Inferring Cis-region Hierarchies from Patterns in Time-Course Gene Expression Data

Vladimir Filkov[1] and Nameeta Shah[1,2]

[1] CS Dept., UC Davis
One Shields Avenue, Davis, CA 95616
filkov@cs.ucdavis.edu
[2] CIPIC, UC Davis
One Shields Avenue, Davis, CA 95616
shahn@cs.ucdavis.edu

Abstract. Resolving the coregulation relationships between genes is a major step toward understanding the underlying topology and dynamics of gene networks. Although coexpression of genes does not directly imply their co-regulation, model-based approaches coupled with the availability of large-scale gene expression data are bringing us closer to attributing expression patterns to cis-regions features. Inspired by studies of transcriptional regulation in sea-urchin, here we report on initial tests of the following simple model: shared patterns in time-course expression profiles of genes are effects of shared binding motifs in their cis-regions. We use a modified version of a prior algorithm for decomposing time-course gene expression patterns into functional events and introduce an order relationship, or a *Regulation Hierarchy* on the genes based on shared events. When tested on actual time-course gene expression data of yeast preliminary results indicate 50% - 71% matches, of high confidence, between our derived and known cis-region regulation hierarchies. This hierarchy structure yields practical predictions when used with other type of genomic data, e.g. location of TF-DNA interactions.

1 Introduction

Gene expression is regulated during transcription by combinations of transfactors (TFs) that bind to corresponding sites in the genes cis-regions. The differential expression of any gene under different experimental conditions is due to the particular regimen (i.e. abundances) of those TFs. Exactly how the cis-regions process the input protein concentration signals is a key question in functional genomics. Methods for grouping genes by similarity of expression profiles across multiple experiments have been partially successful in identifying functionally related genes [1], and the interest in them has grown with the availability of large-scale gene expression data for many organisms. But since coexpression does not imply co-regulation in general their use has been limited in gene networks inference, although some types of experiments (e.g. time-course) are more revealing than others of coregulation.

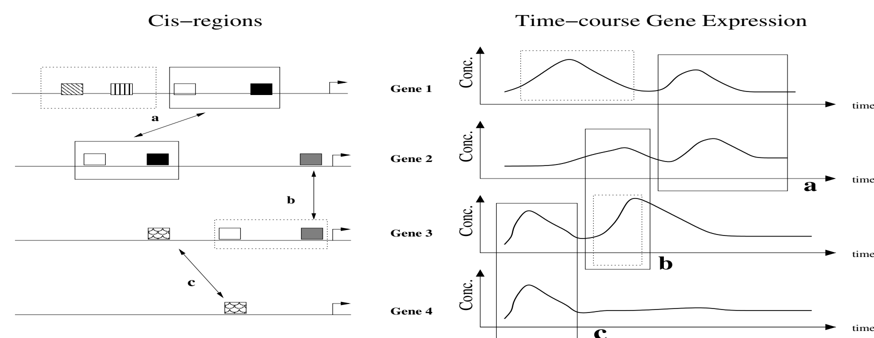

Fig. 1. On the left are given cis-regions for four genes and the cis-elements in them. On the right hand side are the corresponding gene expression signals. **a, b,** and **c** are the modules of cis-elements and their effects on the expression. The dashed boxes indicate expression events for which there might be multiple causes.

The general questions: Can we examine gene expression data to get closer to coregulation? Does gene expression data contain that information at all? are important but difficult to answer since the data itself is very quantitative and the biology behind transcriptional regulation is still very qualitative. Given the same data, ultimately it is about a choice of a modeling approach.

Perhaps one of the best qualitative descriptions of the inner-workings of cis-regions (CR) is given by Eric Davidson [2]. We learn from his work that CRs are logic processing units consisting of binding sites to which TFs bind and effect various patterns of transcription. Each binding site (equivalently TF) has a role in the resulting transcription signal, either alone or as a part of a functional group. Single or groups of binding sites can be thought of as functional units of regulatory systems or regulatory *modules*. The function of any cis-regulatory region could be decomposed hierarchically down to the functions of individual modules, thereby simplifying significantly the overall combinatorial complexity of interaction. Davidson and colleagues have demonstrated on many genes in the Sea Urchin organism [3] that the function of their cis-regions, i.e. gene expression signals, can be decomposed into simpler functions of their sub-regions. The modules formed from the binding sites present in the smallest sub-cis-regions which are still functional supposedly bind to multiple genes and convey their sub-function to each of them. They are, therefore, the building blocks or transcriptional regulation, and their identification is extremely important.

