Discovering Transcriptional Modules from Motif, Chip-Chip and Microarray Data

T. De Bie, P. Monsieurs, K. Engelen, B. De Moor, N. Cristianini, and K. Marchal

Pacific Symposium on Biocomputing 10:483-494(2005)

DISCOVERING TRANSCRIPTIONAL MODULES FROM MOTIF, ChIP-CHIP AND MICROARRAY DATA

TIJL DE BIE, PIETER MONSIEURS, KRISTOF ENGELEN, BART DE MOOR
K.U.Leuven, ESAT-SCD, Kasteelpark Arenberg 10, 3001 Leuven, Belgium

NELLO CRISTIANINI
U.C.Davis Dept. of Statistics, 360 Kerr Hall, One Shields Ave, CA 95616, US

KATHLEEN MARCHAL
Centre of Microbial and Plant Genetics and ESAT-SCD, K.U. Leuven
Kasteelpark Arenberg 20, 3001 Leuven-Heverlee, Belgium

We present a method for inference of transcriptional modules from heterogeneous data sources. It allows identifying the responsible set of regulators in combination with their corresponding DNA recognition sites (motifs) and target genes. Our approach distinguishes itself from previous work in literature because it fully exploits the knowledge of three independently acquired data sources: ChIP-chip data; motif information as obtained by phylogenetic shadowing; and gene expression profiles obtained using microarray experiments. Moreover, these three data sources are dealt with in a new and fully integrated manner. By avoiding approaches that take the different data sources into account sequentially or iteratively, the transparency of the method and the interpretability of the results are ensured. Using our method on biological data demonstrated the biological relevance of the inference.

1 Introduction

Nowadays, data representative of different cellular processes are being generated at large scale. Based on these *omics* data sources, the action of the regulatory network that underlies the organism's behavior can be observed.

Whereas until recently bioinformatics research was driven by the development of methods that deal with each of these data sources separately, the focus is now shifting drastically towards integrative approaches dealing with several data sources simultaneously. Indeed, technological and biological noise in the individual data sources is often so prohibitive and unavoidable that standard methods are bound to fail. Then only a combined use of *heterogeneous* and *independently acquired* information sources can help to solve the problem. Furthermore, these different points of view on the biological system allow gaining a holistic insight into the network studied. Therefore, the integration of heterogeneous data is an important, though non-trivial, challenge of current bioinformatics research.

In this study we focus on 3 types of *omics* data that give independent information on the composition of transcriptional modules, the basic building blocks of transcriptional networks in the cell: *ChIP-chip* data (*chromatin immuno-*

precipitation on arrays) provides information on the direct physical interaction between a regulator and the upstream regions of its target genes; *motif* information as obtained by phylogenetic shadowing describes the DNA recognition sites of these regulators; and *gene expression profiles* obtained using microarray experiments describe the expression behavior in the conditions tested. By integrating these three data sources, we aim at identifying the concerted action of regulators that elicit a characteristic expression profile in the conditions tested, the target genes of these regulators, and the DNA binding sites recognized by these regulators, thus fully specifying the relevant regulatory modules.

Previous successful approaches to integrative analyses in bioinformatics can be found in the class of *kernel methods* [7,19] and methods based on *graphical models* [4,5,13,14,15]. Still, to our knowledge, no successful attempts to solve the problem of module inference exploiting all 3 independently acquired ChIP-chip, motif and expression data have been made so far. Furthermore, most existing approaches that exploit the availability of heterogeneous data sources proceed in a *sequential* or an *iterative* way (see e.g. [8] for simultaneous detection of motifs and clustering of expression data, e.g. [2] for an iterative approach using ChIP-chip and expression data, and e.g. [10] for simultaneous motif detection and analysis of ChIP-chip data).

In this paper, we present an approach that is different in spirit from previous methods, taking the different data sources into account in a highly *concurrent* way. The performance of the algorithm was demonstrated using the Spellman dataset [16] as a benchmark.

