BIOINFORMATICS

Vol. 21 Suppl. 2 2005, pages ii115–ii122
doi:10.1093/bioinformatics/bti1120

Microarrays

Identifying active transcription factors and kinases from expression data using pathway queries

Florian Sohler* and Ralf Zimmer

Department of Informatics, Ludwig-Maximilians-Universität, Amalienstraße 17, 80333 München, Germany

ABSTRACT

Motivation: Although progress has been made identifying regulatory relationships from expression data in general, only few methods have focused on detecting biological mechanisms like active pathways using a single measurement. This is of particular importance when only few measurements are available, e.g. if special cell types or conditions are under investigation. Here we present a method to test user specified hypotheses (pathway queries) on expression data where prior knowledge is given in the form of networks and functional annotations. Based on this method, we develop a scoring function to identify active transcription factors or kinases, thus making a first step toward explaining the measured expression data.

Results: We apply the algorithm to the Rosetta Yeast Compendium dataset, finding that in many cases the results are in concordance with biological knowledge. We were able to confirm that transcription factors and to a lesser degree, kinases identified by our method play an important role in the biological processes affected by the respective knockouts. Furthermore, we show that correlation of inferred activities can provide evidence for a physical interaction or cooperation of transcription factors where correlation of plain expression data fails to do so.

Contact: florian.sohler@bio.ifi.lmu.de

1 INTRODUCTION

As measurement of gene expression using microarrays has become a standard high-throughput method in molecular biology, the analysis of gene-expression data is still a very active area of research in bioinformatics and statistics. There is a wide range of goals addressed by the various methods.

Many analysis methods for gene-expression data are mainly descriptive. Such analyses typically result in a list of genes that exhibit a relevant expression behavior in the experiment under consideration. These lists are often analyzed for common functions to find out which biological processes could be relevant. For instance, Mootha *et al*. (2003) have developed a method that identifies gene sets that are significantly enriched in regulated genes from a predefined list of gene groups. Although these are important steps in understanding the data, they do not answer the question what causes the observed gene expression.

Network inference methods try to identify regulatory relationships as a model for the observed expression data (Friedman, 2004), but result in very general networks that could be inappropriate for higher organisms, such as humans, since regulation of gene expression varies strongly in different cell types and under different conditions. In order to overcome that problem, Segal *et al*. (2003) have proposed a method for regulatory network inference that results in a set of modules which are active only under certain conditions. Furthermore, they do not rely on expression data alone, but make use of sequence data which allows them to incorporate transcription factor binding sites into their model. Other algorithms use prior knowledge in the form of networks to find biologically relevant results: Yeang and Jaakola (2003) suggest a method to extract physical pathways from a database of protein–protein interactions that could best explain expression measurements from yeast knockout strains. Steffen *et al*. (2002) compute signaling pathways using expression data and protein interactions from two-hybrid screens.

All of these methods try to infer general rules and, therefore, work best on large sets of expression data. If only few measurements are available, it is difficult to attain such general rules. Furthermore, biological researchers are often concerned with rather special problems, where they would like to know what is going on with some special cell type in a certain condition and have only a few microarray measurements to investigate this problem. However, the same biologists know a lot about their research area. Using this knowledge, it should be possible to put the measured expression data in a context that facilitates their interpretation. 'Pathway queries' provide a very flexible means of creating such contexts allowing the user to specify hypotheses in the form of network templates. With these templates, a background network containing interactions and annotations for genes and proteins is searched, and instances of the templates are generated which can be scored according to measured expression data and additional criteria.

For instance, for pharmaceutical applications it is important to compare disease models with healthy tissue which can be done using microarray gene expression measurements. To analyze the resulting data, first of all, it is necessary to find genes that are differentially expressed between the healthy and diseased states. However, a more interesting question for the development of disease modifying drugs is finding the molecular mechanism underlying the observed changes in gene expression. Unfortunately, the mechanism itself does not have to be reflected by the changes of gene expression; gene regulation is often influenced by molecular events other than transcription, such as protein modification (phosphorylation, cleavage), translocation, DNA methylation, etc. Nevertheless, if a hypothesis about the relevant mechanism is available, it can be tested on the basis of expression data and prior knowledge in the form of a network model. Such a hypothesis could be that an active kinase X phosphorylates one (or more) transcription factors which causes the observed differences in the expression profiles. This

*To whom correspondence should be addressed.

© The Author 2005. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oxfordjournals.org

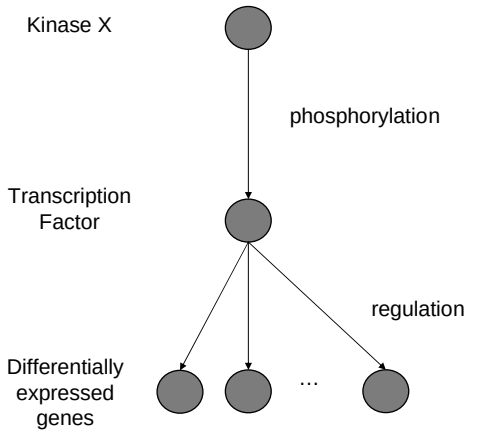

Fig. 1. Example of a simple network template that can be represented by a pathway query.

hypothesis can be visualized as a small network template as shown in Figure 1.

The pathway query language allows specifying such hypotheses in an XML format. In order to test the hypothesis, the pathway search algorithm finds all instances of a query in a given network. If no subnetwork matching the query is found, the hypothesis should be rejected on the basis of the underlying network and annotation data. Otherwise, the different instances can be scored and sorted by significance before they can be visualized and inspected manually.

In this work, we describe the pathway query language and an appropriate pathway search algorithm to enumerate all instances of a network template for a given background network and expression data. We develop suitable pathway queries and scoring schemes that allow us to find relevant kinases and transcription factors in single microarray measurements. The pathway queries are applied to a set of yeast expression data and a background network containing protein–protein and protein–DNA interactions.