Ideally, one would like to formalize their model and use it to resolve the complexity of cis-regions from large-scale functional genomics data, like gene expression. As a lead-in to such a future formalization, here we sought to test how well the following hypothesis holds: *shared events, or sub-signals, of gene expression signals are due to modules of shared binding sites in the cis-regions of the genes.* Fig. 1 gives an illustration. There, the sub-signals **a, b** and **c** are consequences of the actions of the corresponding modules in the cis-regions.

The hypothesis is based on the following simplified assumptions of regulation modularity:

1. cis-modules are responsible for elementary expression signals, i.e. if the corresponding TF are present and bound they will effect a particular signal; and
2. two or more modules on the same cis-region are responsible for expression regulation at different times or places in the organism; otherwise they would be considered a single module

These two rules effectively linearize the combinatorial nature of transcriptional regulation by proposing that a linear decomposition of signals is possible because the bases of the linear space are the signals corresponding to individual modules.

In this paper we propose and test a model for resolving the modularity of cis-regions based on the organization of gene expression signals. To do this, we identify expression events in gene expression signals and then we build a Regulation Hierarchy from them.

We propose identifying expression events by decomposing gene expression signals into Putative Elementary Expression Events (PEEEs) using a modification of our previous approach [4], which was developed for elucidating regulatory relationships between pairs of genes from time-course expression data.

We introduce the *Regulation Hierarchy* structure as a representation of coregulation between genes. The Regulation Hierarchy is a directed acyclic graph, in which genes are partially ordered based on shared cis-modules. In such a graph any two nodes with a common ancestor are co-regulated. Such a hierarchy graph is useful independently as a structure for study of functional elements of gene regulation. As approximations of the Regulation Hierarchy we define two other hierarchies, the *Expression Hierarchy* and the *Transcription Factor Hierarchy* which can be obtained from existing data sets.

To assess the efficacy of the Regulation Hierarchy built from expression data we apply our decomposition strategy on publicly available time-series gene expression data of yeast [5]. The resulting hierarchy graph is compared to an actual TF Hierarchy obtained from a study of TF-DNA binding in yeast [6]. Our results suggest strong correlation between sub-elements of gene expression curves and cis-modules of binding sites: we observed 50% - 71% of matching directed edges between the two herarchies, compared to expected (between 1/6 and 1/8 of that).

Combined with location data of TF-DNA interactions, from the predicted hierarchies we were able to discern basic signals and attribute some to well known TF modules.

This paper is organized as follows. In the next section we talk about related work on effects of modularity of the cis-regions on gene expression, and decomposition of expression signals into basic curves. In Section 2 we review and expand a previous method for identifying elementary expression events. The regulation hierarchies are defined in Sect. 3. We report the results of our preliminary studies in Section 4. In the last section we summarizes the findings and describe

our current and future directions in both expanding the model and utilizing the hierarchy graphs in different ways.

1.1 Coregulation and Coexpression

Out current work is novel in that it proposes a model for coregulation based on attributing identifiable events in expression signals to cis-modules. We also describe an original structure, the Regulation Hierarchy.

Differentiating between coexpressed and coregulated genes is important in particular for gene network inference. In previous, work Pilpel et al. proposed [7] and later improved [8] methods to identify clusters of genes which are co-regulated and co-expressed at the same time. They achieved this by scoring coexpression for genes which share overrepresented elements in the upstream regions. Although our goal is seemingly the same since we are also attempting to resolve coregulation from coexpression, here we are interested in resolving coregulation from expression data, based on the shared events model. We don't aim to resolve actual binding sites in this paper; instead we are after the coregulation hierarchy, and we only use time-course expression data. We discuss later some future uses for the coRegulation Hierarchy.

A few studies recently have focused on identifying modules of genes by considering variety of available data: gene expression, sequence, and TF-DNA location. The working definition for a module in them varies between a group of strongly coexpressed genes in a subset of experiments to a group of genes coregulated by the same factors and sharing a function [9]. In both extremes though the definition of a module is somewhat fuzzy as genes can be taken in or out of it while the module doesn't change. Our definition of a cis-module, a variant of that of Davidson [2], is a group of transcription factors that has an indivisible functional effect on transcription; in other words there is a sense of minimality or atomicity to it. Time-course expression data cannot predict the actual content of a cis-module; other data is needed for that (gene knockout expression data can also be used).