2 Materials and Algorithms

2.1 Data Sources

As microarray benchmark set the Spellman dataset was used [16], which contains 77 experiments describing the dynamic changes of 6178 genes during the yeast cell cycle. The profiles were normalized (subtracting the mean of each profile and dividing by the standard deviation across the time points) and stored in a gene expression data matrix further denoted by $\boldsymbol{A}$ with a row for each gene expression profile and a column for each condition.

Genome-wide location data performed by Lee et al. [9] were downloaded from http://web.wi.mit.edu/young/regulator_network. These contain data on the binding of 106 regulators to their respective target genes in rich medium. The ChIP-chip data matrix (further denoted by $\boldsymbol{R}$) used in our study consists of *one minus* the p-values obtained from combined ratio's between immuno-precipitated and control DNA (see [9]). Thus, a large value (close to one) indicates that the regulator is probably present.

The motif data used in this study were obtained from a comparative genome analysis between distinct yeast species (phylogenetic shadowing) performed by Kellis et al. [6]. The authors describe the detection of 72 putative regulatory motifs in yeast. These motifs, available online as regular expressions, were transformed into the corresponding probabilistic representation (weight matrix): for each motif, the 20 *Saccharomyces cerevisiae* genes in which the motif was most reliably detected according to the scoring scheme of Kellis et al. [6] were selected. The intergenic sequences of these genes were subjected to motif detection based on Gibbs sampling [MotifSampler,18]. If the statistically overrepresented motif in this set of putatively co-expressed genes corresponded to the motif that was detected by the comparative motif search of [6] the motif model was retained. As such 53 of the 71 motifs could be converted into a weight matrix. This weight matrix was subsequently used to screen all intergenic sequences for the presence of the respective regulatory motifs using MotifLocator [11]. Absolute scores were normalized [11]. As the score distribution of the motif hits depends on the motif length and the degree of conservation of the motif, the distribution of the normalized scores differs between motifs. Therefore, normalized scores were converted into percentile values. This allows for an unbiased choice of the thresholds on the motif quality parameter in the algorithm. The matrix containing these percentile values is the motif data matrix ***M*** that will be used in this work.

2.2 Module Construction Algorithm

The aim of the method is to find regulatory modules (which may be overlapping with each other), based on the gene expression, ChIP-chip, and Motif data matrices as specified above.

A module is fully specified by the set of genes it regulates (denoted by an index set ***g***, pointing to the relevant set of rows of ***R***, ***M*** and ***A***), in addition to the set of regulators (corresponding to the columns with indices in a set called ***r*** in the ChIP-chip matrix ***R***) and motifs (corresponding to the columns with indices ***m*** in the Motif matrix ***M***) that are responsible for the regulation of these genes. The goal of our method is to come up with regulatory modules specified in this way, by fully exploiting the heterogeneous data sources available.

We note that the principles behind the method developed here are based on ideas similar to those that laid the foundations for the Apriori algorithm, originally developed in the database community [1]. All implementations used in this paper have been done in matlab.

Seed construction. This is the main step of the algorithm, and allows the construction of a good guess (or *seed*) of the modules. The idealized goal of this step is to find a set of genes ***g***, that have the same expression profile, and such that there exist sufficiently large sets of regulators ***r*** and of motifs ***m*** that are entirely present in all these genes. Since in practice it is not known exactly in which intergenic regions a certain motif occurs or where a regulator binds, we have to resort to the score matrices $\boldsymbol{R}$ and $\boldsymbol{M}$. Furthermore, the expression profiles $\boldsymbol{A}$ of genes in a module will only be approximately equal, and possibly only in a set of conditions, so we relax this constraint to requiring a strong correlation instead of equality between them.