2 METHODS

2.1 Pathway queries

Pathway queries were first introduced at the German Conference on Bioinformatics 2003 using a small case study (Sohler *et al.*, 2004). Here, we give the first detailed description of the algorithm and the new scoring methods developed to identify important transcription factors and kinases.

The pathway query language is an XML query language for biological networks and annotations. The language was designed to help the biologist interpret gene expression and other experimental data by allowing formulating and testing biological hypotheses on a molecular level. Hypotheses are formalized as network templates (the pathway query); the pathway search algorithm then returns a list of all concrete instances in a given biological network (the search graph) that match the template.

2.2 Description of the query language

The search graph is represented as bipartite graph $G = (P, T, E)$ with places P representing genes or proteins or other molecules and transitions T specifying interactions between the molecules. The edges E are used to define which molecules participate in an interaction and (if appropriate) the direction of the interaction.

The constructs used in the pathway query language to restrict sets of places or interactions are basic queries which can represent constraints on all available annotations. These constraints can be combined using boolean operators (and, not, or). For instance, a basic query can place constraints on the function and the expression value of a gene:

(GO Molecular Function *like* kinase) and (Fold Change > 2).

In this example, all kinases that are upregulated at least 2-fold would be selected. Formally, a basic query q encodes a boolean function $\xi(q, .)$ which evaluates to true if the argument is a node in the search graph that satisfies the conditions specified by q.

A pathway query describes a network template as a labeled graph $Q = (N, C)$. Nodes $n \in N$ represent single places or sets of places and edges $c \in C$ represent paths between places. Nodes are labeled by constraints using basic queries as described above. The basic query associated with a query node n is denoted by $\nu(n)$.

There is an additional 'multiplicity' attribute $\mu(n)$ for each query node. If $\mu(n) = 0$ the resulting instances will have exactly one place taking the role of n. Otherwise, one instance will contain all places that are possible in that role, effectively merging all instances that are equal except for the instantiation of n (Fig. 2).

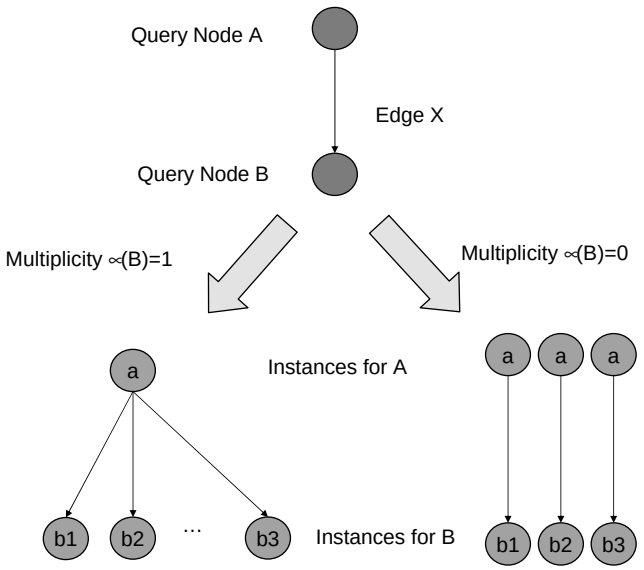

Fig. 2. The multiplicity attribute in pathway queries. Setting the attribute to 1 effectively merges all instances that differ only with respect to that query node.

Connections between two nodes n_1 and n_2 with $\mu(n_1) = 1$ and $\mu(n_2) = 1$ are not allowed, since the meaning of such a construct would be difficult to define and unintuitive.

An edge $c \in C$ in Q corresponds to paths in G. These paths can be restricted in three different aspects. The places on that path must match a basic query denoted by $\gamma_P(c)$ associated with c, the transitions must match another basic query called $\gamma_T(c)$ and the length of the path is limited by $\eta(c)$. Note that there can be more than one edge between two nodes in the query. This is important if we look for instances that have more than one path between two proteins, e.g. different signaling cascades leading from one extracellular input to the same transcription factor.

2.3 The pathway search algorithm

The search algorithm is basically an algorithm for the subgraph-isomorphism (SI) problem on labeled graphs. This problem is known to be NP-complete, and therefore no polynomial algorithm is known (Garey and Johnson, 1979). SI can be reduced elegantly to a clique search problem using a compatibility graph (also called association graph or product graph). The pathway search algorithm also uses this approach. The main difference to the classical

```
Graph buildInstanceGraph(Query q, Graph searchGraph) {
  for (Node n : q.nodes()) {
    instanceNodes=n.findInstanceNodes(searchGraph);
    instanceGraph.addNodes(instanceNodes);
  }
  for (Connection c : q.connections()) {
    paths=c.findPaths(instanceGraph, searchGraph);
    instanceGraph.addEdges(paths);
  }
}

List findPathways(Query q, Graph searchGraph) {
  inst = buildInstanceGraph(q, searchGraph);
  return cliqueSearch(inst);
}
```

Fig. 3. Overview of the pathway search algorithm. Given a search graph and a pathway query it returns all subgraphs (instances) that match the query.

SI problem is that pathway queries are not simply subgraphs of the search graph:

- Edges in the query can correspond to (restricted) paths in the search graph.
- Nodes in the query can correspond to more than one place in the search graph (if the 'multiplicity' is 1).

The pathway search algorithm is summarized in Figure 3. First, we build the instance graph which corresponds to the compatibility graph in the classical algorithm. A node in the instance graph corresponds to a place in the search graph taking the role defined by a query node (such a role could, for example, be the transcription factor in Fig. 1). An edge is added between two nodes if the corresponding places can be in their respective roles in one instance. For instance, if there is a protein in the role of the kinase and another protein in the role of the transcription factor, the kinase must be known to phosphorylate the transcription factor if that is required by the query.

