A METHODOLOGY FOR MOTIF DISCOVERY EMPLOYING ITERATED CLUSTER RE-ASSIGNMENT

Osman Abul* † and Finn Drabløs

Department of Cancer Research and Molecular Medicine, Norwegian University of Science and Technology, Trondheim, Norway
**Email: osman.abul@ntnu.no*
finn.drablos@ntnu.no

Geir Kjetil Sandve

Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway
Email: sandve@idi.ntnu.no

Motif discovery is a crucial part of regulatory network identification, and therefore widely studied in the literature. Motif discovery programs search for statistically significant, well-conserved and over-represented patterns in given promoter sequences. When gene expression data is available, there are mainly three paradigms for motif discovery; *cluster-first, regression*, and *joint probabilistic*. The success of motif discovery depends highly on the homogeneity of input sequences, regardless of paradigm employed. In this work, we propose a methodology for getting homogeneous subsets from input sequences for increased motif discovery performance. It is a unification of *cluster-first* and *regression* paradigms based on iterative cluster re-assignment. The experimental results show the effectiveness of the methodology.

1. INTRODUCTION

Transcription Factors (TF) are proteins that selectively bind to short pieces (5-25nt long) of DNA, so called *Transcription Factor Binding Sites* (TFBS). Although TFs bind in a selective way they allow some degeneracy in their binding sites, forming *Transcription Factor Binding Motifs* (TFBMs) or just *motifs*. This property creates the TFBS representation problem, *i.e.* the choice of language in which motifs are expressed. The most common representations are *motif consensus* over IUPAC codes, *mismatch strings* and *position specific weight matrices* (PSWMs), as well as their variants and specializations.

Finding TFBMs is an important step in elucidation of genetic regulatory networks[a]. There are basically two methods for finding TFBMs, experimental and computational, although they usually benefit from each other. ChIP-chip experiments can analyze the genome-wide binding of a specific TF. For instance, Lee *et al.* [19] have conducted experiments for over 100 TFs for experimental identification of regulatory networks of *Saccharomyces cerevisiae*. Unfortunately their resolution ($\approx$ 1K-nt) is not enough to exactly identify binding locations. Other problems include condition specific binding, measurement noise, and difficulty in finding an optimal consensus motif. TFBMs are functional elements of genes and preserved throughout the evolution. This property, together with available completed genetic maps of many species, has made possible computational identification based solely on sequence data. That is, since these regions have accumulated very few mutations compared to non-functional parts, it is possible to find them computationally by just exploiting the statistical over-representation. Computational ap-

*Corresponding author.
†This work was carried out during the tenure of an ERCIM fellowship.
[a]Regulatory network identification methods are also studied without explicitly focusing on use and discovery of TFBMs[33, 27, 36, 28, 24], in this paper we do not cover these approaches.

proaches built around this fact include *MEME* [2, 1], *BioProspector* [20], *AlignACE* [12], *Consensus* [10], and *MDScan* [21], among many others.

TFs bind to respective TFBSs in promoter regions of their target genes. Each gene can have a number of TFBSs for several different TFs in its promoter sequence. In Eukaryotes, TFBSs are organized in *modules*; sets of TFBSs for a number of TFs. Each TF can function as inducer or repressor and this process is combinatorial, *i.e.* depends on the qualitatively and quantitatively binding of other TFs. This combinatorial behavior can cause non-additive expression behavior for their common targets. In general, intra-module couplings are much stronger than inter-module couplings. Expression behavior also depends on the genome-wide global conditions.

To understand the governing rules for gene expression, we need to know 1) all TFs, 2) abundance and activity of them under varying conditions, 3) their binding sites, and 4) their combinatorial joint regulation of target expression [35, 9]. From this, it is clear that to induce regulatory networks computationally we need both sequence and functional data. Typically, the sequence data employed is the inter-genic promoter regions upstream of transcription start sites while the functional data is obtained from microarray experiments under various conditions. Other useful sources of data for motif (and module) discovery include ChIP-chip experiments (*e.g.* [3]), TFBM databases (*e.g.* [26]), and phylogenetic relations (*e.g.* [14]).

The success of motif discovery programs depends on the quality of input data. That is, they typically give high false-positives/negatives if input genes are heterogenous with respect to regulation. To make the input genes homogeneous, genes are clustered before they are presented to motif discovery programs; hence this is called the *cluster-first* approach. This is because gene expression depends on combinatorial binding of TFs on TFBMs. The co-expressed genes are assumed to be co-regulated, therefore genes are clustered based on their expression profile similarity over a course of microarray experiments. Each cluster (in which sequences are highly probable to contain homogeneous TFBMs) is given as input to motif finding programs (*MEME, BioProspector, MDScan* etc.).

An alternative to the cluster-first approach is to start from a large set of putative motifs and filter them by regressing on expression data. The idea behind this approach is to remove non-relevant motifs and thereby reduce the number of false positives. Examples of this approach include *Reduce* [7], *Motif Regressor* [8, 21], a boosting approach also employing ChIP-chip data (Hong *et al.* [15]) and a logic regression approach by Keleş *et al.* [17].

Although a number of algorithms and programs have been developed for motif discovery, little has been done on designing a methodology for optimal usage. In particular, little attention is paid to the selection of homogeneous subsets from heterogeneous gene sets of interest. In practice, what an experimenter does is 1) cluster the gene sets of interest (using a clustering program like *k-means, hierarchical clustering, Self-organizing maps*, etc), then 2) input them to one or a few motif finding programs, and finally 3) decide on the true motifs among all the candidates, either by further analysis (like regression) or manually. Though clustering before motif discovery improves homogeneity compared to random subsets, it might fail in finding true clusters. Motivated by this, we here study the generation of homogeneous clusters using both sequence and expression data, and we address the issue of methodology for motif discovery.

We define an iterative procedure (a methodology) for the motif discovery process. Briefly, we start with an initial clustering of gene sets from gene expression data and find motifs in these clusters. We then (optionally) refine these motifs by filtering out irrelevant ones. In this step, simple filtering or filtering employing regression analysis is applied. After that, we screen all the genes by motif profiles of each cluster and refine clusters by re-assignment based on screening score. Following this, we restart motif discovery on the new gene clusters and iterate this procedure until convergence. Finally, we output the set of motifs found in the last iteration.

