A Biclique Approach to Reference Anchored Gene Blocks and Its Applications to Pathogenicity Islands

Arnon Benshahar[1], Vered Chalifa-Caspi[3], Danny Hermelin[2(✉)], and Michal Ziv-Ukelson[1(✉)]

[1] Department of Computer Science, Ben-Gurion University of the Negev, Beersheba, Israel
{benarnon,michaluz}@bgu.ac.il
[2] Department of Industrial Engineering and Management, National Institute for Biotechnology in the Negev, Beersheba, Israel
hermelin@bgu.ac.il
[3] National Institute for Biotechnology in the Negev, Ben-Gurion University of the Negev, Beersheba, Israel
veredcc@bgu.ac.il

Abstract. We formalize a new problem variant in gene-block discovery, denoted *Reference-Anchored Gene Blocks* (RAGB). Given a query sequence Q of length n, representing the gene-array of a DNA element, a window size bound d on the length of a substring of interest in Q, and a set of target gene sequences $\mathcal{T} = \{T_1 \dots T_c\}$. Our objective is to identify gene-blocks in $\mathcal{T}$ that are centered in a subset q of co-localized genes from Q, and contain genomes from $\mathcal{T}$ in which the corresponding orthologs of the genes from q are also co-localized. We cast RAGB as a variant of a (colored) biclique problem in bipartite graphs, and analyze its parameterized complexity, as well as the parameterized complexity of other related problems. We give an $O(nm + 2^d nm/\lg m)$ time algorithm for the uncolored variant of our biclique problem, where m is the number of areas of interest that are parsed from the target sequences, and n and d are as defined above. Our algorithm can be adapted to compute all maximal bicliques in the graph within the same time complexity, and to handle edge-weights with a slight $O(\lg d)$ increase to its time complexity. For the colored version of the problem, our algorithm has a time complexity of $O(2^d nm)$. We implement the algorithm and exemplify its application to LEE, a well-known pathogenicity island from the e. coli genome harboring virulence genes. *Our code and supplementary materials, including omitted proofs and figures, are available at* https://www.cs.bgu.ac.il/~negevcb/RAGB/.

1 Introduction

Genomes of bacterial species can evolve through a variety of processes including mutations, rearrangements and horizontal gene transfer. Information gathered over the past few years from a rapidly increasing number of sequenced genomes

M. Frith and C.N.S. Pedersen (Eds.): WABI 2016, LNBI 9838, pp. 14–26, 2016.
DOI: 10.1007/978-3-319-43681-4_2

has shown that besides the core genes which encode essential metabolic functions, bacterial genomes also harbour a variable number of accessory genes acquired by horizontal gene transfer, encoding adaptative traits that might be beneficial for bacteria under certain growth or environmental conditions [15]. Many of the accessory genes acquired by horizontal gene transfer form syntenic blocks recognized as genomic islands - discrete DNA segments that are transferred between closely related strains. During island evolution, several genetic elements have been acquired independently at different time points and from different hosts. Thus, genomic islands often represent mosaic-like structures rather than homogeneous segments of horizontally acquired DNA.

Genomic islands are known to carry genes offering a selective advantage for host bacteria. They play a key role in the emergence of highly virulent and highly resistant pathogenic strains of bacteria, which are of major concern worldwide [15]. This concern motivates new studies, such as the one proposed in this paper, aimed to develop new tools that will help explore the evolution and spreading patterns of pathogenicity, virulence and resistance components harbored within genomic islands, across the bacterial kingdom. Furthermore, applying gene-block discovery approaches to these studies may shed light on the function of unknown proteins which are consistently co-transferred with functional gene cascades.

1.1 The Reference Anchored Gene Block Problem

We propose a new bioinformatic approach that is based on interrogating a given reference DNA element from a specific genome for subsets of genes that are conserved as proximity blocks across other microbial genomes. The subsequent computational problem is called the *Reference Anchored Gene Blocks* problem (RAGB). The input to this problem consists of the gene sequence of a reference DNA element, and a set of target genome sequences. The target sequences are then parsed, either via a simple sliding window approach or according to some a priori biological data, into areas of interest or segments of small proximity. The output of our problem are gene blocks that are clustered together in small vicinity in the reference element, and that have orthologous genes clustered together in segments of sufficiently many target genomes. Note that our model allows paralogous occurrences of genes from the reference elements, and moreover, we do not require that all input genomes be represented in an output block.