A clique in the instance graph that contains nodes for all query nodes is then a valid pathway instance.

The construction of the instance graph $\Pi = (\hat{N}, \hat{E})$ can be formalized as follows: Π contains a node $\hat{n} = (n, p)$ if and only if

$$\xi(\nu(n), p) = 1, \tag{1}$$

i.e. a place p can take the role of query node n if it satisfies the conditions made by the basic query $\nu(n)$.

Let R_Q be a function that recovers the corresponding query node from a node in the instance graph and R_G a function that recovers the corresponding place in the search graph. Then there is an edge $\hat{e}$ between two nodes $\hat{n}_1$ and $\hat{n}_2$ if

$$\forall c \in C, c = (R_Q(\hat{n}_1), R_Q(\hat{n}_2)) \exists (\psi_1 \cdots \psi_l), l < 2\eta(c) + 1,$$

satisfying the following conditions:

(1) $\psi_1 = R_G(\hat{n}_1)$

(2) $\psi_l = R_G(\hat{n}_2)$

(3) $\psi_i \in T, \xi(\gamma_T(c), \psi_i) = 1, \quad i = 2j, \ 0 < j < (l-1)/2$

(4) $\psi_i \in P, \xi(\gamma_P(c), \psi_i) = 1, \quad i = 2j+1, \ 0 < j < (l-3)/2.$

That is, if for all connections required by the query, there is a path in the search graph compatible with the specified restrictions. To keep the product graph small, we do not explicitly add an edge $(\hat{n}_1, \hat{n}_2)$ if there is no edge in the pathway query between $R_Q(\hat{n}_1)$ and $R_Q(\hat{n}_2)$. Nevertheless, $\hat{n}_1$ and $\hat{n}_2$ are compatible; therefore, we define a function θ to represent the compatibility of two nodes in the product graph:

$$\theta(\hat{n}_1, \hat{n}_2) = \begin{cases} 1 & (\hat{n}_1, \hat{n}_2) \in \hat{E} \\ 1 & (R_Q(\hat{n}_1), R_Q(\hat{n}_2)) \notin C \\ 1 & R_Q(\hat{n}_1) = R_Q(\hat{n}_2) \text{ and } \mu(R_Q(\hat{n}_1)) = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Using the compatibility function instead of explicit edges can save a lot of memory when queries have many unspecific nodes and only few edges. For instance, a linear query with 10 nodes, each of which can be instantiated with 100 proteins could result in up to 500 000 edges in the instance graph, most of which will be present as there are no conditions tied to them. Using the compatibility function, there can be no >50 000 and each one is subject to a condition.

Checking the connection conditions involves the largest computational cost. For all pairs of places that have to be connected according to the query, a path must be found containing only places and transitions satisfying the specified conditions. To do that, we build a subgraph $S = (P_S, T_S, E_S)$ of the search graph for each edge $c = (n_1, n_2)$ of the query graph with $P_S = \{p \in P : \xi(\gamma_P(c), p) = 1\}$ and $T_S = \{t \in T : \xi(\gamma_T(c), t) = 1\}$. Given the two corresponding sets of nodes in the instance graph $\hat{n}_1^{(i)} = (n_1, p_1^{(i)})$, $\hat{n}_2^{(i)} = (n_2, p_2^{(i)})$, we use a depth first search on S starting at each $p_1^{(i)}$ to find paths to places $p_2^{(j)}$. In order to avoid unnecessary path computations, we drop nodes from the instance graph as soon as they lack a required connection. Thus, we do not need to check other connections for that node and keep the graph as small as possible.

The clique search uses a rather simple backtracking algorithm. A candidate clique $\mathcal{C}$ and a set $\mathcal{S}$ of all nodes compatible with $\mathcal{C}$ are maintained during the search. At each step, $\mathcal{C}$ is expanded by a node n with $\mu(n) = 0$ and $\mathcal{S}$ is updated as $\mathcal{S}_{i+1} = \mathcal{S}_i \cap \{m : m \in \mathcal{S}_i, \theta(m, n) = 1\}$. When no node n with $\mu(n) = 0$ exists, all remaining nodes are added to the candidate clique. Note that all $\hat{n}$ with $\mu(R_Q(\hat{n})) = 1$ are always compatible with each other as there cannot be a connection in the query between two such nodes.

2.4 Scoring transcription factors and kinases

Transcriptional regulators like transcription factors, kinases and signaling molecules do not need to be regulated transcriptionally. Especially in higher organisms, other modes of regulation, e.g. by phosphorylation or translocation, and combinatorial interaction with other regulators, are more important. Thus, we cannot estimate the activity of a regulator by its expression level alone, but must include the expression levels of its potential regulatory targets.

In order to compute a score for the relevance of a transcription factor or any other regulator F, we need the number of potential target genes $t(F)$. We compute a P-value of the number of significantly regulated genes among the potential targets [denoted by $r(F)$] using Fisher's exact test. The activity score $s(F)$ is then defined as the log of that P-value:

$$s(F) = -\log_{10} \sum_{k=r(F)}^{\min\{R,t(F)\}} \left[\frac{\binom{R}{k}\binom{N-R}{t(F)-k}}{\binom{N}{t(F)}} \right], \tag{3}$$

where N is the total number of measured genes and R the number of genes that are significantly regulated. We will call $s(F)$ the activity score of F, although it rather represents a change of activity. The score will be high when a significantly large number of the regulator's potential targets is differentially expressed in the experiment under consideration.

Of course, it is crucial to have high quality sets of potential targets. To find these, we use pathway queries, but the knowledge has to be encoded in the network. Such networks can be built from databases like TRANSFAC (Wingender *et al.*, 2000); we have also used a combination of text mining and binding site predictions to compute potential targets (Gebauer *et al.*, 2005).

