series. The distance between the expression profiles of gene g_a in three series can be measured using different metrics. The Euclidean metric quantifies the absolute geometric distance between the profiles, whereas the Pearson metric evaluates the correlation of trends in expression. Thus even though the Euclidean distance of g_a between series 1 and series 3 (Eucl (g_{a1}, g_{a3})) is much higher than that between series 1 and series 2 (Eucl (g_{a1}, g_{a2})), their Pearson correlations (Corr(g_{a1}, g_{a2}) and Corr(g_{a1}, g_{a3})) are indeed the same

are replicated, a statistic can be derived by permuting replicated samples between the two treatments. An average distance over all permutations is calculated, and compared to the actual distance to infer a statistical significance level. However, optimizing the difference threshold between the average and the actual distance can be indeed challenging.

This approach was used in a number of studies conducted in *Streptomyces coelicolor*. Genes involved in regulatory circuits related to antibiotic production were identified using Euclidean distance as criterion for differential expression [19]. Euclidean distance was also used in conjunction with principal component analysis (PCA) to reveal genes kinetically perturbed when the *Streptomyces coelicolor* sigma-like protein AfsS was disrupted [64]. In a recent study, more than 900 genes were identified as differentially expressed in an antibody-producing CHO cell line between the butyrate-treated 33°C culture and the non-treated culture [14].

7 Profile Pattern Recognition

Microarray data, with their large size and high dimensionality, are inherently complex. Compared to the number of genes (i.e., dimensionality), the number of samples is almost always small, making it difficult to find an answer to the question being asked. Often, an objective in a microarray experiment is to identify genes with a certain profile or pattern. Sometimes, however, which patterns are present in the data are not even known. In order to identify patterns that exist in the data, two types of techniques can be used: unsupervised and supervised algorithms.

7.1 Unsupervised Classification Methods

Unsupervised pattern recognition consists of organizing data based on the properties of the data themselves without reference to additional information [65]. Mathematical algorithms determine the search for natural patterns existing in the data [66]. The goal of unsupervised pattern recognition is to identify small subsets of genes that display similar expression patterns [67]. Instead of clustering genes, clustering samples based on their expression profiles can also be a goal in clustering analysis. In this case samples with similar expression profiles might help identifying groups, or labels, that can be given to those samples.

Although the term unsupervised pattern recognition is commonly used as a synonym for clustering, it actually encompasses other techniques, such as non-negative matrix factorization (NMF) and principal component analysis (PCA).

7.1.1 Dimensionality Reduction Techniques

Because microarray data is often obtained from only a small number of samples and entails thousands of genes, dimensionality reduction can be helpful for visualization, clustering, and classification. When transcriptome data is represented as an n by m matrix, in which n is the number of genes and m the number of samples ($n \gg m$), dimensionality reduction techniques can be used to identify a smaller number of principal gene expression patterns k (Fig. 5). This can be done by factorizing the original gene expression matrix (A) into two sub-matrices: one containing eigenarrays (W) and the other containing k eigengenes (H). The expression level of each gene in these m samples can be represented as a linear combination of the k eigengenes. Similarly, the overall expression pattern in each sample can be represented as a linear combination of the k eigenarrays.

In PCA, the data is transformed into a new set of variables called principal components (PCs). The principal components are uncorrelated, and, furthermore, they are ranked so that the first PCs contain most of the variation present in all of

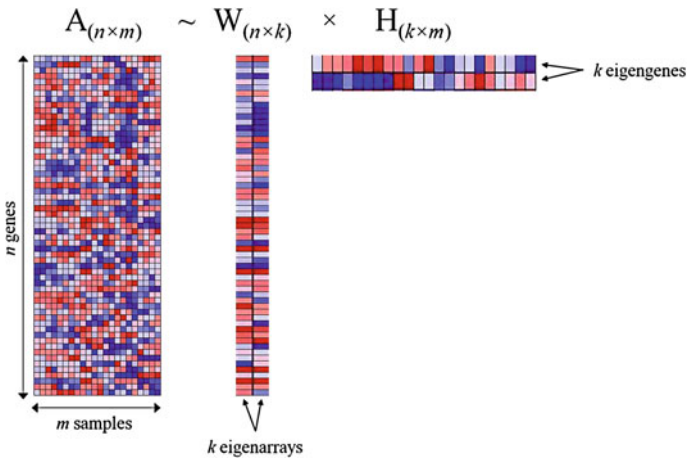


Fig. 5 Matrix factorization in dimensionality reduction techniques: NMF and PCA. Microarray data is organized into a matrix (A) with each row representing the expression levels of a gene in *m* samples. This original matrix can be decomposed into two sub-matrices: one containing *k* eigenarrays (W), and the other containing *k* eigengenes (H). The expression levels of each gene in these *m* samples can be represented as a weighted combination of the *k* eigengenes. Similarly, the overall gene expression pattern in each sample can be represented as a weighted combination of *k* eigenarrays

the original variables [68]. Since the first few PCs capture most of the variation in the original data, it is customary to use only the first few PCs [69]. When the data are projected along the first few PCs (most commonly the first two or three), in many cases it is possible to identify groups.

In PCA, the gene expression values can be reconstructed by a weighted sum of the eigengenes; however, there is no restriction on the sign of the weights. This can cause some variability due to cancellations, if eigengenes with both negative and positive weights are added. In a similar technique, NMF, the coefficients are forced to be non-negative, which ensures that the contributions from principal gene expression patterns are positive and thus additive [70, 71].

Both techniques, PCA and NMF, have been used in the identification of biomarkers; for an example see [72]. PCA has been used to characterize the gene expression of stem cells in different phases [73] and different types of stem cells [74]. As NMF has been found superior to PCA in reducing microarray data [75], it has been used more extensively in the identification of cancer molecular patterns for gene expression data [70, 76, 77].

7.1.2 Clustering

Clustering is one of the most widespread tools for grouping transcripts in microarray data. The concept of clustering is based on the simple idea of grouping similar objects. The goal is to maximize the similarity between objects in the same

cluster, and minimize the similarity of objects in different clusters. How similarity is measured is thus a key part of clustering algorithms. In the case of microarray data, the expression profile of a gene, made up by the different samples, is seen as a series of coordinates that define a vector [78]. Distance metrics can thus compare the similarity of the direction and/or magnitude of two or more vectors.

Traditional clustering algorithms have existed since the 1950s and have been applied to a number of problems, including image analysis, marketing, for document classification (such as books), and for population studies. These traditional algorithms have also been used to cluster transcriptome data. In addition, specialized clustering algorithms have been developed for time-series data.

Clustering for Static Sampling

In the case of static sampling, transcriptome data can be represented as a matrix, with each row representing a gene, and each column representing a single condition. The data can thus be represented as vectors and the distance between these vectors can be determined. Note that there are two ways to organize the data. One is to take the expression value of each gene across different samples as a vector. The other one is to take the expression of all genes in a sample as a vector. Clustering can thus be used to find genes behaving similarly in different samples or samples which are "similar" in overall gene expression. In the following section, all examples are illustrated as clustering genes with similar transcriptional behaviors in different samples. The alternative of classifying samples based on their overall gene expression data is demonstrated in the supervised classification topic.

A distance measure (such as Euclidean, Manhattan, Chebyshev, Mahalanobis, Pearson, cosine, Spearman, or Kendall) is used to assess similarity and the data is then organized into clusters according to clustering rules. These clusters can be of fixed size, the number of clusters determined a priori) or natural clusters can be discovered in the data. The most commonly used clustering algorithms broadly correspond to two categories: hierarchical clustering and partitional clustering.

Hierarchical clustering can be bottom-up, starting with single-gene clusters and joining the most similar clusters until a single cluster with all genes is obtained; or top-down, starting with all genes in a single cluster and dividing them into smaller clusters [79]. In both cases, the result is represented as a hierarchical tree, or dendrogram. Most commonly, the bottom-up approach is used (Fig. 6). Initially, two closest genes (1 and 2; then 3 and 4) are joined using one of the distance metrics. In the next iteration, a linkage or amalgamation rule is needed to join these multiple-gene clusters [80]. This rule can be single linkage, complete linkage, or average linkage. In single linkage (also known as nearest neighbor), the similarity of these two clusters is the shortest distance of all pair-wise comparisons of the genes in one cluster to the other; in this example, the distance between gene 1 and gene 3. In complete linkage (also known as furthest neighbor), the similarity of these two clusters is defined as the largest distance of these pair-wise

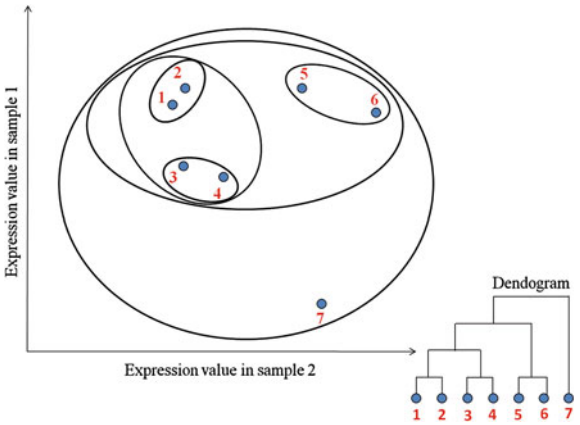


Fig. 6 Hierarchical clustering. The algorithm starts with each gene belonging to its own cluster, followed by joining the two closest genes: 1 and 2. Subsequently, individual genes or multi-gene clusters are joined using single linkage, complete linkage, or average linkage. In this case, single linkage is used, i.e., the distance between two clusters is taken as the shortest distance between any two members of the clusters. Thus the distance between cluster 1–2 and cluster 3–4 is the distance between genes 1 and 3. The two closest clusters are joined accordingly, in this case cluster 1–2 and cluster 3–4. This grouping is continued until all genes are joined into one cluster, and the whole process can be visualized as a dendrogram

comparisons; in this case, the distance between gene 2 and gene 4. In average linkage, the distance between these two clusters is that between their centroids [65]. In this instance, the centroid of the first cluster is a hypothetical gene "in the middle of" gene 1 and gene 2, and thus its expression level is taken as the average expression level of these two genes.