Our framework is based on a bipartite graph modulation. Following the phase where the input element and genomes are parsed into (gene cluster encompassing) segments, a bipartite graph is constructed according to these segments. In this graph, vertices of one side represent subsets of reference genes, nodes in the other side represent segments from the target genomes, and edges connect the subsets of reference genes to segments from the target genomes that contain corresponding orthologs. Based on this, we cast the problem of enumerating reference-anchored gene blocks as a special type of *biclustering problem*: Compute appropriately constrained bicliques in an input bipartite graph. The constraint is a bound that ensures the co-localization of the reference genes participating in a block. When it is necessary to distinguish between segments of

different genomes, we color-code the vertices corresponding to segments in our bipartite graph, and require colorful bicliques as solutions.

1.2 Results

Since our problem translates to a computationally hard problem, we use the theory of parameterized complexity [13] which provides a convenient theoretical framework for analyzing exact algorithms for hard problems. In particular, given a bipartite graph with n vertices corresponding to references genes, and m vertices corresponding to target segments, we show how to compute all maximal constrained bicliques in $O(nm + 2^d nm/\lg m)$ time, where d is the bound on the genomic distance between two genes in a cluster. We then show how to extend this algorithm to a weighted variant of the problem with an $O(\lg d)$ increase to the time complexity. Finally, our algorithm can also be extended to the more challenging vertex-colored variant corresponding to the case where segments are overlapping sliding windows in the target genomes, yet we allow only one segment per each genome in the output. The time complexity increases in this case to $O(2^d nm)$. We also use the theory of parameterized complexity to analyze closely related biclique problems, and show that these are unlikely to admit efficient algorithms with respect to their natural parameterizations.

We implement our algorithm in a program called *RAGB Monitor* (Reference-Anchored Gene Block Monitor). This program enumerates conserved blocks that are centered in small components of a given input DNA element and ranks them according to a probabilistic p-value. The program is exemplified by applying it to the analysis of LEE, a well-known pathogenicity island from the *E. coli* genome, where it identifies components of type III secretion system from LEE that are conserved across several proteobacterial genomes.

1.3 Related Work

On the biological front, previous related works studied the evolution of operons across different species by using either experimentally validated operons, or sets of genes from a pathway of interest, as anchors. A computational method was recently proposed for generalizing such studies in [12]. This method uses alignment-based approaches to measure the distance between the gene maps of orthologous gene clusters in various species and then interprets this information against the phylogeny of the target genomes. The method assumes a model where a chromosome is considered as a permutation of distinct genes. Similarly to these works, we also base our search on the gene map of an anchor DNA element. However, in contrast to these previous works, we consider several orthologs in each target genome, per each gene in the reference element. More importantly, our biological objective is quite different: We apply an exhaustive approach, aimed to discover *all* (possibly overlapping) co-localized *subsets* of genes from the anchor reference element that are conserved as orthologous gene clusters in (possibly overlapping) *subsets* of genomes from the input set.

In this respect, our problem is related to the well-studied *gene team* discovery problem (thoroughly reviewed in [2]) that seeks conserved gene clusters in an input set of genomes. Several models were considered for this problem. In the most basic one, a chromosome is considered as a permutation of distinct genes, and a gene team is defined to be a set of genes that appear in *all* the prespecified species, possibly in a different order, yet with the distance between adjacent genes in the team for each chromosome bounded by a certain threshold. This model is generalized to consider paralogous copies of the same gene in [16]. Polynomial time exact algorithms exist for the problem variants mentioned above. The next step to further generalize the gene team problem is to find teamsthat only occur in a subset of a given set of genomes. This step makes the problem NP-complete, and several heuristic approaches were proposed for this variant [10,11]. Chateau et al. [3] modeled approximate gene clusters as cliques in a graph, where nodes represent intervals (sets of genes co-located in a genomic region) and an edge connecting two nodes indicates that their set-distance is bounded by some predefined constant. The problem introduced in our paper could be viewed as a special variant of gene team discovery, where the sought gene teams are clustered around a predefined team of "centroid" genes. The model we follow in our solution to the problem is the most general one: It allows paralogous occurrences of genes in input strings, does not require gene order conservation, and does not require that all input genomes participate in a candidate solution.