In this paper, we use DNA-binding data for yeast transcription factors from genome-wide location analysis.

The pathway queries that extract the necessary information from the network are particularly simple. For transcription factors, there are only two

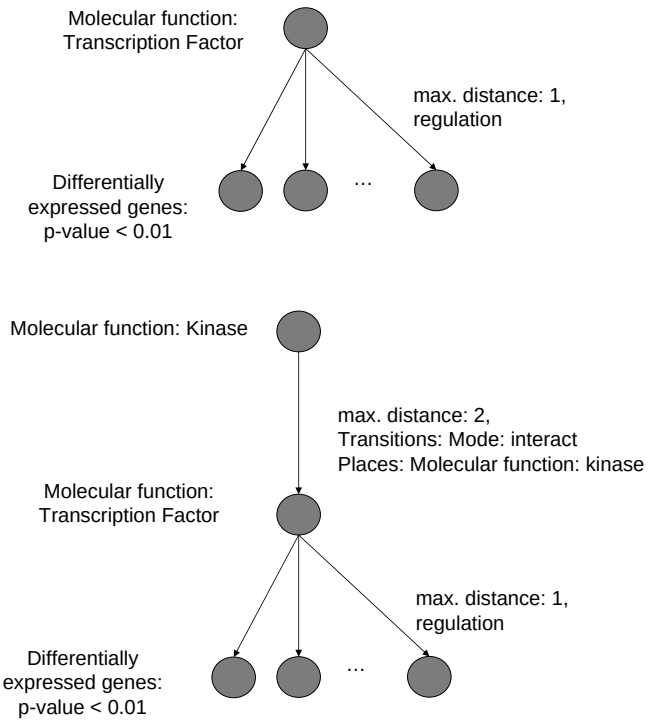

Fig. 4. Pathway queries used to find relevant transcription factors (top) and regulated kinases (bottom). With these simple network templates, regulated genes can be put into context with their regulating transcription factors and kinases.

query nodes, the first representing the transcription factor itself, the second representing the targets. The connection must not be longer than a single transition which should be labeled 'regulates'. We run the query once with no restriction on the targets to select all potential targets, and once for each expression measurement requiring the targets to have a P-value of <0.01 in the experiment under consideration.

For kinases, we use a pathway query that is a little more complicated. We need the first node to represent the kinase which should also be at least slightly regulated, so we require it to have a P-value of <0.1. It must be connected via at most one more kinase to a transcription factor. The transcription factor is again connected to regulated target genes. The two pathway queries are visualized in Figure 4.

There is one instance of the pathway query for kinases for each transcription factor that is connected to that kinase. Therefore, we have to merge all instances to find all potential targets.

3 DATA

To test our method we selected the Rosetta yeast compendium dataset (Hughes *et al*., 2000). It provides expression ratios as well as P-values for differential expression for 300 measurements, most of which are knockout mutations.

We use a combination of two different networks that contain our prior knowledge: the yeast subset of the DIP network of protein interactions (Xenarios *et al*., 2000) and a network containing DNA-binding information constructed from the genome-wide location analysis of Lee *et al*. (2002), adding an edge between a transcription factor and a target gene if the authors reported a P-value of <0.001 for that interaction.

Functional annotations are extracted from gene ontology (The Gene Ontology Consortium, 2001). In our examples, we need annotations only for kinases and transcription factors. The latter could have been extracted from the location data as well; we used gene ontology annotations for simplicity and consistency.

4 RESULTS

4.1 Activity of transcription factors

First, we computed the activity scores of every transcription factor for all 300 expression measurements. In order to visualize the results, we constructed hierarchical clusters using the Spearman rank correlation and average linkage (Fig. 5) on the activity scores for all transcription factors with at least one significant score. Table 1 lists the 25 best scoring results. For evaluation, we manually assess some prominent features of the results and look for evidence in the literature.

The highest value in the matrix is attained by the transcription factor Ste12p for the dig1/dig2 mutant. All targets of Ste12p are upregulated, as expected, since Dig1p and Dig2p are needed for the repression of pheromone-responsive transcription (Bardwell *et al*., 1998). Figure 6 shows Ste12p and its targets with expression data from the dig1/dig2 knockout experiment.

In addition, Ste12p is identified as the most relevant transcription factor for the kss1/fus3 knockout. These two MAP kinases are also known to regulate transcription of pheromone-response genes through Ste12p (Tedford *et al*., 1997).

Bas1p has a very high score in the hpt1 experiment. Again, it can easily be verified that hpt1 mutations affect the purine biosynthesis pathway which is regulated by Bas1p (Guetsova *et al*., 1997; Daignan-Fornier and Fink, 1992).

Arg80p and Arg81p are the transcription factors with the highest scores in the knockout experiment of arg80. Figure 7 shows these two transcription factors with their regulated targets in that experiment. Indeed, the transcription factor that was knocked out should be important for the regulation. Furthermore, it is known that Arg80p and Arg81p are necessary for the repression of anabolic genes in the argenine biosynthesis. The four targets (Arg5,6p, Arg3p, Arg8p and Cpa1p) which are all upregulated catalyze different steps in that metabolic pathway.

These results demonstrate that high scoring hits indeed deliver regulatory contexts that are important for the experiment under consideration.

4.2 Correlation analysis of activity scores

So far, all results have been obtained using only one experiment from the set of expression data at a time. In order to figure out if we can identify cooperating transcription factors, we look at pairs of transcription factors that correlate well in their activity scores. Table 2 shows all pairs of transcription factors with a Spearman rank correlation >0.6 which corresponds to a Z-score of 4.6. Figure 8 shows a histogram of all correlations.

