BIOINFORMATICS ORIGINAL PAPER
Vol. 25 no. 4 2009, pages 490–496
doi:10.1093/bioinformatics/btn658

Gene expression

Module networks revisited: computational assessment and prioritization of model predictions

Anagha Joshi[1,2], Riet De Smet[3], Kathleen Marchal[3,4], Yves Van de Peer[1,2] and Tom Michoel[1,2,*]

[1]Department of Plant Systems Biology, VIB, [2]Department of Molecular Genetics, Ghent University, Technologiepark 927, B-9052 Gent, [3]CMPG, Department of Microbial and Molecular Systems, KULeuven, Kasteelpark Arenberg 20 and [4]ESAT-SCD, KULeuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

Received on October 2, 2008; revised on November 24, 2008; accepted on December 19, 2008

Advance Access publication January 9, 2009

Associate Editor: David Rocke

ABSTRACT

Motivation: The solution of high-dimensional inference and prediction problems in computational biology is almost always a compromise between mathematical theory and practical constraints, such as limited computational resources. As time progresses, computational power increases but well-established inference methods often remain locked in their initial suboptimal solution.

Results: We revisit the approach of Segal *et al.* to infer regulatory modules and their condition-specific regulators from gene expression data. In contrast to their direct optimization-based solution, we use a more representative centroid-like solution extracted from an ensemble of possible statistical models to explain the data. The ensemble method automatically selects a subset of most informative genes and builds a quantitatively better model for them. Genes which cluster together in the majority of models produce functionally more coherent modules. Regulators which are consistently assigned to a module are more often supported by literature, but a single model always contains many regulator assignments not supported by the ensemble. Reliably detecting condition-specific or combinatorial regulation is particularly hard in a single optimum but can be achieved using ensemble averaging.

Availability: All software developed for this study is available from http://bioinformatics.psb.ugent.be/software.

Contact: tom.michoel@psb.ugent.be

Supplementary information: Supplementary data and figures are available from http://bioinformatics.psb.ugent.be/supplementary_data/anjos/module_nets_yeast/.

1 INTRODUCTION

One of the central goals of the top-down approach to systems biology is to infer predictive mathematical network models from high-throughput data. Much of the driving force for the development of network inference methods has come from the availability of various types of large-scale datasets for particular model organisms like *Saccharomyces cerevisiae* and *Escherichia coli*. In contrast, data generation for other organisms has been much slower and mainly focused on gene expression data. These gene expression datasets for typically more complex organisms pose their own challenges, such as a higher number of genes, limited number of experimental conditions, and supposedly a more complex underlying transcriptional network. Therefore, improvement and refinement of methods for network inference from gene expression data continues to be of great interest. Several reviews on a variety of methods have been written (Bansal *et al.*, 2007; Bussemaker *et al.*, 2007; Friedman, 2004; Gardner and Faith, 2005), and development of new methods remains an active area of research (Alter and Golub, 2005; Basso *et al.*, 2005; Bonneau *et al.*, 2006; Faith *et al.*, 2007). Here, we revisit the module network method of Segal *et al.* (2003) to infer regulatory modules and their condition-specific regulators from gene expression data and show that better and more refined module networks can be obtained by using advanced statistical and computational methods. These improvements concern the use of Monte Carlo (Liu, 2004) and ensemble strategies (Carvalho and Lawrence, 2007; Webb-Robertson *et al.*, 2008).

Following Hartwell *et al.* (1999) a 'module' is to be viewed as a discrete entity composed of many types of molecules and whose function is separable from that of other modules. Understanding the general principles that determine the structure and function of modules and the parts they are composed of can be considered one of the main problems of contemporary systems biology (Hartwell *et al.*, 1999). The module network method of Segal *et al.* (2003) addresses this problem using gene expression data as its input. It has yielded novel biological insights in a number of complex eukaryotic systems (Lee *et al.*, 2006; Li *et al.*, 2007; Novershtern *et al.*, 2008; Segal *et al.*, 2003, 2007; Zhu *et al.*, 2007) and has been the source of inspiration for numerous computational approaches to network inference as evidenced by its high number of citations. A module network is a probabilistic graphical model (Friedman, 2004) which consists of modules of coregulated genes and their regulatory programs. A regulatory program uses the expression level of a set of regulators to predict the condition-dependent mean expression of the genes in a module. Segal *et al.* (2003) used a deterministic optimization algorithm that searches simultaneously for a partition of genes into modules and a regulation program for each module. We consider both as separate tasks. When searching for modules, often many local optima exist with partially overlapping modules differing from each other in a few genes. We use a Gibbs

*To whom correspondence should be addressed.

sampling approach for two-way clustering of genes and conditions to generate an ensemble of partially overlapping partitions of genes into modules and produce an ensemble averaged solution (Joshi *et al.*, 2008). This centroid solution consists of so-called *tight clusters* (TCs), subsets of genes which consistently cluster together in almost all local optima. We also use a probabilistic method for learning regulatory programs. These regulatory programs take the form of fuzzy decision trees with regulator expression levels at the decision nodes and generalize the regression tree approach of Segal *et al.* (2003). By summing the strength with which a regulator participates in each member of an ensemble of regulatory programs for a certain module, we obtain a regulator score which gives a statistical confidence measure for the assignment of that regulator. Together, the Gibbs sampling cluster algorithm and probabilistic regulatory program learning provide a computationally efficient method to generate ensembles of module networks from which a centroid-like summarization can be constructed.

