A software system to learn the relationships between over-represented motifs in a set of DNA sequences

Oksana Korol
School of Electrical Engineering and
Computer Science
University of Ottawa
Ottawa, Ontario, Canada
Email: okoro103@uottawa.ca

Marcel Turcotte
School of Electrical Engineering and
Computer Science
University of Ottawa
Ottawa, Ontario, Canada
Email: turcotte@eecs.uottawa.ca

***Abstract*—Finding relationships between motifs in a DNA sequence, such as transcription factor binding sites, is an important step to understand transcription regulation in a particular context. Current computational tools are not ideal for discovering relationships.**

We have developed a software system, ModuleInducer, which integrates motif finding with the analysis of possible interactions between them in the set of related DNA sequences using inductive logic programming. Our method was tested on synthetic and two kinds of real biological data. It has been shown to perform well as a *cis*-regulatory module finder as well as a knowledge mining tool for ChIP-Sequencing data analysis. Our method has proven to be of high suggestive value for future research by uncovering novel motif interactions in ChIP-Seq data, missed in the original study.

**The software is available for use at:
http://induce.eecs.uottawa.ca/.**

I. Introduction

The expression of genes in eukaryotic organisms is achieved by a relatively few transcription factors (about 10 times less than the number of expressed genes) [1], which hints at the importance of co-operative binding of the transcription factors. Current method of exploring this problem is ChIP-Seq experiment, which determines genome-wide bindings of one transcription factor and provides short (about 200 nucleotide) sequence areas that surround the each binding site. Computational analysis is then performed to determine transcription factor (TF) binding complexes, as well as other factors, important for gene expression, such as proximity to a particular gene, DNA methylation and flexibility.

A combination of methods is usually needed to thoroughly analyze the ChIP-Seg data. First the binding site of the immunoprecipitated transcription factor together with possible co-factor binding sites need to be located. A number of methods exist to tackle this task [2]. Some methods search for over-representation of a single motif, such as Weeder [3], Gami [4], and the recently developed DREME [5], which was shown to scale well on ChIP-Seq data. However, even though the predictive accuracy of such methods improves, they are still a subject to Futility Theorem [6], which notes that the majority of *in silico* identified motifs are found to be non-functional *in vivo*. The way to overcome this problem is to use contextual information, like sequence conservation and clustering of TFBS. The methods that fall in this category search for over-represented clusters of motifs (EEL [7], ModuleMiner [8], CREME [9]). However, the largest shortcoming of such methods is that they are unable to detect regulatory modules that are based not on over representation of motifs or clusters of motifs, but on an association of motifs in the cluster. In other words they are unable to discover the underlying structure of the module by ways of finding over-represented relationships between motifs. Figure 1 illustrates one such hypothetical situation, when the experiment and control data both yield the same frequency of single motif occurrence as well as the clusters of several motifs. By closely examining the data, a human expert might notice that motif M1 is always found before motif M2 in the experiment sequences, but not in control. However, current module finding methods will not be able to distinguish these two datasets and due to large size of data generated by ChIP-Seq analysis, visual inspection of the results is often not feasible. A method by Segal and Sharan [10] attempts to circumvent this limitation by looking for spatial motif combinations. However this is done in the windows of pre-determined size, so if a relationship exists between the motifs found at different windows, it will not be found.

In addition, questions like sequence composition and proximity to a particular gene have to be addressed separately from the TFBS cluster analysis.

We provide a method that is able to discover relational information in transcription factor binding complexes as well as combine it with sequence related information, such as CG content and genomic location. Our method, ModuleInducer, uses inductive logic programming (ILP) to determine a set of rules (theory) that distinguish the experiment data from the control. The results can be easily interpreted by a non-computer science expert and have shown to be useful to guide future research in the case of ChIP-Seq study of cancer