Arg80p and Arg81p have the highest correlation; their role was already discussed. Next, we take a closer look at the pair Mcm1p and Ste12p. As with Arg80p and Arg81p, these transcription factors interact according to DIP. The experiments where they are active include knockouts of ste4, ste5, ste7, ste11, ste12, ste18, ste24, fus3/kss1 and dig1/dig2.

Indeed, as these results suggest, mating functions like pheromone maturation, pheromone response and cell fusion are cooperatively

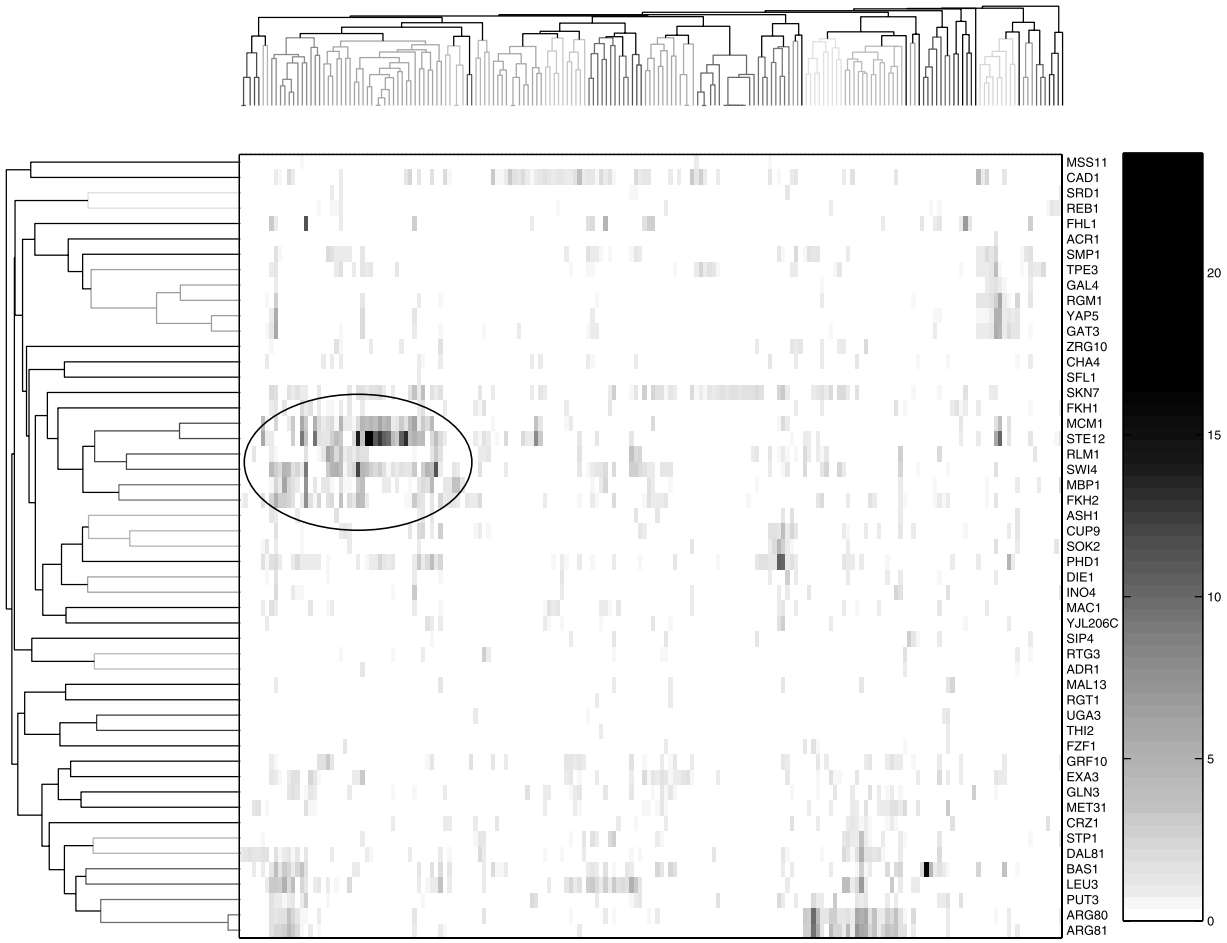

Fig. 5. Inferred activities of 51 transcription factors. Average linkage hierarchical clustering was performed with Spearman rank correlation on 300 expression measurements. Values are—$\log_{10} p$, where p is a P-value quantifying the significance of a transcription factor for the expression data. The marked area contains mainly experiments that affect the MAPK signaling pathway (e.g. knockouts of ste4, ste5, ste7, ste12, ste18, ste24, fus3/kss1 and dig1/dig2).

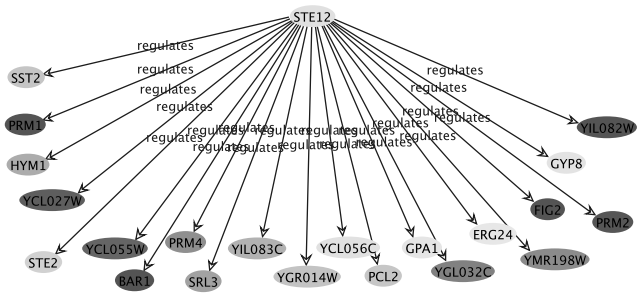

Fig. 6. Ste12p and its regulated targets in the dig1/dig2 knockout strain. The shade of the nodes represents the magnitude of the expression ratio (mutant versus wild-type). All genes are upregulated.