2 Problem Definition and Formulations

Let Σ denote a finite set of characters representing genes. A genome is represented by a *sequence* $S = \sigma_1 \cdots \sigma_n$ of concatenated characters $\sigma_1, \ldots, \sigma_n \in \Sigma$. For a sequence $S = \sigma_1 \cdots \sigma_n$, we use $|S| = n$ to denote the *length* of S, and $S[i] = \sigma_i$ to denote the i'th character of S. A *subsequence* of S is any non-empty sequence S' that can be obtained by deleting zero or more characters from S. An *interval* of S is a subsequence of S with consecutive characters. For $1 \leq i \leq j \leq |S|$, we let $S[i, j] = \sigma_i \cdots \sigma_j$ denote the interval of S beginning at position i and ending at position j. We call a sequence where all characters are different a *permutation*. Two sequences S_1 and S_2 are said to be *equivalent*, denoted $S_1 \equiv S_2$, if $|\{S_1[i] = \sigma : 1 \leq i \leq |S_1|\}| = |\{S_2[i] = \sigma : 1 \leq i \leq |S_2|\}|$ for all $\sigma \in \Sigma$. In other words, $S_1 \equiv S_2$ if both sequences have the same number of occurrences of each character $\sigma \in \Sigma$. Clearly, for two equivalent sequences S_1 and S_2 we have $|S_1| = |S_2|$.

Let Q denote a sequence representing our designated reference element, and let $\mathcal{T} = \{T_1, \ldots, T_C\}$ denote a set of sequences representing the target genomes. An instance of our problem is defined by a triplet $(Q, \mathcal{I}, d)$, where d is a positive integer, and $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_C\}$ is a family of interval sets where each $\mathcal{I}_i = \{T_i^1, \ldots, T_i^{t_i}\}$ contains intervals of T_i. Each interval T_i^j represents an area of interest in the target genome T_i, and d represents the length of intervals that are of interest in Q. Our goal is to find subsequences in intervals of length d

in Q, representing operons in the reference element modeled by Q, that have equivalent occurrences in areas of interest of the target genomes. We formalized this in the following definition:

Definition 1 (Block). A *block* in $(Q, \mathcal{I}, d)$ is a set of sequences $\{q, t_{i_1}, \ldots, t_{i_k}\}$ satisfying:

1. q is a subsequence of some interval of length d in Q,
2. t_{i_j} is a subsequence of some interval in $\mathcal{I}_{i_j}$ for each $1 \leq j \leq k$,
3. $i_1 \neq i_2 \neq \cdots \neq i_k$, and
4. $q \equiv t_{i_j}$ for each $1 \leq j \leq k$.

We say that a block $\{q, t_{i_1}, \ldots, t_{i_k}\}$ is *maximal* in $(Q, \mathcal{I}, d)$ if there is no other block $\{q', t'_{j_1}, \ldots, t'_{j_\ell}\}$ in this instance where q is a subsequence of q' and $\{i_1, \ldots, i_k\} \subseteq \{j_1, \ldots, j_\ell\}$.

Definition 2 (Reference Anchored Gene Blocks Problem (RAGB)). The Reference Anchored Gene Blocks problem is the problem of computing all maximal blocks in a given problem instance $(Q, \mathcal{I}, d)$.

We consider two distinct approaches to parse the intervals of our target genomes. The first approach, which we call the *sliding window* approach, is an exhaustive approach where each target genome in T_i is parsed into all its substrings of length d, *i.e.* $T_i[1, d], T_i[2, d+1], \ldots, T_i[n-d, n]$, and each such substring yields an interval in $\mathcal{I}_i$. The second approach takes into account biological signals to parse the genome into non-overlapping intervals. Another modeling option to be considered is whether we allow one or more orthologous genes in each of our input genomes; that is, whether or not our input sequences are permutations. This leads to the following two RAGB problem variants:

RAGB1. Compute reference anchored gene clusters in the following model: Intervals in $\mathcal{I}$ are parsed biologically into non-overlapping intervals. All input sequences are permutations.

RAGB2. Compute reference anchored gene clusters in the following model: Intervals in $\mathcal{I}$ are parsed via the sliding window approach. The input sequences are not necessarily permutations.