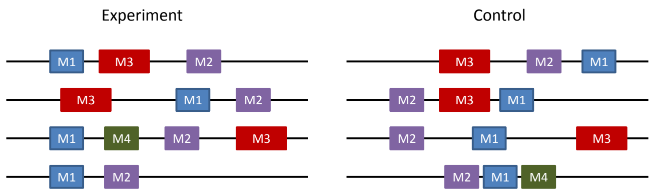

Fig. 1. Example of experimental data that will be hard to classify. M1, .. M4 are hypothetical transcription factors (TFs), matched inside experiment and control sequences. Single TFs are not over-represented in experiment vs. control data, neither are clusters of any TF combinations. The difference lies in the order of TF binding, i.e. M1 always precedes M2 in all experiment and none of the control sequences.

expression in humans by Palii et al. [11]. The method was tested on synthetic and two types of real biological data. Apart from ChIP-Seq data analysis, the method can be applied to finding *cis*-regulatory modules in a set of related DNA sequences.

The combination of ILP with current motif finding methods offers an improvement of motif prediction in the context of the individual experiment.

II. Methods

A. Overview

Our approach, ModuleInducer, is an end-to-end analysis application that is able to describe related experimental sequences in terms of relationships between commonly found biological markers. The input to our application is a set of related DNA sequences, obtained in an experiment such as ChIP-Seq, and an optional set of control sequences. ModuleInducer uses the existing motif finding method DREME [5] to discover and locate frequently occurring motifs. It then utilizes inductive logic programming to induce a theory (a set of logical rules), which describes the experiment sequences.

The application can be divided into three main modules: user interface, data management, and ILP engine. The high level diagram of the module interaction and data flow through the system is presented in Figure 2. Each module is described in detail in the following sections.

B. User Interface

ModuleInducer is a web application, which can be accessed at http://induce.eecs.uottawa.ca/. The minimal information, required by the interface, is a set of experiment sequences in a FASTA format. Control sequences could be either supplied or generated. Possible motifs could also be either supplied in a position specific scoring matrix (PSSM) format or automatically discovered using DREME [5]. The user interface also exposes some of the advanced application parameters, such as DREME matching score and ILP accuracy parameters.

C. Data Management

The Data Management module offers several ways to extract the necessary data for the ILP engine. Control sequences, if not supplied, can be randomly generated with the same composition, as experiment sequences or using a Markov chain order 1 model. There are two ways to locate motifs inside the sequences:

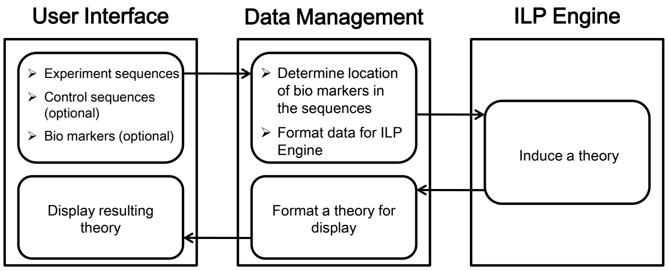
User Interface
Data Management
ILP Engine
Experiment sequences
Control sequences (optional)
Bio markers (optional)
Display resulting theory
Determine location of bio markers in the sequences
Format data for ILP Engine
Format a theory for display
Induce a theory

Fig. 2. High Level Project Diagram. The data flow is as follows. Positive and negative sequences and possible biological markers (i.e. PSSMs for TFBSs) are entered through User Interface. For simplification purposes, other system parameters are omitted in this diagram. The data are sent to Data Management, where position of each biological marker in each sequence is determined and resulting data are formatted for ILP. ILP Engine is then invoked, which induces a theory based on the positive and negative examples and system defined background knowledge. The resulting theory is then formatted in the Data Management and sent to User Interface for display.

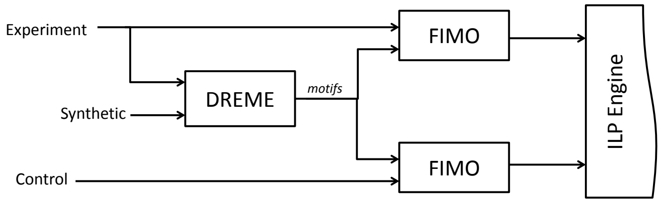

Fig. 3. DREME/FIMO Data Flow. Experiment and Control are two sets of sequences, supplied by the user. Synthetic is the DREME default background (see [5]).

