Rule-based clustering for gene promoter structure discovery

T Curk,[a,*] U Petrovic,[b] G Shaulsky,[c] B Zupan[a,c]

a) University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia

b) J. Stefan Institute, Department of Molecular and Biomedical Sciences, Ljubljana, Slovenia

c) Baylor College of Medicine, Department of Molecular and Human Genetics, Houston, TX

*) Contact: Tomaz Curk, University of Ljubljana, Faculty of Comp. and Inf. Science, Trzaska c. 25, SI-1000 Ljubljana, Slovenija, tomaz.curk@fri.uni-lj.si, phone: +386-1-4768267,

Lemma

Rule-based clustering for promoter analysis

Summary

Background

The genetic cellular response to internal and external changes is determined by the sequence and structure of gene regulatory promoter regions.

Objectives

Using data on gene regulatory elements (*i.e.*, either putative or known transcription factor binding sites) and data on gene expression profiles we can discover structural elements in promoter regions and infer the underlying programs of gene regulation. Such hypotheses obtained *in silico* can greatly assist us in experiment planning. The principal obstacle for such approaches is the combinatorial explosion in different combinations of promoter elements to be examined.

Methods

Stemming from several state-of-the-art machine learning approaches we here propose a heuristic, rule-based clustering method that uses gene expression similarity to guide the search for informative structures in promoters, thus exploring only the most promising parts of the vast and expressively rich rule-space.

Results

We present the utility of the method in the analysis of gene expression data on budding yeast *S. cerevisiae* where cells were induced to proliferate peroxisomes.

Conclusions

We demonstrate that the proposed approach is able to infer informative relations uncovering relatively complex structures in gene promoter regions that regulate gene expression.

Keywords

promoter analysis, gene expression analysis, machine learning, rule-based clustering

1 Introduction

Regulation of gene expression is a complex mechanism in the biology of eukaryotic cells. Cells carry their function and respond to the environment by an orchestration of transcription factors and other signaling molecules that influence gene expression. The resulting products regulate expression of other genes thus forming diverse sets of regulatory pathways. To better understand gene function and gene interactions we need to uncover and analyze the programs of gene regulation. Computational analysis (1) of gene regulatory regions that can use information from known gene sequences, putative binding sites and sets of gene expression studies, can greatly speed-up and automate the tedious discovery process performed by classical genetics.

The regulatory region of a gene is defined as a stretch of DNA, which is normally located upstream of the gene's coding region. Transcription factors are special proteins that bind to specific sequences (binding sites) in the regulatory regions, thus inhibiting or exciting gene expression of target genes. Regulation by binding of transcription factors is just one of the many regulatory mechanisms. Expression is also determined by chromatin structure (2), epigenetic effects, post-transcriptional, translational, post-translational and other forms of regulation (3). Because there is a general lack of these kinds of data, most current computational studies focus on inference of relations between gene regulatory content and gene expression measured using DNA microarrays (4).

Determination of the regulatory region and putative binding sites are the first crucial steps in such analyses. Regulatory and coding regions differ in nucleotide and codon frequency. This fact is successfully exploited by many prediction algorithms (5), and promoter (regulatory) sequences are readily available in public data bases for most model organisms. The next crucial, well studied, and notoriously difficult step is to determine the transcription factors' putative binding sites in promoter regions. These are 4 to 20 nucleotide long DNA sequences (3) which are highly conserved in the promoter regions of regulated genes. A matrix representation of the frequencies of the four nucleotides (A, T, C, G) at each position in the binding site is normally used in computational analysis. The TRANSFAC data base (6) is a good source of experimentally confirmed and computationally inferred binding sites. Candidate binding sites for genes with unknown regulations can be found using local sequence alignment programs such as MEME (7). A detailed description and evaluation of such tools is presented in the paper by Tompa *et al.* (8).

Most contemporary methods that try to relate gene structure and expression start with gene expression clustering and then determine cluster-specific binding sites (4, 9). The success of such approaches strongly relies on the number and composition of gene clusters. Slight parameter changes in clustering procedures can lead to significantly different clustering (10, 11), and consequently to inference of different cluster-specific binding sites. Most often these methods search for non-overlapping clusters and may miss interesting relations, as it is known that genes can respond in many different ways and perform various functions (12).