The small number of different patterns evident in time-course gene expression data, especially the cycling genes set by Spellman et al. [5], has motivated several studies into evaluating the possibility of decomposing the expression signals into a combination of a few basic signals. In particular the study by Holter et al. [10] identifies a small number of characteristic modes in microarray time-series data [5], as discovered by Singular Value Decomposition. Such studies although informative about the range of the transcriptional signals under specific conditions, and arguably successful in correlating functional gene categories with specific modes of regulation, do not address the issue of coregulation.

2 Gene Expression Events

We define Putative Elementary Expression Events (PEEEs) as functionally important part of the expression signals. For our purposes, these are parts of the

signals that either increase or decrease. They are identified using a modified version of the edge detection algorithm by Filkov et al. [4]

There, events were defined as biologically meaningful changes in expression with time. In the ideal case, with no fluctuations in the signals, events would correspond to monotonically increasing or decreasing smooth curves between local optima. Because large-scale gene expression data is far from ideal, signals are smoothed out as follows. Starting from the initial time point, and proceeding to the right iteratively, over the rest of the time points, the events are identified, grown, and possibly merged, so long as the expression change is in the same direction (i.e. increase or decrease) as the rest of the event, with tolerances for default and random fluctuations in expression levels, as well as with a biologically significant cap on the maximum length of an event. The original method uses one neighbor on both sides of time points to label them local minima, local maxima and in-between. But edges can be missed that way because of noise in the data. We improve on this by using two neighbors on both sides of a local optimum to label the points more accurately.

The result is a list of putative events, or PEEEs, for each gene. Each event is a run of points that either increases or decreases in expression. We showed previously [4] that these lists of events can be used to identify gene regulatory relationships between genes with greater fidelity than the whole expression signals. In addition, the putative events lists for pairs of genes can be aligned to discover any shared events.

3 Regulation Hierarchy Graphs

The idea behind the *Regulation Hierarchy* is to build a structure that captures the shared regulator information between genes. The Regulation Hierarchy is meant to be an invariant view of regulation from both the sequence and gene expression.

The *Regulation Hierarchy (RH)* is defined as a directed graph, $G_r = (V, E_r)$ over the genes in an organism, $V = \{g_1, g_2, \ldots, g_n\}$, where there is an edge between two nodes if the set of TF modules regulating one is a subset of the set of the TF modules regulating the other, and the direction of the edge is from the smaller toward the larger set of regulators. That is, if $Mod(x)$ is the set of modules regulating node x, then for every pair of genes (nodes) i and j, $(i, j) \in E$ if $Mod(i) \subseteq Mod(j)$. If i and j share regulating modules but none dominates the other, then neither $(i, j) \in E$ nor $(j, i) \in E$. For example, there is only one relationship between the genes in Fig. 1, and that is $Gene4 \leq Gene3$. The rest of the gene pairs don't have order relationships although they share regulators (and sub-signals).

We define the following two additional hierarchy graphs, which in contrast to the regulation hierarchy can be easily obtained from existing data. First is the *TF hierarchy (TFH)*, defined as $G_{tf} = (V, E_{tf})$, where if $Tf(x)$ is the set of transcription factors that can bind to the cis-region of gene x then $(i, j) \in E_{tf}$ if $Tf(i) \subseteq Tf(j)$. The second hierarchy is the *Expression Hierarchy (EH)*, defined

as $G_e = (V, E_e)$, where if $Peee(x)$ is the set of PEEEs present in the expression signal of gene x then $(i, j) \in E_e$ if $Peee(i) \subseteq Peee(j)$.

It should be clear that $E \subset E_{tf}$ since the modules are hierarchies of TFs and that there is no overlap between modules (by the second assumption of regulation modularity above). Also, since PEEEs correspond to modules, ideally $E_e \subset E$.

With ideal data, these three hierarchies would be directed, and transitively closed graphs, where if $(i, j) \in E$ and $(j, k) \in E$ then $(i, k) \in E$. They would also be acyclic except for the trivial cycles which will happen between two genes sharing exactly the same regulators.

3.1 Utility of the Regulation Hierarchies

From the RH one can readily answer if two genes are coregulated by looking up if they have the same ancestor. Also, with the RH and the TFH one can explore TF modules, whereas from the RH and EH the basic expression signals corresponding to modules can be found.

In addition, RH can be a powerful tool for building regulatory networks. Namely, the RH establishes classes of coregulated genes–information that can help bound the in-degrees of nodes during inference.

