Finding Promoter Profiles with Multiobjective identification of cis-regulatory modules based on constraints

R. Romero-Zaliz
DECSAI - UGR
Granada, Spain
e-mail:
rocio@decsai.ugr.es

J. Arnedo-Fdez
DECSAI - UGR
Granada, Spain
e-mail:
arnedo@decsai.ugr.es

I. Zwir
Washington University
St. Louis, USA
DECSAI-UGR
Granada, Spain
e-mail:
igor@decsai.ugr.es

C. del Val
DECSAI – UGR
CITIC-UGR
Granada, Spain
e-mail:
delval@decsai.ugr.es

Abstract— **Gene expression regulation is an intricate, dynamic phenomenon essential for all biological functions. The necessary instructions for gene expression are encoded in cis-regulatory elements that work together and interact with the RNA polymerase to confer spatial and temporal patterns of transcription. Therefore, the identification of these elements is currently an active area of research in computational analysis of regulatory sequences. However, the problem is difficult since the combinatorial interactions between the regulating factors can be very complex. Here we present a web server that identifies cis-regulatory modules given a set of transcription factor binding sites and, additionally, also RNA polymerase sites for a group of genes.**

Multi-objective; cis-regulatory modules; gene regulation

I. INTRODUCTION

Gene expression regulation is an intricate, dynamic phenomenon essential for all biological functions. The necessary instructions for gene expression are encoded in cis-regulatory elements, or modules (CRMs) in the region upstream of a gene, the so-called promoter or intergenic region (IGR). A CRM consist of a set of transcriptional factors (TFs) that work together [1] and interact with the RNA polymerase (RNA pol) to confer specific spatial and temporal patterns of transcription. Generally, a CRM ranges from a few hundred basepairs long to a few thousand basepairs long; several transcription factors bind to it, and each of these transcription factors can have multiple binding sites. Additionally they can be influenced by the presence of repetitive elements or other elements in their neighborhood within the same IGR such as non-annotated non-coding RNAs in bacteria. A recent study has confirmed the importance of CRM functions, and revealed how subtle changes to the original arrangement of module elements can affect its function [2].

Therefore, the identification of CRMs is currently an active area of research in computational analysis of regulatory sequences [3].

Nowadays there are many available methods, which perform analysis of promoter regulatory features predicting transcription factors binding sites (TRANSFAC [4][5], JASPAR [6], RegTrans [7]), ribosomal binding sites (RBS) or RNA polymerase recognition sites [8]. Moreover, the new next generation sequencing (NGS) technologies, such as ChIP-Seq that combines chromatin immuno-precipitation (ChIP) with massively parallel DNA sequencing is able to precisely map global binding sites for any protein of interest. Concerning other cis and trans regulatory elements that could interact with TFBS and RNA polimerase are the small RNAs in bacteria and riboswitches, which are predicted with a high number of false positives making difficult a selection to test in laboratory. In definitive, this big amount of high throughput data together with the growing number of available genomes makes necessary a filter through hypothesis generation in order to make feasible experimental tests.

Our Promoter Profile Finder is able to identify profiles, which are sets of promoters (or IGRs) described by common sets of features, thus identifying genes which their expression are controlled under the same conditions. However, the problem is difficult since the combinatorial interactions between the regulating factors can be very complex and also because the vast majority of spatiotemporal relationships between transcription factors and other features remain unknown.

Here we present a tool that identifies CRMs given a set of TFs binding sites and, additionally, also other features for a group of genes. It takes into account spatial constraints on the arrangement of cis-elements such as relative position, distance and strand orientation, and deal with more than one binding site for a certain TF in the same promoter region. The web server infers these modules using multi-objective and multi-modal techniques [9]. Our algorithm is able to identify not only the most obvious solutions (i.e., the most frequent), but also those that are not locally dominated and that represent less frequent but relevant ones.

II. METHODOLOGY

A. Definitions

Association Rules [10] learning is a popular and well-researched method for discovering interesting relations between variables in large databases. This task is pursued by a modification of the Apriori algorithm [11] on Borgelt's eclat implementation [12] where candidate itemsets are counted in a pass and not generated on-the-fly. Borgelt's implementation is based on the idea to organize the counters for the itemsets in a special sort of prefix tree, which not only allows us to store them efficiently with little use of memory, but also supports the processing of transactions as well as the rule generation.