Formally, then the task to solve is:

Find all maximal gene sets $\boldsymbol{g}$ for which there exist an $\boldsymbol{r}$ of size $|\boldsymbol{r}| \geq r_{\min}$ and a set $\boldsymbol{m}$ of size $|\boldsymbol{m}| \geq m_{\min}$, such that the following 3 constraints are satisfied:

1. $\boldsymbol{R}(i,j) > t_{\boldsymbol{r}}$ for all $i \in \boldsymbol{g}$ and $j \in \boldsymbol{r}$
2. $\boldsymbol{M}(i,j) > t_{\boldsymbol{m}}$ for all $i \in \boldsymbol{g}$ and $j \in \boldsymbol{m}$
3. $\mathrm{corr}(\boldsymbol{A}(i,:), \boldsymbol{A}(j,:)) > t_{\boldsymbol{a}}$ for all $i,j \in \boldsymbol{g}$

where $r_{\min}$, $m_{\min}$ and thresholds $t_{\boldsymbol{r}}$, $t_{\boldsymbol{m}}$ and $t_{\boldsymbol{a}}$ are parameters of the method.

Here, a maximal set $\boldsymbol{g}$ is defined as a set that cannot be extended with another gene without violating one or more of these constraints. In the following, we will use the term *valid set* for a gene set $\boldsymbol{g}$ that satisfies these constraints.

Clearly it is computationally impossible to tackle this problem with a naive approach: the number of gene sets is exponentially large in the number of genes in the dataset, which is prohibitive even for the smallest genomes. However, it is trivial to verify that:

Observation 1: When a gene set does not satisfy the constraints, none of its supersets satisfy the constraints.

This means that we can build up the maximal sets incrementally, starting with valid sets of size one, and gradually expanding them. Concretely, the (already less naive) algorithm would then look like[1]:

[1] Notationally, we will use $\boldsymbol{L}^i$ to denote the list containing all valid gene sets with i genes. For an individual valid gene set we will use a bold face $\boldsymbol{g}_k^i$, with a superscript i to specify that it is an element of $\boldsymbol{L}^i$ and thus contains i genes, and with a subscript k to distinguish it from the other gene sets in $\boldsymbol{L}^i$. The x-th gene in this gene set is denoted as $g_k(x)$, for brevity without superscript.

```
- For all single genes, check if they satisfy constraints 1 and 2 (constraint 3 is
  trivially satisfied for singleton gene sets). Make a list $\boldsymbol{L}^1$ of all singleton
  gene sets that contain such a valid gene.
- Set $i = 2$.
- While size($\boldsymbol{L}^{i-1}$) $\neq 0$
      For $k=1$:size($\boldsymbol{L}^{i-1}$), expand set $\boldsymbol{g}_k^{i-1} =\{g_k(1), g_k(2), \dots , g_k(i-1)\} \in \boldsymbol{L}^{i-1}$ once
          for each gene $g$ that is not yet contained in $\boldsymbol{g}_k^{i-1}$. Put the thus
          expanded sets $\{g_k(1), g_k(2), \dots , g_k(i-1), g\}$ that satisfy the 3
          constraints (to be verified in $\boldsymbol{R}$, $\boldsymbol{M}$ and $\boldsymbol{A}$), in a list $\boldsymbol{L}^i$.
      Set $i = i+1$.
```

Notice that following this strategy, a gene set can be constructed in different ways, by adding the genes to it in a different ordering (i.e. in different iterations i). This can be avoided by adding a gene to a gene set $\boldsymbol{g}_k^{i-1}$ only whenever its row number g is larger than that of all other genes already in $\boldsymbol{g}_k^{i-1}$. Thus for every $\boldsymbol{g}_k^{i}=\{g_k(1), g_k(2), \dots , g_k(i)\} \in \boldsymbol{L}^i$ we always have that $g_k(x) < g_k(y)$ for $x < y$.

Additionally, in this way we can easily keep the list $\boldsymbol{L}^i$ of gene sets $\boldsymbol{g}_k^i$ sorted as well, where the sorting is carried out first according to the first added gene and last according to the last added gene. More formally: $\boldsymbol{g}_k^i$ preceeds $\boldsymbol{g}_l^i$ in $\boldsymbol{L}^i$ if and only if $g_k(\text{argmin}_x(g_k(x) \neq g_l(x))) < g_l(\text{argmin}_x(g_k(x) \neq g_l(x)))$ (this ordering of the list $\boldsymbol{L}^i$ is indeed a total ordering relation.)