In this paper we show how to obtain the Expression Hierarchy and the TF hierarchy, and explore how well they coincide. Again ideally $E_e \subset E_{tf}$. We use both to illustrate how one can identify TF modules.

3.2 Adjusting for Real Data

Real data of course is noisy. Thus we have to allow for some fuzziness in the regulation hierarchies. In addition, data sets with which we will be working below will be clusters of gene expression signals as opposed to individual signals. Thus each cluster contains a number of coexpressed genes, whose curves are to us indistinguishable. Here we describe how to derive the expression and TF hierarchies from noisy clusters of genes.

Expression Hierarchy The data used for the EH is time-course expression data (see below), from which PEEEs have been identified for each gene. An edge is created between two nodes based on the overlap score of their PEEE lists. If the nodes are clusters of genes, the average of their expression signals is the representative signal for that cluster. Then the PEEE list for the average is the PEEE list for that node.

We define the overlap score for edge (i, j) by using a combination of following two scores:

(i) $S_{ij} = |Peee(i) \cap Peee(j)|$, i.e number of common PEEEs present in the event sets of both nodes;
(ii) $S_{i-j} = |Peee(i)| - |Peee(i) \cap Peee(j)|$, i.e. number of PEEEs present in node i but not in j.

Then,

$$(i,j) \in E_e \text{if } S_{ij} > \overline{S_{ij}} + Z_e\sigma \text{ and } \frac{S_{i-j}}{S_{ij}} < 0.3 \quad (1)$$

We consider an edge present if the overlap is Z_e standard deviations more than average. This z-score will serve as our threshold for the edges in E_e. To ensure the containment relationship but allow for noisy data we add the constraint that S_{i-j} is less than 30% of the overlap. In other words, for an edge (i,j) we allow for some PEEEs to be in i but not in j. The 30% we determined to be a well balanced cap on such events.

TF hierarchy The data used for the TFH is TF-DNA interaction location data (see below). An edge (i,j) in the TFH graph is established by carefully evaluating the overlapping and non-overlapping sets of TFs between nodes i and j. The nodes here are clusters of genes, so we count the overlap between pairs within and between clusters.

An edge (i,j) is defined by using a combination of following three scores:

(i) A, intercluster overlap of TFs. The score is obtained by counting the number of common TFs for all pairs formed by genes in cluster i and cluster j.
(ii) B, intracluster overlap of TFs. The score is obtained by counting the number of common TFs for all pairs formed by genes in cluster i.
(iii) C, intracluster overlap of TFs. The score is obtained by counting the number of common TFs for all pairs formed by genes in cluster j.

Then,

$$(i,j) \in E_{tf} \text{if } A > \overline{A} + Z_{tf}\sigma \text{ and } B < C \quad (2)$$

We consider an edge present if the TF overlap is Z_{tf} standard deviations more than average. To ensure the containment relationship we add the constraint that B is less than C.

4 Preliminary Studies

Here we use two separate data sets of yeast. The first is a time-course, genome-wide, gene expression data, known as the Cell Cycling Genes data, from Spellman et al. [5] Although somewhat dated, we used this data set because it is still one of the best time-course data expression data sets around, mostly because of the length of the series (i.e. number of measurements) as well as the sampling times which are small enough to capture some of the important processes in yeast. The data set consists of four separate time-series measurements of expression for each gene, totaling 76 measurements, for about 6200 genes of yeast. We impute the missing values using KNNimpute [11]. We concatenate all the measurements and obtain a 76 length real-valued vector for each gene. The second data set is the TF-DNA data by Lee et al. [6]. The set consists of 6200×106 P-values

indicating the confidence of binding for each of 106 TFs to all 6200 genes of yeast. By selecting a confidence value for each gene we obtain a TF profile of binding (i.e. a list of TFs that bind to the closest intergenic region to that gene). We used $P = 0.001$ which was used in the original analysis by the experimenters.

The Cell-Cycling Genes data set does not have too many features, i.e. the expression signals do not have many degrees of freedom as the conditions to which the genes were exposed in that experiment were not diverse. Thus one need to lower the dimensionality of the expression matrix, since over 6000 signals presents an overkill and will result in a large number of spurious events identified. So we clustered the data into a smaller number of clusters which should all be sufficiently different and offer variety of sub-signals. The genes' expression vectors were clustered using average-link hierarchical clustering with the Pearson's correlation as the distance measure. The clustering goal was indistiguishability of curves within clusters under visual observation. The resulting 87 clusters are the nodes in the hierarchy graphs.