We have applied this ensemble method to the very same dataset as Segal *et al.* (2003) and performed several comparison tasks. First, we considered the probabilistic models and evaluated them on training as well as test data. We show that the model inferred by Segal *et al.* (2003) is equivalent to a single instance of the ensemble of models inferred by our algorithm. The TCs obtained from the ensemble solution generate a quantitatively better model than each of the single instances, including the model of Segal *et al.* (2003). Second, we compared the clustering of genes. TCs are in general more functionally coherent and improve the original modules in two ways. They can remove spurious profiles and fetch only the core of tightly coexpressed genes from a single module, or they can merge separate but related modules into one cluster. Third, we used the regulator score to analyze the network of modules and their associated regulators from Segal *et al.* (2003). We show that this network contains both high- and low-scoring regulators and that several high-scoring regulators are missed by the solution of Segal *et al.* (2003). In general, regulator assignments which can be validated by external sources such as ChIP data or literature are highly ranked. In combination with the TCs, the probabilistic method assigns more regulators supported by literature and the clusters to which they are assigned contain a higher ratio of known targets compared with the module network of Segal *et al.* (2003). Fourth, we show that the regulator scoring scheme can also be used to infer context-specific and combinatorial regulation by identifying pairs of regulators which occur significantly often together in the same regulation program.

Finally, we have applied the ensemble method to a bHLH module network that was recently inferred for mouse brain (Li *et al.*, 2007). Li *et al.* (2007) used their module network to make several hypotheses about modes of combinatorial regulation among different brain tissues. We show that only few of these hypotheses are statistically supported by the ensemble method. This example illustrates the usefulness of an approach which can generate internal significance measures, in particular if no other data sources are available to validate hypotheses generated by a single local optimum.

Together all these results convincingly show that the ensemble method for learning module networks significantly improves the direct optimization method of Segal *et al.* (2003). Unlike a single optimum, ensemble averaging allows the assessment and prioritization of the statistically most reliable modules and their condition-specific regulators. Such high-confidence modules can be used directly for generating experimentally verifiable hypotheses or can be integrated with other, perhaps smaller scale, data sources to create a more comprehensive view of the underlying networks.

2 RESULTS AND DISCUSSION

2.1 Data and procedure

We obtained all data from the supplementary website of Segal *et al.* (2003), including expression data, gene modules and regulatory programs. Using the Gibbs sampler we generated 12 different partitions of genes into modules which were combined into one set of TCs. The number of clusters is determined automatically by the Gibbs sampler and ranges from 65 to 78 in the different runs, compared with the predefined value of 50 of Segal *et al.* (2003). Of the 2355 genes in the dataset, 1892 could be assigned with high confidence to 69 TCs. To generate regulator assignment scores, we learned 10 probabilistic regulation (PR) programs per module with 100 regulator and split value pairs sampled per regulation program node. More details about these procedures are given in Section 4. This resulted in four different module network models:

(1) SCSR: Segal clusters with Segal regulation (SR) programs, corresponding exactly to the results of Segal *et al.* (2003).
(2) SCPR: Segal clusters with PR programs.
(3) GCPR: Gibbs sampler clusters (single run) with PR programs.
(4) TCPR: TCs (multiple Gibbs sampler runs combined) with PR programs.

2.2 Model evaluation

A module network infers a probabilistic model which explains relations between expression levels of a set of genes. More precisely, there is a probability distribution $p(x_1, \ldots, x_N)$ which computes the probability (density) to observe a particular combination of expression levels x_i for a set of N genes. This probabilistic model predicts the response in expression of genes in a module upon perturbations of its regulators, such as knockout or overexpression, and thus yields biologically verifiable hypotheses. For a module network, the distribution $p(x_1, \ldots, x_N)$ is a product of N factors (see Section 4), so we consider the normalized quantity $\mathcal{L} = (1/N)\log p(x_1, \ldots, x_N)$ which can be compared between models with potentially different numbers of genes. Higher values of $\mathcal{L}$ mean better explanation of the data by the model, i.e. more accurate prediction of the outcome of new experiments.

First, we performed evaluations on each of the conditions in the original dataset. Figure 1a shows that the histogram of $\mathcal{L}$-values for SCPR fits well within a non-parametric curve fit of the histogram for GCPR. This implies that the clusters found by Segal *et al.* (2003) are equivalent to one local optimum identified by the Gibbs sampler procedure. Figure 1b shows the histogram of $\mathcal{L}$-values for SCSR (red) overlayed on the histogram for SCPR (blue), both with non-parametric curve fits. The mean $\mathcal{L}$-values obtained by SCPR are higher than SCSR by a one-tailed t-test ($\alpha = 0.01$) proving that PR programs give a better explanation of the data. In Figure 1c, we compared SCPR with TCPR. TCPR has a higher mean $\mathcal{L}$ than SCPR with a one-tailed t-test ($\alpha = 0.01$). This shows that TCs are selecting a subset of genes which are the most informative and therefore generate a better model.