Still the number of expanded gene sets can be huge in every iteration: each of the gene sets $\boldsymbol{g}_k^{i-1}$ in $\boldsymbol{L}^{i-1}$ must be expanded by all genes $g>g_k(i-1)$, after which the validity has to be checked by looking at the matrices $\boldsymbol{R}$, $\boldsymbol{M}$ and $\boldsymbol{A}$. This can still be too expensive. However, we can exploit the converse of Observation 1:

Observation 2: Whenever a gene set satisfies the constraints, all of its subsets satisfy the constraints.

Using this so-called *hereditary property* of the constraint set, in some cases we can conclude *a priori* —i.e. without checking in $\boldsymbol{R}$, $\boldsymbol{M}$ and $\boldsymbol{A}$— if an extended gene set of size i can possibly be valid or not: we simply have to check if all of its size $i-1$ subsets belong to $\boldsymbol{L}^{i-1}$. Only if this is the case, we still have to access the data in $\boldsymbol{R}$, $\boldsymbol{M}$ and $\boldsymbol{A}$; if it is not the case, we know without further investigation that the extended subset is invalid.

Specifically, assume that we expand the gene set $\boldsymbol{g}_k^{i-1}=\{g_k(1), g_k(2), \dots , g_k(i-2), g_k(i-1)\} \in \boldsymbol{L}^{i-1}$ with g, leading to $\{g_k(1), g_k(2), \dots , g_k(i-2), g_k(i-1), g\}$. Then, since for a valid size i set each of its size $i-1$ subsets must be contained in $\boldsymbol{L}^{i-1}$, also $\{g_k(1), g_k(2), \dots , g_k(i-2), g\}$ must be contained in $\boldsymbol{L}^{i-1}$. In other words: there has to exist a $\boldsymbol{g}_l^{i-1}=\{g_l(1), g_l(2), \dots , g_l(i-2), g_l(i-1)\} \in \boldsymbol{L}^{i-1}$ for which $g_k(x)=g_l(x)$ for $x \leq i-2$, and $g= g_l(i-1)$. This can efficiently be ensured *constructively*, by exploiting the fact that the list $\boldsymbol{L}^{i-1}$, and all $\boldsymbol{g}_k^{i-1}$ themselves are sorted. Indeed, thanks to this, all gene sets $\boldsymbol{g}_k^{i-1}$ that have the first *i-2* genes in common occur consecutively in $\boldsymbol{L}^{i-1}$. Therefore, to expand $\boldsymbol{g}_k^{i-1}$ with an additional gene, we only have to screen the list $\boldsymbol{L}^{i-1}$ starting at $\boldsymbol{g}_{k+1}^{i-1}$ and move forward in $\boldsymbol{L}^{i-1}$ for as long as the first *i-2* genes are equal to $g_k(1)$, $g_k(2),\ldots$ and $g_k(i-2)$. For every gene set $\boldsymbol{g}_l^{i-1}$ screened in this way, read the last gene $g_l(i-1)$ and append it to $\boldsymbol{g}_k^{i-1}$, thus resulting in a candidate gene set of size *i*, potentially to be appended to $\boldsymbol{L}^i$. To find out whether this candidate gene set is valid indeed, one still has to check the constraints explicitly. However, thus constructively exploiting the hereditary property, the number of queries to ***R***, ***M*** and ***A*** is drastically reduced. Note that this strategy also ensures that $\boldsymbol{L}^i$ is sorted automatically.

Module validation. In some cases the first step described above is not sufficient for adequate module inference. There are three reasons for this:

First, the seed construction method can be rather conservative in recruiting genes, since each of the genes in the module has to satisfy all 3 of the constraints. Therefore, in a second step, we calculate the mean of the expression profiles of the seed modules found in the first step, further called the *seed profile*. Then we can additionally recruit all genes with a high correlation with the seed profile to be incorporated in the module. In order to determine an optimal threshold value for this correlation, we compute the enrichment of each of the motifs and regulators in the genes that have an expression profile that achieves this threshold correlation with the seed profile. The logarithm of the p-value of the enrichment is then plotted as a function of this threshold (Figure (1)), and the threshold can be chosen such that this value is minimal.