2. POWERING MOTIF DISCOVERY USING GENE EXPRESSION DATA

The three main paradigms for incorporating gene expression data into motif discovery are *cluster-*

first, *regression* and *joint probabilistic*.

Brazma *et al.* [6] presented one of the earliest methods within the cluster-first paradigm. They look for over-represented oligos with limited degeneracy, both genome-wide and for clusters generated from gene expression clustering based on the time series data. The approach taken by Beer *et al.* [5] also use a cluster-first approach. The genes are clustered using expression data with *k-means* clustering and *AlignACE* [12] is used for motif discovery. A very similar approach using a custom clustering algorithm is presented in [23].

A variant of the cluster-first approach is TFCC (Transcription Factor Centric Clustering) of Zhu *et al.* [37]. The idea is to find a set of genes showing similar expression profiles to the expression profile of a particular TF over a set of expression experiments, and then look for motifs in that cluster using *AlignACE* [12]. Similarly, Hvidsten *et al.* [13] find similar genes to a selected gene using the expression data, and construct logical rules (in the form of *if-then* rules) in terms of the absence/presence of *a priori* given motifs. The objective in this approach is not to find novel motifs but motif modules. An indirect cluster-first approach is presented in Tamada *et al.* [34] where the objective is finding regulatory networks. Motif discovery is an intermediate step used to refine the network. Briefly, they construct a regulatory network from gene expression data, and from the induced network they identify TFs and search for motifs in the sequence data of subtrees of TFs.

One of the earliest work using regression on gene expression data for motif discovery is the *Reduce* method of Bussemaker *et al.* [7]. The objective is to find the best minimal set of motifs (*K*-mers) capable of explaining the gene expression data. The method uses single gene expression experiments over which oligo scores are linearly regressed. The model is fit in an iterative manner, *i.e.* starting with an empty set and adding the most significant motif to the model in each iteration until there is no statistical improvement. Similarly, Conlon *et al.* [8] introduced a linear regression method called *Motif Regressor*. They employ *MDScan* [21] to extract features, sets of candidate motifs, from sequence data. From the resulting large number of putative motifs the insignificant ones are eliminated through regression. The *LogicMotif* approach of Keleş *et al.* [17] uses two-step logistic regression on a single gene expression experiment. In the first step, the set of all over-represented oligos (allowing limited degeneracy) in the input sequences are identified as candidate motifs. In the second step, for each sequence a binary score vector (serving as a covariate vector) is constructed in which each entry corresponds to existence of a motif type (or a logical function of a subset of all motif types, a so called logic tree) and this vector is regressed on expression data. The *Rim-Finder* system of Zilberstein *et al.* [38] is another method using the regression approach. Identification of synergistic effects of pairs of motifs using co-expression has also been studied [26].

Methods for binary regression (classification) have also been developed. A large-margin classification approach, called *Medusa*, using boosting together with alternating decision trees is given in [22]. Likewise, the recent study by Hong *et al.* [15] presents a boosting approach for motif discovery. They formulate the problem of motif discovery as a classification of ChIP-chip data, and find motifs accordingly.

The idea of using a *joint probabilistic* paradigm was first proposed by Holmes and Bruno [11]. The idea is to model probabilistic interactions between sequence features (motifs) and expression data. The approach has been extensively studied by Segal *et al.* [30, 29, 32, 31, 4] on a few model variants all employing Bayesian reasoning. The basic variant assumes that transcriptional modules are dependent on sequence information and that they in turn determine gene expression. The approach learns transcriptional module motif profiles using the *Expectation-maximization* (EM) algorithm. Another similar probabilistic clustering algorithm jointly using sequence and time series expression data is presented in [18], where each cluster represent transcriptional modules and in turn determine motif profile and gene expression of genes in the modules. However, they assume an initial set of motifs given *a priori* and assign motif profiles to modules after clustering finishes, *i.e.* as a post-processing step.

3. A MOTIF DISCOVERY METHODOLOGY

In the cluster-first approaches, clustering based on gene expression data is assumed to represent true

functional clusters. Due to the noise in data, uncertainty of the number of clusters and lack of true knowledge of optimal distance measures, the results only partially represent true clusters. It is also the case that genes with TFBSs for the same TF are not necessarily co-expressed during a specific time-course as gene expression is combinatorial and therefore depends on several factors.

To explore the claims above we have conducted experiments on some subsets of *S.cerevisiae* gene clusters reported by Harbison *et al.* [9] and on genome-wide gene expression data by Gasch *et al.* [25]. More information is provided for these datasets in the Experiments section. In Figure 1 we show the *Silhouette index* of true clusterings and clustering induced by *k-means* clustering for two subsets. *Silhouette index*, ranging -1 to 1, measures how similar a point is to points in its cluster compared to points in other clusters. Larger index values indicate good cluster separation. The results agree with our claims that genes having similar motifs need not be co-expressed, and that co-expression clustering therefore can be deceptive for motif discovery.

On the other hand, as shown in Figure 2, clustering (using gene expression) before motif discovery improves the quality of discovered motifs. In the analysis, we have used *MDScan* for motif discovery and we have selected random subsets with 500 genes from over 6000 genes of the Gasch *et al.* dataset. The number of clusters is 5. Note also that, Figure 2:b scores are higher than Figure 2:a scores. This makes sense as selection of homogeneous clusters instead of random clusters gives better candidates for motif discovery, as already discussed.