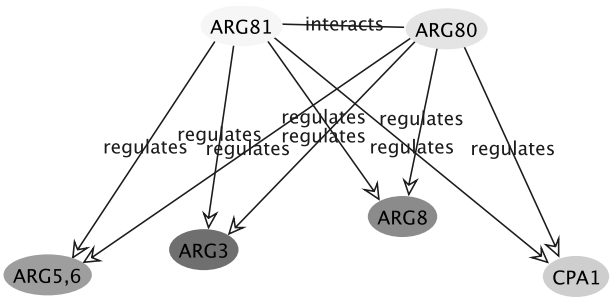

Fig. 7. Arg80p and Arg81p and their regulated targets in the arg80 knockout strain. The shade of the nodes represents the magnitude of the expression ratio (mutant versus wild-type). All genes except arg80 are upregulated. Arg80p and Arg81p are the two most relevant transcription factors for the arg80 mutant.

controlled by the transcription factors Ste12p and Mcm1p (Hwang-Shum *et al*., 1991; Dolan *et al*., 1989). Interestingly, this cooperative behavior could not have been identified using only the expression data of STE12 and MCM1, as these do not correlate. Also, the potential targets of Ste12p and Mcm1p do not overlap to a large extent (88 potential targets for Mcm1p, 51 for Ste12p and 8 overlapping). Only the combination of potential targets and their expression data creates the evidence that leads to the correct conclusion (Fig. 9).

For the other pairs we did not find additional literature evidence for a functional relationship, but we believe that this is worth investigating.

Table 1. The 25 top scoring transcription factors together with their activity scores as described in Section 2.4

Experiment	Transcription factor	Score
dig1,dig2(haploid)	STE12	23.7
fus3,kss1(haploid)	STE12	17.4
hpt1	BAS1	15.7
dig1,dig2	STE12	14.0
sst2(haploid)	STE12	13.3
ste18(haploid)	STE12	12.9
dig1,dig2	SWI4	12.1
ERG11(tetpromoter)	FHL1	11.3
kin3	SWI4	11.2
ste12(haploid)	STE12	10.9
ste7(haploid)	STE12	10.6
ssn6(haploid)	PHD1	10.5
ste4(haploid)	STE12	10.4
tup1(haploid)	PHD1	10.3
FR901,228	STE12	10.1
ste5(haploid)	STE12	9.9
arg80	ARG80	9.8
ste24(haploid)	STE12	9.6
arg80	ARG81	9.5
ERG11(tetpromoter)	SWI4	9.2
sod1(haploid)	STE12	9.0
ste11(haploid)	STE12	7.9
ERG11(tetpromoter)	FKH2	7.7
ERG11(tetpromoter)	MBP1	7.2
pep12	ARG81	7.2

The table displays the most relevant transcription factors for individual experiments, e.g. Phd1p was identified for the ssn6 knockout experiment.

Table 2. The six most highly correlated pairs of transcription factors

Transcription factor 1	Transcription factor 2	Spearman correlation
ARG80	ARG81	0.85
YAP5	GAT3	0.79
RGM1	GAT3	0.70
RGM1	GAL4	0.67
MCM1	STE12	0.66
RGM1	YAP5	0.62

Correlation was computed using Spearman rank correlation on the activity scores of the transcription factors. Interestingly, there are only few pairs with high correlation (see Fig. 8), although related transcription factors often cluster together nicely (Fig. 5 left).

4.3 Activity of kinases

Using the same scoring method and the pathway query described in Figure 4, we computed activity scores for kinases. Our algorithm detects much fewer high scoring kinases than transcription factors. The 10 highest scores are listed in Table 3.

The highest score is attained for the dig1/dig2 mutant and the MAP kinase Kss1p. As was already mentioned, the Kss1p regulates transcription through Ste12p.

The network that constitutes the second best score contains the kinase Slt2p and its regulated targets in the kin3 knockout experiment

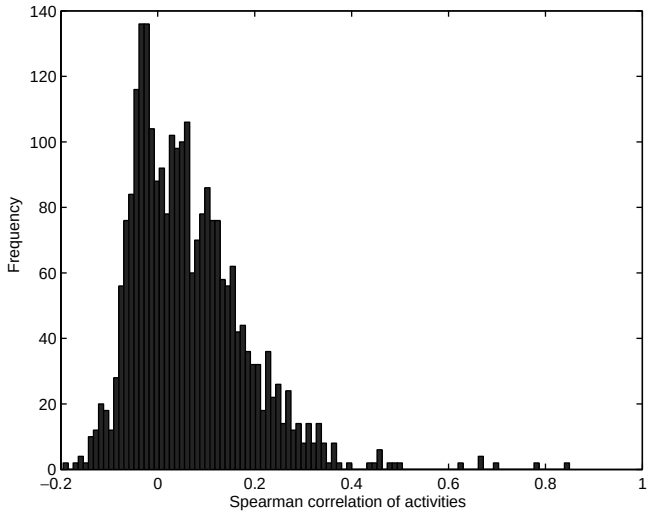

Fig. 8. Histogram of the correlations between the activities of transcription factors.

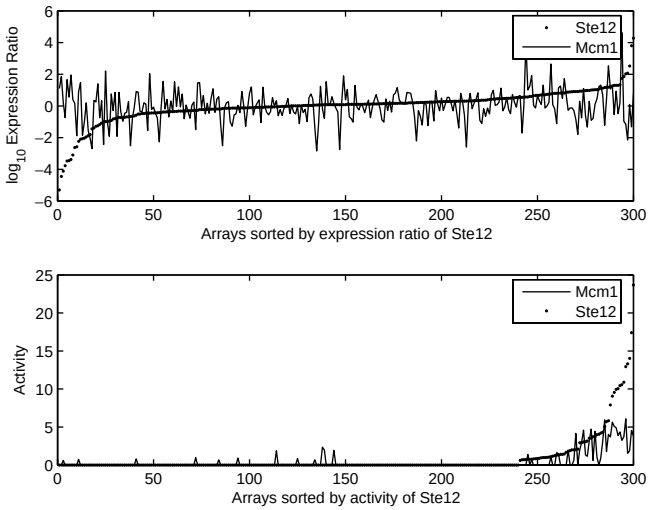

Fig. 9. Correlation of Ste12 and Mcm1 with respect to expression values and activity score. Although no correlation for the expression data can be detected, inferred activities clearly correlate.