An alternative to clustering-first approaches are methods that start with information on binding sites and search for descriptions shared by similarly expressed genes. For example, in an approach by Chiang *et al.* (13) the group's pair-wise gene expression intra-correlation is computed for each set of

genes comprising a specific binding site in the promoter region. Their method reports on binding sites where this correlation is statistically significant, but fails to investigate the combinations of two or more putative binding sites: it is known that regulation of gene expression can be highly combinatorial and requires the coordinated presence of many transcription factors. There are other approaches where combinations of binding sites are investigated, but they are often limited to the presence of two sites due to the combinatorial explosion of the search (4, 14). For example, the number of all possible combinations of three binding sites, from a base of a thousand binding sites available for modeling, quickly grows into hundreds of millions. Transcription is also affected by absolute or relative orientation and distance between binding sites and other landmarks in the promoter region (*i.e.*, the translation start ATG), further complicating the language that should be used to model promoter structure and subsequently increasing the search space.

To overcome the limitations described above, we have devised a new algorithm that can infer potentially complex promoter sequence patterns and relate them to gene expression. In the approach, which we call rule-based clustering (RBC), we essentially borrowed from several approaches developed within machine learning that use heuristic search to cope with potentially huge search space. The uniqueness of the presented algorithm is its ability to discover groups of genes that share any combination of promoter elements that can be in placement and orientation specific to the start of the gene or to another promoter element. Below, we first define the language we use to describe the constitution of promoter region, then describe the RBC algorithm and finally illustrate its application on the analysis of peroxisome proliferation data on *S. cerevisiae*.

2 Rule-based clustering method

The inputs to the proposed rule-based clustering (RBC) method are gene expression profiles and data on their promoter regulatory elements. The algorithm does not include any preprocessing of expression data (*e.g.*, normalization, scaling) and considers the data as provided. For each gene, the data on regulatory elements is given as a set of sequence motifs with their position relative to the start of the gene and orientation. The motifs are represented either by a position weight matrix (7) or a single line consensus; the former was used in all our experiments. The RBC algorithm aims to find clusters of similarly expressed genes with structurally similar promoter regions. The output of the algorithm are rules of the form "IF *structure* THEN *expression profile*", where *structure* is an assertion over the regulatory elements in the gene promoter sequence and *expression profile* is a set of expression profiles of matching genes.

2.1 Descriptive language for assertions on promoter structure

RBC discovers rules that contain assertions-conditions on the structure of the promoter region that include the presence of binding sites, the distance of the binding sites from transcription and translation start site (ATG), the distance between binding sites, and the orientation of binding sites. We have devised a simple language to represent these assertions. For instance, the expression "S_1" says that site S_1 (in whichever orientation) must be present in the promoter, and the expression "$S_1{-}@{-}d_1(\text{ref}{:}S_2)$" asserts that both sites S_1 and S_2 should be present in the promoter region such that S_1, in the non-sense direction, appears d_1 nucleotides upstream of S_2.

The proposed description language is not unequivocal: the same promoter structure may often be described in several different ways. For example, any of the following rules may describe the same structure : "S_1+@−d_1(ref:ATG) and S_2−@d_2(ref:s_1)," "S_2−@−d_3(ref:ATG) and S_1+@−d_2(ref:S_2)," and "S_1+@−d_1(ref:ATG) and S_2−@−d_3(ref:ATG)". All three descriptions require sites S_1 and S_2 to be oriented in the sense and non-sense directions, respectively. The first rule requires site S_1 to be positioned at distance d_1 from the reference ATG (translation start site) and the position S_2 to be relative to S_1. According to the second rule, the position of S_1 is relative to the absolutely positioned S_2 at distance d_3 from ATG. The third rule defines the position of both sites relative to ATG. In such cases, the RBC algorithm will return only one of the semantically equivalent descriptions, depending on the order in which they were found in the heuristic search.