To obtain the best set of CRMs we will use a multi-objective evolutionary approach: the NSGA-II algorithm. This algorithm has been demonstrated as one of the most efficient algorithms for multi-objective optimization on a number of benchmark problems [13]. It uses non-dominated sorting for fitness assignments. All individuals not dominated by any other individuals, are assigned front number 1. All individuals only dominated by individuals in front number 1 are assigned front number 2, and so on. Selection is made, using tournament between two individuals. The individual with the lowest front number is selected if the two individuals are from different fronts. The individual with the highest crowding distance is selected if they are from the same front, i.e. a higher fitness is assigned to individuals located on a sparsely populated part of the front. There are N parents and in every iteration N new individuals (offspring) are generated. Both parents and offspring compete with each other for inclusion in the next iteration.

B. Algorithm

The developed methodology follows a set of steps. The first step is the generation of Association Rules, which implies the discovering of sets of features that appear frequently together in the set of cis-features in the input file. The Apriori's method used can be interpreted as a search problem with one objective function: the *support* of the itemset. In order to deal with multi-objective problems we have introduced a new objective to be maximized: *complexity*. Support is given by the number of genes that have the same cis-elements as the itemset, while complexity is calculated by the number of different cis-elements of the itemset. We extract itemsets from a database of transactions that maximize both support and complexity functions by post-processing the information generated by the Apriori algorithm. As the number of features and genes (transactions) grows, the number of possible solutions increases exponentially. At a certain level of features and gene combination the eclat implementation is unable to produce results halting by a memory problem.

For such cases we have developed a heuristic method. It uses an evolutionary approach to the problem of finding the best itemsets in a database of transactions. We propose a genetic algorithm (GA) based on the NSGA-II multi-objective approach. The proposed GA instantiates the NSGA-II algorithm using a simple chromosome composed by a list of module structures (e.g.,<(AP2A,0.3,D,0), (SP1,0.6,D,1), (SP1,0.9,D,110), (SP1,0.5,D,25), (SP1,0.7,D,32)>). A module structure consists of five features: name, score, orientation, pattern and distance (e.g., name: CAAT, score: 0.7, orientation: D, distance: 30: (CAAT,0.76,D,30)). Some of the features of the chromosome may be not used in some cases, for instance, when strand is ignored, then the orientation feature will be ignored. One-point crossover is used, that is, the chromosomes of the parents are cut at some randomly chosen common point (between module structures) and the resulting sub-chromosomes are swapped. Different mutation operators are used: add, erase and modify. An add mutation operator simply extends the chromosome by adding a new module structure at the beginning or the end of the chromosome. Erase mutation operator selects any module structure from the chromosome and deletes it. Finally, a modification mutation operator selects one of the features of the module structure and changes it. If the name is chosen, then a random name from the set of valid names is chosen and replaced in the module structure; if the orientation is chosen, then strand changes from D to R and vice versa. Score feature is not used in the evaluation process; therefore it is not used in any mutation operator. Finally, if distance is chosen, a small integer is added or subtracted from the one in the selected module structure from the chromosome. Again, strand and distance features are used only if the user selected them in the original parameters.

III. THE TOOL

The tool created with this methodology has been implemented in a web server: http://gps-tools2.its.yale.edu/modulos/modules.html.

A. Input

There is a large number of DNA motif finding algorithms and a lack of standards to measure their correctness. Most of the algorithms perform better in lower organisms, including yeast, as compared to higher organism [14]. For this reason, we leave to the user the selection of one or multiple DNA motif finding algorithms or databases available. These algorithms may use combinatorial enumeration, probabilistic modeling (Stormo, Gibbs, AlineACE, ANN-Spec, MEME), mathematical programming, neural networks and/or genetic algorithms (EC, GAME), while databases can be TRANSFAC, REGULON DB, JASPAR, etc as well as the programs or sources to extract information about repeats or riboswitches in the same region.