Second, sometimes it is undesirable to a priori decide how many motifs and regulators we want in the module, or it may be difficult to choose the thresholds t_r, t_m and t_a (even though experiments show little dependence on these). Then one can first use the seed construction algorithm requiring only 1 regulator and motif, and with stringent thresholds, after which again the enrichment of all motifs and regulators can be plotted as a function of the correlation threshold with the mean profile of the seed module. For each such seed profile, the corresponding enrichment plot will visually hint at the number of motifs and regulators (namely the number of significantly enriched motifs and regulators).

Third, similarly, the enrichment plot allows excluding false positive motifs or regulators: when they are selected in step 1, but appear not to be enriched in the validation step, they are considered as a false positives and discarded.

To calculate the enrichment, we first calculate the mean score of the module for the particular motif or regulator. Note that the mean score of a module by random gene selection is approximately Gaussianly distributed (central limit theorem), with mean equal to the mean over all genes, and variance equal to the overall variance divided by the size of the module. Thus, we can calculate the enrichment as the logarithm of the p-value based on a Gaussian approximation.

Note that the p-values have been computed based on profiles that have been obtained from the data, such that they do not have a rigorous probabilistic interpretation here. Hence, we can only use them as explained above.

2.3 Calculating Overrepresentation of Functional Classes

Functional categories for each gene were obtained from MIPS [12]. Functional enrichment of the modules was calculated using the hypergeometric distribution [17], which assigns to each functional class a p-value.

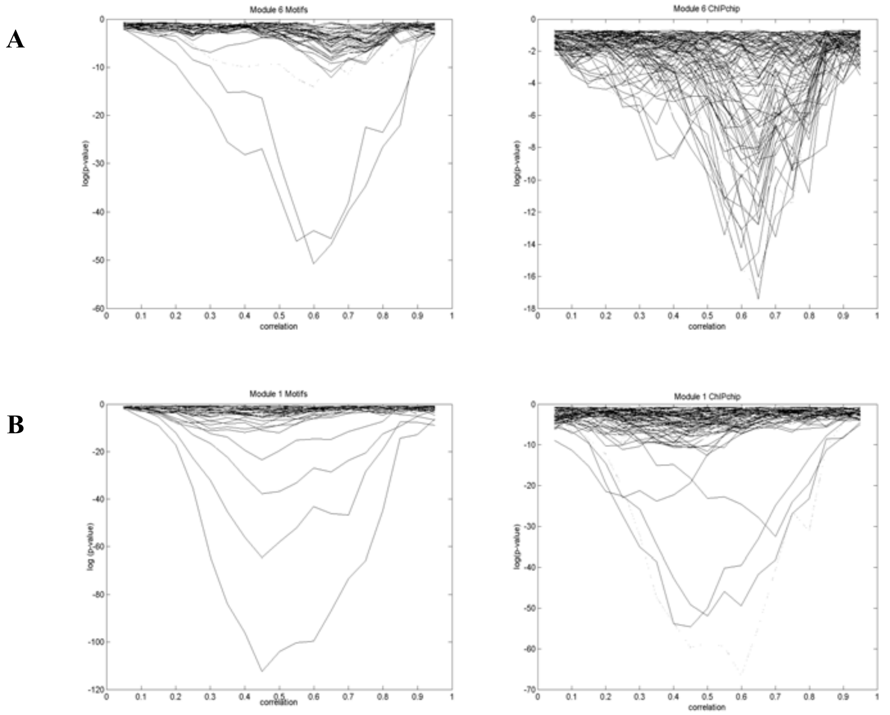


Figure 1: Two examples (panels A and B) of the module validation step for two seed profiles: on the left, the logarithms of the p-values (vertical axes) are plotted for all motifs as a function of the correlation threshold (horizontal axes). I.e. each line in the plot shows the p-values for the enrichment of its corresponding motif, as a function of the gene expression correlation threshold used. The right column shows similar plots for the regulators. Clearly panel A shows the results for a false positive prediction (module 6): the regulator (right figure) of the identified seed module turns out not to be significantly overrepresented in genes correlated with the seed profile. In panel B the results are displayed for the positive example described in the text.