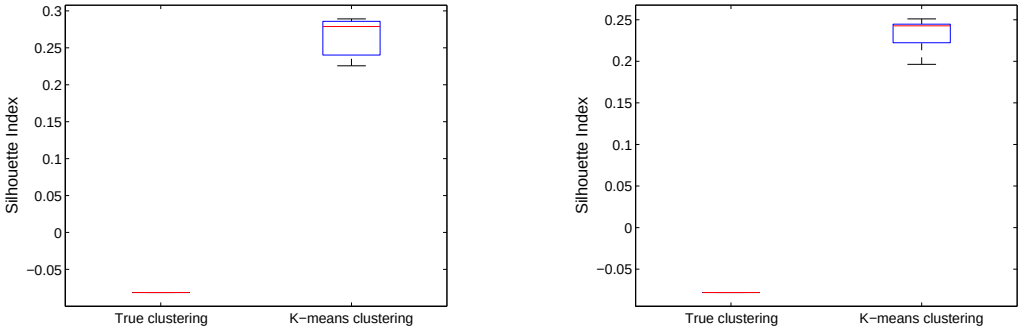
To get the advantages of gene expression clustering, while at the same time avoiding its potential deceptiveness, we propose a methodology for discovering regulatory motifs using both gene expression and upstream sequence data.

The methodology is illustrated in Figure 3. It starts with the initial clustering from gene expression data. Following this, a motif discovery algorithm is used to find candidate motifs for each cluster. Then, these motifs are optionally regressed over the gene expression values for motif filtering; only the significant motifs are retained. Since motif discovery is applied to each cluster, the discovered motifs can be regarded as motif profiles of their respective clusters. Motifs from the motif discovery algorithm are assumed to be putative and further refined by filtering. In this way, only significant and relevant motifs are kept. After that, the motif profiles are used to screen all the genes; *i.e.* a score for each gene for each motif profile is computed. Based on the motif scores, genes are re-assigned to clusters. The idea here is that if a gene is closer to the motif profile of a different cluster rather than its current one, then its cluster membership should be changed based on this new evidence. These steps (motif discovery, filtering, screening and cluster re-assignment) are iterated until the clusters converge and a final set of motifs are output as motif profiles for each cluster. Note that, we do not force the use of any particular clustering, motif discovery, filtering or screening algorithm.

We will now define a basic vocabulary to be used in the remaining parts of this section. Let $G = \{G_g\}_{g=1}^{g=|G|}$ be the set of genes and $E = \{E_e\}_{e=1}^{e=|E|}$ be the set of gene expression experiments (either time series or various treatment conditions). Our input data is DNA sequence data extracted from regions upstream of transcription start site and gene expression data. Define $S = \{S_g : g \in G\}$ as the sequence data such that $S_g = \{S_{gl} : l = 1, \ldots, |S_g|\}$ where $S_{gl} \in \{A, C, G, T\}$ is the nucleotide in the l'th position and $|S_g|$ is the length of the sequence for g, respectively. Finally, define the gene expression matrix as $Y = \{Y_g^e : g = 1, \ldots, |G|; e = 1, \ldots, |E|\}$ where Y_g^e is the pre-processed gene expression value for gene g under experiment e. For convenience, we also define Y_g as the expression vector for g over all experiments and Y^e as the expression vector for e over all genes.

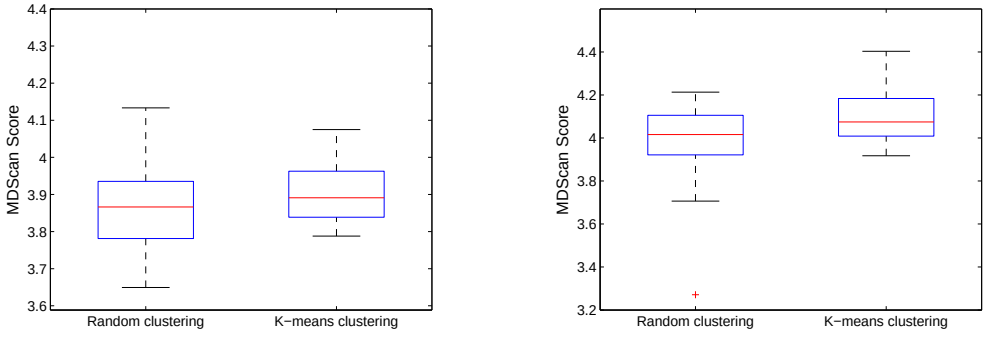
3.1. Clustering

The input to the clustering step is Y. The task is to partition G into a given number of partitions based on similarity computed from Y_g. In the literature a number of clustering algorithms for gene expression data have been designed and employed. A crucial point in clustering is to decide on the clustering method (*k-means*, *self-organizing maps* and *hierarchical clustering* are more common) and similarity measure (*e.g. Euclidean distance*, *Mahalanobis distance*, *Pearson correlation*). For instance, in [37] and [5], modified *k-means* algorithms with Pearson corre-



a) gene cluster {CBF1,FHL1,BAS1,INO4,MBP1} b) gene cluster {CBF1,BAS1,MBP1,MSN2,REB1}

Fig. 1. *Silhouette index* scores



a) random 500 genes b) gene cluster {CBF1,BAS1,MBP1,MSN2,REB1}

Fig. 2. *MDScan* scores

lation coefficient are used. It is also important to decide on the number of clusters. Some methods estimate the number of clusters by applying model selection (*e.g.* using cross-validation [18], adaptive-quality based clustering [23]). Since the hierarchical clusters are exploratory and flexible they are usually the preferred choice.

Since the clustering step is done once in our approach, and serves only as a source of good initial clusters, we leave the selection of clustering algorithm to the user, as different clustering algorithms may be optimal depending on the specific dataset. We denote initial clustering results as $\{C_c^1\}_{c=1}^{c=|C^1|}$, where $|C^1|$ is the number of clusters.

3.2. Motif Discovery

The motif discovery methods basically differ in their motif representation (*e.g.* IUPAC codes, regular expressions, PSWMs) search algorithm (*e.g.* Gibbs sampling, Expectation-maximization, word counting), and exploitation of biological knowledge (*e.g.* fixed/flexible gaps, bi-modality, palindromic motifs, motifs in modules, inter-dependence of motif locations).

For our purpose, any PSWM based motif finding method like *AlignACE, MEME, MDScan* and *BioProspector* can be used in this step. We apply the motif discovery algorithm for each cluster separately and independently. Let us denote the clusters at the i'th step by C^i and the motif set output from cluster C_c^i by M_c^i.

3.3. Motif Filtering

Given the resulting motifs M_c^i of the motif discovery step, the filtering step outputs a subset of the motifs denoted by $M_c^{i'}$.