The server requires as input a file, provided by the user in csv format. Each entry of the csv input file contains the information about which features are present in which promoters. The csv file contains the following fields:

- Sequence name: promoter region identifier (e.g., E_coli_K12_flhDC).
- Feature type: free election for example "dnapat" for DNA patterns.
- Feature name: cis-feature identifier (e.g., rpod18).
- Strand: a character 'D' for direct strand or 'R' for reverse strand.
- Feature start: start position of the cis-feature.
- Feature end: last position of the cis-feature.
- Pattern: DNA pattern (e.g., TTGACA). Optional.
- Score: score given by the cis-feature prediction program or by the annotator (e.g., 0.652860653). Optional.

As input parameters, the user can select which restrictions to apply to the promoter profile search process. It is important to mention that searching for a cis-regulatory module consists of searching for two properties: a set of signals, and the spatiotemporal relationships (constraints) between this set of signals. Unfortunately, except for a small number of specific, well-characterized, interactions, the vast majority of spatiotemporal relationships (constraints) between transcription factors remain unknown. In order to select the type of constraints that could have an impact on the expression of a transcript we focused on the group previous work on RNA polymerase motifs [8], small RNAs prediction in bacteria [15] and co-expression and location of transcription factors in bacteria and human [16], [17]. These publications bring up the importance of the distance distributions (close, medium, and remote) between RNA polymerase and transcription factor binding sites in activating and repressing promoters; Thus the selected constraints were:

- *Order*. The order takes into account not only the presence of a feature but also the relative order of appearance of all the features given in a promoter.
- *Distance*: If order is taken into account, the distance between the features inside the modules can be specified as crisp or fuzzy. Crisp distances will search for modules in the input file with strict distances between their cis-features. Fuzzy distances will search for modules in the input file with distances equal or +/- the given value. Fuzzy distances ranges between 1 and 50 with a default value of 1.
- *Orientation*, which characterizes the binding boxes as either in direct or opposite orientation relative to the open reading frame.
- *Maximal number of items*: Maximal number of different cis-features (e.g., TFBS, RNApol sites) per module to search for. This value must belong to the interval [1-100] with a default value of 5. Bigger values will slow down the algorithm but will give you modules with the number of required features.

There are two versions available: the *exhaustive*, for small input files (bacteria), and *heuristic* for larger datasets (eukaryotes). In case of selecting the heuristic version there are some additional options:

- *Population size*: Evolutionary algorithms rely on a population of abstract representations (called chromosomes) of candidate solutions (called individuals) to an optimization problem, and evolve toward better solutions. Bigger population sizes will result in slower performance, but in better results. As the size of the population increases, the number of evaluations performed must also increase. The user can change the population size in the range [10-1000]. This field is mandatory and the default value is 200.
- *Number of evaluations*: usually, an initial population of randomly generated candidate solutions comprises the first generation. During each successive generation, a proportion of the existing population is selected to breed a new generation. A cost function is applied to the candidate solutions and any subsequent offspring to quantify the optimality of a solution (that is, a chromosome) in an EA so that that particular chromosome may be ranked against all the other chromosomes. Each of these cost function evaluations can be used to determine when to stop an EA execution. The user can specify the maximal number of evaluations for the EA, where values range from 1 to 99999. This field is mandatory and the default value is 200. As a rule of dumb, the number of evaluations should be a multiple of the population size. This multiple number will be approximately the number of generations to perform.

B. Output

There is an intermediate output with two links corresponding to the features.txt and modules.txt file, respectively. By right-click on any of them the user can save them in their local drive first and then can proceed to the visual inspection of the individual promoters or the obtained modules. The features file has eleven columns, the ones in the input file plus:

- *ColorRGB*: a specific color for each feature identifier.
- *IGR start*: intergenic region start position.
- *IGR end*: intergenic region end position.

The modules.txt file includes the following information for every module:

- *Module name*: module identifier (e.g., Module1).
- *Elements*: promoter region identifiers of those elements in the module.
- *Feature type*: free election for example "dnapat" for DNA patterns.
- *Feature name*: cis-feature identifier (e.g., rpod18).
- *Strand*: a character 'D' for direct strand or 'R' for reverse strand.