2.2 RBC algorithm

The proposed algorithm is outlined in Fig. 2. In its input it requires data on gene expression profiles P_{all} and data on promoter elements in the corresponding gene regulatory regions. The algorithm returns a list of inferred rules of the form $R = (C, P)$ with condition on the promoter structure C contained in genes with similar gene expression profiles P.

RBC uses a beam-search approach (lines 3-12) followed by two post-processing steps (lines 13 and 14 of the algorithm). *Beam* is a list of at most L currently inferred rules considered for further refinement that are ordered according to their associated scores (see below). Parameter L is a user-defined parameter (with a default value of 1000) that affects the scope of the search and thus the runtime. At the start of the search *Beam* is initialized with a rule "IF *True* THEN P_{all}" that covers all genes under consideration.

In every iteration of the main loop (lines 3 to 12), the search focuses on the best-scored rule $R = (C, P)$ from *Beam* and considers all possible single-term extensions of its condition C, which are allowed by the given descriptive language. Each such refinement results in a new candidate rule, which is added into the list of *Candidates* (line 6). The refinements include adding the terms with assertion on the presence of a site, presence of a site with its orientation, or the presence of a site (with or without the information on orientation) at a relative distance of a specific landmark (another site or start of gene). Refined rules are then represented in a simplified form. For instance, adding a single-site presence condition S_1 to the initial rule "(True, P_{all})" yields a rule "True and S_1" which is simplified to its logical equivalent "S_1." Adding a term with the same site but non-sense orientation to the latter yields the rule "S_1 and S_1−" which is simplified to "S_1−." Similarly, adding a term with the same site but with information on a distance of 100 to 80 nucleotides to the ATG may result in a rule such as "S_1@−100..−80(ref:ATG)." Requirements of other binding sites may be added, either simply by requiring their presence (*e.g.*, rule "S_1 and S_2") or by adding them as a reference to the presently included sites in conditions (*e.g.*, "S_1@−100..−80(ref:S_2)"). Candidate rules will include those with matching at least N genes, where N being a user-defined parameter with a default value of six.

Candidate rules are then compared to their (non-refined) parent rule based on the intra-cluster pair-wise gene expression profiles distance of the covered genes. To identify co-expressed genes, the algorithm uses Pearson correlation as a default distance measure, which – when computing the dis-

tance between two genes – ignores experiments where for any of these two genes the expression is missing. The user can replace it with any other type of distance function that suits the particular type of expression profiles or the biological question addressed. For a set of candidate rules, only those with a significant reduction of this distance are retained in the list of *Candidates* (line 7). This decrease of variance in the intra-cluster pair-wise distances is tested using the *F-test* statistic:

$$F = \frac{SS_R}{n_R - 1} \Big/ \frac{SS_{Candidate}}{n_{Candidate} - 1}$$

where SS_R and $SS_{Candidate}$ are sums of squared differences from mean inside the cluster of genes covered by the parent rule *R* and by a refined *Candidate* rule, respectively, and values n_R and $n_{Candidate}$ are the total number of genes in each of the two clusters. A *p-value* is calculated from the *F score* and used to determine the significance of change (the threshold, α_F, defaults to 0.05). Figure 1 shows an example of explored refinements during rule search that may lead to the identification of profile-coherent gene clusters.

The resulting refined rules stored in the *Candidates* list are added to *Beam* (line 9), which retains at most *L* best-scored rules (line 10). Because the goal is to discover the most homogeneous clusters, each rule is scored according to the potential coherence of its corresponding sub-cluster potentially obtained after the refinement of the rule. Potential coherence estimates how promising the cluster is in terms of finding a good subset of genes. While examining all subgroups of genes in the cluster would be an option, such an estimate is computationally expensive because of potentially large number of subgroups. Instead, we define the potential coherence of a cluster as the average of $k \cdot N \cdot (k \cdot N - 1)/2$ minimal pair-wise profile distances. This in a way approximates a choice of a subset with $k \cdot N$ most similar genes. If the cluster being estimated contains less then $k \cdot N$ genes, its estimated potential equals to the average of all pair-wise gene distances.

Rules for which the above procedure finds no suitable refinements and whose intra-cluster pair-wise distance is below a user-defined threshold *D* are added to *Rules*, the list that stores the terminal rules discovered by RBC algorithm (line 12). Note that a process of taking the best-scored rule from the *Beam*, refining it and adding newly found rules (if any) with improvements in intra-cluster profile distances is repeated until *Beam* is left empty.