(Fig. 10). We could find evidence that Slt2p regulates Swi4p (Baetz *et al*., 2001) and Rlm1p (de Nobel *et al*., 2000), but the connection to the kin3 mutant is not clear, since not much has been published about the function of Kin3p.

In general, the results for the kinases are harder to validate; however, the implied hypotheses are more detailed and more interesting.

4.4 Cooperating transcription factors

In section 4.2 we tried to identify cooperating transcription factors by correlating activity scores of single transcription factors. We also defined another pathway query that uses the protein interaction data from our background network to find pairs of transcription factors that cooperatively regulate sets of genes (Fig. 11). We computed all

Table 3. Top 10 scores for kinases together with their activity scores as described in Section 2.4

Experiment	Kinase	Score
dig1,dig2(haploid)	KSS1	17.0
kin3	SLT2	10.2
dig1,dig2	KSS1	9.3
ERG11(tetpromoter)	DUN1	9.2
ERG11(tetpromoter)	ESR1	9.2
swi4	ELM1	7.9
swi4	IPL1	7.9
ERG11(tetpromoter)	MKK2	7.8
ERG11(tetpromoter)	SLT2	7.8
ERG11(tetpromoter)	CKB1	6.6

The table displays the most relevant transcription factors for individual experiments, e.g. Kss1p was identified for the dig1,dig2 knockout experiment.

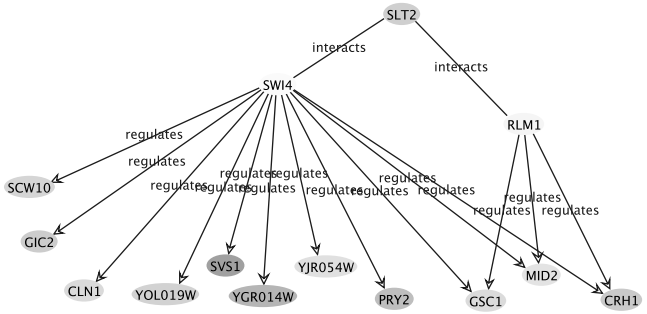

Fig. 10. Kinase SLT2, interacting transcription factors and regulated genes in the kin3 mutant. The shade of the nodes represents the magnitude of the expression ratio (mutant versus wild-type). All genes are upregulated.

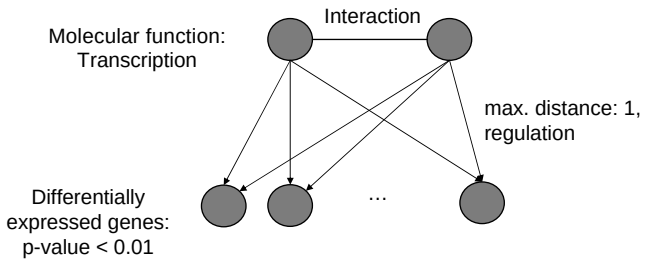

Fig. 11. Pathway query for finding cooperating transcription factors. This query is used to find interacting pairs of transcription factors that both have protein–DNA interactions with a common set of regulated genes.

instances of this pathway query with the pathway search algorithm and scored pairs of transcription factors using the same method as before for single transcription factors. The 25 transcription factor pairs with the highest scores are shown in Table 4. As expected from the previous results, we could find many high scoring instances with Arg80p, Arg81p and Ste12p, Mcm1p. In addition, we found two instances with Hir1p and Hir2p which are involved in cell-cycle regulated transcription of histone genes (Sherwood *et al*., 1993); the knocked out genes are hir2 itself and swi4 which is also a

Table 4. Top 25 scores for cooperating transcription factors together with their activity scores as described in Section 2.4

Experiment	Transcription factors	Score
arg80	ARG80, ARG81	11.2
sod1(haploid)	STE12, MCM1	7.7
pep12	ARG80, ARG81	7.0
vps8	ARG80, ARG81	7.0
fus3,kss1(haploid)	STE12, MCM1	6.9
rtg1	ARG80, ARG81	6.7
FR901,228	STE12, MCM1	5.8
yor080w	STE12, MCM1	5.6
AUR1(tetpromoter)	ARG80, ARG81	5.6
ste18(haploid)	STE12, MCM1	5.5
dig1,dig2(haploid)	STE12, MCM1	5.2
yhl029c	ARG80, ARG81	5.0
ste12(haploid)	STE12, MCM1	4.9
erg3(haploid)	ARG80, ARG81	4.7
ste24(haploid)	STE12, MCM1	4.5
HMG2(tetpromoter)	STE12, MCM1	4.4
hir2	HIR2, HIR1	4.4
ymr010w	ARG80, ARG81	4.3
ste5(haploid)	STE12, MCM1	4.1
KAR2(tetpromoter)	STE12, MCM1	4.0
yjl107c(haploid)	STE12, MCM1	3.9
yer044c(haploid)	ARG80, ARG81	3.7
yer044c(haploid)	STE12, MCM1	3.7
swi4	HIR2, HIR1	3.7
ERG11(tetpromoter)	FKH2, MCM1	3.6

The table displays the most relevant transcription factors for individual experiments, e.g. Ste12p and Mcm1p were identified for the ssn6 knockout experiment.

cell cycle-dependent transcription factor. The last pair in the list is Fkh2p, Mcm1p which are known to bind cooperatively to their targets (Hollenhorst *et al*., 2001).

Again, these results demonstrate how our algorithm can extract relevant contexts from the data in a very flexible way.