- *Feature start*: start position of the cis-feature.
- *Feature end*: last position of the cis-feature.

```
#Date:Thu Jun 16 01:32:36 CDT 2011
#Input:/tmp/eDDEqMyKZN.input
#Modules:presence_of_features
#Columns:Module_name;Feature_type;Feature_name;St
rand;Start;End START #Module#:1
Elements:Y_pestis_CO92;Y_pestis_KIM_flhDC Module1
Module2   dnapat    3_ygix    D         1
          15 Module2          dnapat    PhoP_3    D
Module3   dnapat    rpod18    D         55
          61 Module3          dnapat    rpod18    D
```

Figure 1. Example of module.txt output, each module contain the name of the promoter/IGR region that belong to it. In the next lines are specified the common features.

The graphical panel/s contain a detailed description of all cis-features founded in a CRM. The user can explore individually each module. All views can be saved as graph or txt in different formats such as: txt, csv, png, html, jpg. There is also a possibility of zooming down to bp level.

There are available tutorials along with example test sets. The tutorials explain in detail which parameters can be modified and cover the following help topics: input, maximal number of features to be considered in a combinatorial module, order and distance, orientation, e-mail results, and output results. The results can also be received by e-mail, where in case of error, a human-readable message is displayed.

IV. Results

The here described approach has been already successfully used to identify PhoP regulated promoters harboring more than one binding site for the TF PhoP and sharing an atypical orientation and distance of the PhoP box/boxes to the RNA polymerase site (Figure 2) [8], leading the results to the inference of PhoP transcription control over acid resistance genes in *Salmonella typhimurium* [17], [18]. These different modules have been recently proved to be related with differences in the speed of expression (submitted Zwir et al., 2011).

V. Discussion

The development of the tool presented here has been user-driven from the beginning. The here described approach has been already successfully used to identify PhoP regulated promoters harboring more than one binding site for the TF PhoP and sharing an atypical orientation and distance of the PhoP box/boxs to the RNA polymerase site leading the results to the inference of PhoP transcription control over acid resistance genes in Salmonella typhimurium [17], [18]. The server has been active for the last 2 years and the number of sequences used by de users varied very much from 10-500.

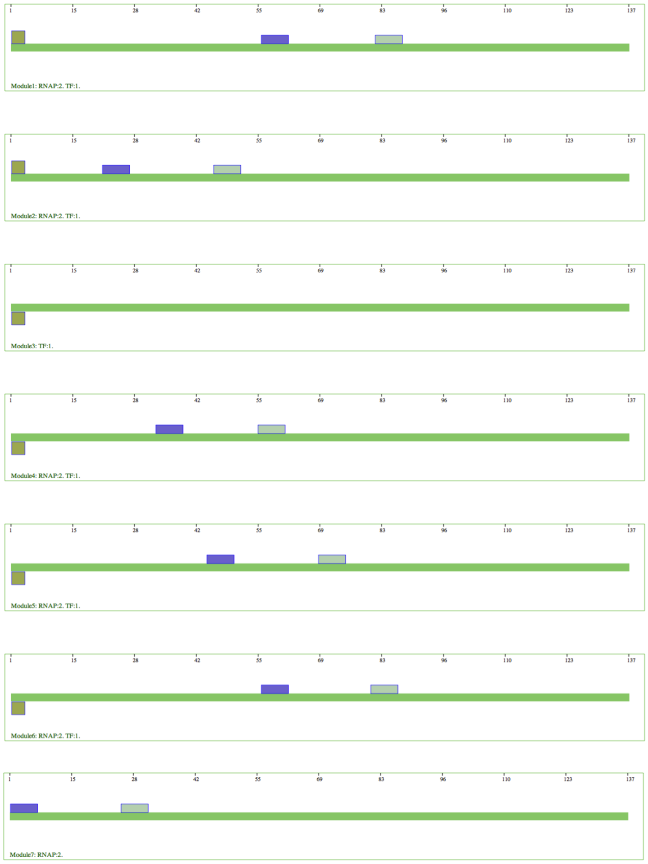

Figure 2. Graphical Module output

There are a number of tools that search for specific CRMs based on a set of co-regulated genes but they are focused in the discovery of new signals and binding sites as well as, their organization in regulatory modules. However our approach only predicts CRM from regions for which the user has provided a defined set of signals. There are only a couple of these methods but evaluating them in this scenario is difficult because there are only a few genomic regions where we are certain that all regulatory elements have been discovered. Thus it is hard to accurately estimate the false positive rate.