Hierarchical clustering has been used extensively to compare cell types and tissues, including diseased vs. healthy cells, and drug effects, for example [81–85]. Hierarchical clustering has also been used to classify proteomic profiles of serum, plasma, and modified media supplements used in cell culture [86], and metabolomic profiles of extracellular metabolites in recombinant CHO fed-batch cultures [87].

In partitional clustering, data points are separated into a pre-defined number of clusters. In the first step of these iterative algorithms, data points are randomly assigned to clusters. The distance between individual data points and the cluster is then calculated and used to reassign the data points to the cluster to which they are closest. This process continues until all data points are assigned to the closest cluster [88]. *K*-means clustering, Self-Organizing Maps (SOM), and Fuzzy C-means (FCM) clustering are among the best known clustering algorithms in this category. One limitation of these algorithms is that the number of clusters has to be fixed from the beginning, and thus the results are dependent on it [89].

In *k*-means clustering [90], *k* is the number of clusters, and is a required input. *k* random points are used as cluster centers (or means) at initialization. All data

points are assigned to these initial clusters by finding the one with the closest distance. In iterative steps, the mean of each cluster is recalculated and the data points reassigned to new clusters [91]. This process continues until the assignment does not change markedly. As the value of k greatly influences the final outcome, several algorithms include a procedure to determine the best k. k-means clustering has been used to analyze transcriptome data of cancer cells [92] and stem cells [74, 93] among others.

Similar to k-means clustering, in the case of SOMs [94], the number of clusters is also a required input. In addition, their geometry must be specified (grid size). Thus not only the number of clusters but also their geometry has an effect on the final clustering result. A seed vector is first assigned to each cluster, and data assigned to these clusters in an iterative process. In each iteration, randomly selected gene expression data is compared to the seed vectors. The gene is assigned to the cluster that has the more similar seed vector. The value of the seed vector is updated, so that it is more similar to the expression of the gene used in the comparison. Because the cluster centers are part of a grid, the values of the other seed vectors are also modified, although to a lower extent. SOMs have been used to analyze monolayers of cultured rat hepatocytes [95], to study hematopoietic differentiation [96], to investigate saline osmotic tolerance in yeast [97], and to investigate hepatic differentiation [98], among others.

Whereas k-means and SOM assign each gene to a single cluster (hard clustering), FCM [99] links each gene to all clusters using a series of values. Values close to one indicate strong association to a cluster, and values close to 0 indicate absence of association. These indexes define the membership of each gene with respect to all clusters [100]. In addition to the number of clusters, the fuzziness parameter is also a required input. Kim et al. [101] have reported that the fuzziness parameter is sensitive to the normalization method used, and thus the clustering results vary with the normalization method. Recently, a method for the determination of the optimal parameters for FCM has been proposed [102]. FCM has been used to analyze gene expression profiles in high-grade gliomas [103] and in tumor sample classification [104].

Clustering for Dynamic Sampling

Clustering algorithms such as hierarchical clustering, k-means, and SOM are also commonly used to analyze time-series data. However, these algorithms do not take into account the sequential aspect of time-series data [105]. Thus clustering of time series requires specialized algorithms. Some of the specialized algorithms require long series (>10 time points), whereas others have been developed specifically for short time series.

B-splines [20, 106, 107], linear splines [108], ordered restricted inference [109], hidden Markov models [110], and gene expression dynamics using regression [111] are examples of clustering algorithms that can be used for long time-series data. Fuzzy C-Varieties with Transitional State Discrimination preclustering

(FCV-TSD) [112], ASTRO and MiMeSR [105], and Short Time-series Expression Miner (STEM) [113] are examples of clustering algorithms developed specifically for short time-series data.

STEM selects a set of potential expression profiles, each representing a unique pattern. Each gene is then assigned to the profile that best represents it. The significance of each profile is determined using hypothesis testing. The number of genes assigned to each profile under the true ordering is compared to the average number of genes assigned to each profile when permutated data is used. The significant profiles can then be analyzed independently or grouped into clusters. STEM has been used to cluster time-course microarray data collected in the study of egg development in *Drosophila melanogaster* [114], salt stress in *Medicago truncatula* [115], and muscle differentiation [116].

Biclustering takes clustering algorithms a step further. It consists of simultaneous clustering of both genes (rows) and conditions (columns) [117]. The goal in biclustering is to find submatrices [118], that is, to identify subgroups of genes and/or subgroups of conditions with highly correlated behaviors. Thus biclustering can find correlations in certain datasets where other algorithms cannot. Biclusters can be of constant row, constant column, or both constant row and column.

Among the software that can perform biclustering are Gene Expression Mining Server (GEMS) [119], Expression Analyzer and DisplayER (EXPANDER) [120], Phase-shifted Analysis of Gene Expression (PAGE) [121], Biclustering Gene Expression Time Series (BIGGEsTS) [122], Biclustering algorithm and Visualization (BiVisu) [123], and Biclustering Analysis Toolbox (BicAT) [124], which integrates several biclustering algorithms.

7.2 Supervised Classification Methods

Unsupervised classification methods are used for the identification of naturally existing clusters within the data. Supervised approaches, on the other hand, are designed to address the following question: given a set of samples categorized into pre-defined groups (training set), can we use the gene expression data of these samples to construct a rule, or a function, to differentiate these groups? This also implies the ability to use this rule for classification of new, uncategorized samples (test set) based on their expression data.

Since the classification rule is built upon the training set, it may fit this dataset "too well" and thus have poor performance on unclassified samples in the test set (Fig. 7a). In this example, the high producer clones (blue circles) and the low producer clones (black squares) can be simply separated by a linear model (solid line), allowing several samples to be misclassified (outliers). Yet the model can become over-complicated (dashed line) when trying to classify correctly all outliers and thus often results in a higher error rate in classifying regular samples. This is known as "overfitting", and ideally should be assessed using an independent test set. However, in situations where acquiring additional data is

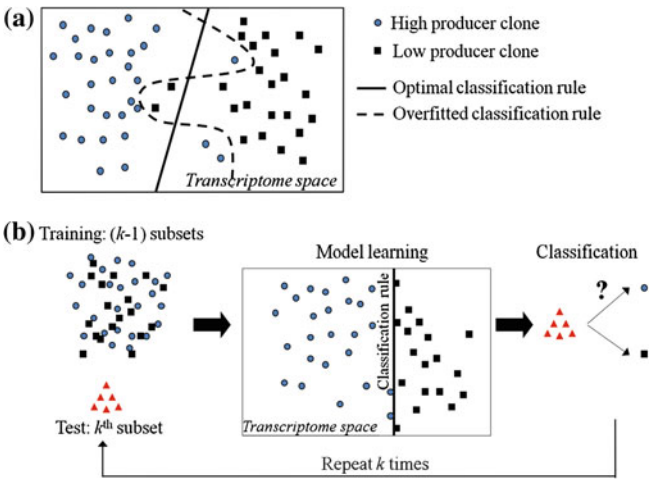


Fig. 7 Overfitting of training data and *k*-fold cross-validation scheme. **a** High producer clones (*blue circles*) and low producer clones (*black squares*) can be separated using a linear model (*solid line*) with a few outliers. Yet the model can become complex (*dashed line*) when all outliers are taken into consideration. This overfitted model will have a high error rate when classifying new samples. **b** The data is split into *k* subsets: $(k - 1)$ subsets are used for training the model, and testing is performed on the *k*th subset. This process is repeated *k* times until all data have been used for testing

expensive or not feasible, various cross-validation schemes can be used. The leave-one-out scheme allocates one sample for testing whereas the rest are used to train the classification model. In the hold-out scheme, the data is split into two equal sets—one is used for training, and the other for testing. Another frequently used method is the *k*-fold cross-validation, in which the data is divided into *k* sets—the first $(k - 1)$ sets are used for training, and the last one for testing (Fig. 7b). This process is repeated until all data have been used for testing. Commonly used supervised classifiers for gene expression data include K-Nearest Neighbors (KNNs), decision trees, Artificial Neural Networks (ANNs), and Support Vector Machines (SVMs). These algorithms have been implemented in several code libraries and various downloadable packages in Matlab and R.