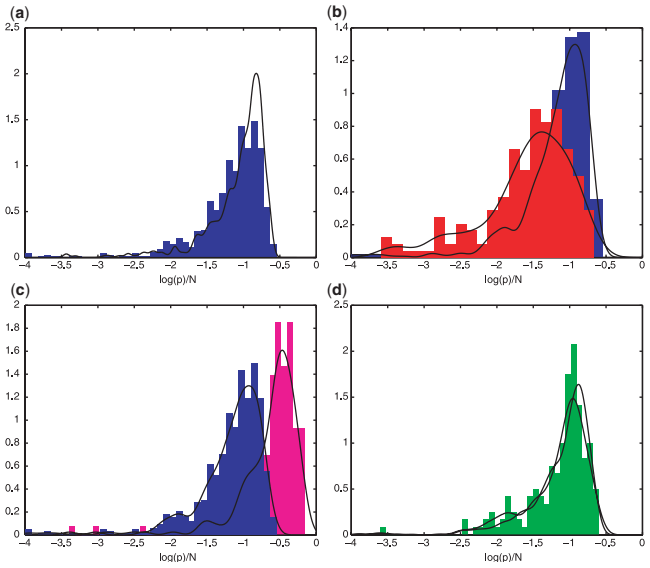

Fig. 1. Model evaluation experiments. **(a)** Histogram of $\mathcal{L} = (1/N)\log p(x_1, \ldots, x_N)$ for SCPR (blue) and non-parametric fit of the histogram for GCPR (black curve). **(b)** Histogram of $\mathcal{L}$ for SCSR (red) overlayed on histogram of $\mathcal{L}$ for SCPR (blue), with non-parametric fits (black curves). **(c)** Histogram of $\mathcal{L}$ for SCPR (blue) overlayed on histogram of $\mathcal{L}$ for TCPR (magenta), with non-parametric fits (black curves). **(d)** Histogram and non-parametric fit (left black curve) of $\mathcal{L}$ for GCPR learned on training data and evaluated on test data (green) and non-parametric fit of the same models evaluated on training data (right black curve). All histograms and curves are normalized to have area equal to 1.

Next we tested how well these models explain unseen data by performing a cross-validation experiment. We removed 10% of the conditions at random from the complete data (the test set) and ran the Gibbs sampler once on the remaining 90% (the training set). The resulting model was then evaluated on the test set. This procedure was repeated 10 times and all test set evaluation values were collected in one histogram and compared with the training set values (Fig. 1d). The curve of the test set is slightly shifted to the left with respect to the training set curve, as one would expect, but both curves have the same mean with a one-tailed t-test ($\alpha = 0.01$). This shows that the probabilistic models indeed generalize to unseen data.

2.3 Gene clustering improvement

We have shown in the previous section that SCPR is equivalent to GCPR but TCPR gives a better model over SCPR. We also observe that TCs are overall more functionally coherent than the clusters obtained in Segal *et al.* (2003) (SC). Figure 2 shows the fraction of genes in a cluster belonging to a MIPS functional category which is significantly overrepresented ($P < 0.001$) in SC and TC. Several examples illustrate the general trend seen in this figure. In TC-40, 4/7 genes are involved in amino acid transport compared with SC-27 with 8/53 genes. In TC-27, 7/9 genes belong to purine nucleotide anabolism compared with SC-11 with 6/53 genes.

Segal cluster 1 (SC-1) contains 55 genes, out of these 32 (58%) are validated targets of Hap4, a global regulator of respiratory genes, according to the YEASTRACT database (Teixeira *et al.*, 2006). This cluster has maximum overlap with TC-7 with 30 genes out

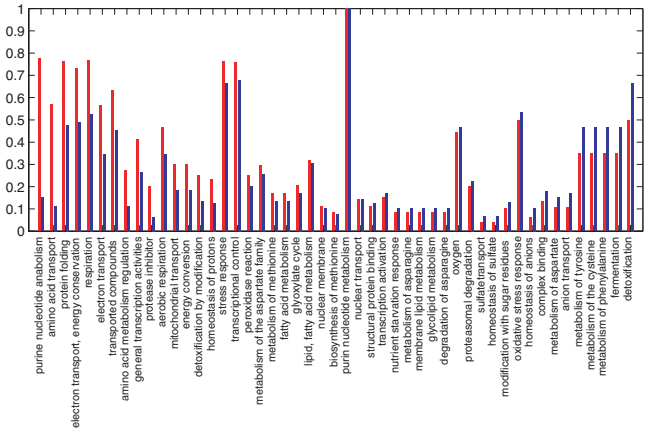

Fig. 2. Histogram of the highest fraction of genes in one module in a MIPS functional category for TC (red) and SC (blue), sorted by ratio difference.

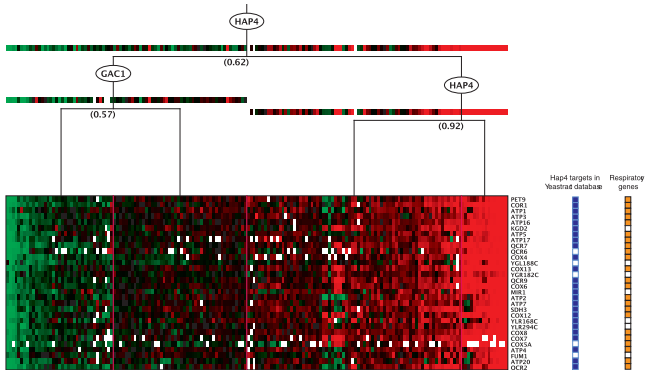

Fig. 3. TC-7 with Hap4 assigned as a top regulator. Genes known to be regulated by Hap4 in YEASTRACT are marked in blue and those involved in respiration are marked in orange.

of which 25 (83%) are known Hap4 targets. The five remaining genes are Qcr6, Cox5a and Fum1, all located in mitochondrion and involved in respiration, and two unknown genes Ygl188c and Ygr182c. With 24/30 respiratory genes (80%), TC-7 even improves on COGRIM (Chen *et al.*, 2007) which combines multiple data sources. Using expression data alone [the same dataset as Segal *et al.* (2003)], Chen *et al.* (2007) obtain a cluster with 32/51 (62%) genes belonging to MIPS respiration category. Using both ChIP and expression data, they obtain a cluster with 23/34 (68%) respiratory genes, significantly lower than TC-7. Figure 3 shows TC-7 with known Hap4 targets and respiratory genes marked in blue and orange, respectively.