1) When biological markers are supplied, Patser [12] is automatically invoked to find their location in experiment and control sequences.
2) If no biological markers are supplied, ModuleInducer invokes DREME [5] to discover them in experiment sequences and then calls FIMO [13] to find the location of discovered motifs in both experiment and control sequences. The data flow of this scenario is presented in Figure 3.

Sequence names, motif names, their location in the sequence and other related information is encoded in logic programs and passed on to ILP Engine to induce a theory. Once the ILP engine is invoked and the result is processed, the Data Management module takes care of the formatting of the theory to display it to the user.

D. ILP Engine

The term inductive logic programming was first introduced by Stephen Muggleton in 1991 [14]. It works by deriving a general description from specific instances. The input to the ILP system is the background knowledge B, a set of positive examples E^+ and a set of negative examples E^-. The output is a theory T (also called hypothesis) that describes all positive and none of the negative examples in terms of the background knowledge. All data in ILP, including examples, background knowledge, as well as induced hypothesis are represented as logic programs, consisting of rules of the following form:

$$g \leftarrow b_1, b_2, ...b_n$$

where $g, b_1, b_2, ...b_n$ are predicates. For example, a rule for hydrocyanic acid (HCN) will be: $hydrogen\_cyanide(x) \leftarrow sungle\_bond(x, H, C), triple\_bond(x, C, N)$.

In ModuleInducer, positive and negative examples are the user supplied experiment and control sequences respectively, which were transformed into logic programs by the Data Management module. The background knowledge is a set of interactions that are possible between motifs in a DNA sequence. If we think of the DNA sequence as time and motifs as temporal intervals, then we can represent these interactions as relationships between temporal intervals, described by Allen [15]. Since the search space of the ILP algorithm grows exponentially with the addition of every background knowledge rule, specifying all possible interactions would not be feasible. Therefore we have selected a subset of interactions, based on our knowledge and discussions with experts. The background rules are described in a general way, as generic interactions. For example, an interaction "before" describes a relationship between two motifs such that one motif is located before another in a sequence. Other background rules include distance interval between two motifs (determined at the time of learning), motif belonging to a sequence, sequence location on a particular chromosome, . When the ILP engine is executed, all possible rules and their combinations are tested for specific motif locations in every supplied sequence. Thus a theory is formed consisting of the subset of background rules instantiated with the subset of relevant motifs. Therefore apart from providing a description of the experiment data, ILP has the ability to root out *in silico* discovered motifs in the context of the experiment.

The ILP Engine is implemented on the Aleph [16] system.

E. Availability

ModuleInducer is available for use at http://induce.eecs.uottawa.ca/.
Source code of the framework is available upon request from the authors.

III. Results

ModuleInducer can be thought of as an automated expert analyzing the motifs, found in the experiment sequences and presenting the results in the human-readable relational form.

A. Improving the quality of motif matches

A large number of *in silico* found motifs do not have a biological function, therefore taking the context of the experiment into consideration is a useful way to weed out false positive matches. A recent algorithm, DREME, allows to supply the control sequences, as opposed to using synthetic background, to improve the quality of the matches. ModuleInducer, which relies on DREME to discover motifs in experiment sequences, uses DREME default background sequences (see [5]). However, the ILP Engine contrasts the motif information, found in experiment sequences with the supplied control sequences and hence finds a theory that discriminates experiment against control. We have run ModuleInducer on the jurkat vs. erythroid data from Palii et al. [11] and compared the motifs selected in the results with the DREME run on the same data.

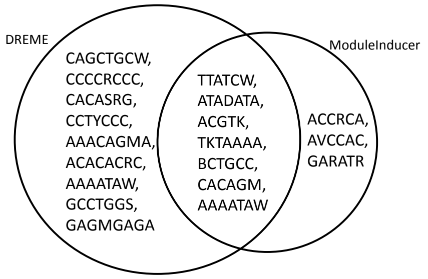

Fig. 4. Motifs excluded by DREME and ModuleInducer on experiment vs. control dataset compared to experiment vs. synthetic dataset.