We cast both RAGB problem variants as biclique enumeration problems in bipartite graphs. The input to our framework consists of a sequence Q representing our designated genome, and $T_1, \ldots, T_C$ sequences representing the target genomes. Each genome T_i is parsed into intervals, and the ensemble of intervals from all the genomes yields the interval set $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_C\}$.

Based on Q and the intervals in $\mathcal{I}$ a bipartite graph $G = (A \uplus B, E)$ is constructed: Each node in $A = \{a_1, \ldots, a_n\}$ represents a single character in Q, such that node $a_i \in A$ corresponds to character $Q[j]$. Each node in $B = \{b_1, \ldots, b_m\}$ represents a distinct interval in $\mathcal{I}$. We then connect vertex $a_i \in A$ to vertex $b_j \in B$ iff the character $Q[i]$ appears in the interval corresponding to b_j.

We can also add a *weight measure* to this edge to indicate the level of similarity between the gene $Q[i]$ and its occurrence in the interval corresponding to b_j.

Next, for the second variant of RAGB, RAGB2, we further need to distinguish between intervals of different genomes. For this, we introduce a *coloring function* $c : B \to \{1, \ldots, C\}$ for the vertices of B, where $c(b_j) = i$ iff b_j corresponds to an interval of T_i. The reason we do not need this function for RAGB1 is that each sequence T_i is a permutation, so any character of Q can appear at most once in any of these sequences.

A *biclique* in G is a pair of non-empty vertex subsets $A' \subseteq A$ and $B' \subseteq B$ where $\{a, b\} \in E$ for each pair of vertices $a \in A'$ and $b \in B'$. We say that a biclique (A', B') is *maximal* if for any biclique (A'', B'') in G with $A' \subseteq A''$ and $B' \subseteq B''$ we have $A' = A''$ and $B' = B''$. In case G is equipped with a coloring function for the vertices in B, we say that a biclique (A', B') is *colorful* if no two distinct vertices in B' have the same color. For $1 \leq i \leq n - d$, let $A[i, i + d]$ denote the subset of vertices $\{a_i, a_{i+1}, \ldots, a_{i+d}\} \subseteq A$.

Observation 1. *There is one-to-one bijection between maximal (colorful) bicliques* (A', B') *in* G *with* $A' \subseteq A[i, i + d]$ *for some* $1 \leq i \leq n - d$ *and maximal blocks in* $(Q, \mathcal{I}, d)$.

3 Block Bicliques

In this section we present algorithms for our model for the Reference Anchored Gene Blocks problem, as well as analyze related possible models. We are interested in computing bicliques of certain properties in a bipartite graph. Since computing a biclique with a certain number of edges or vertices in a bipartite graph is NP-complete [7], any meaningful model for our problem will be NP-hard as well. Thus, we use the theory of parameterized complexity [13] to cope with this hardness.

Recall that $G = (A \uplus B, E)$ denotes a bipartite graph with $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$. For a vertex $v \in A \uplus B$, let $N(v)$ denote the set of neighbors of v, *i.e.* $N(v) = \{u : \{u, v\} \in E\}$. For a subset of vertices $A' \subseteq A$, denote the set of *common neighbors* of A' by $B_{A'} = \bigcap_{a \in A'} N(a)$. Similarly, let $A_{B'} = \bigcap_{b \in B'} N(b)$ denote the set of common neighbors of any $B' \subseteq B$. In this way, a pair of non-empty subsets $A' \subseteq A$ and $B' \subseteq B$ is a biclique in G iff $B_{A'} = B'$ and $A_{B'} = A'$. Clearly, the number of edges in a biclique (A', B') is $|A'||B'|$.

3.1 Three Biclique Problems

We next consider three possible candidates for biclique computation problems. For the sake of simplicity, we consider only decision problems for now.

Bipartite Biclique :
Input: A bipartite graph $G = (A \uplus B, E)$ and an integer k.
Question: Is there a biclique (A', B') in G with $|A'||B'| \geq k$?

Bipartite Balanced Biclique :
Input: A bipartite graph $G = (A \uplus B, E)$ and an integer k.
Question: Is there is a biclique (A', B') in G with $|A'| = |B'| \geq k$?