To further reduce the potentially large number of rules found by the beam search, RBC uses two post-processing steps (lines 13 and 14). RBC may infer rules that describe exactly the same cluster of genes. Each such rule set is considered individually, with the aim to retain only the most general rules from it. That is, for each pair of rules with conditions C_1 and C_2, only the first rule from the pair is retained in the rule set if its condition C_1 subsumes condition C_2, that is, it covers the same genes but is more general in terms of logic. For instance, condition "S_1" subsumes condition "S_1 and S_2." The remaining list of *Rules* is further filtered by keeping only the most coherent rules so that on average no more than a limited number of rules describe any gene (parameter M set by the user, default is five). The final set of rules is formed by selecting the rules with lowest intra-cluster distance first, and adding them to the final set only if their inclusion does not increase the rule-coverage for any gene beyond M.

Alternatively to considering all the genes in its input data, RBC can additionally deal with the information on a set of target genes for which the user wants to focus the analysis. Typically, target genes would comprise a subgroup of similarly annotated genes, or a subset of differentially expressed genes. If a target set is given, discovered rules are included in *Beam* and in the final set only if they cover at least *N* target genes. Because the algorithm starts with one rule (line 1), which describes all genes, the discovered rules can cover genes outside the target set. The method is thus able to identify genes that were initially left out of a target set but should have been included based on their regulatory content and gene expression.

The proposed rule-based clustering method was inspired by the beam-search procedure successfully used in a well known, supervised machine learning algorithm CN2 (15), and by an unsupervised approach of clustering trees developed by Blockeel *et al.* (16), but is in its implementation and application substantially different from both. CN2 infers rules that relate attribute-value based description of the objects to their discrete class, while clustering trees identify attribute-value based description of non-overlapping clusters of similar objects.

RBC combines both approaches by using a beam search to infer symbolic descriptions of potentially overlapping clusters of similarly regulated genes. Compared to beam search in CN2, where the size of the beam is relatively small (ten to twenty best rules are most often considered for further refinements), RBC uses a much wider beam but also generates potentially overlapping rules in a single loop. In contrast, in CN2, only the best-found rule is retained, objects covered by it removed from the data, and the procedure is restarted until no objects to be explored remain. Similar to CN2, the essence of our algorithm is rule refinement, for which, in the area of machine learning, the beam search proved to be an appropriate heuristic method.

3 A case study and experimental validation

We applied the proposed RBC method to data from a microarray transcription profiling study where budding yeast *S. cerevisiae* cells were induced to proliferate peroxisomes – organelles that compartmentalize several oxidative reactions – due to the cell's regulated response to the exposure to oleic fatty acid (oleate) and to the absence of glucose, which causes peroxisome repression (17). The transcriptional profile of each gene consists of six microarray measurements on oleate induction time course, and two measurements in "oleate *vs.* glucose" and "glucose *vs.* glycerol" growth conditions. In total, gene pair-wise distance was calculated on gene expression profiles consisting of eight microarray measurements. We defined the pair-wise distance function to be $1.0-r$, where r is the Pearson correlation between two gene profiles.

For the target group we selected a set of 224 genes identified by the study to have similar expression profiles to those of genes involved in peroxisome biogenesis and peroxisome function. The goal of our analysis was to further divide the target group into smaller subgroups of genes with common promoter structure and possibly identify genes that were inadvertently left out of the target group but should have been included based on their expression and promoter structure similarity.

We analyzed data on 2,135 putative binding sites which were identified using a local sequence alignment software tool MEME (7). We searched for presence of these binding sites in 1Kb promoter regions taken upstream from the translation start site (ATG) for ~6,700 genes. The search identified ~302,000 matches of putative binding sites that were then used to infer rules with RBC. The algorithm was run with the default values of parameters. Distances between binding sites were rounded to increments of 40 bases; the maximum possible range of 2Kb (for the given promoter length, relative distances can be from –1Kb to +1Kb) was thus reduced to 50 different values ($= 2000b/40b$). This largely reduced the number of possible subintervals that needed to be considered during rule inference.