The reason we introduce a filtering step is because statistical over-representation does not necessarily imply biological significance. In other words, some statistically over-represented motifs may be ei-

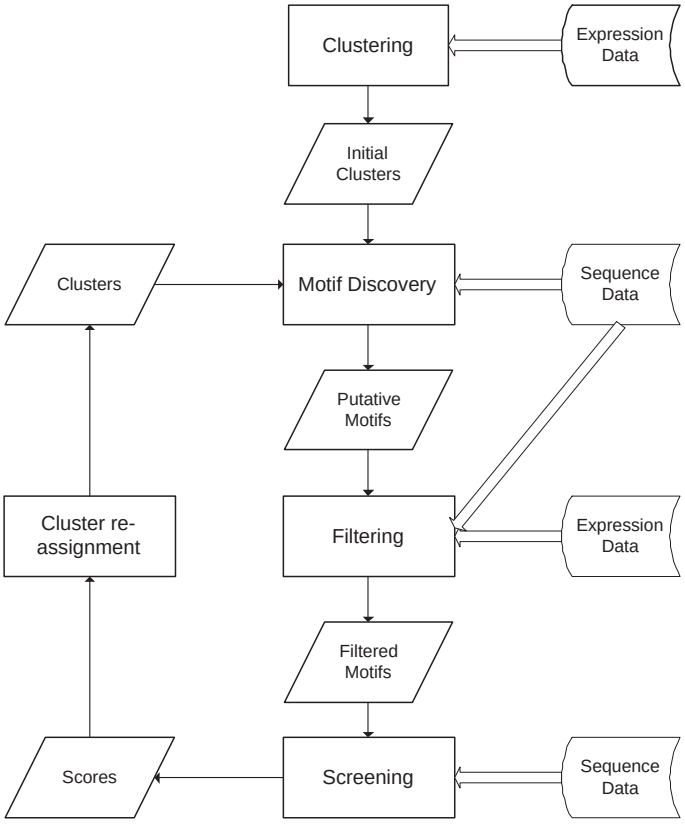


Fig. 3. Iterative Motif Discovery Approach.

ther artifacts of the motif discovery program or simple tandem repeats. As those artifact motifs are not generally consistent with expression, this filtering step has the potential of eliminating most of them. Since in regression based approaches the success depends highly on the initial putative motifs, feeding these programs with the output of statistically over-represented motifs usually give better results. Note that it is also possible to have a simple filter that does not employ expression data, *e.g.* filtering nothing or filtering putative motifs based only on their motif scores.

3.4. Screening and Cluster Re-assignment

Given the motif profile $M_c^{i'}$ of each cluster, we score all the genes, including the genes in other clusters, measuring the conformance of them to the motif profile. This creates a vector of cluster motif profile conformance measures for each gene. We use the matrix similarity score metric reported in Kel *et al.* [16]. The metric basically uses the information content of PSWM and scales each *k-mer* within maximum possible and minimum possible match.

The genes are assigned to the cluster to which they have highest conformance, thus creating a cluster re-assignment. If the cluster re-assignment is the same or very similar to previous clustering, the set of motifs for each cluster is output, otherwise the iteration continues with the gene clusters C^{i+1}.

4. EXPERIMENTS

To assess the merit and relevance of the methodology presented we conduct several experiments on real datasets for *S.cerevisiae*.

We use the gene expression dataset by Gasch *et al.* [25]. The dataset contains over 150 gene expression arrays (measured under several conditions with repetitions) for 6371 ORFs of *S.cerevisiae*. We pre-process the dataset by *log* transforming the background corrected intensities. Since the dataset contains missing values, we eliminate arrays and genes with considerable number of missing entries. This gives 149 arrays and 6107 ORFs, which can be con-

sidered as a 6107×149 matrix. There are still missing values in this matrix and we impute these missing values with $k - nn$ imputation method. The method, for each missing-valued gene, identifies closest k genes over non-missing entries and then imputes the missing value by the average of column values for the k genes. After this we get a complete expression matrix. As for the sequence data, we use at most 500 (-500 to -1) base sequences from upstream of transcription start site for all of the 6107 genes.

There are many alternative methods and tools that can be used in different steps of our methodology. Since our objective here is to show the effectiveness of it, we do not experiment with an extensive set of methods and tools, but rather a few practical ones. In all of the experiments conducted we have selected *k-means* as clustering and *MDScan* as motif discovery algorithms, and use either a trivial identity or linear regression based filters. *k-means* and *MDScan* algorithms have been chosen mainly because they are fast. This is particularly important for the choice of motif discovery algorithm, as it is run for each cluster in every iteration. Although *MDScan* is originally designed for ChIP-chip experiments, it also works well without ChIP-chip data.

As performance measures we use *MDScan* scores, *Convergence*, *Jaccard index* and *Silhouette index*. *MDScan* score is used to quantify the strength of motifs within clusters. In cases where experimentally determined binding sites for motifs are available, the correspondence between predicted and known sites could have been used as performance measure. We rather preferred *MDScan* score as it is more objective and more general. *Convergence* is defined as the number of re-assigned genes so it is a natural metric for our methodology. We use *Silhouette index* and *Jaccard index* as cluster separation and similarity metrics, respectively.

4.1. Random Clusters

In this experiment, we randomly select 500 genes among 6107 genes and cluster them into 5 clusters by *k-means* clustering. For each cluster we use the same parameter setting for *MDScan* as follows (and default values for other parameters);

- motif width=8
- number of motifs to report=2
- number of top sequences to seed=20
- number of motifs to be kept for refinement = 4× number of motifs to report

The order of genes presented to *MDScan* is relevant. In our experiments we have used random orders to avoid any bias (this is because we do not use ChIP-chip data). On the other hand we conjecture that the genes could have been sorted based on distance to their cluster centroids, thereby possibly improving motif discovery.

Figure 4 gives the results for 20 runs (Iteration 1 is the result for the initial *k-means* clustering). In all of these runs we employ *k-means* as the initial clustering algorithm and use trivial identity filters. In all of the runs we start from converged *k-means* results at iteration number 1. From the figure we see that the number of re-assigned genes decreases along iterations, suggesting a convergence. *MDScan* scores of clusters also increases with the iterations. It is clear from both figures that our approach is able to correct some deceptiveness of the initial clustering.