We found that compared to the default DREME run of jurkat vs. synthetic, MI excludes 10 motifs, found in the former run. While executing DREME on jurkat vs. erythroid excludes 15 motifs. Out of 10 motifs, excluded by MI, 7 were also excluded by DREME jurkat vs. erythroid run (see Figure 4). This supports our expectation that MI is able to eliminate some irrelevant motif matches.

B. Discovering the underlying structure

To verify that our method can uncover the underlying structure of motif complexes, we have run it on a set of sequences known to contain muscle regulating *cis*-regulatory modules, as described in [17]. The clusters of motifs inside these sequences were identified by authors as muscle regulating modules if a cluster contained at least two motifs, if the distance between motifs was no more than 40 nucleotides and if at least two motifs corresponded to muscle-specific binding sites. We have executed ModuleInducer in the mode where we supply the suspected motifs as position specific scoring matrices (PSSM). The matrices that we have supplied were identified in the study and ranked according to their muscle-specificity. The resulting theory is presented below.

```
[theory]

[Rule 1] [Pos cover = 10 Neg cover = 1
positive(A) :-
   has_feature(A, m1, 'R'),
   distance_interval(A, m9, m16, 20, 100).

[Rule 2] [Pos cover = 5 Neg cover = 0]
positive(A) :-
   distance_interval(A, m4, m3, 40, 10).
```

Our method was able to recover the assumptions of the original study. The rules cover a total of 5 motifs (`m1`, `m9`, `m16`, `m4`, `m5`). Four out of these motifs (`m1`, `m3`, `m4` and `m16`) correspond to the top 5 muscle-specific motifs, identified in the study. The Rule 2 identifies the distance between `m3` and `m4` to be 40 nt +/- 10 nt.

C. Synthetic data

To verify that our method is capable of discovering patterns in the data, we executed it on a specially designed synthetic sequences. Inside the sequences we have planted two motifs at random distances but one always located before the other. The dataset consisted of 10 experiment and 15 control sequences of length 200, to keep the noise associated with the change occurrence of one of the two selected motifs to a minimum.

We repeated the experiment with this data 60 times and found that 57% of times the induced theory contained `before` term with the two planted motifs, with 37% of theories consisting of only the `before` rule. The 43% of theories that did not contain `before` term, contained instead a `distance_interval` rule with the planted motifs. This is because the randomly generated data for these theories placed the PSSMs at a particular distance, which is not surprising, given the small amount of sequences that we tested. Based on this result we conclude that our method can uncover significant rules in the data.

We have also repeated this experiment for the larger number of sequences, to investigate how our method would perform on a more realistic dataset. Running it with 600 positive and 600 negative examples of length 200 a total of 20 times, we have found that 41% of the runs contained a planted `before` term, while 14% contained the whole rule. The 58% contained either `distance_interval` term, similar to the previous experiment, or `has_feature` term with one of the two planted PSSMs. Thus we also obtain confidence that introducing noise to the data, by means of larger number of sequences, our approach is still able to discover significant rules.

D. Human data

We have used ModuleInducer to further analyze the data from Palii et al. [11]. The study looked at the expression of the leukemia cancer in humans with relation to TAL1 transcription factor binding. The experiment data was taken from jurkat cell lines, derived from the patients with cancer, and the control consisted of erythroid cell lines taken from the healthy patients. When looking for the binding pattern of TAL1 relative to other TFs, the computational analysis focused on the distance between TAL1 and three other significant factors. However, finding a biologically significant rule that contrasted jurkat with erythroid data has proven to be difficult. Expanding the search to rules other than distance and including more TFs would have taken a lot of time and most likely yielded similar results.

We have executed ModuleInducer on jurkat vs. erythroid data in a mode where we do not supply PSSMs for suspected binding sites, but let DREME discover over-represented motifs. Consistent with the original findings, the theory did not contain rules with high (>2%) coverage. However it contained a number of rules that could be interesting for future experiments. GREAT [18] was used to find out if certain rules were associated with genes, corresponding to specific GO terms. We found that the Rule 1, presented below, correspond to abnormal bone marrow cell morphology/development, while Rule 2 corresponds to abnormal T cell differentiation in mouse phenotype.

```
[Rule 1]
positive(A) :-
before(A, ’CAGCTGCW’, ’CCCCRCCC’),
before(A, ’CAGCTGCW’, ’GCCTGGS’),
has_feature(A, ’CCTYCCC’),
has_feature(A, ’CTTCCTBY’),
has_feature(A, ’CAGCTGCW’, ’R’).

[Rule 2]
positive(A) :-
before(A, ’CTTCCTBY’, ’ARAGAAA’),
has_feature(A, ’CACASRG’),
has_feature(A, ’MGGAARY’).
```

These findings suggest that our method is able to discover specific rules of transcription factor binding in the context of the experiment.