7.2.1 K-Nearest Neighbors

KNN is among the simplest and most fundamental classification methods, and is often the first choice when prior knowledge about the dataset is minimal. Given that a set of samples have been classified into different groups, a new sample will be assigned into the group whose members constitute the majority in the neighborhood of the sample [125, 126]. The choice of distance metric thus becomes vital in this case —a new sample can be assigned to a different group when a different distance metric is applied. In addition, if a certain group is dominant in size compared to the others,

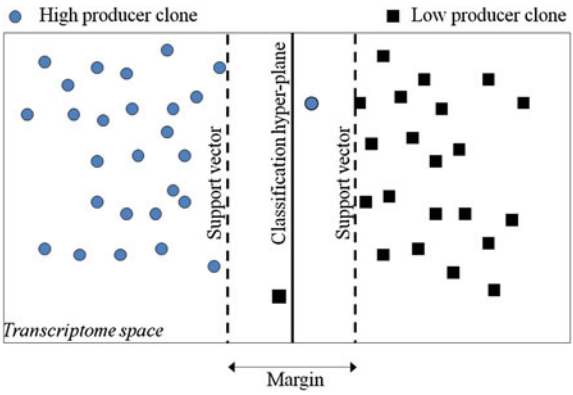


Fig. 8 Support Vector Machines with soft margin. Binary SVM algorithms search for a separation hyper-plane that maximizes the margin (or distance) between two groups: in this case, high producer clones and low producer clones. Samples on the margins are referred to as "support vectors". A few samples can be misclassified in order to obtain a maximal margin ("soft margin")

a bias in assigning new samples into that group is likely to occur. One way to circumvent this problem involves giving each "neighbor" a weight inversely proportional to its distance to the new sample. Furthermore, the distance threshold and the number of "neighboring" samples *k* also have an effect on the final classification, and thus should be optimized using cross-validation.

In the FDA MicroArray Quality Control (MAQC) project, a KNN data analysis protocol was developed to predict the clinical outcome of about 500 new neuroblastoma patients [127]. These KNN models were built using a large gene expression dataset obtained from approximately 700 breast cancer, neuroblastoma, and multiple myeloma samples. In another example, gene expression signatures from 4413 probes in 37 colorectal cancer samples were also used to train a KNN model which was further validated using a leave-one-out scheme [128]. This model successfully classified these samples into serrated and conventional colorectal cancer samples using the expression data of 10 genes.

7.2.2 Decision Trees

Decision trees are built using an iterative scheme in which a question about the gene expression signatures of the training samples is posed at each node [126, 129–131]. The entire tree is obtained by repeated splitting of those samples into two or multiple descendant subsets. The training samples will guide the choice of splitting rules such that each terminal node of the tree, i.e., leaf, is assigned a group label. Thus decision trees are often more interpretable than other classifiers, and naturally support multiple-group assignment. Furthermore, multiple decision trees can be combined into an ensemble, e.g., random forest, to increase the classification accuracy [132, 133]. When applying decision trees, it is critical to control the complexity of the tree, i.e., avoid overfitting the training data. In addition to using cross-validation, one can also prune the tree by collapsing several internal nodes into one leaf, or stop branching the tree when there is no substantial improvement in the homogeneity of the final group assignment.

Several decision-tree algorithms were applied to 869 genes differentially expressed in earthworms in response to explosive compounds TNT or RDX [134]. 354 genes were subsequently selected by these algorithms as classifiers, and ranked according to their significance in the assembled tree. In another application, hierarchical clustering results of gene expression data from three different cohorts of 481 breast cancer samples were further analyzed using decision trees [135]. Four groups with different expression levels of osteopontin (OPN), activated leukocyte cell adhesion molecule (ALCAM), human epidermal growth factor 2 (HER2), and estrogen receptor (ER) were found. Patients with high OPN and low ER, HER2 and ALCAM were placed in a particularly high-risk group.

7.2.3 Artificial Neural Networks

ANNs were developed based on the computation principles occurring in the network of neurons within the human brain [126, 136, 137]. An ANN model can be considered as an assembly of interconnected nodes in which all input sources, in this case the expression values of all genes on the array, are weighted and combined. This weighted average is compared to a threshold, yielding an output value based on a step function. If the average exceeds the threshold, the output value will be one, corresponding to one group; zero, which corresponds to the other group. During the training process, the weighting factors and the threshold can be estimated iteratively, and a linear decision boundary (i.e., separating hyper-plane) can be obtained. Yet when the data are not linearly separable, hidden layers of intermediate nodes can be added to the network. A partial classification is performed at each layer, and assembled to achieve the final classification at the output node. Furthermore, alternative functions such as sigmoid or linear model can be utilized in place of the simple step function in these feed-forward neural networks.

Using gene expression data obtained from 63 training samples of small, round blue cell tumors (SRBCTs), 3750 ANNs have been constructed and cross-validated [138]. Without overfitting, these models successfully classified the samples into four diagnostic categories of tumors. ANNs have also proven efficient in tracking transcriptional changes responsible for progression from the chronic stage to a highly aggressive acute stage of adult T-cell leukemia (ATL) [139]. Using gene expression data from more than 44,000 probe sets and 10-fold cross-validation on 37 samples, 44 “predictor” genes could be identified, offering the possibility of diagnosing different ATL stages.

7.2.4 Support Vector Machines

In binary SVM, two groups (for example, high producer clones and low producer clones) are separated in such a way that the distance between the training samples and the decision boundary is maximized [126, 140, 141] (Fig. 8). This optimization process results in the construction of a separating hyper-plane, i.e., a

linear line in 2-dimensional space, which maximizes the margin between the two groups. In several cases where the samples are not linearly separable in the original space, a kernel function can be chosen to transform the data to a higher-dimensional space in which a "linear" hyper-plane can be found. Furthermore, a few anomalous samples are often allowed to be misclassified to achieve a larger margin. Thus a cost function has to be selected and optimized such that the size of this "soft" margin is balanced with the allowable degree of hyper-plane violation.

Gene expression data from 97,802 clones was used to construct several SVM models using the simple dot-product kernel and validated through the leave-one-out scheme [142]. 31 human tissue samples were successfully classified by these models into cancerous ovarian and normal tissues. Interestingly, an SVM model was also built using gene expression profiles from seven high and four low recombinant IgG-producing NS0 cell lines. Through the leave-one-out cross-validation process, the transcriptomic differences between these high and low producers were indeed highlighted, supporting the molecular basis of productivity trait [143].

8 Pathway Analysis

Microarray analysis results in a list of differentially expressed genes or genes with a dynamic trend over time. It is possible that the transcriptional changes seen on those genes might not be independent, but rather have occurred in a coordinated manner. Thus understanding the physiological relevance of these changes requires analysis in a biological context, beyond what differential expression analysis can determine. Furthermore, examining genes in each pathway as a whole allows one to detect subtle, yet consistent, transcriptional changes that would otherwise be neglected by differential gene expression analysis.

Pathway analysis involves mapping the list of differentially expressed genes onto known pathways in order to elucidate a whole chain of events which might have occurred during the experiment. Depending on the microarray platform, probe identifiers can be linked to different sources of annotation, for instance, Gene Ontology (GO) [144], Kyoto Encyclopedia of Genes and Genomes (KEGG) [145], and Gene Map Annotator and Pathway Profiler (GenMAPP) [146]. This retrieval of pathway information allows all differentially expressed genes in a certain pathway to be highlighted. However, statistical tests need to be performed to confirm whether the entire pathway is indeed enriched or under-represented rather than occurring by random chance. A number of methods and software have been developed to assess the statistical significance of this functional enrichment/under-representation, including Ingenuity's IPA [147], GeneGo's MetaCore [148], GenMAPP's MAPPFinder [146], Gene Set Enrichment Analysis (GSEA) [149], and Gene Set Analysis [150]. Those methods differ in the calculation of the enrichment score and the corresponding significance level, usually *p*-value or FDR. For illustrative purposes, two representative methods, MAPPFinder and GSEA, are described in the following section.

8.1 MAPPFinder

In order to assess the degree of enrichment for each pathway (or gene set), MAPPFinder calculates a *z*-score using the number of differentially expressed genes in the set, the number of genes in the set, the number of differentially expressed genes in total, and the total number of genes on the array [151–153]. A high positive *z*-score indicates that the pathway of interest is significantly enriched, and an extreme negative *z*-score suggests that it is under-represented. Furthermore, if a *p*-value is desired, a *z*-score of 1.96 or −1.96 can be converted to a *p*-value of 0.05 given that the data strictly follows a hyper-geometric distribution. It is important to note that, similar to several other pathway analysis tools, MAPPFinder also requires a pre-defined list of differentially expressed genes. This is sometimes challenging since the list can vary considerably depending on the selected fold-change and the *p*-value cutoff.

Prickett et al. have demonstrated the use of MAPPFinder in uncovering several immune-system pathways affected in chicken infected with a protozoan parasite [154]. 1,175 genes, accounting for about 10% of the total unique Ensembl genes present on the array, were mapped to 85 inferred chicken pathways in GenMAPP, 18 of which were either up- or down-regulated at a *p*-value cut-off of 0.05. In another study, functional enrichment information obtained from MAPPFinder was linked automatically to the original gene expression data to calculate the average intensity or ratio of all differentially expressed genes in each pathway [155]. This quantitative evaluation of dose- and time-dependent microarray data in rats exposed to toxicants thus allows one to calculate an effective dose (ED_{50}) for each pathway, which plays an important role in risk assessment.