TC-27 contains nine genes which form a subset of SC-11 containing 53 genes (Fig. 4a). Six genes (67%) in this cluster are known Bas1 targets compared with only 18% Bas1 targets in SC-11. TC-28 and TC-37 contain 70% and 100% known targets of Msn4, respectively. These clusters have a large overlap with SC-3 and SC-41, respectively, which have 55% and 93% known targets of Msn4. TC-1 consists of 51 genes, out of which 28 (55%) are known to be Swi4 targets. This module merges genes from SC-10, 29 and 30. They have 4/37 (11%), 19/41 (46%)

and 8/30 (27%) Swi4 targets, respectively. TC-11 contains genes of SC-8 and SC-9 whose highest ranked regulator is Gat1 (see Section 2.4) (Fig. 4b). YEASTRACT data confirms 17% of these targets, while for SC-8 and 9 overall 15% targets are confirmed by YEASTRACT. TC-35 is overrepresented for genes involved in RNA export from nucleus (P-value 10^{-8}). It overlaps with SC-19, 31 and 36 (P-values $\sim 10^{-3}$). TC-31 contains genes mainly involved in ribosomal biogenesis (P-value 10^{-13}) and combines relevant genes from SC-13, 14 and 15 (P-values $\sim 10^{-4}$).

We conclude that TCs improve clustering results obtained by Segal *et al.* (2003) in two ways. They can fetch only the core of tightly coexpressed genes from a SC (Fig. 4a), or they can merge clusters which were separate in SC (Fig. 4b).

2.4 Regulator assignment prioritization

The ensemble approach generates multiple equally plausible regulatory programs for a single module in a probabilistic fashion. The regulator assignment score which takes into account how often a regulator is assigned to a module, with what score, and at which level in the regulation tree, can therefore be used to prioritize regulators (highest regulator score gets topmost rank).

First, we consider only the difference between probabilistic regulator assignment and the original method by comparing SCSR with SCPR, hence keeping the gene modules the same for both methods. Figure 5 shows regulator–module links in SCSR [cf. Fig. 5 in Segal *et al.* (2003)]. The edges colored red are the ones supported by literature [data from Segal *et al.* (2003)]. To each edge we add the rank with which it is assigned in SCPR. Regulator–module links supported by literature have often a higher rank. SCSR assigns Hap4, a global regulator of respiratory genes, to SC-1. This cluster contains 58% known Hap4 targets and Hap4 has second highest rank in SCPR. SCSR also assigns Hap4 to SC-10 which contains genes involved in amino acid metabolism. SC-10 has only 2/37 (5%) known Hap4 targets according to YEASTRACT and this assignment is ranked very low (rank 73) in SCPR. Several high-ranking SCPR assignments which were missed by SCSR could also be validated using Harbison *et al.* (2004) data ($P < 0.005$). We assign Gal80, a transcriptional regulator involved in the repression of Gal genes in the absence of glucose, with second rank to SC-6. This is a cluster of four Gal genes, Gal1, Gal2, Gal7 and Gal10. Met32, a zinc-finger DNA-binding protein involved in transcriptional regulation of the methionine biosynthetic genes assigned with third rank to SC-8, and Gis1, a histone demethylase assigned to SC-3 with 5th rank, are supported by YEASTRACT (respectively, 5/29 and 6/31 known targets).

Next, we compared TCPR with SCSR to analyze the combined improvement made by the ensemble averaging at the level of gene clustering as well as at the level of regulator assignment. For TCPR, we selected the top six regulators for each cluster. This rank cutoff was determined as follows. We computed the significance for the overlap between each TC and each transcription factor target set using the YEASTRACT database. A reference module network was formed by keeping all transcription factor—TC edges below a certain P-value cutoff. By comparison with this reference network,

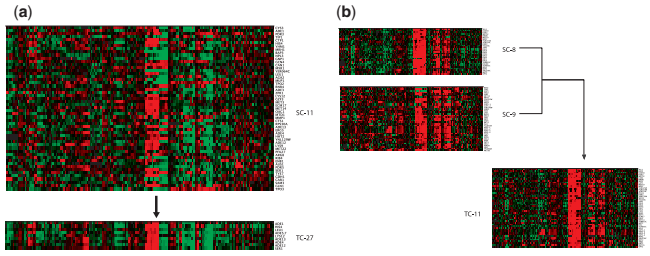

Fig. 4. **(a)** TC-27 fetches the core of tightly coexpressed genes from SC-11; 67% genes in TC-27 are known to be Bas1 targets. **(b)** SC-8 and SC-9 which have similar expression are merged into TC-11. SC-8, SC-9 and TC-11 all are enriched for Gat1 targets.