IV. Conclusion

We have presented ModuleInducer - an application aimed at discovering relationships between biological markers in a set of related sequences. This problem is broadly defined and can include a variety of important biological questions, such as finding *cis*-regulatory modules, studying the impact of transcription factors binding on the gene transcription in the context of the experiment, or analyzing the results of ChIP-Sequencing experiments. We have applied our method in all these contexts as well as tested it on a specially designed synthetic dataset, which has shown that it is able to discover significant patterns in the data. In addition, there are several advantages to using our method.

First, it is an end-to-end solution, which requires for an input as little as a set of related DNA sequences. However, providing a set of real experimental control sequences, instead of relying on the automatically generated synthetic ones, will provide a more realistic result. Supplying a set of suspected motifs (i.e. transcription factor binding sites) in the form of PSSMs is also possible.

Second, MI does not require a large number of co-associated motifs to provide a classification. The method is aimed at discovering the most general rules (covering as many examples, as possible), describing the data. However, if high coverage of individual rules is not possible, the resulting theory will contain a large number of rules with lower coverage. This ability can also prove useful in case when no significant patterns exist in experiment data, compared to the control, as discovered in the Section III-D. Less significant rules can also be found useful at describing the data, especially when combined with other analysis, such as gene proximity.

Third, current methods that predict transcription regulation by discovering over-represented motifs or their clusters in the context of expression data. Our method is able to discover over-represented relationships between motifs and other data (such as chromosome binding). Thus it is able to discover the rules where none of the motifs or their cluster is significant, while the specific interactions between the motifs, perhaps combined with other metadata is significant. In the case of comparing the performance of our method to DREME executions (III-A), this can explain why the number of motifs, eliminated in ModuleInducer run is smaller then the DREME, since DREME looks at the over-representation of single motifs, while ModuleInducer might find the combination of weak motifs significant.

One important feature of our method is extensibility. In the recent study by Karczewski et al. [19] the authors place a special significance on incorporating different analysis, other than motif association, to understand the nuances of TF binding. In one respect our method already does it, by including the information of chromosomal location of the motifs in the background knowledge (in addition to the usual motif association rules). However the big advantage of our method compared to other classification algorithms is the ability to add other types of analysis, such as functional information about protein-protein interaction and the proximity to genes, without modifying the format of the input or any significant pre-processing. We are currently working on extending our pipeline to include GO ontology and the algorithms capable of proving the information on gene proximity of the analyzed motifs.

Limitation of our method is long running time, which is the result of a computational complexity of ILP. For instance, the running time of the run on the Palii et al. [11] dataset, which included 2238 experiment and 5707 control sequences took about 5 hours on a dual core 2.5 GHz processor with 4G of RAM. The factors that contribute to this are the number and length of analyzed sequences, number of PSSMs, number of rules in the background knowledge and Aleph search parameters. Since the size of the dataset is determined by the biological experiment, the ways to address the execution time may include including fewer rules in the background knowledge or modifying Aleph search parameters.

Acknowledgment

We thank Marjorie Brand (The Sprott Center for Stem Cell Research, Department of Regenerative Medicine, Ottawa Hospital Research Institute and Department of Cellular and Molecular Medicine, University of Ottawa, Ottawa, Ontario, Canada) for sharing her data with us, for her contribution to the selection of the biologically relevant background knowledge, and for inspiring conversations.

References

[1] A. Reményi, H. R. Schöler, and M. Wilmanns, "Combinatorial control of gene expression," *Nature Structural and Molecular Biology*, vol. 11, p. 812815, 2004.