For a fixed positive integer d, a biclique (A', B') is called a *d-block biclique* if $A' \subseteq A[i, i + d]$ for some $1 \leq i \leq n - d$.

Block Bipartite Biclique :
Input: A bipartite graph $G = (A \uplus B, E)$ and two positive integers d and k.
Question: Is there a d-block biclique (A', B') in G with with $|A'||B'| \geq k$?

Clearly, the latter of these problems is tailor suited for the RAGB problem, but the other two might *a priori* be of use in this context as well. In Bipartite Biclique we wish to find a biclique with at least k edges, and in Bipartite Balanced Biclique we wish to find a biclique where each side has at least k vertices. Solutions to both of these problems are clearly meaningful in our context. Note that we could have also considered a third variant where the goal is to find a biclique with k vertices altogether (*i.e.* on both sides), but in the setting of parameterized complexity of which we analyze all our problems, this problem is quite similar to Bipartite Biclique.

Lemma 1. Bipartite Biclique can be solved in $O(2^k n)$ time.

Lemma 2. Bipartite Balanced Biclique *is* W[1]-hard with respect to parameter k.

Thus, the Bipartite Biclique problem is FPT with respect to parameter k, while Balanced Bipartite Biclique is not (under the widely believed assumption that FPT $\neq$ W[1]). Note however that the main issue with the Bipartite Biclique problem is that we assume that the number of edges in a solution biclique will be rather small, and can thus be taken as a parameter. This is not the case for the Block Bipartite Biclique problem. As we will see in the next section, this latter problem is fixed-parameter tractable with respect to d, which for our purposes is much smaller than the number of edges in a solution biclique. The biological motivation for RAGB1 and RAGB2 naturally yields small bounds on d.

3.2 Solving RAGB1: An Algorithm for Computing d-block Bicliques

The Block Bipartite Biclique problem trivially has a fixed parameter algorithm with respect k, the number of edges in the solution biclique, by using the

same arguments used in proving Lemma 1. We next show that this problem also has a fixed parameter algorithm with respect to parameter d, the size of the block, which is expected to be much smaller in practice than k. In fact, we will show a much stronger result in that we can compute in FPT time the set of all maximal d-block bicliques of our input graph.

Lemma 3. *Given a bipartite graph $G = (A \uplus B, E)$ with $|A| = n$ and $|B| = m$, and an integer d, one can compute the set of all maximal d-block bicliques of G in $O(nm + 2^d nm/\lg m)$ time.*

Algorithm for computing d-block bicliques:

- For each $i \in \{1, \ldots, n-d\}$ and $A' \subseteq A[i, i+d]$ do
  a. Compute the set $B_{A'}$ of common neighbors of A'.
  b. Return (A', B').

Note that the set of all bicliques produced by this algorithm contains all maximal bicliques of G. These can be easily weeded out at a post-processing stage, or during the computation of the algorithm.

To bound the running time of the algorithm above, first observe that we need $O(nm)$ time just to read the entire input. Next, notice that the algorithm has $O(n)$ iterations, where in each iteration it computes 2^d bicliques (A', B'). Starting in each iteration with bicliques (A', B') where $|A'| = 1$, and increasing the size of A' by one each time, each set of common neighbors $B_{A'}$ can be computed with a single *set intersection* operation between $N(a) \subseteq B$ and $B_{A' \setminus \{a\}} \subseteq B$ for some $a \in A'$. This set intersection operation can be naively performed in $O(m)$ time, giving a total running time of $O(2^d nm)$ to our algorithm. However, using standard bit-tricks of the RAM model, we can improve the running time of each such operation to $O(m/\lg m)$, reducing the total running time of our algorithm to the one stated in Lemma 3.

In the full version of the paper, we show how to use the "four russians technique" in order to adapt the algorithm above the case where the edges of G are weighted. This allows us to compute all maximal d-block bicliques, along with their weight, with only a factor of $O(\lg d)$ increase to the time complexity of the algorithm. Details are omitted due to space constraints.