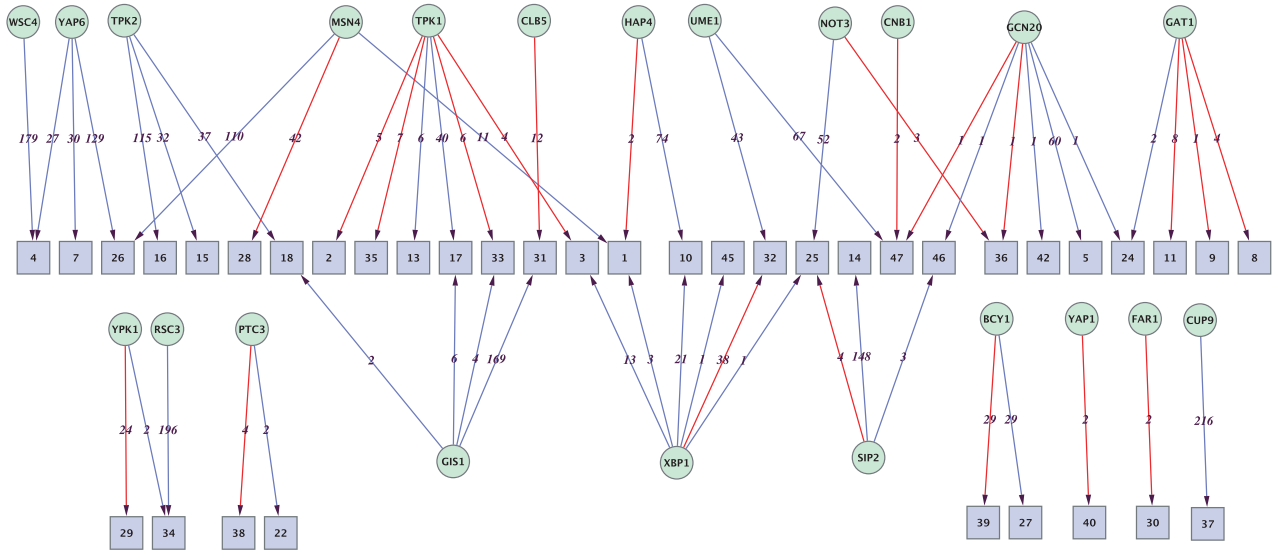

Fig. 5. Module network inferred by Segal *et al.* (2003) with edge-ranks computed by the ensemble method described in the current article. Red edges mean the module is overrepresented in known targets of the connected regulator.

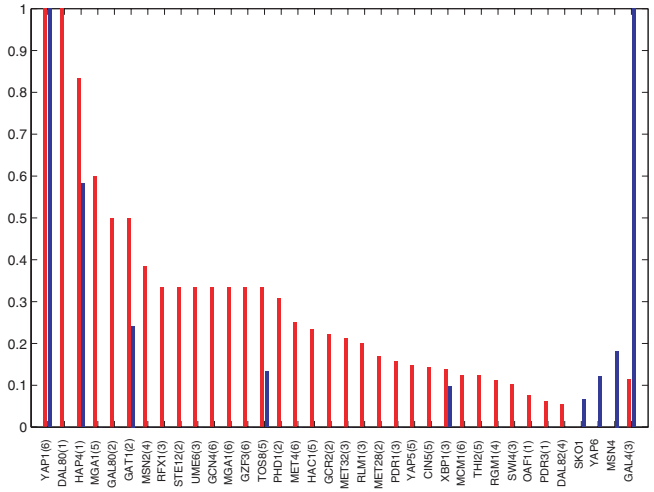

Fig. 6. Histogram of the highest fraction of known targets of transcription factors in a module using TCPR (red) and SCSR (blue) according to the YEASTRACT database. The rank with which a regulator is assigned to a module in TCPR is indicated in brackets.

we found that a rank cutoff of six gives the best overall F-measure score at different P-value cutoffs (see Supplementary Material). A similar analysis for SCSR shows that the F-measure for TCPR is consistently higher (see Supplementary Material). To compare TCPR and SCSR in more detail, we identified for each regulator the cluster with the highest fraction of known targets in YEASTRACT. Likewise we find the best cluster for each regulator in SCSR. Figure 6 shows that TCPR assigns more regulators supported by YEASTRACT and also that the clusters contain a higher ratio of known targets. There are six regulators assigned by both methods, four of which HAP1, GAT1, TOS8 and XBP1 all are assigned to clusters more enriched in their known targets in the TCPR solution.

2.5 Context-specific and combinatorial regulation

Segal *et al.* (2003) used a decision tree approach to model regulatory programs because it can represent, at least in principle, context-specific and combinatorial regulation. In the ensemble language, context-specificity means a regulator gets a high overall score by being assigned consistently to a lower, non-root level in the set of decision trees for a certain module. In SCPR, 59 regulator assignments divided over 39 (out of 50) modules have a significant score contribution (value >100) from a non-root level (see Section 4 for the decomposition of the score function over different tree levels). Combinatorial regulation means two (or more) regulators are consistently assigned together at different levels in all decision trees. Although this form of combinatorial regulation may correspond to genuine biological combinatorial regulation, we take a strictly data driven definition here: combinatorial regulation in the decision tree sense means the expression levels of both regulators are needed *together* to explain the expression level of the module ('AND' regulation). Alternatively, two (or more) high-scoring regulators may achieve their high rank from the same decision tree level (usually the root level). In this case both regulators explain the module equally well *alone* ('OR' regulation). In SCPR, there are a total of 100 regulator assignments with significant score contribution from the root level (OR regulation), which can be combined with the 59 assignments at level 1 for potential AND combinatorial regulation.