[2] P. Van Loo and P. Marynen, "Computational methods for the detection of *cis*-regulatory modules," *Briefings in Bioinformatics*, vol. 10, no. 5, pp. 509–524, 2009. [Online]. Available: http://bib.oxfordjournals.org/content/10/5/509.abstract

[3] G. Pavesi, G. Mauri, and G. Pesole, "An algorithm for finding signals of unknown length in DNA sequences." *Bioinformatics*, vol. 17, pp. s207–s214, 2001.

[4] C. Congdon, C. Fizer, N. Smith, H. Gaskins, J. Aman, G. Nava, and C. Mattingly, "Preliminary results for GAMI: A genetic algorithms approach to motif inference," *Computational Intelligence in Bioinformatics and Computational Biology*, 2005.

[5] T. L. Bailey, "DREME: Motif discovery in transcription factor ChIP-Seq data," *Bioinformatics*, 2011.

[6] W. W. Wasserman and A. Sandelin, "Applied bioinformatics for the identification of regulatory elements," *Nature Reviews Genetics*, vol. 5, pp. 276–287, 2004.

[7] O. Hallikas, K. Palin, N. Sinjushina, R. Rautiainen, J. Partanen, E. Ukkonen, and J. Taipale, "Genome-wide prediction of mammalian enhancers based on analysis of transcription-factor binding affinity," *Cell*, vol. 124, pp. 47–59, 2006, eEL.

[8] P. Van Loo, S. Aerts, B. Thienpont, B. De Moor, Y. Moreau, and P. Marynen, "ModuleMiner - improved computational detection of *cis*-regulatory modules: are there different modes of gene regulation in embryonic development and adult tissues?" *Genome Biology*, vol. 9, 2008, moduleMiner.

[9] R. Sharan, I. Ovcharenko, A. Ben-Hur, and R. M. Karp, "CREME: a framework for identifying *cis*-regulatory modules in human-mouse conserved segments," *Bioinformatics*, vol. 19, p. i283i291, 2003.

[10] E. Segal and R. Sharan, "A discriminative model for identifying spatial *cis*-regulatory modules," *Journal of Computational Biology*, vol. 12, pp. 822–834, 2005.

[11] C. G. Palii, C. P.-I. Z. Yao, Y. Cao, F. Dai, J. Davidson, H. Atkins, D. Allan, F. J. Dilworth, R. Gentleman, S. J. Tapscott, and M. Brand, "Differential genomic targeting of the transcription factor TAL1 in alternate haematopoietic lineages," *The EMBO Journal*, p. 116, 2010.

[12] G. Z. Hertz and G. D. Stormo, "Identifying DNA and protein patterns with statistically significant alignments of multiple sequences," *Bioinformatics*, vol. 15, pp. 563–577, 1999.

[13] C. E. Grant, T. L. Bailey, and W. S. Noble, "FIMO: Scanning for occurrences of a given motif," *Bioinformatics*, vol. 27(7), p. 10171018, 2011.

[14] S. H. Muggleton, "Inductive logic programming," *New Generation Computing*, vol. 8, p. 295318, 1991.

[15] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, pp. 832–843, 1983.

[16] A. Srinivasan, "The Aleph manual. version 4 and above." 2007. [Online]. Available: http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/aleph.html

[17] G. Zhao, L. A. Schriefer, and G. D. Stormo, "Identification of muscle-specific regulatory modules in caenorhabditis elegans," *Genome Research*, vol. 17, pp. 348–357, 2007.

[18] C. Y. McLean, D. Bristor, M. Hiller, S. L. Clarke, B. T. Schaar, C. B. Lowe, A. M. Wenger, and G. Bejerano, "GREAT improves functional interpretation of *cis*-regulatory regions," *Nature Biotechnology*, vol. 28(5), pp. 495–501, 2010.

[19] K. J. Karczewski, N. P. Tatonettia, S. G. Landt, X. Yang, T. Slifer, R. B. Altman, and M. Snyder, "Cooperative transcription factor associations discovered using regulatory variation," *PNAS (Proceedings of the National Academy of Sciences of the United States of America)*, vol. 108, p. 1335313358, 2011.