The search returned 41 rules that described and divided 114 target genes (51% of target genes) into 37 subgroups (see Fig. 3b). No rule could be found to describe the remaining 110 target genes. Most of the discovered gene groups are composed of five genes with high pair-wise intra-group correlation (above 0.927). Many genes are shared (overlap) between the 37 discovered groups, resulting in six major gene groups visible in Figure 3a and 3b. Seven genes outside the target set were also identified by the method (marked in black in Fig. 3a). For example, the smallest eight-gene group in the top-left corner in Fig. 3a includes two outsiders (*INP53* and *YIL168W* - also named *SDL1*). Gene ontology annotation shows that *INP53* is involved together with two target genes (*ATP3* and *VHS1*) in the biological process *phosphate metabolism*. Gene *SDL1* is annotated to function together with the group's target gene *LYS14* in the biological process *amino acid metabolism* and other similar parent GO terms (results not shown). Details on the promoter structure and gene expression are given in Fig. 3c and 3d. These examples confirm the method's ability to identify functionally related genes that were not initially included in the target set.

The majority of the discovered rules in the case study include conditions that are composed of three terms, describing the binding site's orientation and distance relative to ATG or other binding sites. There is no general binding site that would appear in many rules; only two rules include the same binding site (results not shown).

Exhaustive search of even relatively simple rules can quickly grow into a prohibitively hard problem due to combinatorial explosion. Exhaustive search for all possible rules composed of three binding sites with defined orientation (three possible values: positive, negative, no preference) and distance (distance range is reduced into 50 different values) would, for this case study, require checking a huge number of rules:

$$\binom{2135 \times 3}{3} \times 50^3 \approx 5.47 \times 10^{15}$$

Our method checked $2 \times 11 \times 10^9$ of the most promising rules, or less than 0.00004% of the entire three-term rule space. The search took 40 minutes on a Pentium 4, 3.4 GHz workstation. This demonstrates RBC's ability to efficiently derive potentially complex rules within reasonable time frame.

To evaluate the predictive ability of the approach we used a data set on 1364 *S. cerevisiae* genes that includes accurate binding sites data for 83 transcription factors (18). We modeled the regulatory region spanning from −800bp to 0bp relative to ATG. Pair-wise gene distance was calculated as the

average pair-wise distance across nineteen gene expression microarray studies available at SGD's Expression Connection data base (http://www.yeastgenome.org/). All genes were considered to be target genes.

Five-fold cross-validation was used that randomly splits genes into five sets. Clustering and testing of the inferred rules was repeated five times, each time with a different set of genes for validation of a model constructed using the remaining four sets. Each discovered rule was tested on genes in the test set. If a rule matched the promoter region of a test gene, then we calculated the prediction error by calculating the distance between the true gene expression of the test gene and its predicted expression. When more than one rule could be applied to predict the expression of a test gene, the average prediction error was returned for that gene. Overall, the method successfully predicted the expression of 286 genes (21% of all genes considered), with an average cross-validation prediction error of 0.75. If we were to use "random" rules, which would randomly cluster genes into groups of the same size as those by inferred rules, we could expect the prediction error to be 0.96. We believe that the achieved prediction error is a good indication of the predictive quality of inferred rules.

4 Conclusion

The proposed rule-based clustering method can efficiently find rules of gene regulation by searching for groups of similarly expressed genes and with similar structure of the regulatory region. Starting from a target set of genes of interest, the method was able to cluster them into subgroups. Concurrently, RBC may expand the target set by identifying other similarly regulated genes that were initially overlooked by the user. Rule-search is guided and is made efficient by the proposed search heuristics. An important feature of RBC is its ability to discover overlapping groups of genes, potentially indicating common regulation or function.

The algorithm uses a number of parameters that essentially determine the size of the search space being examined. The default values provided with the algorithm were set according to particular characteristics of the domain (*e.g.*, about ten thousand genes, small subset of genes sharing some motif pattern, most known patterns include from one to five motifs (19)). The choice of parameters also affects the run time, and the defaults were chosen to make implementation practical and to infer the rules within one hour of computational time on a standard personal computer.