Only few of the significant AND combinatorial regulation pairs are present in the single-optimum solution of SCSR (see edge ranks in Fig. 5). SC-47 has Gcn20 as the highest ranked regulator at level 0 and Cnb1 at level 1, and both assignments are supported by literature. SC-36 has two validated regulators Gcn20 and Not3 ranked first and third, respectively in SCPR, but the score of Not3 is low and not deemed significant. SC-4 is an example of OR regulation wrongly assigned in SCSR. In SCSR, Ypl230w is assigned at level 0 and Gac1 at level 1, but in SCPR both are assigned at level 0 with first and third rank, respectively, and no high-scoring regulator is found at level 1. Some of the AND combinatorial regulation pairs in SCPR that were missed in SCSR can be validated by YEASTRACT. SC-40 has Tos8 assigned at level 0 (overall rank 1) and Yap1 at level 1 (overall rank 2). Tos8 has 3/15 known targets in this module, while Yap1 has all known targets (15/15). SC-26 has Gac1 at root level (overall rank 1) and Mal13 at level 1 (overall rank 2). Mal13 has two known targets (out of six known) in SC-26.

Due to the high number of possible regulator combinations, identifying statistically significant regulation of AND-type is an even more complex problem than simple regulator assignment. These examples show that also for this problem, the ensemble approach is well suited.

2.6 Module network in mouse brain

Recently, Li *et al.* (2007) reconstructed a bHLH transcription factor regulatory network in mouse brain by a direct application of the method of Segal *et al.* (2003). They selected a small dataset of 198 genes and 22 conditions, built a module network using 22 bHLH transcription factors as candidate regulators and assigned 15 different regulators to 28 modules (denoted again by SC), out of which 12 (43%) have at least two genes in the same GO category. Based on the co-occurrence of regulators in the regulation programs of individual modules, Li *et al.* (2007) make hypotheses about different modes of coregulation among brain tissues which are currently not confirmed by other data sources. We applied the ensemble method on this dataset and got 17 TCs, out of which 11 (65%) have at least two genes in the same GO category.

Only 11/28 SC have a high-scoring regulator with a significant score contribution from a non-root level, compared with 39/50 for yeast. Li *et al.* (2007) use the co-occurrence of Neurod6 and Hey2 in the SR regulation programs of SC-10, 15 and 27 to predict a cross-repression between Neurod6 and Hey2 with different modes of coregulation in different brain tissues. In the PR programs, Hey2 is the highest ranked regulator for SC-10, consistently assigned to the root level. However, at level 1, there are three equally good regulators Hes5 (overall rank 4), Neurod6 (overall rank 5) and Npas4 (overall rank 2). For SC-15, Neurod6 is the highest ranked regulator, consistently assigned to the root level, but the assignment of Hey2 at level 1 has a very low score (overall rank 4). For SC-27, we find consistent assignments of Hey2 at root level with overall rank 1 and Neurod6 at level 1 with overall rank 2. Thus the cross-repression mechanism predicted by Li *et al.* (2007) is supported only in the case of SC-27 and not SC-10 and 15. This example underscores the usefulness of an ensemble method to assess confidence levels of predicted interactions, especially in cases with limited amount of expression data and no other validation sources available.

3 CONCLUSIONS

We have re-examined the module network method of Segal *et al.* (2003) and compared an ensemble-based strategy with the standard direct optimization-based strategy. Ensemble averaging selects a subset of most informative genes and builds a quantitatively better model for them. It finds functionally more coherent tight gene clusters and is able to determine the statistically most significant regulator assignments. The difficult problem of identifying multiple regulators which explain together, but not separately, the expression of a module can be addressed in a reliable way. The ensemble method is thus able to deliver the promise to infer context-specific and combinatorial regulation through the probabilistic module network model.

4 METHODS

4.1 Bayesian two-way clustering

We associate to each gene i a continuous valued random variable X_i measuring the gene's expression level. For a data matrix $\mathcal{D}=(x_{im})$ with expression values for N genes in M conditions, the module network model of Segal *et al.* (2003) gives rise to a probabilistic model for two-way clusters, where a two-way cluster k is defined as a subset of genes $\mathcal{A}_k \subset \{1,\ldots,N\}$ with a partition $\mathcal{E}_k$ of the set $\{1,\ldots,M\}$ into condition clusters. The Bayesian posterior probability for a set of coclusters $(\mathcal{A}_k,\mathcal{E}_k)$, denoted $\mathcal{C}$, is given by

$$P_{\text{post}}(\mathcal{C}) \propto \prod_k \prod_{E\in\mathcal{E}_k} \iint \mathrm{d}\mu \mathrm{d}\tau \; p(\mu,\tau) \prod_{i\in\mathcal{A}_k} \prod_{m\in E} p(x_{i,m}\,|\,\mu,\tau),$$

where $p(x\,|\,\mu,\tau)$ is a normal distribution with mean μ and precision τ and $p(\mu,\tau)$ is a normal-gamma distribution [see Segal *et al.* (2005) or Joshi *et al.* (2008) for more details]. We use the Gibbs sampler strategy developed in Joshi *et al.* (2008) to sample multiple high-scoring coclusterings from this posterior distribution. From these multiple solutions we extract tight gene clusters using the procedure outlined in Joshi *et al.* (2008). It consists of a graph spectral method extracting densely connected regions from the graph on the set of genes with edge-weights p_{ij}, the frequency that gene i and j belong to the same cocluster in each of the sampled solutions.