Lemma 4. *Given a bipartite graph $G = (A \uplus B, E)$ with $|A| = n$ and $|B| = m$, a function $w : E \to \{1, \ldots, x\}$ assigning weights to the edges of G, and an integer d, one can compute the set of all weighted maximal d-block bicliques of G in $O(nm + 2^d \lg d \cdot nm/\log m)$ time.*

3.3 Solving RAGB2: The Colorful Variant

For the purposes of solving RAGB2, we consider the *colorful variant* of the Block Bipartite Biclique problem where the vertices in B have colors, and we wish to find a biclique that contains at most one vertex $b \in B$ of each color. For this purpose, let $c : B \to \{1, \ldots, C\}$ be a coloring function of the vertices

in B. Recall that a biclique (A', B') is said to be *colorful* if $c(b_1) = c(b_2)$ implies $b_1 = b_2$, for every $b_1, b_2 \in B'$. The Colorful Block Bipartite Biclique problem is the variant of Block Bipartite Biclique where we wish to find a colorful block biclique with a certain number of edges.

Unfortunately, we can no longer apply dynamic programming and the four russians trick in this case. This is because the colors make the problem harder to handle. In fact, it turns out that the colorful variants of the two other bipartite biclique problems discussed above are W[1]-hard. This is not surprising for Colorful Balanced Bipartite Biclique, as Balanced Bipartite Biclique is W[1]-hard by Lemma 2, and there is a generic parameterized reduction from any problem to its colorful variant using the color-coding technique [1]. For Colorful Bipartite Biclique we need a slightly more elaborate argument:

Lemma 5. Colorful Bipartite Biclique *is* W[1]*-hard when parameterized by* k*.*

Regardless of the above, we can still compute in $O(m)$ time a maximum sized subset $B' \subseteq B_{A'}$ in a set of common neighbors of some $A' \subseteq A$. This means that we can adapt the algorithm above to compute maximum size colorful biclique (A', B') in $O(m)$ time per each biclique. Moreover, note that in the same amount of time we can actually count the number of different colorful bicliques (A', B') corresponding to some $A' \subseteq A[i, i + d]$. This is done by a simple combinatorial computation that considers all possibilities of picking a single vertex out of each color in $B_{A'}$.

Lemma 6. Colorful Block Bipartite Biclique *can be solved in* $O(2^d nm)$ *time.*

4 Methods

We implemented the algorithm for the RAGB2 problem in a program called *RAGB Monitor* (Reference-Anchored Gene Blocks Monitor). Given the gene map of a reference element and a set of target genomes, both in GenBank file format, our program first BLASTs each gene from the reference element against each gene from the target genomes, and considers the two genes to be orthologous if their BLAST score is below 10^{-8}. Upon a successful BLAST result, genes from the target genome are re-labeled with the gene id of the corresponding gene from the reference element. If a gene in a target genome is found to be orthologous with more than one gene from the reference element, we map it to the one with the highest BLAST score. Genes from the reference element, on the other hand, are allowed to be mapped to more than one gene in each target genome. For each target genome, a sequence of gene ids is then created, consisting only of the genes that were labeled by an ortholog from the reference element genes, and preserving the gene order in the original target genome.

Our program also takes as input several parameters, including an upper bound d on the length (measured as number of genes) of an interval in the

reference element, an upper bound d' (measured as number of genes) on the length of an interval in the target genomes, and quorums q_1 and q_2 on the minimal number of anchor genes and target genomes, respectively, required in a bicluster. For segmenting the target genomes into intervals, biological segmentation is applied: The distance between two consecutive genes in an interval is bounded from above by 2000bp, and in addition, an interval length is bounded from above by parameter d'. The tool was implemented in Python 2.8.3 and the experiments performed on an Intel Xeon X5680 machine with 192 GB RAM. For a query reference element consisting of 42 genes versus 33 target proteobacterial genomes (see Sect. 5), the running time of our program ranged from 0.19 s for $d = 2$, up to 379.8 s for $d = 20$.

Finally, we define a p-value that determines the probabilistic likelihood of each gene block found. Let m denote the number of target genomes, and let n denote the length of each target genome. We define our p-value as the probability that k genes appear together in d-blocks of c out of the m genomes. We denote the probability of this event by $\Pr[k, d, c]$. Here, we assume that each genome is a permutation on $\{1, \ldots, n\}$ drawn uniformly and independently at random.