We have experimentally confirmed the ability of RBC algorithm with default settings to infer rules that describe a complex regulatory structure and which can be used to reliably predict gene expression from regulatory content. In contrast with other contemporary methods that mainly use information on the presence of binding sites, a principal novelty of our approach is the use of a rich descriptive language to model the promoter structure. The language can be easily extended to accommodate other descriptive features, such as chromatin structure, when such kinds of data become available on a genome-wide scale.

To summarize and display the findings of the analysis at different levels of abstraction we have applied different visualizations, which proved useful for understanding and biological interpretation. We believe that the main application of RBC is an exploratory search for additional evidence that genes, in

theoretically or experimentally defined groups, actually share a common regulatory mechanism. The biologist can then gain insight by looking at the presented evidence and can better decide which inferred patterns are worth testing in the laboratory.

Acknowledgments

This work was supported in part by Program and Project grants from the Slovenian Research Agency (P2-0209, J2-9699, P1-0207) and by a grant from the National Institute of Child Health and Human Development (P01-HD39691).

References

1. Bellazzi R, Zupan B. Intelligent data analysis--special issue. Methods Inf Med 2001;40(5):362-4.

2. Segal E, Fondufe-Mittendorf Y, Chen L, Thastrom A, Field Y, Moore IK, et al. A genomic code for nucleosome positioning. Nature 2006;442(7104):772-8.

3. Wasserman WW, Sandelin A. Applied bioinformatics for the identification of regulatory elements. Nat Rev Genet 2004;5(4):276-87.

4. Beer MA, Tavazoie S. Predicting gene expression from sequence. Cell 2004;117(2):185-98.

5. Bajic VB, Tan SL, Suzuki Y, Sugano S. Promoter prediction analysis on the whole human genome. Nat Biotechnol 2004;22(11):1467-73.

6. Wingender E, Dietze P, Karas H, Knuppel R. TRANSFAC: a database on transcription factors and their DNA binding sites. Nucleic Acids Res 1996;24(1):238-41.

7. Bailey TL, Elkan C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proc Int Conf Intell Syst Mol Biol 1994;2:28-36.

8. Tompa M, Li N, Bailey TL, Church GM, De Moor B, Eskin E, et al. Assessing computational tools for the discovery of transcription factor binding sites. Nat Biotechnol 2005;23(1):137-44.

9. Down TA, Bergman CM, Su J, Hubbard TJ. Large-scale discovery of promoter motifs in Drosophila melanogaster. PLoS Comput Biol 2007;3(1):e7.

10. Bolshakova N, Azuaje F. Estimating the number of clusters in DNA microarray data. Methods Inf Med 2006;45(2):153-7.

11. Rahnenfuhrer J. Clustering algorithms and other exploratory methods for microarray data analysis. Methods Inf Med 2005;44(3):444-8.

12. Ihmels J, Friedlander G, Bergmann S, Sarig O, Ziv Y, Barkai N. Revealing modular organization in the yeast transcriptional network. Nat Genet 2002;31(4):370-7.

13. Chiang DY, Brown PO, Eisen MB. Visualizing associations between genome sequences and gene expression data using genome-mean expression profiles. Bioinformatics 2001;17 Suppl 1:S49-55.

14. Pilpel Y, Sudarsanam P, Church GM. Identifying regulatory networks by combinatorial analysis of promoter elements. Nat Genet 2001;29(2):153-9.

15. Clark P, Nibblet T. The CN2 induction algorithm. Machine Learning 1989;3(4):261-83.

16. Blockeel H, De Raedt L, Ramon J. Top-down induction of clustering trees. Machine Learning 1998.

17. Smith JJ, Marelli M, Christmas RH, Vizeacoumar FJ, Dilworth DJ, Ideker T, et al. Transcriptome profiling to identify genes involved in peroxisome assembly and function. J Cell Biol 2002;158(2):259-71.

18. MacIsaac KD, Wang T, Gordon DB, Gifford DK, Stormo GD, Fraenkel E. An improved map of conserved regulatory sites for Saccharomyces cerevisiae. BMC Bioinformatics 2006;7:113.