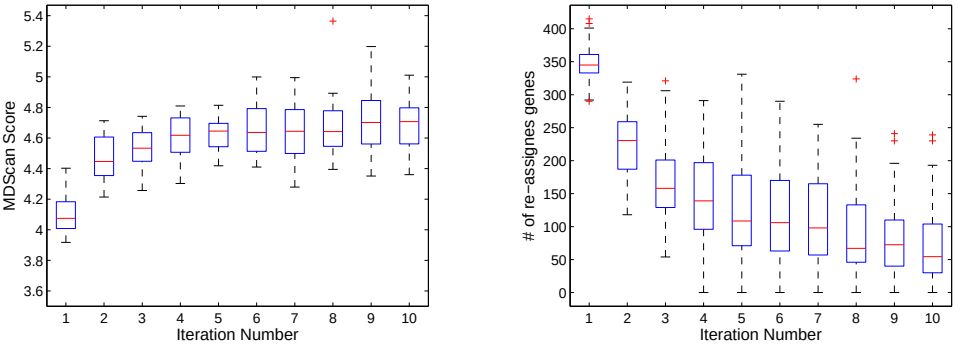
We have tested how sensitive our methodology is to initial clustering by running with random initial clusterings. We have also tested the approach by changing the random gene numbers, number of clusters and *MDScan* parameters. In all of these cases, similar results are observed to those reported in Figure 4. This means that improvement of the methodology is not dependent on particular settings of initial clusters or motif discovery tools.

4.2. Harbison et al. Clusters

Harbison *et al.* [9] identified *S.cerevisiae* target genes for a number of TFs by collecting results from the following resources, ChIP-chip data, published data from literature and phylogenetic conservation. As a result, they defined several dozens of (overlapping) gene clusters for each binding motif. They also confirmed the results by applying several motif discovery programs (*MEME, MDScan, AlignACE*, etc.). We therefore assume these clusters as true clusters for our purpose.

We conduct experiments with the following three gene subsets drawn from Table 1;

- Subset 1: {CBF1,FHL1,BAS1,INO4,MBP1} (5 clusters),



a) *MDScan* scores b) Convergence

Fig. 4. Performances results on random clusters

- Subset 2: {CBF1,BAS1,MBP1,MSN2,REB1} (5 clusters)
- Subset 3: {CBF1, REB1} (2 clusters)

As a pre-processing, we remove genes not contained within the 6107 genes of the Gasch *et al.* dataset. We also remove genes found in more than one cluster. This way we ensure that each gene belongs exactly one cluster. As a result, subsets 1, 2, and 3 contain 399, 403, and 253 genes, respectively.

Table 1. Clusters used in experiments

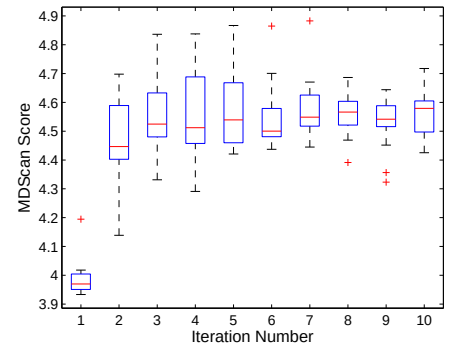
Regulator	# of genes in the cluster
CBF1	195
FHL1	131
BAS1	17
INO4	32
MBP1	92
MSN2	74
REB1	99

We show general utility of the methodology on the subsets 1 and 2. Number of clusters parameter for the *k-means* is set to 5. *MDScan* parameters used are same as given in Section 4.1 and trivial identity filter are employed.

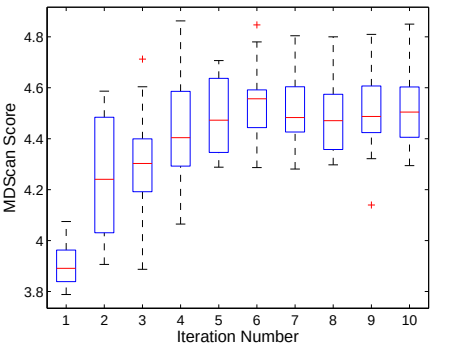
In Figure 5:a-d , the average *MDScan* scores and convergence performances are shown for subset 1 and 2 over 20 runs. The results clearly indicate the improved *MDScan* scores and convergence over iterations. With the same parameters, the true clustering for subset 1 (2) has *MDScan* score of 4.00 (4.14) and number of re-assigned genes is 230 (248). This additionally shows that even though their performance is better than *k-means* clustering, there is a potential to increase performance by cluster re-assignment.

Figure 5:e-f shows the *Silhouette index* of clusterings for subset 1 and 2 through iterations and also for the true clustering. We reason from the figure that our method achieves the similar *Silhouette index* as true clusters while *k-means* clustering (iteration number 1) destroys the original clustering. To measure the cluster similarity between the true clustering and clusterings over iterations we measure the *Jaccard index*. This index, ranging from 0 to 1, is an external cluster validation method measuring how similar a clustering is to another. A high index value indicates high cluster similarity. The results which are shown on Figure 5:g-h are inconclusive. This might be due to the similarity in sequences (like *TATA box*) in different clusters of the true clustering, and failure of *MDScan* to find specific motifs for the considered clusters. For instance, it finds specific binding sites for true clusters of CBF1, FHL1, INO4, MBP1 and REB1 while it fails to find specific binding sites for true BAS1 and MSN2 clusters.