Theorem 1. *The following bound holds:*

$$\Pr[k, d, c] \leq \binom{m}{c} \left(\frac{\binom{n-k}{d-k}}{\binom{n}{d}} (n - d) \right)^c .$$

5 Preliminary Bioinformatics Results

Enteropathogenic Escherichia coli (EPEC) is a major cause of food poisoning, leading to significant morbidity and mortality. EPEC virulence is dependent on a type III secretion system (T3SS), a molecular syringe employed by EPEC to inject effector proteins into host cells [8]. The hallmark of T3SS is the needle apparatus it forms, also called "injectisome". Bacterial effector proteins that need to be secreted pass from the bacterial cytoplasm through the needle directly into the host cytoplasm. Three membranes separate the two cytoplasms: the double membrane (inner and outer membranes) of the Gram-negative bacterium and the eukaryotic membrane. The needle provides a smooth passage through those highly selective and almost impermeable membranes. The injected effector proteins subvert host cellular functions to the benefit of the infecting bacteria. A single bacterium can have several hundred needle complexes spread across its membrane. It has been proposed that the needle complex is a universal feature of all T3SSs of pathogenic bacteria. More than 15 proteins are needed to build the T3SS, some of which are highly conserved in all known T3SSs. In EPEC, the T3SS and related genes reside in several operons clustered in the Locus of Enterocyte Effacement (LEE), which is a stable pathogenicity island [5]. We exemplify our tool based on LEE (EPEC) as the reference element and on representative proteobacteria species as the target genomes.

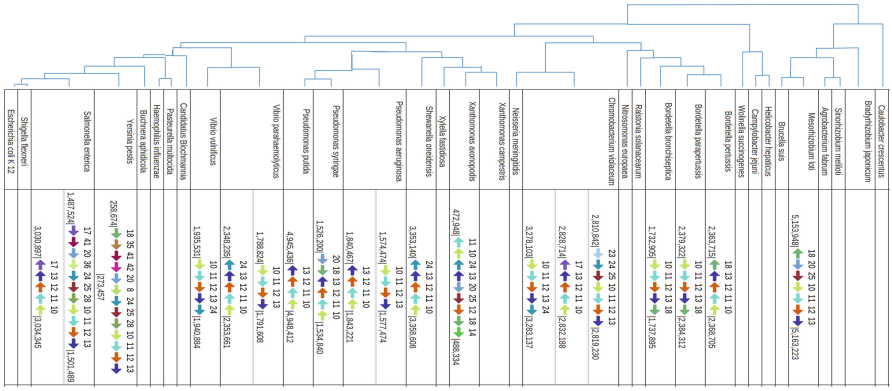

Fig. 1. The main result bicluster, anchored by genes 10–13 (escR, escS, escT, and escU) from the LEE pathogenicity Island. For each co-localized instance of the anchor genes in a target proteobacterial genome, the figure also shows additional homologs of genes from the query island that were identified within the same genomic interval.

Sequences of LEE proteins from EPEC strain O127:H6 (str. E2348/69) [6], NC 011601, were downloaded from the IslandViewer database [9], based on prediction by SIGI-HMM [17], spanning genomic coordinates 4103271-4138312 of the main chromosome. Islandviewer predicted 42 genes for this island. 33 target genomes of proteobacteria were downloaded from NCBI, as described in Ream et al. [12], who used this set of genomes in a study of the evolution of E. coli operons.

Our objective was to identify biclusters: sets of genes, that are located in LEE within a small proximity (i.e. contained within a window of size d in the gene map of LEE), and that have orthologous genes conserved within small proximity in some of the genomes of the proteobacteria species in our target set. For this, we ran sliding windows of increasing sizes on the gene map of LEE, starting with $d = 2$ (two genes) and converging at $d = 17$, after three runs in a row gave consistent clusters. For each value of d, we considered all sliding windows of length d (i.e. d consecutive genes) in the gene-map of LEE. For each sliding window, we computed all biclusters, and selected the one with the lowest p-value, computed according to the method described in Sect. 4.