One of our strengths is the possibility of controlling constraints, while in the other tools any subtle change in the combination or the order of the founded/provided signals may yield different results. The other strength is that our algorithm is able to identify not only the most obvious solutions (i.e., the most frequent), but also those that are not locally dominated and that represent less frequent but relevant ones.

However the selected constraints are a very small set, due to the fact that the vast majority of spatiotemporal relationships between transcription factors remain unknown. Information of the distance, orientation and the conservation, are far from being enough to identify a functional module. To answer this request, novel analysis strategies and prediction methods that integrate sequence information and chromatin signatures could be a major step forwards.

Acknowledgment

The Spanish Ministerio de Educación y Ciencia (TIN2009-13950) and Junta de Andalucía (TIC-02788), funded this work. R. Romero-Zaliz is also supported by GENIL program (PYR-2010-28).

References

[1] E. H. Davidson, *Genomic Regulatory Systems: Development and Evolution*, 29th ed. Academic Press, San Diegelements. Nat. Genet., 2001, pp. 153-159.

[2] Document_not_found, “Document not found ([2]).” .

[3] B. P. Berman et al., “Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 2, p. 757, 2002.

[4] M. Markstein and M. Levine, “Decoding cis-regulatory DNAs in the Drosophila genome,” *Current opinion in genetics & development*, vol. 12, no. 5, pp. 601-606, 2002.

[5] E. Wingender et al., “TRANSFAC: an integrated system for gene expression regulation.,” *Nucleic acids research*, vol. 28, no. 1, pp. 316-319, 2000.

[6] A. Sandelin, W. Alkema, P. Engström, W. W. Wasserman, and B. Lenhard, “JASPAR: an open-access database for eukaryotic transcription factor binding profiles.,” *Nucleic acids research*, vol. 32, no. Database issue, p. D91--4, 2004.

[7] A. E. Kazakov et al., “RegTransBase--a database of regulatory sequences and interactions in a wide range of prokaryotic genomes.,” *Nucleic acids research*, vol. 35, no. Database issue, p. D407--12, 2007.

[8] V. Cotik, R. Romero Zaliz, and I. Zwir, “A hybrid promoter analysis methodology for prokaryotic genomes,” *Fuzzy Sets and Systems*, vol. 152, no. 1, pp. 83-102, May. 2005.

[9] K. Deb, *Multi-objective optimization using evolutionary algorithms*. New York, New York, USA: John Wiley & Sons, 2001.

[10] R. Agrawal, T. Imielinski, and A. Swami, “Mining Association Rules Between Sets of Items in Large Databases,” in *ACM SIGMOD Record*, 1993, vol. 22, no. 2, pp. 207-216.

[11] C. Borgelt and R. Kruse, “Induction of association rules: Apriori implementation,” in *Compstat: Proceedings in Computational Statistics: 15th Symposium Held in Berlin, Germany*, 2002, p. 395.

[12] C. Borgelt, “Efficient Implementations of Apriori and Eclat.”

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II.” 2000.

[14] M. K. Das and H.-K. Dai, “A survey of DNA motif finding algorithms.,” *BMC bioinformatics*, vol. 8 Suppl 7, p. S21, 2007.

[15] C. del Val, E. Rivas, O. Torres-Quesada, N. Toro, and J. I. Jiménez-Zurdo, “Identification of differentially expressed small non-coding RNAs in the legume endosymbiont Sinorhizobium meliloti by comparative genomics.,” *Molecular microbiology*, vol. 66, no. 5, pp. 1080-1091, 2007.

[16] C. Val, O. Pelz, K.-H. Glatting, E. Barta, and A. Hotz-Wagenblatt, “PromoterSweep: a tool for identification of transcription factor binding sites,” *Theoretical Chemistry Accounts*, vol. 125, no. 3-6, pp. 583-591, Oct. 2009.

[17] I. Zwir et al., “Dissecting the PhoP regulatory network of Escherichia coli and Salmonella enterica.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 8, pp. 2862-7, Feb. 2005.

[18] O. Harari et al., “Identifying promoter features of co-regulated genes with similar network motifs,” *BMC Bioinformatics*, vol. 10 Suppl 4, p. S1, 2009.