4.2 Probabilistic regulatory programs

For each set of conditions E in the condition partition $\mathcal{E}_k$ for a given module k, we have an associated normal distribution with parameters (μ_E,τ_E) which can be estimated from the posterior distribution. Hence such a condition set can be interpreted as a discrete expression state for the module. A regulatory program 'predicts' the expression state of any condition in terms of the expression levels of a small set $\mathcal{R}_k$ of regulators, i.e. there is a conditional distribution

$$p\big(x_i\,|\,\{x_r, r\in\mathcal{R}_k\}\big) = p(x_i\,|\,\mu_E,\tau_E).$$

The selection of an expression state is done by constructing a decision tree with the states $E\in\mathcal{E}_k$ at the leaves. To each internal node t, we associate a regulator r_t and split value z_t. In Segal *et al.* (2003), the decision at the node is based on the test $x_{r_t}\geq z_t$ or $x_{r_t}<z_t$. Here, we extend this model to allow *fuzzy* decision trees. More precisely, we sort the expression states $E\in\mathcal{E}_k$ by their mean μ_E, and link this ordered set hierarchically. Then we can associate to each internal node a binary variable $y_t=\pm 1$, where $y_t=-1$ means 'decrease expression state' (go 'left' in decision tree) and $y_t=+1$ means 'increase expression state' (go 'right' in decision tree). Again we also associate a regulator r_t and split value z_t to node t, and a conditional probability

$$p(y_t\,|\,x_{r_t},z_t,\beta_t) = \frac{1}{1+e^{-\beta_t y_t (x_{r_t}-z_t)}}. \tag{1}$$

Given expression values x_r for all $r\in\mathcal{R}_k$, we traverse the decision tree in a probabilistic fashion, taking the decision $y_t=\pm 1$ at each node t by tossing a biased coin with bias Equation (1). The original model with *hard* decision trees is recovered if $\beta_t=\pm\infty$ for each node.

The conditional distribution or regulatory program now becomes a normal mixture distribution

$$p\big(x_i\,|\,\{x_r, r\in\mathcal{R}_k\}\big) = \sum_{E\in\mathcal{E}_k} \alpha_E\big(\{x_r, r\in\mathcal{R}_k\}\big) p(x_i\,|\,\mu_E,\tau_E) \tag{2}$$

where

$$\alpha_E\big(\{x_r, r\in\mathcal{R}_k\}\big) = \prod_t p(y_t\,|\,x_{r_t},z_t,\beta_t)$$

with the values y_t determined by the unique path through the decision tree that ends at leaf E.

For a cocluster $(\mathcal{A}_k,\mathcal{E}_k)$ inferred from a dataset $\mathcal{D}=(x_{im})$ by the method summarized in the previous section, we can derive a posterior probability function for each regulator at each node t as follows. First note that each condition m belongs to exactly one set E in $\mathcal{A}_k$ and hence determines a unique path through the decision tree, or in other words a set of values $y_{t,m}$ at each node t. Furthermore, each node t has an associated condition set E_t consisting of the union of all condition sets E which can be reached from node t. Hence, we can define at each node a posterior probability by

$$P_{\text{post}}(r,z) \propto \max_\beta \Big(\prod_{m\in E_t} p(y_{t,m}\,|\,x_{r,m},z,\beta) \Big), \tag{3}$$

where for computational simplicity we maximize over β instead of marginalizing over a prior distribution. By allowing only a discrete set of split values, Equation (3) becomes a discrete distribution from which it is easy to sample. Typically, we consider as possible split values z the expression values $x_{r,m}$ for $m\in E_t$, but simpler schemes such as only allowing one or two split values can be used to reduce computation time for large datasets.

The posterior probability Equation (3) measures how well the expression values of a regulator 'predict' the partition into two sets of E_t induced by the condition partition $\mathcal{E}_k$. We define the average prediction probability of (r,z) at node t by the geometric average

$$p_t(r,z) = \Big(\prod_{m\in E_t} p(y_{t,m}\,|\,x_{r,m},z,\beta_{\max}) \Big)^{1/|E_t|}, \tag{4}$$

where $\beta_{\max}$ is the maximizer in Equation (3).

4.3 Regulator assignment score

To assess the significance $Z_t(r)$ for assigning a regulator r to a node t in a certain regulation program, we use the average prediction probabilities [Equation (4)] and define:

$$Z_t(r) = w_t \sum_z p_t(r,z). \tag{5}$$

A typical choice for the weight factor w_t is $w_t=(|E_t|/M)$, expressing that we have more confidence in assignments to nodes supported on more conditions. The sum $\sum_z$ runs over the discrete set of split values for regulator r at node t. The overall significance $Z(r)$ for assigning a regulator r to a module is defined by summing Equation (5) over all nodes of all regulation programs for that module:

$$Z(r) = \sum_{T\in\mathcal{T}} \sum_{t\in T} Z_t(r).$$

4.4 Model evaluation

For an experiment with expression levels $(x_1,\ldots,x_N)$, we can evaluate the probability distribution

$$p(x_1,\ldots,x_N) = \prod_{i=1}^{N} p\big(x_i\,|\,\{x_r, r\in\mathcal{R}_{k(i)}\}\big),$$

with $k(i)$ the module to which gene i belongs and $\mathcal{R}_k$ the regulator set of module k, using the conditional distributions (2). We only consider genes for

which the model makes actual predictions, i.e. genes belonging to clusters with a regulation tree. For the cross-validation experiment, we removed 10% of the conditions randomly from the total of 173 conditions. We learned module networks on the remaining 90% data and repeated this procedure 10 times.