The results shown in Fig. 1 were obtained for $d = 17$. The top-scoring cluster, consistently across all window sizes, was the cluster containing the four genes: 10,11,12, and 13. The p-value for this bicluster (e^{-78} for $d = 17$) was significantly lower than the p-values for all the other biclusters, based on its orthologous occurrences in 14 out of the 33 genomes in the target set. Furthermore, in some of these genomes: Chromobacterium violaceum, Pseudomonas aeruginosa, Vibrio parahaemolyticus, and Salmonella enterica serovar Typhi CT18, our tool identified two or three copies of this bicluster in distinct locations on the chromosome. To interpret this result, we checked the literature and the Virulence Factor Data Base [4]. The genes participating in the main bicluster predicted by our

program were identified as four consecutive genes within the first operon of LEE: escR, escS, escT, and escU. These genes are annotated as conserved T3SS proteins: assembly of an inner membrane complex containing these proteins might represent a critical early step in the biogenesis of the "syringe" apparatus mentioned above [14]. The other (less significant) biclusters yielded by our program were also combinations of T3SS genes.

Acknowledgements. The research of D.H. and A.B. was partially supported by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement number 631163.11, and by the Israel Science Foundation (grant No. 551145/). The research of M.Z-U. and A.B. was partially supported by the Israel Science Foundation (grant No. 179/14.) and by the Frankel Center for Computer Science at Ben Gurion University.

References

1. Alon, N., Yuster, R., Zwick, U.: Color-coding. J. ACM **42**(4), 844–856 (1995)
2. Bergeron, A., Chauve, C., Gingras, Y.: Formal models of gene clusters. Bioinform. Algorithms Tech. Appl. **8**, 177–202 (2008)
3. Chateau, A., Riou, P., Rivals, E.: Approximate common intervals in multiple genome comparison. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 131–134. IEEE (2011)
4. Chen, L., et al.: VFDB: a reference database for bacterial virulence factors. Nucleic Acids Res. **33**(suppl 1), D325–D328 (2005)
5. Deng, W., et al.: Dissecting virulence: systematic and functional analyses of a pathogenicity island. Proc. Natl. Acad. Sci. U.S.A. **101**(10), 3597–3602 (2004)
6. Elliott, S.J., et al.: The complete sequence of the locus of enterocyte effacement (LEE) from enteropathogenic escherichia coli e2348/69. Mol. Microbiol. **28**(1), 1–4 (1998)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H Freeman, New York (1979)
8. Hazen, T.H., et al.: Refining the pathovar paradigm via phylogenomics of the attaching and effacing escherichia coli. Proc. Natl. Acad. Sci. U.S.A. **110**(31), 12810–12815 (2013)
9. Dhillon, B.K.: Islandviewer 3: more flexible, interactive genomic island discovery, visualization and analysis. Nucleic Acids Res. **43**, W104–W108 (2015). gkv401
10. Kim, S., et al.: A hybrid gene team model and its application to genome analysis. J. Bioinform. Comput. Biol. **4**(02), 171–196 (2006)
11. Ling, X., He, X., Xin, D.: Detecting gene clusters under evolutionary constraint in a large number of genomes. Bioinform. **25**(5), 571–577 (2009)
12. Ream, D.C., et al.: An event-driven approach for studying gene block evolution in bacteria. Bioinform. **31**(13), 2075–2083 (2015)
13. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, New York (1999)
14. Samuel, W., et al.: Organization and coordinated assembly of the type III secretion export apparatus. Proc. Nat. Acad. Sci. **107**(41), 17745–17750 (2010)

15. Schmidt, H., Hensel, M.: Pathogenicity islands in bacterial pathogenesis. Clin. Microbiol. Rev. **17**(1), 14–56 (2004)
16. Schmidt, T., Stoye, J.: Quadratic time algorithms for finding common intervals in two and more sequences. In: Sahinalp, S.C., Muthukrishnan, S.M., Dogrusoz, U. (eds.) CPM 2004. LNCS, vol. 3109, pp. 347–358. Springer, Heidelberg (2004)
17. Waack, S., et al.: Score-based prediction of genomic islands in prokaryotic genomes using hidden Markov models. BMC Bioinform. **7**(1), 142 (2006)

http://www.springer.com/978-3-319-43680-7

Algorithms in Bioinformatics
16th International Workshop, WABI 2016, Aarhus, Denmark, August 22-24, 2016. Proceedings
Frith, M.C.; Storm Pedersen, C.N. (Eds.)
2016, XVII, 322 p. 92 illus., Softcover
ISBN: 978-3-319-43680-7