8.2 Gene Set Enrichment Analysis

GSEA is a powerful tool for pathway analysis which calculates gene set enrichment using all genes present on the array instead of a pre-defined set of differentially expressed genes [149, 156, 157]. An ordered list is first generated by ranking all genes in the dataset based on their signal-to-noise ratio (Fig. 9). This ratio is often the quotient between the difference in average expression levels and the overall variability of measurement. In the second step, a running-sum statistic is measured for each pathway (or gene set S) by travelling down the ordered list. If the gene encountered is a part of the gene set of interest, the statistic is increased; otherwise it is decreased. The magnitude of this change is set to be proportional to the signal-to-noise of that gene and the size of the gene set it belongs to. The maximum deviation from zero of the running-sum statistic is chosen as the enrichment score (ES), and an associated statistical significance (*p*-value) can be calculated using a permutation scheme. Concurrently, a leading-edge subset of genes which are key contributors to enrichment of the function represented by the gene set can also be exported.

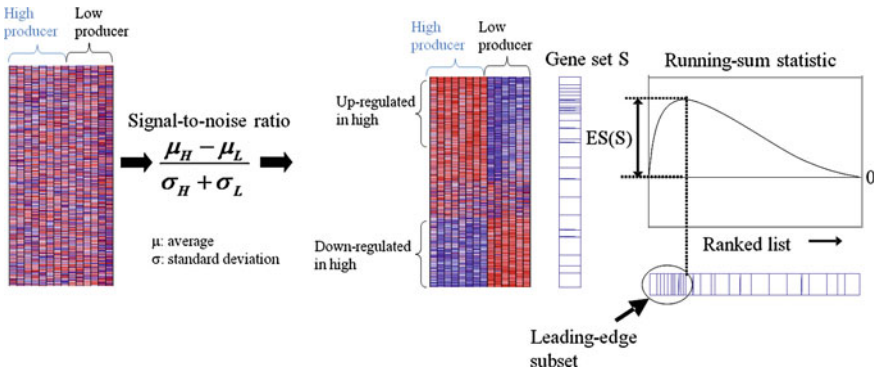


Fig. 9 Gene Set Enrichment Analysis (GSEA). Genes are ranked based on their signal-to-noise ratios to create an ordered list. A running sum statistic is calculated by walking down this list. If the gene encountered is part of the gene set of interest, the running sum statistic is increased; otherwise, it is decreased. The enrichment score (ES) of each gene set (S) is chosen as the maximum deviation of this statistic from zero. Genes with key contributions to the enrichment of the gene set are listed in the leading-edge subset

Deregulated functional categories in Ewing's sarcoma family tumors (ESFT) cell lines under hypoxia were identified by applying GSEA with three different gene sets [158]. Hypoxia-related functions such as angiogenesis, vasculature development, and glucose metabolism were shown to be up-regulated under hypoxic conditions. GSEA was also used alongside other pathway analysis tools to investigate the biological relevance of transcriptional differences between neurofibromatosis type 1 (NF1)-haploinsufficient lymphoblastoid cell lines (LCLs) and mouse B lymphocytes [159]. Despite the modest changes in gene expression detected using the *t*-test, several pathways were shown to experience perturbations including cell cycle, DNA replication and repair, transcription and translation, and immune response.

9 Network Reconstruction

Gene network inference attempts to reconstruct gene networks reflecting their interactions from high-throughput data, especially microarray data. Network reconstruction is a challenging task as gene interactions are dynamic and membership of particular elements in a network is not always permanent. In this regard, the use of microarray data compiled under a wide range of conditions, or from a variety of mutants, can help unveil interactions. Also, time-series microarray data is of particular relevance in reverse engineering regulatory networks. In addition to algorithms for constructing regulatory networks using static gene expression data, special algorithms have also been developed for data obtained from time-series microarrays.

9.1 Network Reconstruction From Static Gene Expression Data

9.1.1 Information Theoretic Methods

Several methods based on information theory have been used for reverse engineer cellular networks from microarray expression profiles. These methods calculate mutual information (MI) between pairs of gene expression profiles. An advantage of MI over other measures of relatedness is that it can detect non-linear interactions. Although these algorithms can be used on time-series data, the sequential aspect is lost, as each sample time point would be considered a different condition.

The original algorithm, relevance networks (RELNET) [160], infers an interaction if MI for a pair is larger than a threshold. RELNET has been applied to reconstruct networks in yeast [160], in cancer cell lines [161], in human hepatoma cells [162], and to identify hub cancer genes [163]. This approach, however, can result in many false positives, and thus extensions which discriminate between direct and indirect interactions have been developed.

Extensions to RELNET proceed in two steps. The first is common to all methods, and consists of calculating MI between pairs of gene expression profiles. In the second step the MI values are assessed and compared, and interactions inferred. The second step is unique to each method.

Context Likelihood of Relatedness (CLR) [164] is an algorithm that removes false correlations by comparing MI for each pair with a background distribution of MI scores. CLR was used to reconstruct parts of the transcriptional regulatory network of the pathogen *Salmonella typhimurium* [165].

A second algorithm based on relevance networks, the Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE) [166–168], eliminates indirect relationships by using data process inequality (DPI), a characteristic of mutual information. ARACNE has been used in reverse engineering the regulatory networks of human B cells [166], in the identification of the targets of the transcriptional repressor BCL6 [169], in the reconstruction of red blood cell metabolism from metabolic data [170], and in the genome-wide reconstruction of the regulatory networks of *Streptomyces coelicolor* [171], an antibiotic producer. CLR and ARACNE were both used to identify genes regulated by Nrf2 in response to oxidative stress [172], and to infer the connectivity of phosphorylation sites in receptor tyrosine kinases [173].

A third algorithm, Minimum Redundancy Networks (MRNET) [174], performs a series of maximum relevance/minimum redundancy (MRMR) selection procedures for each gene and selects the genes having the highest MI with the target.

RELNET, CLR, ARACNE, and MRMR are included in the R package minet (Mutual Information NETwork inference) [175]. The networks resulting from these algorithms can be visualized using the R package Rgraphviz [176]. In addition, the Java implementation of ARACNE includes Cytoscape [177] for network visualization.

9.1.2 Bayesian Networks

Bayesian networks have recently emerged as promising approaches for inferring gene regulatory networks using microarray data. These methods are particularly suitable for the reconstruction of cellular networks due to their ability to capture the stochastic nature of gene regulation and allow causality inference [178, 179]. Furthermore, prior knowledge can be incorporated to improve the accuracy of the final network structure.

A Bayesian network can be represented as a directed acyclic graph, in which each node is a gene, and the edge between two nodes denotes the dependency between two corresponding genes [180, 181]. A joint probability for the network is thus calculated as a product of multiple conditional probabilities for each gene, given that it is regulated by a defined set of parent genes. These probability functions can be either discrete, e.g., binomial distributions, or continuous, e.g., normal density function. Among the several possible networks being reconstructed, an optimal network can be chosen by maximizing the corresponding posterior probability.

Bayesian predictive networks have been constructed using gene expression in combination with genotypic, transcription factor binding site, and protein–protein interaction data in yeast [182]. These networks were shown to successfully predict regulators causing hot spots of gene expression activity in a dividing yeast population. Molecular mechanisms underlying transcriptome reprogramming in cyanobacteria under altered environments were also revealed using Bayesian networks [183]. A large number of genes in the core transcriptional response (CTR) are associated with oxidative stress under most perturbations, indicating the important role of reactive oxygen species in the regulation of these genes.

9.2 Network Reconstruction from Dynamic Gene Expression Data

9.2.1 Information Theoretic Method: TimeDelay-Aracne

Some of the algorithms originally used for network reconstruction using static sampling data have been extended to take advantage of the dependency information contained in time-series data. One such example is an extension of ARACNE. This extension, implemented in the TimeDelay-ARACNE algorithm [184], uses time-course data to retrieve time statistical dependencies between gene expression profiles. This algorithm considers the possibility that the expression of a gene at a certain time could depend on the expression level of another gene at an earlier time point; that is, it detects time-delayed dependencies. The algorithm performs three steps: firstly, it detects the time point of the initial changes in the expression for all genes; secondly, it constructs networks by calculating time-dependent MIs; and thirdly, it performs network pruning using DPI. TimeDelay-ARACNE, which has been implemented in R, also attempts to infer edge directionality.