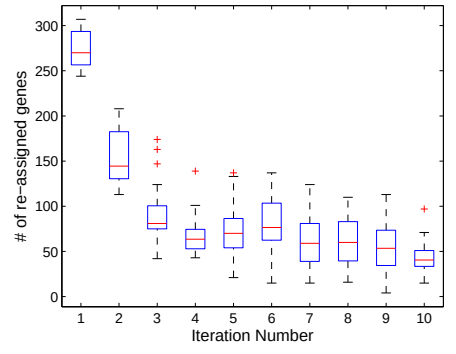
We test the effect of using different filters on subset 3. Filter 1 is a trivial identity filter, while Filter 2 and 3 are stepwise multiple linear regression filters. The only difference between Filter 2 and 3 is the scoring function for computing the covariate vector; Filter 2 uses the function defined in [16] and Filter 3 uses the the one employed in *Motif Regressor*. For the case of Filter 1 the number of motifs reported by *MDScan* is set to 2, while for Filter 2 and 3 it is set to 15 and the other parameters are the same as in Section 4.1. The number of clusters parameter for *k-means* is set to 2 for all filters. Since



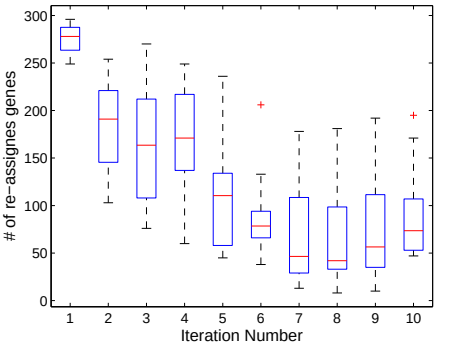
a) *MDScan* score



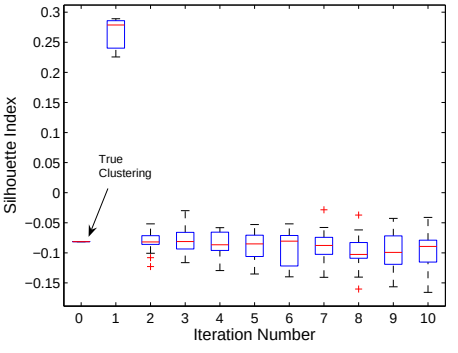
b) *MDScan* score



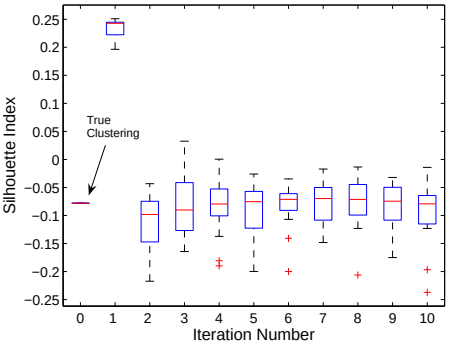
c) Convergence



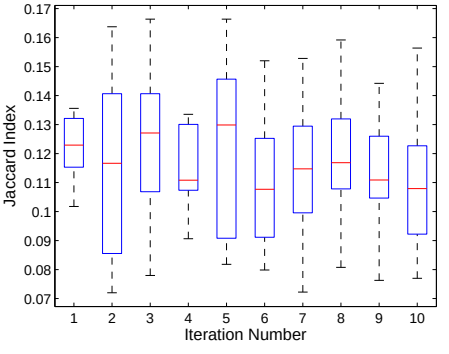
d) Convergence



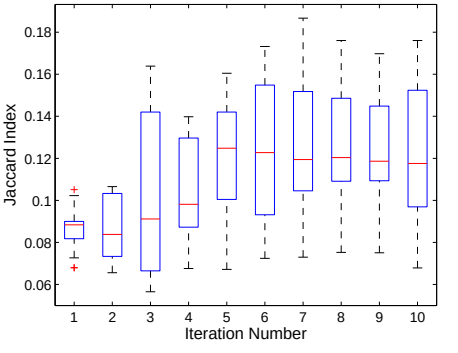
e) *Silhouette index*



f) *Silhouette index*

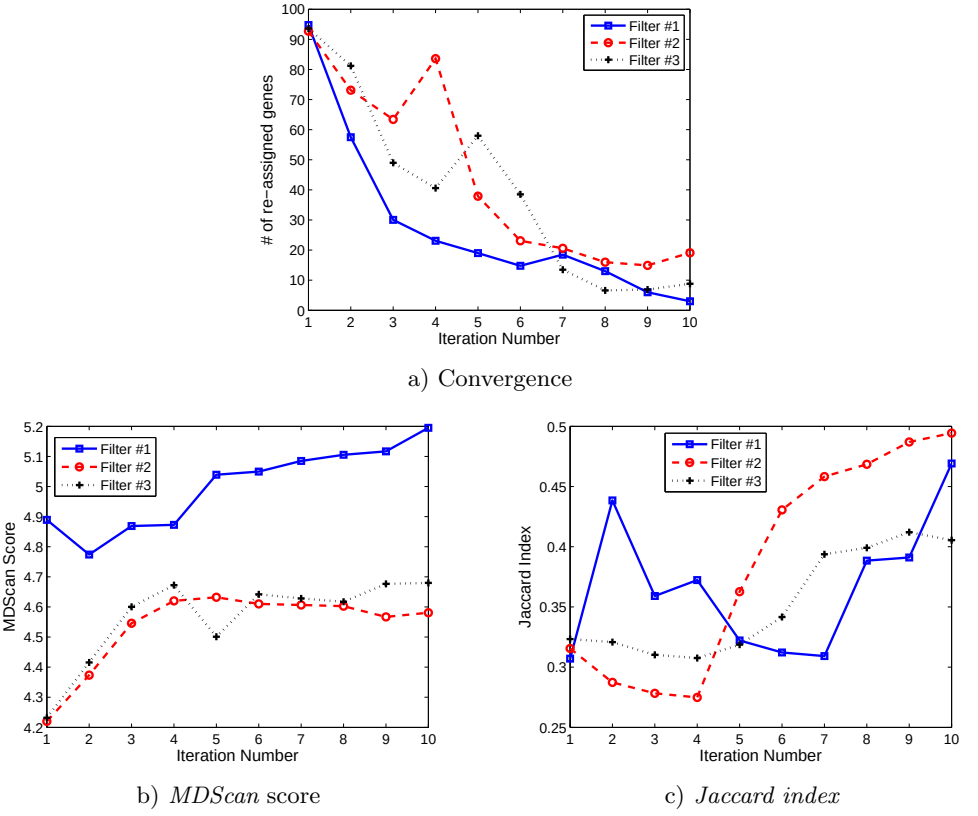


g) *Jaccard index*



h) *Jaccard index*

Fig. 5. Performance results for Subset 1 (left) and Subset 2 (right)



a) Convergence

b) *MDScan* score

c) *Jaccard index*

Fig. 6. Performance results for subset 3

we have 149 expression experiments, we regress once for each experiment and count the motifs selected in the stepwise regression. Finally, we use the 2 most frequently selected motifs out of 15 (*i.e.* 13 motifs are filtered out) as a cluster motif signature and use them for cluster conformance computation in the re-assignment step.