5 DISCUSSION

Biologists using microarray technology are often confronted with the problem of interpreting lists of regulated genes. Sorting these lists with respect to functional annotation and identifying overrepresented classes is often not sufficient to provide insight into the mechanisms leading to the observed expression patterns. Differentially expressed genes have to be interpreted in context with their regulators like transcription factors and signaling molecules in order to derive causal relationships and networks. Other interesting contexts could include proteins from the same metabolic pathway or even metabolites. The biological expert should be able to examine his data in a context that appears meaningful to him.

The pathway query language provides a formalism to formulate biological hypotheses that can provide such a context for the analysis of expression data. In general, conducting an analysis with pathway queries on a new dataset involves four steps:

(1) develop a pathway query that describes the context one is interested in;

(2) assemble networks that contain the relevant information;

(3) devise a scoring scheme to identify significant contexts;

(4) run the pathway search algorithm and evaluate the results.

All of these steps are critical for a successful analysis. The first step can best be done by a biological expert. Steps two and three will, in most cases, need the cooperation of a computer scientist and a biologist, although some simple scoring schemes can be specified in the pathway query and some general networks can be easily supplied.

In this paper we have performed the necessary steps for several research questions on public datasets with encouraging results. Using a background network containing protein–protein interactions and DNA binding information, we were able to show that information on regulatory contexts can be valuable for the interpretation of expression data. With the presented method, we could identify active transcription factors, active interacting pairs of transcription factors and to some degree, active kinases in single expression measurements. In addition, the method can deliver a clear interpretation of the data or at least exhibit a testable hypothesis as the results include not only the regulators but also the regulated genes and can be visualized as networks.

Although one of the strengths of the approach is its ability to achieve results with single or few measurements available, we could demonstrate its merits also for large-scale analyses. Correlation analysis of the computed activity scores revealed the potential to use the method to predict interactions between transcription factors. Further large scale analyses are needed to assess this potential.

Conflict of Interest: none declared.

REFERENCES

Baetz,K. *et al*. (2001) Transcriptional coregulation by the cell integrity mitogen-activated protein Kinase Slt2 and the cell cycle regulator Swi4. *Mol. Cell. Biol.*, **21**, 6515–6528.

Bardwell,L. *et al*. (1998) Differential regulation of transcription: repression by unactivated mitogen-activated protein kinase Kss1 requires the Dig1 and Dig2 proteins. *Proc. Natl Acad. Sci. USA*, **95**, 15400–15405.

Daignan-Fornier,B. and Fink,G. (1992) Coregulation of purine and histidine biosynthesis by the transcriptional activators BAS1 and BAS2. *Proc. Natl Acad. Sci. USA*, **89**, 6746–6750.

de Nobel,H. *et al*. (2000) Cell wall maintenance in fungi. *Trends Microbiol.*, **8**, 344–345.

Dolan,J.W. *et al*. (1989) The yeast ste12 protein binds to the DNA sequence mediating pheromone induction. *Proc. Natl Acad. Sci. USA*, **86**, 5703–5707.

Friedman,N. (2004) Inferring cellular networks using probabilistic graphical models. *Science*, **303**, 799–805.

Garey,M.R. and Johnson,D.S. (eds) (1979) *Computers and Intractability : A Guide to the Theory of NP-Completeness*. A Series of Books in the Mathematical Sciences, W.H. Freeman, San Francisco.

Gebauer,M. *et al*. (2005) Comparison of the chondrosarcoma cell line sw1353 with primary human adult articular chondrocytes regarding their gene expression profile and their reactivity to il-1b. *Osteoarthr. Cartil.*, **13**, 697–708.

Guetsova,M.L. *et al*. (1997) The isolation and characterization of *Saccharomyces cerevisiae* mutants that constitutively express purine biosynthetic genes. *Genetics*, **147**, 383–397.

Hollenhorst,P.C. *et al*. (2001) Mechanisms controlling differential promoter-occupancy by the yeast forkhead proteins Fkh1p and Fkh2p: implications for regulating the cell cycle and differentiation. *Genes Dev.*, **15**, 2445–2456.

Hughes,T.R. *et al*. (2000) Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109–126.

Hwang-Shum,J.J. *et al*. (1991) Relative contributions of MCM1 and STE12 to transcriptional activation of a- and alpha-specific genes from *Saccharomyces cerevisiae*. *Mol. Gen. Genet.*, **227**, 197–204.

Lee,T. *et al*. (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, **298**, 799–804.

Mootha,V.K. *et al*. (2003) Pgc-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.*, **34**, 267–273.

Segal,E. *et al*. (2003) Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, **19**(Suppl 1), 273i–282i.

Sherwood,P.W. *et al*. (1993) Characterization of HIR1 and HIR2, two genes required for regulation of histone gene transcription in *Saccharomyces cerevisiae*. *Mol. Cell Biol.*, **13**, 28–38.

Sohler,F. *et al*. (2004) New methods for joint analysis of biological networks and expression data. *Bioinformatics*, **20**, 1517–1521.

Steffen,M. *et al*. (2002) Automated modelling of signal transduction networks. *BMC Bioinformatics*, **3**, 34.

Tedford,K. *et al*. (1997) Regulation of the mating pheromone and invasive growth responses in yeast by two map kinase substrates. *Current Biol.*, **7**, 228–238.

The Gene Ontology Consortium (2001) Creating the gene ontology resource: design and implementation. *Genome Res.*, **11**, 1425.

Wingender,E. *et al*. (2000) Transfac: an integrated system for gene expression regulation. *Nucleic Acids Res.*, **28**, 316–319.

Xenarios,I. *et al*. (2000) Dip: the database of interacting proteins. *Nucleic Acids Res.*, **28**, 289–291.

Yeang,C.-H. and Jaakola,T. (2003) Physical network models and multi-source data integration. In *The Seventh Annual International Conference on Research in Computational Molecular Biology*, Berlin, Germany.