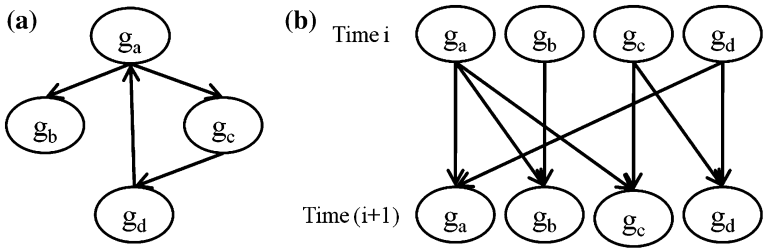


Fig. 10 Gene regulatory network with feedback loop deciphered using DBN. A regulatory network containing four genes (g_a, g_b, g_c, and g_d), three of which form a feedback loop ($g_a \rightarrow g_c \rightarrow g_d \rightarrow g_a$). **a** The feedback loop among g_a, g_c, and g_d is deciphered by allowing cross-interactions along the time axis. **b** The expression level of g_c at time point $(i + 1)$ is dependent on that of g_a at time point i. Similarly, the expression level of g_d at time point $(i + 1)$ is dependent on that of g_c at time point i. The loop is closed by allowing g_d's expression level at time point i to have an effect on that of g_a at time point $(i + 1)$. Note that each gene's expression level at a certain time point is always dependent on its own expression level at the previous time point

9.2.2 Dynamic Bayesian Networks

Built upon Bayesian networks, Dynamic Bayesian Networks (DBNs) also calculate a joint probability using the conditional probability of each gene, and select the optimal network based on the posterior probability. DBNs further allow time delay and modeling of feedback loops by incorporating temporal information associated with time-series data. For instance, the cyclic regulation among genes g_a, g_c, and g_d shown in Fig. 10a can be represented by allowing these genes to cross-interact from time point i to time point $(i + 1)$ (Fig. 10b). To further enhance the prediction accuracy and reduce the computational complexity of DBNs, a number of modifications have also been proposed [185]. For example, potential regulators are limited to those genes with either preceding or simultaneous expression changes. Transcriptional time lags between regulators and target genes can also be estimated, and statistical analysis is thereby restricted within that time frame to improve the accuracy of the prediction.

DBNs have been used successfully to construct gene regulatory networks in yeast using cell cycle time-series microarray data in two independent studies [179, 185]. Main regulatory nodes in the S.O.S DNA repair network in *E. coli* were also extracted using DBNs [186]. Compared to other methods for inferring gene regulatory networks such as Granger causality and probabilistic Boolean network, DBNs consistently displayed enhanced performance. This was especially the case for short time series, as exemplified with data obtained from muscle development in fruit fly [187], normal and infected *Arabidopsis* leaves [188], and food intake effect on human blood [189]. Furthermore, the causality inference power of DBNs was substantially improved when time-series gene perturbation data was also incorporated [190].

10 Concluding Remarks

In this review, methods for the analysis of microarray data are summarized, with a focus on their use in mammalian cell culture. Whereas specific algorithms used for each step depend on the type of data and the question being asked, the general steps for microarray data analysis remain valid. These steps include data pre-processing followed by identification of differentially expressed genes at a minimum, but greater biological insight can be gained by using other types of analysis such as profile pattern recognition, pathway analysis, and network reconstruction.

Even though transcriptome studies of antibody-producing cell lines have been few compared to other cell types, the next few years will see an increase in the resources available for studying genomes and transcriptomes, and this will greatly benefit the understanding of these relevant cell lines.

References

1. Schena M, Shalon D, Davis RW, Brown PO (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270:467
2. Lashkari DA, DeRisi JL, McCusker JH, Namath AF, Gentile C, Hwang SY, Brown PO, Davis RW (1997) Yeast microarrays for genome wide parallel genetic and gene expression analysis. Proc Natl Acad Sci USA 94:13057
3. Agilent http://www.genomics.agilent.com/GenericB.aspx?PageType=Custom&SubPageType=Custom&PageID=2011
4. Affymetrix http://www.affymetrix.com/browse/brand/affymetrixMicroarraySolutions/brandAffymetrixMicroarraySolutions-overview.jsp?category=35722&categoryIdClicked=35722&rootCategoryId=35677&navMode=35722&parent=35722&aId=affymetrixmicroarraybrandsNav
5. Nimblegen http://www.nimblegen.com/products/expression/index.html
6. Wang Z, Gerstein M, Snyder M (2009) RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet 10:57
7. Seth G, Charaniya S, Wlaschin KF, Hu W-S (2007) In pursuit of a super producer–alternative paths to high producing recombinant mammalian cells. Curr Opin Biotechnol 18:557
8. Krampe B, Swiderek H, Al-Rubeai M (2008) Transcriptome and proteome analysis of antibody-producing mouse myeloma NS0 cells cultivated at different cell densities in perfusion culture. Biotechnol Appl Biochem 50:133
9. Spens E, Häggström L (2009) Proliferation of NS0 cells in protein-free medium: the role of cell-derived proteins, known growth factors and cellular receptors. J Biotechnol 141:123
10. Swiderek H, Logan A, Al-Rubeai M (2008) Cellular and transcriptomic analysis of NS0 cell response during exposure to hypoxia. J Biotechnol 134:103
11. Tai YC, Speed TP (2006) A multivariate empirical Bayes statistic for replicated microarray time couse data. Ann Stat 34:6
12. Schaub J, Clemens C, Schorn P, Hildebrandt T, Rust W, Mennerich D, Kaufmann H, Schulz TW (2010) CHO gene expression profiling in biopharmaceutical process analysis and design. Biotechnol Bioeng 105:431
13. Lee YY, Wong KTK, Nissom PM, Wong DCF, Yap MGS (2007) Transcriptional profiling of batch and fed-batch protein-free 293-HEK cultures. Metab Eng 9:52

14. Kantardjieff A, Jacob NM, Yee JC, Epstein E, Kok Y-J, Philp R, Betenbaugh M, Hu W-S (2010) Transcriptome and proteome analysis of Chinese hamster ovary cells under low temperature and butyrate treatment. J Biotechnol 145:143
15. Kerr MK, Churchill GA (2001) Statistical design and the analysis of gene expression microarray data. Genet Res 77:123
16. Wang X, Wu M, Li Z, Chan C (2008) Short time-series microarray analysis: methods and challenges. BMC Syst Biol 2:58
17. Quackenbush J (2002) Microarray data normalization and transformation. Nat Genet 32:496–501
18. Bolstad BM, Irizarry RA, Astrand M, Speed TP (2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics 19:185
19. Mehra S, Lian W, Jayapal K, Charaniya S, Sherman D, Hu W-S (2006) A framework to analyze multiple time series data: a case study with Streptomyces coelicolor. J Ind Microbiol Biotechnol 33:159
20. Bar-Joseph Z, Gerber GK, Gifford DK, Jaakkola TS, Simon I (2003) Continuous representations of time-series gene expression data. J Comput Biol 10:341
21. Bar-Joseph Z (2004) Analyzing time series gene expression data. Bioinformatics 20:2493
22. Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech Signal Process 26:43
23. Kassidas A, MacGregor JF, Taylor PA (1998) Synchronization of batch trajectories using dynamic time warping. AIChE J 44:864
24. Ramaker H-J, van Sprang ENM, Westerhuis JA, Smilde AK (2003) Dynamic time warping of spectroscopic BATCH data. Anal Chim Acta 498:133
25. Smith AA, Vollrath A, Bradfield CA, Craven M (2009) Clustered alignments of gene-expression time series data. Bioinformatics 25:i119
26. Storey JD, Xiao W, Leek JT, Tompkins RG, Davis RW (2005) Significance analysis of time course microarray experiments. Proc Natl Acad Sci USA 102:12837
27. Bonferroni CE (1936) Teoria statistica delle classi e calcolo delle probabilità. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze 8:3
28. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B (Methodological) 57:289
29. Korke R, Gatti MDL, Lau ALY, Lim JWE, Seow TK, Chung MCM, Hu W-S (2004) Large scale gene expression profiling of metabolic shift of mammalian cells in culture. J Biotechnol 107:1
30. De Leon Gatti M, Wlaschin KF, Nissom PM, Yap M, Hu W-S (2007) Comparative transcriptional analysis of mouse hybridoma and recombinant Chinese hamster ovary cells undergoing butyrate treatment. J Biosci Bioeng 103:82
31. Oleksiak MF, Churchill GA, Crawford DL (2002) Variation in gene expression within and among natural populations. Nat Genet 32:261
32. Kalinka AT, Varga KM, Gerrard DT, Preibisch S, Corcoran DL, Jarrells J, Ohler U, Bergman CM, Tomancak P (2010) Gene expression divergence recapitulates the developmental hourglass model. Nature 468:811
33. Tusher VG, Tibshirani R, Chu G (2001) Significance analysis of microarrays applied to the ionizing radiation response. Proc Natl Acad Sci USA 98:5116
34. Zheng H, Ying H, Yan H, Kimmelman AC, Hiller DJ, Chen A-J, Perry SR, Tonon G, Chu GC, Ding Z, Stommel JM, Dunn KL, Wiedemeyer R, You MJ, Brennan C, Wang YA, Ligon KL, Wong WH, Chin L, DePinho RA (2008) p53 and Pten control neural and glioma stem/progenitor cell renewal and differentiation. Nature 455:1129
35. Liu W, Tanasa B, Tyurina OV, Zhou TY, Gassmann R, Liu WT, Ohgi KA, Benner C, Garcia-Bassets I, Aggarwal AK, Desai A, Dorrestein PC, Glass CK, Rosenfeld MG (2010) PHF8 mediates histone H4 lysine 20 demethylation events involved in cell cycle progression. Nature 466:508
36. Lonnstedt I, Britton T (2005) Hierarchical Bayes models for cDNA microarray gene expression. Biostatistics 6:279