Figure 6 gives the averaged results over 20 runs. It is clear that all the performance parameters show increased performance over the *k-means* clustering for all filters used. Note that in Figure 6:a, there is full convergence, and from Figure 6:c, the *Jaccard index* increases suggesting recovery of true clustering for all filters. Recall that the last issue was inconclusive from subsets 1 and 2. We conclude from the results that the methodology works well for all of these filters. It is also possible to compare filters based on the performance parameters. For the average *MDScan* scores, Filter 1 scores best while in terms of *Jaccard index* Filter 2 scores slightly better. This clearly shows the tradeoff of selecting the filter among multiple filters. Note that, this is why we have introduced a filtering step in our methodology.

5. CONCLUSION

In this work, we have addressed the problem of developing a methodology for motif discovery. It is organized around the idea of getting highly homogeneous gene clusters using both sequence and expression data. We do this by screening all genes and re-assigning clusters in several iterations.

The analysis and experimental results show that clustering based on gene expression is a better basis for motif discovery than random clustering, but not perfect. It is also shown that it might mislead. Our method is developed to compensate these two issues and thereby improve the quality of motif discovery. The conducted experiments clearly suggest the utility of our approach.

The methodology is quite flexible, *e.g.* not developed around a particular motif discovery, filtering, screening or clustering algorithm. In other words, a

broad range of algorithms developed in the field can be used in our methodology.

The methodology presented here can also be considered as a unification of the cluster-first and regression based motif discovery paradigms into a single framework. Our approach is similar to the joint probabilistic approaches, especially to Tamada *et al.* [34] where their main motivation is finding regulatory networks rather than discovering motifs. However, it is in general different from these approaches, in that our approach does not establish any probabilistic relationships between gene expression and sequence information.

We have also shown the importance of the filtering step. It has been shown that regardless of the actual filtering method used, the methodology works well, *i.e.* improves over the initial clustering.

Future work will focus on assessment of general utility and performance of our methodology as compared to *joint probabilistic* modeling.

References

1. Timothy L. Bailey and Charles Elkan. The Value of Prior Knowledge in Discovering Motifs with MEME. In *Proc. of the ISMB'95*, Menlo Park, CA, 1995.
2. Timothy L. Bailey and Charles Elkan. Unsupervised Learning of Multiple Motifs in Biopolymers using Expectation Maximization. *Machine Learning*, (21):51–80, 1995.
3. Yoseph Barash, Gal Elidan, Nir Friedman, and Tommy Kaplan. Modeling Dependencies in Protein-DNA Binding Sites. In *Proc. of the 7th International Conf. on Research in Computational Molecular Biology*, Berlin, Germany, 2003.
4. Alexis Battle, Eran Segal, and Daphne Koller. Probabilistic Discovery of Overlapping Cellular Processes and Their Regulation. In *Proc. of 9th RECOMB*, San Diego, CA, 2004.
5. Michael A. Beer and Saeed Tavazoie. Predicting gene expression from sequence. *Cell*, 117:185–198, 2004.
6. Alvis Brazma, Inge Jonassen, Jaak Vilo, and Esko Ukkonen. Predicting Gene Regulatory Elements in Silico on a Genomic Scale. *Genome Research*, 8:1202–1215, 1998.
7. Harmen J. Bussemaker, Hao Li, and Eric D. Siggia. Regulatory Element Detection using Correlation with Expression. *Nature Genetics*, 27:167–171, February 2001.
8. Erin M. Conlon, X. Shirley Liu, Jason D. Lieb, and Jun S. Liu. Integrating Regulatory Motif Discovery and Genome-wide Expression Analysis. *PNAS*, 100(6):3339–3344, 2003.
9. Harbison CT, Gordon DB, Lee TI, Rinaldi NJ, Macisaac KD, Danford TW, Hannett NM, Tagne JB, Reynolds DB, Yoo J, Jennings EG, Zeitlinger J, Pokholok DK, Kellis M, Rolfe PA, Takusagawa KT, Lander ES, Gifford DK, Fraenkel E, and Young RA. Transcriptional regulatory networks in Saccharomyces Cerevisiae. *Nature*, 431(7004):99–104, 2004.
10. Gerald Z. Hertz and Garry D. Stormo. Identifying DNA and Protein Patterns with Statistically Significant Alignments of Multiple Sequences. *Bioinformatics*, 15(7/8):563–577, 1999.
11. Ian Holmes and William J. Bruno. Finding regulatory elements using joint likelihoods for sequence and expression profile data. In *Proc. of Eighth International Conference of Intelligent Systems for Molecular Biology*, pages 202–210, 2000.
12. Jason D. Hughes, Preston W. Estep, Saeed Tavazoie, and George M. Church. Computational Identification of Cis-regulatory Elements Associated with Groups of Functionally Related Genes in Saccharomyces Cerevisiae. *Journal of Molecular Biology*, (296):1205–1214, 2000.
13. Torgeir R. Hvidsten, Bartosz Wilczynski, Andriy Kryshtafovych, Jerzy Tiuryn, Jan Komorowski, and Krzysztof Fidelis. Discovering Regulatory Binding-site Modules using Rule-based Learning. *Genome Research*, (15):856–866, 2005.
14. Shane T. Jensen, Lei Shen, and Jun S. Liu. Combining Phylogenetic Motif Discovery and Motif Clustering to Predict Co-regulated Genes. *Bioinformatics*, 21(20):3832–3839, 2005.
15. Katherina J. Kechris, Erik van Zwet, Peter J. Bickel, and Michael B. Eisen. A Boosting Approach for Motif Modeling using ChIP-chip Data. *Bioinformatics*, 21(11):2636–2643, 2005.
16. A.E. Kel, E. Gobling, I. Reuter, E. Cheremushkin, O.V. Kel-Margoulis, and E. Windenger. MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Research*, 31(13):3576–3579, 2003.
17. Sündüz Keleş, Mark J. van der Laan, and Chris Vulpe. Regulatory Motif finding by Logic Regression. *U.C. Berkeley Biostatistics Working Paper Series*, (145), 2004.
18. Anshul Kundaje, Manuel Middendorf, Feng Gao, Chris Wiggins, and Christina Leslie. Combining sequence and time series expression data to learn transcriptional modules. *IEEE Transactions on Computational Biology and Bioinformatics*, 2(3):194–202, 2005.
19. T. Lee, N. Rinaldi, F. Robert, D. Odom, Z. Bar-Joseph, G. Gerber, N. Hannett, C. Harbison, C. Thompson, I. Simon, J. Zeitlinger, E. Jennings, H. Murray, D. Gordon, B. Ren, J. Wyrick, J. Tagne, T. Volkert, E. Fraenkel, D. Gifford, and R. Young. Transcriptional regulatory networks in Saccharomyces Cerevisiae. *Science*, (298):799–804, 2002.