For the Expression Hierarchy we create an average expression profile for each cluster by averaging the expression vectors from the Cell Cycling genes data. We run our modified edge detection algorithm to detect events (see Section 2). An event profile is created for each gene, consisting of runs of points labeled as increasing or decreasing. Pairs of profiles are overlapped and the events matching in location and direction are counted as common events. For the TF Hierarchy we create a regulation profile for each gene using the TF-DNA data. Then we calculate the overlap score between two clusters, from the expression data clustering, as defined above.

To test our model of regulation we compare the two resulting hierarchies, the EH and TFH. In ideal conditions, $E_e \subset E_{tf}$, i.e. all the edges in EH should be in TFH.

4.1 Efficacy of the Model: Comparing the Hierarchies

As a part of our preliminary studies we built several different EH and TFH, for different values of the z-scores Z_e and Z_{tf}. We also generated random graphs on the 87 nodes by permuting the expression data and running the algorithms to identify PEEEs and score them on this permuted data. The results for $Z_{ef} > 0.5$ and $Z_{ef} > 1$ are shown in Fig. 2.

Several things are evident from the figure. First of all the edges in the Expression Hierarchy correspond very well to the edges in the TF Hierarchy. As a matter of fact, with increasing Z_e we get up to 71% matches. The second observation is that the results are significant: the random graphs have many fewer edges (down to about 10%) that match the TFH. So we did in fact get most of the edges from the EH in TFH.

We also notice that the number of edges in EH is about 10% to 20% of those in TFH. This was to be expected as the genes were not exposed to many different conditions during the Cell Cycling Genes experiment, so the actual range of expression signals captured only a small number of modules' effects.

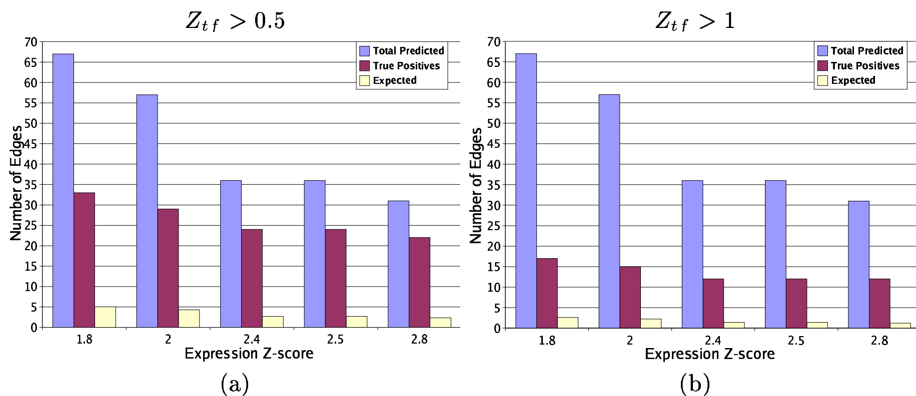

Fig. 2. Comparing the inferred Expression Hierarchy to a known TF Hierarchy, at two different thresholds of TF factor overlap $Z_{tf} > 0.5$ (560 edges) and $Z_t f > 1$ (290 edges). The total number of edges in E_e, for 5 varying thresholds of overlap, together with the true positives and the expected (random) matches is given. The correct predictions increase from 48% and 71%, with the expected number decreasing to 1/6 of that (for $Z_{tf} > 0.5$). In (b) although the statistical significance of the results is better than in (a) the sensitivity is lowered.

4.2 Inspecting the Hierarchy

The resulting clusters and hierarchy graphs can be obtained from our Web site: `graphics.cs.ucdavis.edu/~nyshah/Regulation`. Here we omit them for space considerations and report on some initial observations. First of all the hierarchy of the yeast genome is shallow with the longest paths being of length at most 5. This is well in agreement with other studies [12].

Next we examined coregulated nodes in the partially ordered graph by comparing the TFs from the TF-DNA location data set. Note that each node is a cluster of genes. The following are modules of TFs that were common to at least three genes between coregulated nodes. Together with the genes we give some annotation either from SGD [13] or individual references. Some of the TFs in the modules are known to act together while the others are not. More TF modules are available at our Web site.