37. Smyth GK (2004) Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Stat Appl Genet Mol Biol 3:article3
38. Feldser DM, Kostova KK, Winslow MM, Taylor SE, Cashman C, Whittaker CA, Sanchez-Rivera FJ, Resnick R, Bronson R, Hemann MT, Jacks T (2010) Stage-specific sensitivity to p53 restoration during lung cancer progression. Nature 468:572
39. Bonfanti P, Claudinot S, Amici AW, Farley A, Blackburn CC, Barrandon Y (2010) Microenvironmental reprogramming of thymic epithelial cells to skin multipotent stem cells. Nature 466:978
40. Storey JD, Dai JY, Leek JT (2007) The optimal discovery procedure for large-scale significance testing, with applications to comparative microarray experiments. Biostatistics 8:414–432
41. Storey JD, Tibshirani R (2003) Statistical significance for genome wide studies. Proc Natl Acad Sci USA 100:9440
42. Leek JT, Monsen E, Dabney AR, Storey JD (2006) EDGE: extraction and analysis of differential gene expression. Bioinformatics 22:507
43. Zhan M, Yamaza H, Sun Y, Sinclair J, Li H, Zou S (2007) Temporal and spatial transcriptional profiles of aging in Drosophila melanogaster. Genome Res 17:1236
44. White P, Lee May C, Lamounier RN, Brestelli JE, Kaestner KH (2008) Defining pancreatic endocrine precursors and their descendants. Diabetes 57:654
45. Conesa A, Nueda MJ, Ferrer A, Talon M (2006) maSigPro: a method to identify significantly differential expression profiles in time-course microarray experiments. Bioinformatics 22:1096
46. Sanges R, Cordero F, Calogero RA (2007) oneChannelGUI: a graphical interface to Bioconductor tools, designed for life scientists who are not familiar with R language. Bioinformatics 23:3406
47. Tarraga J, Medina I, Carbonell J, Huerta-Cepas J, Minguez P, Alloza E, Al-Shahrour F, Vegas-Azcarate S, Goetz S, Escobar P, Garcia–Garcia F, Conesa A, Montaner D, Dopazo J (2008) GEPAS, a web-based tool for microarray data analysis and interpretation. Nucleic Acids Res 36:W308
48. Nueda MJ, Sebastian P, Tarazona S, Garcia-Garcia F, Dopazo J, Ferrer A, Conesa A (2009) Functional assessment of time course microarray data. BMC Bioinformatics 10(6):S9
49. Brusniak MY, Bodenmiller B, Campbell D, Cooke K, Eddes J, Garbutt A, Lau H, Letarte S, Mueller LN, Sharma V, Vitek O, Zhang N, Aebersold R, Watts JD (2008) Corra: computational framework and tools for LC-MS discovery and targeted mass spectrometry-based proteomics. BMC Bioinformatics 9:542
50. Levin AM, de Vries RP, Conesa A, de Bekker C, Talon M, Menke HH, van Peij NN, Wosten HA (2007) Spatial differentiation in the vegetative mycelium of Aspergillus niger. Eukaryot Cell 6:2311
51. Wong CE, Singh MB, Bhalla PL (2009) Molecular processes underlying the floral transition in the soybean shoot apical meristem. Plant J 57:832
52. Wong CE, Singh MB, Bhalla PL (2009) Floral initiation process at the soybean shoot apical meristem may involve multiple hormonal pathways. Plant Signal Behav 4:648
53. Pascual L, Blanca JM, Canizares J, Nuez F (2009) Transcriptomic analysis of tomato carpel development reveals alterations in ethylene and gibberellin synthesis during pat3/pat4 parthenocarpic fruit set. BMC Plant Biol 9:67
54. Hoogerwerf WA, Sinha M, Conesa A, Luxon BA, Shahinian VB, Cornelissen G, Halberg F, Bostwick J, Timm J, Cassone VM (2008) Transcriptional profiling of mRNA expression in the mouse distal colon. Gastroenterology 135:2019
55. Smilde AK, Jansen JJ, Hoefsloot HCJ, Lamers ANR-J, van der Greef J, Timmerman ME (2005) ANOVA-simultaneous component analysis (ASCA): a new tool for analyzing designed metabolomics data. Bioinformatics 21:3043
56. Jansen JJ, Hoefsloot HCJ, Greef JVD, Timmerman ME, Westerhuis JA, Smilde AK (2005) ASCA: analysis of multivariate data obtained from an experimental design. J Chemometr 19:469

57. Smilde AK, Hoefsloot HCJ, Westerhuis JA (2008) The geometry of ASCA. J Chemometr 22:464
58. Vis D, Westerhuis J, Smilde A, van der Greef J (2007) Statistical validation of megavariate effects in ASCA. BMC Bioinformatics 8:322
59. Wang J, Reijmers T, Chen L, Van Der Heijden R, Wang M, Peng S, Hankemeier T, Xu G, Van Der Greef J (2009) Systems toxicology study of doxorubicin on rats using ultra performance liquid chromatography coupled with mass spectrometry based metabolomics. Metabolomics 5:407
60. Nueda MJ, Conesa A, Westerhuis JA, Hoefsloot HCJ, Smilde AK, Talon M, Ferrer A (2007) Discovering gene expression patterns in time course microarray experiments by ANOVA SCA. Bioinformatics 23:1792
61. Heard NA, Holmes CC, Stephens DA (2006) A quantitative study of gene regulation involved in the immune response of anopheline mosquitoes. J Am Stat Assoc 101:18
62. Angelini C, De Canditiis D, Mutarelli M, Pensky M (2007) A Bayesian approach to estimation and testing in time-course microarray experiments. Stat Appl Genet Mol Biol 6: Article24
63. Angelini C, Cutillo L, De Canditiis D, Mutarelli M, Pensky M (2008) BATS: a Bayesian user-friendly software for analyzing time series microarray experiments. BMC Bioinformatics 9:415
64. Lian W, Jayapal K, Charaniya S, Mehra S, Glod F, Kyung Y-S, Sherman D, Hu W-S (2008) Genome-wide transcriptome analysis reveals that a pleiotropic antibiotic regulator, AfsS, modulates nutritional stress response in Streptomyces coelicolor A3(2). BMC Genomics 9:56
65. Gollub J, Sherlock G, Alan K, Brian O (2006) Clustering microarray data. Academic Press, London
66. Morrison DA, Ellis JT (2003) The design and analysis of microarray experiments: applications in parasitology. DNA Cell Biol 22:357
67. Boutros PC, Okey AB (2005) Unsupervised pattern recognition: an introduction to the whys and wherefores of clustering microarray data. Brief Bioinform 6:331
68. Jolliffe I (2005) Principal component analysis. Wiley, NY
69. Yeung KY, Ruzzo WL (2001) Principal component analysis for clustering gene expression data. Bioinformatics 17:763
70. Brunet J-P, Tamayo P, Golub TR, Mesirov JP (2004) Metagenes and molecular pattern discovery using matrix factorization. Proc Natl Acad Sci USA 101:4164
71. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788
72. Schachtner R, Lutter D, Stadlthanner K, Lang EW, Schmitz G, Tome AM, Gomez Vilda P (2007) Routes to identify marker genes for microarray classification. In: Engineering in medicine and biology society, 2007 EMBS 2007 29th Annual International Conference of the IEEE
73. Aiba K, Sharov AA, Carter MG, Foroni C, Vescovi AL, Ko MSH (2006) Defining a developmental path to neural fate by global expression profiling of mouse embryonic stem cells and adult neural stem/progenitor cells. Stem Cells 24:889
74. Ulloa-Montoya F, Kidder B, Pauwelyn K, Chase L, Luttun A, Crabbe A, Geraerts M, Sharov A, Piao Y, Ko M, Hu W-S, Verfaillie C (2007) Comparative transcriptome analysis of embryonic and adult stem cells with extended and limited differentiation capacity. Genome Biol 8:R163
75. Liu W, Yuan K, Ye D (2008) Reducing microarray data via nonnegative matrix factorization for visualization and clustering analysis. J Biomed Inform 41:602
76. Han X (2008) Improving gene expression cancer molecular pattern discovery using nonnegative principal component analysis. Genome Inf 21:200
77. Frigyesi A, Hoglund M (2008) Non-negative matrix factorization for the analysis of complex gene expression data: identification of clinically relevant tumor subtypes. Cancer Inform 6:275