3 Results

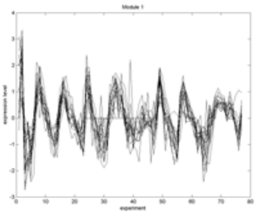
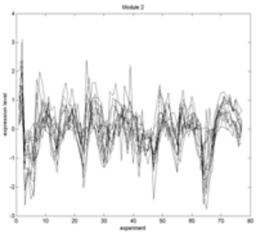
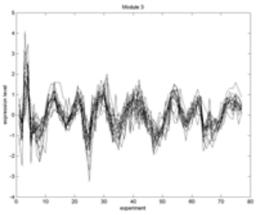
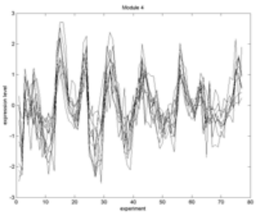
Cell cycle related modules. To test the reliability of our method, we used the well-studied Spellman dataset as benchmark. The analysis we performed using our two-step algorithm is illustrated by elaborating on the detection of the cell cycle related module 1. Using the seed detection step we searched for modules of genes having at least 1 common motif (1M) in their intergenic sequences and 1 common regulator (1R) showing a small p-value in the ChIP-chip data, and of which the expression profiles were mutually correlated with a minimal correlation of 0.7. This seed identification step then predicts several potential modules, and for each of them a seed profile can be calculated. For each of these modules we performed the module validation step.

Fig. 1A (right) shows how this validation step allows to detect that the regulator associated with this module is probably a false positive: when recruiting genes that correlate stronger than a certain threshold with the seed profile, it is not significantly enriched in the recruited gene set, no matter what the threshold is.

In Fig 1B, using the parameter settings of 1M/1R, we identified a potential seed module containing regulator 98 (Swi4) and motif M_11 (known as a Swi4 motif). Calculating the statistical overrepresentation of all motifs and regulators in genes correlated with the seed profile of this putative module showed that in this subset of genes indeed M_11 and Swi4 were overrepresented. The identified module seed thus is likely to be biologically relevant. These results also show that besides Swi4 and M_11, 3 additional motifs and regulators were overrepresented in subsets of genes correlated with the module seed profile, indicating the probable underestimation of the real module size. To verify whether these other regulators/motifs co-occur in the same subsets of genes and therefore comprise a larger module, we repeated the seed identification step using additional parameter settings (see Table 1 in the online supplement). From this result it appeared that we could recover a complete module consisting of the 3 overrepresented regulators (Mbp1, Swi4, Swi6) and 2 motifs (M_16, M_10) and that this module is present in genes displaying an expression profile that shows a correlation of at least 0.7 with the average seed profile. Checking the identities of the regulators and the motifs (regulators Mbp1, Swi4, Stb1 combined with the regulatory motifs Mbp1 (M_18, M_12) and Swi4 (M_11 and M67)) showed that we identified a previously extensively described regulatory module of the yeast cell cycle.

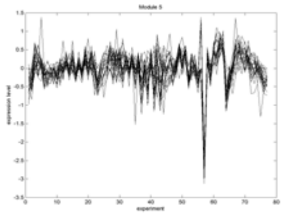
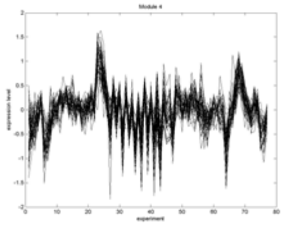
Besides this first module, 3 additional related cell cycle (Table 1) modules could be retrieved. Additional information on each of the separate modules can be found in the online supplement. Genes in the different modules showed peak expressions shifted in time relative to each other, as shown in Figure 1 of the online supplement. All of the predicted modules are conform the previously described knowledge on the cell cycle [2,3,9].