19. Harbison CT, Gordon DB, Lee TI, Rinaldi NJ, Macisaac KD, Danford TW, et al. Transcriptional regulatory code of a eukaryotic genome. Nature 2004;431(7004):99-104.

Figures

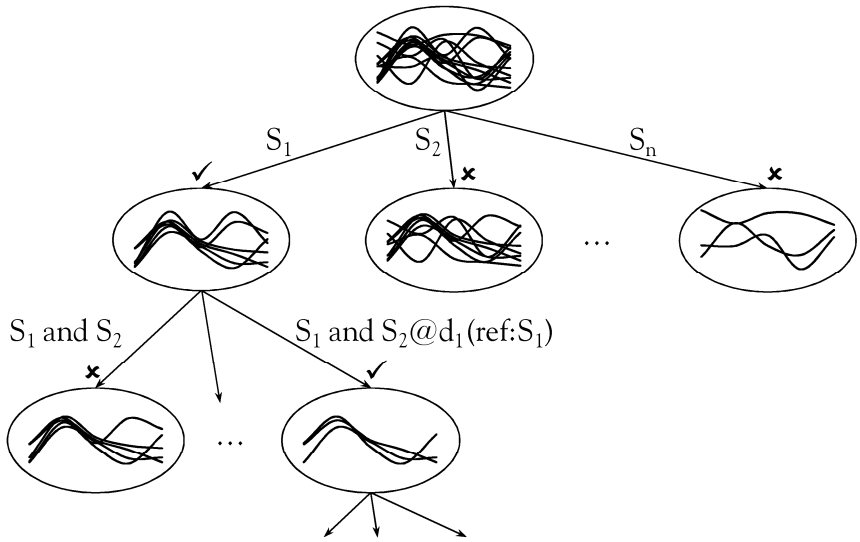

Figure 1. Example of a rule search trace. Rule refinements that result in a significant increase in gene expression coherence (check mark) are explored further. Search along unpromising branches is stopped (cross).

Input: a set of gene expression profiles P_{all} (for every gene, a vector of gene expressions), and a set of promoter elements for observed genes (for every gene, a set of tuples (motif_id, position relative to ATG, orientation) that define the list of motifs in the regulatory region of the gene).

Output: list of inferred rules that relate promoter structure and gene expression; each rule $R = (C, P)$, which can be read as "**IF** *C* **THEN** *P*," is a pair of condition on promoter structure *C* and rule's expression profile *P* (a collection of expression profiles of genes that match *C*).

Parameters:

L size of the search beam (default: 1,000 rules),
N minimum number of genes that a rule's condition has to match (default: 6),
D maximum average intra-cluster pair-wise distance (0.5, for 1-Pearson correlation used in our applications),
k used in computation of cluster's potential coherence, estimated as the smallest intra-cluster average pair-wise distance for a subset of $k \cdot N$ genes (default: 2),
α_F significance level for acceptable change in cluster's coherence after rule refinement (default: 0.05),
M average number of rules retained during post-processing, which are used to describe a gene (default: 5)

```
1   Beam ← [(True, P_all)]
2   Rules ← []; is a list of discovered rules
3   while Beam not empty
4       R=(C, P) ← highest scored rule from Beam
5       remove R from Beam
6       Candidates ← rules covering at least N genes with all possible extensions of C with a single new term in
                     condition and an associated matching subset of gene expression profiles P
7       Candidates ← subset of rules from Candidates with intra-cluster distances significantly (α_F) lower than R
8       if Candidates not empty
9           add rules from Candidates to Beam
10          Beam ← L best-scored candidates from Beam (uses k)
11      else
12          if intra-cluster distance of R < D then add R to Rules
13  from subsets of completely overlapping rules in Rules keep only most general ones
14  from Rules remove rules with low scores and high overlap with higher-scoring rules (uses M)
15  return Rules
```

Figure 2. Outline of the RBC algorithm.