78. Sherlock G (2000) Analysis of large-scale gene expression data. Curr Opin Immunol 12:201
79. Nugent R, Meila M (2010) An overview of clustering applied to molecular biology. Humana Press, Clifton
80. Frades I, Matthiesen R (2010) Overview on techniques in cluster analysis. Bioinf Methods Clin Res 593:81
81. Anichini A, Scarito A, Molla A, Parmiani G, Mortarini R (2003) Differentiation of CD8+T cells from tumor-invaded and tumor-free lymph nodes of melanoma patients: role of common i^3-chain cytokines. J Immunol 171:2134
82. Vega F, Coombes KR, Thomazy VA, Patel K, Lang W, Jones D (2006) Tissue-specific function of lymph node fibroblastic reticulum cells. Pathobiology 73:71
83. Ambrosi DJ, Tanasijevic B, Kaur A, Obergfell C, O'Neill RJ, Krueger W, Rasmussen TP (2007) Genome-wide reprogramming in hybrids of somatic cells and embryonic stem cells. Stem Cells 25:1104
84. Secco M, Moreira Y, Zucconi E, Vieira N, Jazedje T, Muotri A, Okamoto O, Verjovski-Almeida S, Zatz M (2009) Gene expression profile of mesenchymal stem cells from paired umbilical cord units: cord is different from blood. Stem Cell Rev R 5:387
85. Fortier JM, Payton JE, Cahan P, Ley TJ, Walter MJ, Graubert TA (2010) POU4F1 is associated with t(8;21) acute myeloid leukemia and contributes directly to its unique transcriptional signature. Leukemia 24:950
86. Ayache S, Panelli M, Byrne K, Slezak S, Leitman S, Marincola F, Stroncek D (2006) Comparison of proteomic profiles of serum, plasma, and modified media supplements used for cell culture and expansion. J Translational Med 4:40
87. Chong WPK, Goh LT, Reddy SG, Yusufi FNK, Lee DY, Wong NSC, Heng CK, Yap MGS, Ho YS (2009) Metabolomics profiling of extracellular metabolites in recombinant Chinese Hamster Ovary fed-batch culture. Rapid Commun Mass Spectrom 23:3763
88. De Bruyne V, Al-Mulla F, Pot B (2005) Methods for microarray data analysis. Humana Press, Clifton
89. Dopazo J, Zanders E, Dragoni I, Amphlett G, Falciani F (2001) Methods and approaches in the analysis of gene expression data. J Immunol Methods 250:93
90. Everitt BS (1974) Cluster analysis. Heinemann Educational [for] the Social Science Research Council, London
91. Do JH, Choi D-K (2008) Clustering approaches to identifying gene expression patterns from DNA microarray data. Mol Cells 25:279
92. Liu Y, Yang Y, Xu H, Dong X (2010) Implication of USP22 in the regulation of BMI-1, c-Myc, p16INK4a, p14ARF, and cyclin D2 expression in primary colorectal carcinomas. Diagn Mol Pathol 19:194
93. Way KJ, Dinh H, Keene MR, White KE, Clanchy FIL, Lusby P, Roiniotis J, Cook AD, Cassady AI, Curtis DJ, Hamilton JA (2009) The generation and properties of human macrophage populations from hemopoietic stem cells. J Leukoc Biol 85:766
94. Kohonen T (2001) Self-organizing maps. Springer, Berlin
95. Baker TK, Carfagna MA, Gao H, Dow ER, Li Q, Searfoss GH, Ryan TP (2001) Temporal gene expression analysis of monolayer cultured rat hepatocytes. Chem Res Toxicol 14:1218
96. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. Proc Natl Acad Sci USA 96:2907
97. Pandey G, Yoshikawa K, Hirasawa T, Nagahisa K, Katakura Y, Furusawa C, Shimizu H, Shioya S (2007) Extracting the hidden features in saline osmotic tolerance in Saccharomyces cerevisiae from DNA microarray data using the self-organizing map: biosynthesis of amino acids. Appl Microb Biotechnol 75:415
98. Li W, You P, Wei Q, Li Y, Fu X, Ding X, Wang X, Hu Y (2007) Hepatic differentiation and transcriptional profile of the mouse liver epithelial progenitor cells (LEPCs) under the induction of sodium butyrate. Front Biosci 12:1691
99. Bezdek J (1981) Pattern Recognition with Fuzzy Objective Function Algorithms (Advanced Applications in Pattern Recognition). Springer, Berlin

100. Dembele D, Kastner P (2003) Fuzzy C-means method for clustering microarray data. Bioinformatics 19:973
101. Kim S, Lee J, Bae J (2006) Effect of data normalization on fuzzy clustering of DNA microarray data. BMC Bioinformatics 7:134
102. Schwammle V, Jensen ONJ (2010) A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics 26:2841
103. Czernicki T, Zegarska J, Paczek L, Cukrowska B, Grajokowska W, Zajaczkowska A, Brudzewski K, Ulaczyk J, Marchel A (2007) Gene expression profile as a prognostic factor in high-grade gliomas. Int J Oncol 30:55
104. Wang J, Bø T, Jonassen I, Myklebost O, Hovig E (2003) Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data. BMC Bioinformatics 4:1
105. Tchagang A, Bui K, McGinnis T, Benos P (2009) Extracting biologically significant patterns from short time series gene expression data. BMC Bioinformatics 10:255
106. Luan Y, Li H (2003) Clustering of time-course gene expression data using a mixed-effects model with B-splines. Bioinformatics 19:474
107. Gaffney, S and P Smyth (2005) Joint probabilistic curve clustering and alignment. Adv Neural Inf Process Syst
108. De Hoon MJ, Imoto S, Miyano S (2002) Statistical analysis of a small set of time-ordered gene expression data using linear splines. Bioinformatics 18:1477
109. Peddada SD, Lobenhofer EK, Li L, Afshari CA, Weinberg CR, Umbach DM (2003) Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. Bioinformatics 19:834
110. Schliep A, Schonhuth A, Steinhoff C (2003) Using hidden Markov models to analyze gene expression time course data. Bioinformatics 19(1):255
111. Ramoni MF, Sebastiani P, Kohane IS (2002) Cluster analysis of gene expression dynamics. Proc Natl Acad Sci USA 99:9121
112. Moller-Levet CS, Cho KH, Wolkenhauer O (2003) Microarray data clustering based on temporal variation: FCV with TSD preclustering. Appl Bioinform 2:35
113. Ernst J, Bar-Joseph Z (2006) STEM: a tool for the analysis of short time series gene expression data. BMC Bioinformatics 7:191
114. Baker DA, Russell S (2009) Gene expression during Drosophila melanogaster egg development before and after reproductive diapause. BMC Genomics 10:242
115. Li D, Su Z, Dong J, Wang T (2009) An expression database for roots of the model legume Medicago truncatula under salt stress. BMC Genomics 10:517
116. Ozbudak E, Tassy O, Pourquie O (2010) Spatiotemporal compartmentalization of key physiological processes during muscle precursor differentiation. Proc Natl Acad Sci USA 107:4224
117. Cheng Y, Church GM (2000) Biclustering of expression data. Proc Int Conf Intell Syst Mol Biol 8:93
118. Madeira SC, Oliveira AL (2004) Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans Comput Biol Bioinform 1:24
119. Wu C-J, Kasif S (2005) GEMS: a web server for biclustering analysis of expression data. Nucleic Acids Res 33:W596
120. Shamir R, Maron-Katz A, Tanay A, Linhart C, Steinfeld I, Sharan R, Shiloh Y, Elkon R (2005) EXPANDER—an integrative program suite for microarray data analysis. BMC Bioinformatics 6:232
121. Leung E, Bushel PR (2006) PAGE: phase-shifted analysis of gene expression. Bioinformatics 22:367
122. Goncalves JP, Madeira SC, Oliveira AL (2009) BiGGEsTS: integrated environment for biclustering analysis of time series gene expression data. BMC Res Notes 2:124
123. Cheng KO, Law NF, Siu WC, Lau TH (2007) BiVisu: software tool for bicluster detection and visualization. Bioinformatics 23:2342

124. Barkow S, Bleuler S, Prelic A, Zimmermann P, Zitzler E (2006) BicAT: a biclustering analysis toolbox. Bioinformatics 22:1282
125. Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13:21
126. Tan P-N, Steinbach M, Kumar V (2005) Introduction to data mining. Addison-Wesley, Reading
127. Parry RM, Jones W, Stokes TH, Phan JH, Moffitt RA, Fang H, Shi L, Oberthuer A, Fischer M, Tong W, Wang MD (2010) k-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction. Pharmacogenomics J 10:292
128. Laiho P, Kokko A, Vanharanta S, Salovaara R, Sammalkorpi H, Jarvinen H, Mecklin JP, Karttunen TJ, Tuppurainen K, Davalos V, Schwartz S Jr, Arango D, Makinen MJ, Aaltonen LA (2006) Serrated carcinomas form a subclass of colorectal cancer with distinct molecular basis. Oncogene 26:312
129. Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth International Group, Belmont
130. Quinlan JR (1993) C4.5: Programs for machine learning. Morgan Kaufmann Publishers, Los Altos
131. Kingsford C, Salzberg SL (2008) What are decision trees? Nat Biotechnol 26:1011
132. Breiman L (2001) Random forests. Mach Learn 45:5
133. Tong W, Hong H, Fang H, Xie Q, Perkins R (2003) Decision forest: combining the predictions of multiple independent decision tree models. J Chem Inf Comput Sci 43:525
134. Li Y, Wang N, Perkins EJ, Zhang C, Gong P (2010) Identification and optimization of classifier genes from multi-class earthworm microarray dataset. PLoS One 5:e13715
135. Ihnen M, Wirtz RM, Kalogeras KT, Milde-Langosch K, Schmidt M, Witzel I, Eleftheraki AG, Papadimitriou C, Janicke F, Briassoulis E, Pectasides D, Rody A, Fountzilas G, Muller V (2010) Combination of osteopontin and activated leukocyte cell adhesion molecule as potent prognostic discriminators in HER2- and ER-negative breast cancer. Br J Cancer 103:1048
136. Minsky ML, Papert SA (1969) Perceptrons. MIT Press, Cambridge
137. Krogh A (2008) What are artificial neural networks? Nat Biotechnol 26:195
138. Khan J, Wei JS, Ringner M, Saal LH, Ladanyi M, Westermann F, Berthold F, Schwab M, Antonescu CR, Peterson C, Meltzer PS (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nat Med 7:673
139. Choi YL, Tsukasaki K, O'Neill MC, Yamada Y, Onimaru Y, Matsumoto K, Ohashi J, Yamashita Y, Tsutsumi S, Kaneda R, Takada S, Aburatani H, Kamihira S, Nakamura T, Tomonaga M, Mano H (2006) A genomic analysis of adult T-cell leukemia. Oncogene 26:1245
140. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. Association for Computing Machinery, New York
141. Noble WS (2006) What is a support vector machine? Nat Biotechnol 24:1565
142. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer ML, Haussler D (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16:906
143. Charaniya S, Karypis G, Hu W-S (2009) Mining transcriptome data for function–trait relationship of hyper productivity of recombinant antibody. Biotechnol Bioeng 102:1654
144. Gene Ontology. http://www.geneontology.org
145. Kyoto Encyclopaedia of Genes and Genomes. http://www.genome.jp/kegg/
146. GenMAPP. http://www.genmapp.org
147. Ingenuity. http://www.ingenuity.com/
148. MetaCore. http://www.genego.com/metacore.php
149. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci USA 102:15545