Table 1: Cell cycle related modules. Column 'R' contains the regulators, column 'M' the motifs, the column 'Functional Class: p-value' contains p-values for several functional classes, and the 'Seed Profile' column contains a plot with the expression profiles of the genes regulated by the module.

	R	M	Functional Class: p-value	Seed Profile
Module 1	Mbp1 Swi6 Swi4 Stb1	M_18 (Mbp1) M_12 (Mbp1) M_11 (Swi4) M_67 (Swi4)	**10** CELL CYCLE AND DNA PROCESSING: 0 **10.03** cell cycle: 2.7e-5 **10.01** DNA processing: 1.3e-4 **42.04** cytoskeleton: 4.2e-3	
Module 2	Swi4 Mbp1 Swi6 FKH2	M_18 (Mbp1) M_12 (Mbp1) M_11 (Swi4) M_8 (Mcm)	**40** CELL FATE: 5.2e-4 **40.01** cell growth / morphogenesis: 2.6e-3 **43** CELL TYPE DIFFERENTIATION: 5.2e-3 **43.01** f ungal/microorganismic cell type differentiation: 5.2e-3 **34.11** cellular sensing and response:5.3e-3 **01.05.01** C-compound and carbohydrate utilization: 6.8e-3 **10.03.04.03** chromosome condensation: 9.4e-3	
Module 3	NDD1 FKH2 Mcm1	M_8 (Mcm) M_30 (Mcm)	**43** CELL TYPE DIFFERENTIATION: 3.6e-3 **43.01** fungal/microorganismic cell type differentiation: 3.6e-3 **10.03.03** cytokinesis (cell division) /septum formation: 4.8e-3	
Module 4	Swi5 (Ace2)	M_8 (Mcm)	**32.01** stress response: 3.2e-3 **10.03** cell cycle: 8.7e-3	

Non cell cycle related modules. Besides the modules primarily involved in cell cycle, other modules could be identified in the Spellman dataset (see Table 2). Module 5, consisting of Fhl1, Rap1 and Yap5, involved in the regulation of ribosomal proteins was previously also identified [9]. Note that it was identified from a profile that does not change significantly and consistently with the cell cycles. By our analysis we could pinpoint motif M_54 [6], as the regulatory motif correlated with this regulatory module. A second non cell cycle related module consisted of the genes regulated by the motifs M_7 and M_3 (identified as ESR1 and ESR2 [6]). For this module, related to transcription and ribosomal RNA processing only the motifs seemed informative (see module 6 in Table 2, and Figure 1A).

Table 2: Non cell cycle related modules.

	R	**M**	**Functional Class: p-value**	**Seed Profile**
M **o** **d** **u** **l** **e** **5**	FKL1 Yap5 Rap1	M_54	**12** PROTEIN SYNTHESIS: 0 **12.01** ribosome biogenesis: 0	
M **o** **d** **u** **l** **e** **6**	/	M_3 (ESR1) M_7 (ESR2)	**11** TRANSCRIPTION: 0.000002 **11.04** RNA processing: 0 **11.04.01** rRNA processing: 0	

4 Discussion

We described a 2-step methodology combining ChIP-chip, motif and expression data to infer complete descriptions of transcriptional modules. The seed construction step predicts the putative modules consisting of regulators, their corresponding motifs and the elicited expression profile. The validation step filters false positive predictions and gives further insight into the module size.

The problem is attacked in a very direct way: the integration of the data sources is achieved in a one-shot-algorithm, and requires no iteration over the different data sources. While the running time was very reasonable for all experiments carried out for this paper, it heavily depends on the parameters. The more stringent they are set, the smaller the lists $\boldsymbol{L}^i$ will be and the faster the algorithm will run. Further speed-ups are possible, but not needed for the experiments reported in this paper (for all parameter setting used, it remained below 1 hour for the slowest step, which is the seed identification, on an intel pentium 2GHz laptop with 512Mb RAM). Therefore we will not go into these here.