4.5 Datasets

Yeast expression data for 2355 differentially expressed genes in 173 stress conditions, gene clusters, their regulators, split values and regression trees were downloaded from the supplementary website of Segal *et al.* (2003) at http://robotics.stanford.edu/~erans/module_nets/. MIPS functional catagories were downloaded from ftp://ftpmips.gsf.de/catalogue/annotation_data. For TC and SC we calculated the P-value whether the overlap between a given cluster and a given functional catagory is statistically significant. We used data on genome-wide binding and phylogenetically conserved motifs for 102 transcription factors from Harbison *et al.* (2004). For a given transcription factor, only genes that were bound with high confidence (significance level $\alpha = 0.005$) and showed motif conservation in at least one other *Saccharomyces* species (besides *S.cerevisiae*) were considered true targets. We also downloaded all known regulator target interactions from the YEASTRACT database http://www.yeastract.com. We calculated the P-value whether the overlap between a given cluster and a given transcription factor target set is statistically significant.

Mouse expression data by Su *et al.* (2004) was downloaded from http://wombat.gnf.org and the data selection and normalization was done as described in Li *et al.* (2007).

ACKNOWLEDGEMENTS

We would like to acknowledge Eric Bonnet for useful discussions.

Funding: Early-Stage Marie Curie Fellowship (to A.J.); IWT (SBO-BioFrame); IUAP P6/25 (BioMaGNet).

Conflict of Interest: none declared.

REFERENCES

Alter,O. and Golub,G.H. (2005) Reconstructing the pathways of a cellular system from genome-scale signals by using matrix and tensor computations. *Proc. Natl Acad. Sci. USA*, **102**, 17559–17564.

Bansal,M. *et al*. (2007) How to infer gene networks from expression profiles. *Mol. Syst. Biol.*, **3**, 78.

Basso,K. *et al*. (2005) Reverse engineering of regulatory networks in human b cells. *Nat. Genet.*, **37**, 382–390.

Bonneau,R. *et al*. (2006) The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets *de novo*. *Genome Biol.*, **7**, R36.

Bussemaker,H.J. *et al*. (2007) Predictive modeling of genome-wide mRNA expression: from modules to molecules. *Annu. Rev. Biophys. Biomol. Struct.*, **36**, 329–347.

Carvalho,L.E. and Lawrence,C.E. (2007) Centroid estimation in discrete high-dimensional spaces with applications in biology. *Proc. Natl Acad. Sci. USA*, **105**, 3209–3214.

Chen,G. *et al*. (2007) Clustering of genes into regulons using integrated modeling-COGRIM. *Genome Biol.*, **8**, R4.

Faith,J.J. *et al*. (2007) Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol.*, **5**, e8.

Friedman,N. (2004) Inferring cellular networks using probabilistic graphical models. *Science*, **308**, 799–805.

Gardner,T.S. and Faith,J.J. (2005) Reverse-engineering transcription control networks. *Phys. Life Rev.*, **2**, 65–88.

Harbison,C.T. *et al*. (2004) Transcriptional regulatory code of a eukaryotic genome. *Nature*, **431**, 99–104.

Hartwell,L.H. *et al*. (1999) From molecular to modular cell biology. *Nature*, **402**, C47–C52.

Joshi,A. *et al*. (2008) Analysis of a Gibbs sampler for model based clustering of gene expression data. *Bioinformatics*, **24**, 176–183.

Lee,S. *et al*. (2006) Identifying regulatory mechanisms using individual variation reveals key role for chromatin modification. *Proc. Natl. Acad. Sci. USA*, **103**, 14062–14067.

Li,J. *et al*. (2007) Regulatory module network of basic/helix-loop-helix transcription factors in mouse brain. *Genome Biol.*, **8**, R244.

Liu,J.S. (2004) *Monte Carlo Strategies in Scientific Computing*. Springer, New York.

Novershtern,N. *et al*. (2008) A functional and regulatory map of asthma. *Am. J. Respir. Cell Mol. Biol.*, **38**, 324–336.

Segal,E. *et al*. (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.*, **34**, 166–167.

Segal,E. *et al*. (2005) Learning module networks. *J. Mach. Learn. Res.*, **6**, 557–588.

Segal,E. *et al*. (2007) Decoding global gene expression programs in liver cancer by noninvasive imaging. *Nat. Biotech.*, **25**, 675–680.

Su,A. *et al*. (2004) A gene atlas of the mouse and human protein-encoding transcriptomes. *Proc. Natl. Acad. Sci. USA*, **101**, 6062–6067.

Teixeira,M. *et al*. (2006) The YEASTRACT database: a tool for the analysis of transcription regulatory associations in Saccharomyces cerevisiae. *Nucleic Acids Res.*, **34**, D446–D451.

Webb-Robertson,B.-J.M. *et al*. (2008) Measuring global credibility with application to local sequence alignment. *PLoS Comput. Biol.*, **4**, e1000077.

Zhu,H. *et al*. (2007) Combined microarray analysis uncovers self-renewal related signaling in mouse embryonic stem cells. *Syst. Synth. Biol.*, **1**, 171–181.