150. Efron B, Tibshirani R (2007) On testing the significance of sets of genes. Ann Appl Stat 1:107
151. Dahlquist KD, Salomonis N, Vranizan K, Lawlor SC, Conklin BR (2002) GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. Nat Genet 31:19
152. Dahlquist KD (2002) Using GenMAPP and MAPPFinder to view microarray data on biological pathways and identify global trends in the data. Wiley, NY
153. Doniger S, Salomonis N, Dahlquist K, Vranizan K, Lawlor S, Conklin B (2003) MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. Genome Biol 4:R7
154. Prickett D, Watson M (2009) Use of GenMAPP and MAPPFinder to analyse pathways involved in chickens infected with the protozoan parasite Eimeria. BMC Proc 3:S7
155. Yu X, Griffith WC, Hanspers K, Dillman JF, Ong H, Vredevoogd MA, Faustman EM (2006) A system-based approach to interpret dose- and time-dependent microarray data: quantitative integration of gene ontology analysis for risk assessment. Toxicol Sci 92:560
156. Mootha VK, Lindgren CM, Eriksson KF, Subramanian A, Sihag S, Lehar J, Puigserver P, Carlsson E, Ridderstrale M, Laurila E, Houstis N, Daly MJ, Patterson N, Mesirov JP, Golub TR, Tamayo P, Spiegelman B, Lander ES, Hirschhorn JN, Altshuler D, Groop LC (2003) PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. Nat Genet 34:267
157. Gene Set Enrichment Analysis. http://www.broadinstitute.org/gsea/index.jsp
158. Aryee DNT, Niedan S, Kauer M, Schwentner R, Bennani-Baiti IM, Ban J, Muehlbacher K, Kreppel M, Walker RL, Meltzer P, Poremba C, Kofler R, Kovar H (2010) Hypoxia modulates EWS-FLI1 transcriptional signature and enhances the malignant properties of ewing's sarcoma cells in vitro. Cancer Res 70:4015
159. Pemov A, Park C, Reilly K, Stewart D (2010) Evidence of perturbations of cell cycle and DNA repair pathways as a consequence of human and murine NF1-haploinsufficiency. BMC Genomics 11:194
160. Butte, A J and I S Kohane (2000) Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. Pac Symp Biocomput 418
161. Butte AJ, Tamayo P, Slonim D, Golub TR, Kohane IS (2000) Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. Proc Natl Acad Sci USA 97:12182
162. Moriyama M, Hoshida Y, Otsuka M, Nishimura S, Kato N, Goto T, Taniguchi H, Shiratori Y, Seki N, Omata M (2003) Relevance network between chemosensitivity and transcriptome in human hepatoma cells. Mol Cancer Ther 2:199
163. Jiang W, Li X, Rao S, Wang L, Du L, Li C, Wu C, Wang H, Wang Y, Yang B (2008) Constructing disease-specific gene networks using pair-wise relevance metric: application to colon cancer identifies interleukin 8, desmin and enolase 1 as the central elements. BMC Syst Biol 2:72
164. Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, Cottarel G, Kasif S, Collins JJ, Gardner TS (2007) Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biol 5:e8
165. Taylor RC, Singhal M, Weller J, Khoshnevis S, Shi L, McDermott J (2009) A network inference workflow applied to virulence-related processes in Salmonella typhimurium. Ann N Y Acad Sci 1158:143
166. Basso K, Margolin AA, Stolovitzky G, Klein U, Dalla-Favera R, Califano A (2005) Reverse engineering of regulatory networks in human B cells. Nat Genet 37:382
167. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Favera R, Califano A (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7(1):S7
168. Margolin AA, Wang K, Lim WK, Kustagi M, Nemenman I, Califano A (2006) Reverse engineering cellular networks. Nat Protoc 1:662
169. Basso K, Saito M, Sumazin P, Margolin AA, Wang K, Lim WK, Kitagawa Y, Schneider C, Alvarez MJ, Califano A, Dalla-Favera R (2010) Integrated biochemical and computational

approach identifies BCL6 direct target genes controlling multiple pathways in normal germinal center B cells. Blood 115:975
170. Nemenman I, Escola GS, Hlavacek WS, Unkefer PJ, Unkefer CJ, Wall ME (2007) Reconstruction of metabolic networks from high-throughput metabolite profiling data: in silico analysis of red blood cell metabolism. Ann N Y Acad Sci 1115:102
171. Castro-Melchor M, Charaniya S, Karypis G, Takano E, Hu W-S (2010) Genome-wide inference of regulatory networks in Streptomyces coelicolor. BMC Genomics 11:578
172. Taylor RC, Acquaah-Mensah G, Singhal M, Malhotra D, Biswal S (2008) Network inference algorithms elucidate Nrf2 regulation of mouse lung oxidative stress. PLoS Comput Biol 4:e1000166
173. Ciaccio MF, Wagner JP, Chuu CP, Lauffenburger DA, Jones RB (2010) Systems analysis of EGF receptor signaling dynamics with microwestern arrays. Nat Methods 7:148
174. Meyer PE, Kontos K, Lafitte F, Bontempi G (2007) Information-theoretic inference of large transcriptional regulatory networks. EURASIP J Bioinform Syst Biol 79879
175. Meyer PE, Lafitte F, Bontempi G (2008) minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information. BMC Bioinformatics 9:461
176. Carey VJ, Gentry J, Whalen E, Gentleman R (2005) Network structures and algorithms in Bioconductor. Bioinformatics 21:135
177. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13:2498
178. Murphy K, Mian S (1999) Modelling gene expression data using dynamic Bayesian networks
179. Kim SY, Imoto S, Miyano S (2003) Inferring gene networks from time series microarray data using dynamic Bayesian networks. Brief Bioinform 4:228
180. Heckerman D (1998) A tutorial on learning with Bayesian networks. Kluwer Academic, Boston
181. Needham CJ, Bradford JR, Bulpitt AJ, Westhead DR (2006) Inference in Bayesian networks. Nat Biotechnol 24:51
182. Zhu J, Zhang B, Smith EN, Drees B, Brem RB, Kruglyak L, Bumgarner RE, Schadt EE (2008) Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. Nat Genet 40:854
183. Singh A, Elvitigala T, Cameron J, Ghosh B, Bhattacharyya-Pakrasi M, Pakrasi H (2010) Integrative analysis of large scale expression profiles reveals core transcriptional response and coordination between multiple cellular processes in a cyanobacterium. BMC Syst Biol 4:105
184. Zoppoli P, Morganella S, Ceccarelli M (2010) TimeDelay-ARACNE: reverse engineering of gene networks from time-course data by an information theoretic approach. BMC Bioinformatics 11:154
185. Zou M, Conzen SD (2005) A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. Bioinformatics 21:71
186. Perrin B-E, Ralaivola L, Mazurie A, Bottani S, Mallet J, d'Alche-Buc F (2003) Gene networks inference using dynamic Bayesian networks. Bioinformatics 19:ii138
187. Li P, Zhang C, Perkins E, Gong P, Deng Y (2007) Comparison of probabilistic Boolean network and dynamic Bayesian network approaches for inferring gene regulatory networks. BMC Bioinformatics 8:S13
188. Zou C, Feng J (2009) Granger causality vs dynamic Bayesian network inference: a comparative study. BMC Bioinformatics 10:122
189. Zhu J, Chen Y, Leonardson AS, Wang K, Lamb JR, Emilsson V, Schadt EE (2010) Characterizing dynamic changes in the human blood transcriptional network. PLoS Comput Biol 6:e1000671
190. Dojer N, Gambin A, Mizera A, Wilczynski B, Tiuryn J (2006) Applying dynamic Bayesian networks to perturbed gene expression data. BMC Bioinformatics 7:249

http://www.springer.com/978-3-642-28349-9

Genomics and Systems Biology of Mammalian Cell Culture

Hu, W.-S.; Zeng, A.-P. (Eds.)
2012, XXII, 290 p., Hardcover
ISBN: 978-3-642-28349-9