The Spellman dataset was used as a benchmark to test the performance of our method. Since this dataset and the yeast cell cycle have extensively been studied before [2,9], it is ideally suited for testing the reliability and biological relevance of the predictions. We were able to reconstruct 4 important modules known to be involved in cell cycle and also 2 non cell cycle related modules without using any prior biological knowledge or prior data reduction. These results indicate that predictions passing the module validation step are likely to be biologically relevant (no false positives present).

5 Conclusion

The 3 data types mutually agreeing with each other on the prediction of a module not only results in the most reliable predictions (as was the case for the cell cycle related modules), but also allows correlating a set of regulators with their corresponding regulatory motifs and elicited profiles in a very natural and direct way. On the other hand, because of the restricted number of experimental data yet available (ChIP-chip data not known for all regulators and tested in a limited set of conditions, expression data for specific conditions not available), and the questionable quality of the motif models, the presence of a signal in 1 data type can compensate for the lack of it in another data type, allowing still to retrieve the module.

While to our knowledge this is the first time these 3 independently acquired data sources are exploited in such a concurrent way for module identification, the approach is further extendible towards any number of information sources, and in principle towards the use of other data types. The only condition for an efficient method to exist is that the constraints the gene sets have to satisfy must be hereditary. This extension will be the subject of future work.

Acknowledgments

KM is a post-doctoral researcher of the Fund for Scientific Research – Flanders (F.W.O–Vlaanderen). TDB is a research assistant with the Fund for Scientific Research – Flanders (F.W.O.–Vlaanderen). KE is a research assistant with the IWT.

This work is partially supported by: 1. IWT projects: GBOU-SQUAD-20160; 2. Research Council KULeuven: GOA Mefisto-666, GOA-Ambiorics, IDO genetic networks; 3. FWO projects: G.0115.01 (microarrays/oncology), G.0413.03 (inference in bioi), G.0388.03 (microarrays for clinical use), G.0229.03 (ontologies in bioi), and G.0241.04 (Functional Genomics); 4. IUAP V-22 (2002-2006).

Supplementary information:
www.esat.kuleuven.ac.be/~kmarchal/Supplementary_Info_PSB2005/SuppWebsiteYeastPSB.html

References

1. R. Agrawal et al, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207-216 (1993)
2. Z. Bar-Joseph et al, *Nature Biotechnology*, **21**(11): 1337-1342 (2003)
3. M.C. Costanzo et al, *Nucleic Acids Research*, **28**(1): 73-76 (2000)
4. N. Friedman, *Science,* **303**(5659): 799-805 (2004)
5. A. Hartemink et al, *Proceedings of the Pacific Symposium on Biocomputing*, **7**: 437–449 (2002): 166-176 (2003)
6. M. Kellis et al, *Nature*, **423**(6937): 241-254 (2003)
7. G. Lanckriet et al, *Bioinformatics* Advance Access, bth294 (2004)
8. M. Lapidot and Y. Pilpel, *Nucleic Acids Research*, **31**(13): 3824-3828 (2003)
9. T.I. Lee et al, *Science*, **298**(5594): 799-804 (2002)
10. X.S. Liu et al, *Nature Biotechnology*, **20**(8): 835-839 (2002)
11. K. Marchal et al, *Genome Biology*, **5**(2): R9 (2004)
12. H.W. Mewes et al, *Nucleic Acids Research*, **32** Database issue D41-D44 (2004)
13. N. Narai et al, *Proceedings of the Pacific Symposium on Biocomputing*, **9**: 336-347 (2004)
14. E. Segal et al, *Bioinformatics*, **19** (Suppl 1): 264-272 (2003)
15. E. Segal et al, *Nature Genetics*, **34**(2)
16. P.T. Spellman et al, *Molecular Biology of the Cell*, **9**: 3273–3297 (1998)
17. S. Tavazoie et al, *Nature Genetics*, **22**(3):281-285 (1999)
18. G. Thijs et al, *Journal of Computational Biology*, **9**(2): 447-464 (2002)
19. J.P. Vert and M. Kanehisa, *Bioinformatics*, **19**: 238ii-234ii (2003)