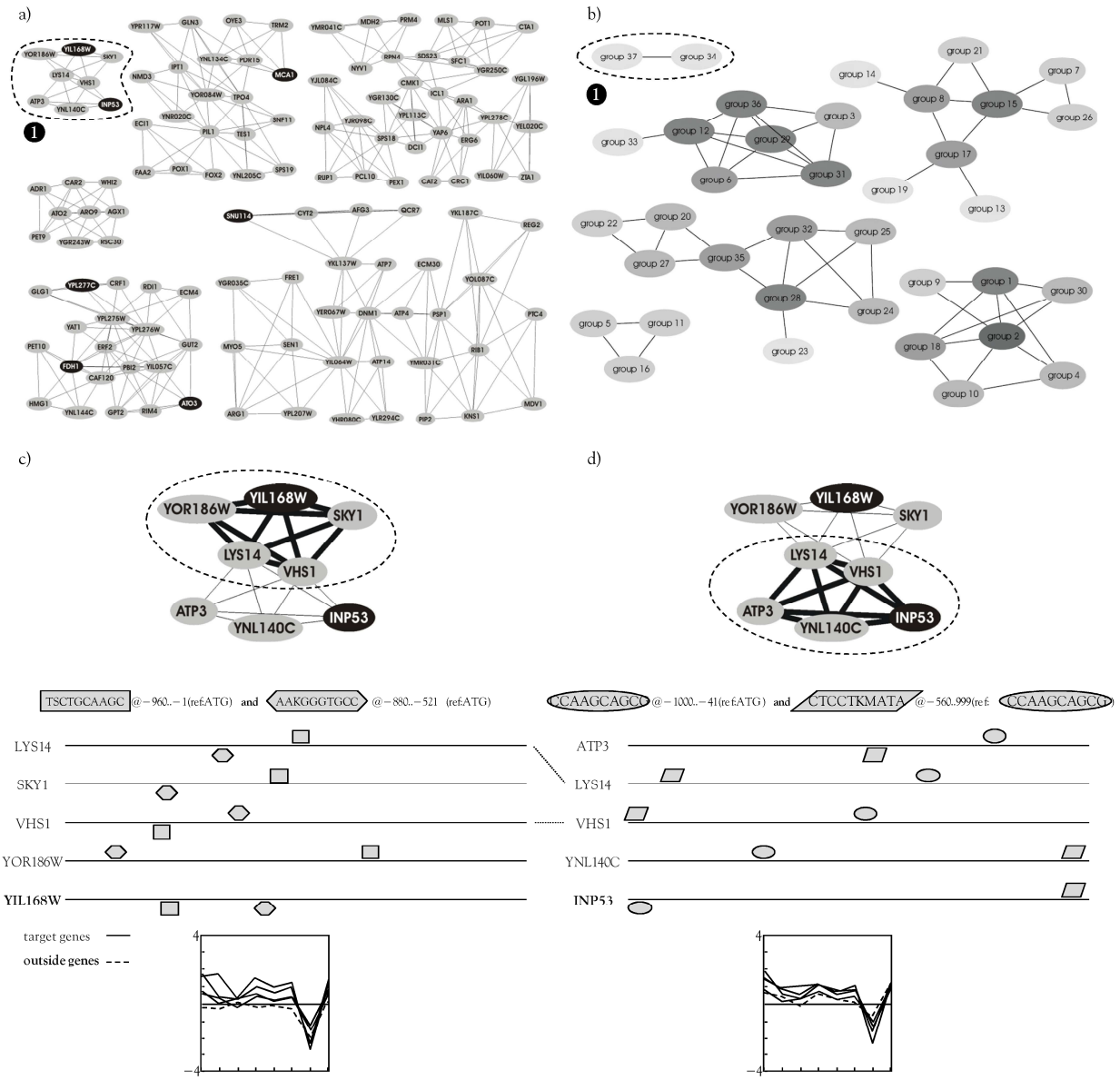

Figure 3. a) Gene network, where we connect genes from same rule, for the peroxisome data set (target genes in gray, genes outside target in black). It includes 114 target genes and 7 outside genes, which are clustered in six major groups. b) Group graph of the discovered 37 clusters (two groups are connected if sharing a subset of genes). c and d) Inferred promoter structure and gene expression of the two sub-clusters forming the eight-gene cluster, marked “1” in figure 3a (also shown as clusters “group 37” and “group 34” in 3b).