20. X. Liu, D.L. Brutlag, and J.S. Liu. Bioprospector: Discovering Conserved DNA Motifs in Ppstream Regulatory Regions of Co-expressed Genes. In *Proc. of Pacific Symposium on Biocomputing*, 2001.
21. X. Shirley Liu, Douglas L. Brutlag, and Jun S. Liu. An Algorithm for Finding Protein-DNA Binding Sites with Applications to Chromatin-Immunoprecipitation Microarray Experiments. *Nature Biotechnology*, 20:835–839, 2002.
22. Manuel Middendorf, Anshul Kundaje, Mihir Shah, Yoav Freund, Chris H. Wiggings, and Christina Leslie. Motif Discovery through Predictive Modeling of Gene Regulation. In *Proc. of 9th RECOMB*, Cambridge, MA, 2005.
23. Yves Moreau, Gert Thijs, Kathleen Marchal, Frank De Smet, Janick Mathys, Magali Lescot, Stephane Rombauts, Pierre Rouze, and Bart De Moor. Integrating Quality-based Clustering of Microarray Data with Gibbs Sampling for the Discovery of Regulatory Motifs. *JOBIM 2002*, pages 75–79, 2002.
24. Naoki Nariai, Yoshinori Tamada, Seiya Imoto, and Satoru Miyano. Estimating Gene Regulatory Networks and Protein-protein Interactions of Saccharomyces Cerevisiae from Multiple Genome-wide Data. *Bioinformatics*, 21(2):206–212, 2005.
25. Audrey P.Gasch, Paul T. Spellman, Camilla M. Kao, Orna Carmel-Harel, Michael B. Eisen, Gisela Storz, David Botstein, and Patrick O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11:4241–4257, 2000.
26. Yitzhak Pilpel, Priya Sudarsanam, and George M. Church. Identifying Regulatory Networks by Combinatorial Analysis of Promoter Elements. *Nature Genetics*, 29:153–159, 2001.
27. Jiang Qian, Jimmy Lin, Nicholas M. Luscombe, Haiyuan Yu, and Mark Gerstein. Prediction of Regulatory Networks: Genome-wide Identification of Transcription Factor Targets from Gene Expression Data. *Bioinformatics*, 19(15):1917–1926, 2003.
28. John Jeremy Rice, Yuhai Tu, and Gustavo Stolovitzky. Reconstructing Biological Networks using Conditional Correlation Analysis. *Bioinformatics*, 21(6):765–773, 2005.
29. E. Segal, R. Yelensky, and D. Koller. Genome-wide Discovery of Transcriptional Modules from DNA Sequence and Gene Expression. *Bioinformatics*, 19(1):273–282, 2003.
30. Eran Segal, Yoseph Barash, Itamar Simon, Nir Friedman, and Daphne Koller. From Promoter Sequence to Expression: A Probabilistic Framework. In *Proc. of 6th RECOMB*, Washington, DC, 2001.
31. Eran Segal, Dana Peer, Aviv Regev, Daphne Koller, and Nir Friedman. Learning Module Networks. *Journal of Machine Learning Research*, 6:557–588, 2005.
32. Eran Segal, Michael Shapira, Aviv Regev, Dana Peer, David Botstein, Daphne Koller, and Nir Friedman. Module Networks: Identifying Regulatory Modules and Their Condition-specific Regulators from Gene Expression Data. *Nature Genetics*, 34(2):166–176, 2003.
33. Lev A. Soinov, Maria A. Krestyaninova, and Alvis Brazma. Towards Reconstruction of Gene Networks from Expression Data by Supervised Learning. *Genome Biology*, 4(1):R6.1–R6.10, 2003.
34. Yoshinori Tamada, Su Yong Kim, Hideo Bannai, Seiya Imoto, Kousuke Tashiro, Satoru Kuhara, and Satoru Miyano. Estimating Gene Networks from Gene Expression Data by Combining Bayesian Network Model with Prometer Element Detection. *Bioinformatics*, 19(2):227–236, 2003.
35. Biao Xing and Mark J. van der Laan. A Statistical Method for Constructing Transcriptional Regulatory Networks using Gene Expression and Sequence Data. *U.C. Berkeley Biostatistics Working Paper Series*, (144), 2004.
36. Biao Xing and Mark J. van der Laan. A Causal Inference Approach for Constructing Transcriptional Regulatory Networks. *Bioinformatics*, 21(21):4007–4013, 2005.
37. Zhou Zhu, Yitzhak Pilpel, and George M. Church. Computational Identification of Transcription Factor Binding Sites via a Transcription-factor-centric Clustering (TFCC) Algorithm. *Journal of Molecular Biology*, (318):71–81, 2002.
38. Chaya Ben-Zaken Zilberstein, Eleazar Eskin, and Zohar Yakhini. Sequence Motifs in Ranked Expression Data. *Technion CS Dept. Technical Report*, (CS-2003-09), 2003.