Cluster 4 (12 genes) 4 regulated by FKH2, MCM1 and NDD1 MCM1 is a known yeast cell cycle regulator during the M and M/G1 phases. FKH2 is involved in the regulation of the SIC1 cluster, whose member genes are expressed in the M/G1 phase of the yeast cell cycle, and are involved in mitotic exit [14]. NDD1 is a high-dosage suppressor of cdc28-IN, essential for expression of a subset of late S phase specific genes in yeast [15].

Cluster 6 (12 genes) 4 regulated by all of GAL4, GAT3, RGM1, YAP5 GAL4 is a well known transcription factor for the GAL structural genes, which

encode galactose metabolic proteins. GAT3 (YLR013w) is a protein encoding GATA-family zinc finger motifs, known transcription factors [16]. RGM1 is a putative transcriptional repressor with proline-rich zinc fingers. YAP5 is a bZIP protein and a known transcription factor.

Cluster 13 (25 genes) 6 regulated by MBP1, SWI6 SWI6 is a transcription cofactor, forms complexes with DNA-binding proteins Swi4p and Mbp1p to regulate transcription at the G1/S transition. MBP1 is a cell-cycle regulating transcription factor.

5 Discussion and Directions

We presented here a model based approach to elucidating coregulation from time-course gene expression measurements. We introduce the Regulation Hierarchy as a structure that usefully summarizes transcriptional regulation, show how it relates to two more practical structures, the expression and TF hierarchies, and approximate it using gene expression data. Using publicly available data on gene expression and TF-DNA binding in yeast we were able to get encouraging results supporting the utility of the Regulation Hierarchy, and its derivation from expression data. We demonstrate one particular use for the RH by combining it with the TF-DNA data and identifying TF regulatory modules.

Again these are preliminary studies, and there are many things on which we need to improve. Our method for identifying PEEE is ad hoc and dated; better methods from time-series analysis will likely yield better PEEEs. The overlap scores can also be improved upon by doing alignment and maximal overlaps for example. But in spite of the deficiencies, our approach works as a proof of concept.

The biggest goal in front of us is building the regulation hierarchies from static expression data, of which there is thousands of available sets for yeast. If that is possible, which we do not know yet, the resulting hierarchies would have many more edges, as the genes would have been exposed to significantly more conditions than the ones in the data we used.

And of course as we mentioned before the regulation hierarchies are not ends in themselves but stepping stones toward identifying interactions between genes and gene products on a large-scale. In particular, they can be used jointly with gene expression data to limit the in-degree of nodes during network inference, which can speed up the process significantly.

References

1. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Science **85** (1998) 14863–14868
2. Davidson, E.: Genomic Regulatory Systems. Academic Press (2001)

3. Davidson, E., et al.: A genomic regulatory network for development. Science **295** (2002) 1669–1678
4. Filkov, V., et al.: Analysis techniques for microarray time-series data. Journal of Computational Biology **9** (2002) 317–330
5. Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Molecular Biology of the Cell **9** (1998) 3273–3297
6. Lee, T., et al.: Transcriptional regulatory networks in *saccharomyces cerevisiae.* Science **298** (2002) 799–804
7. Pilpel, Y., Sudarsanam, P., Church, G.: Identifying regulatory networks by combinatorial analysis of promoter elements. Nature Genet. **29** (2001) 153–159
8. Lapidot, M., Pilpel, Y.: Comprehensive quantitative analyses of the effects of promoter sequence elements on mrna transcription. Nucleic Acids Research **31** (2003) 3824–3828
9. Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., Friedman, N.: Module networks: Identifying regulatory modules and their condition specific regulators from gene expression data. Nature Genetics **34** (2003) 166–76
10. Holter, N., et al.: Fundamental patterns underlying gene expression profiles:simplicity from complexity. PNAS **97** (2000) 8409–8414
11. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.: Missing value estimation methods for dna microarrays. Bioinformatics **17** (2001) 520–525
12. Alon, U. Internationl Conference on Systems Biology (2003) Invited Talk.
13. Dolinski, K., et al.: Saccharomyces genome database (2004) http://www.yeastgenome.org/.
14. Zhu, G., et al.: Two-yeast forkhead genes regulate the cell-cycle and pseudohyphal growth. Nature **406** (2000) 90–94
15. Loy, B., et al.: Ndd1, a high-dosage suppressor of cdc28-in, in sacc. cerevisiae. Mol. Cell. Biol. (1999)
16. Cox, K., Pinchak, A., Cooper, T.: Genome-wide transcriptional analysis in s. cerevisiae by mini-array membrane hybridization. Yeast **15** (1999) 703–713