Inferring Transcriptional Networks by Mining 'Omics' Data

Tim Van den Bulcke[1], Karen Lemmens[1], Yves Van de Peer[2], and Kathleen Marchal[*,1,3]

[1]*BIOI@SCD, Dept. Electrical Engineering, K.U.Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee (Leuven), Belgium*

[2]*Department of Plant Systems Biology, Flanders Interuniversity Institute for Biotechnology (VIB), Ghent University, Technologiepark 927, B-9052 Ghent, Belgium*

[3]*CMPG, Dept. Microbial and Molecular Systems, K.U.Leuven, Kasteelpark Arenberg 20, B-3001 Heverlee (Leuven), Belgium*

Abstract: Inferring comprehensive regulatory networks from high-throughput data is one of the foremost challenges of modern computational biology. As high-throughput expression profiling experiments have gained common ground in many laboratories, different techniques have been proposed to infer transcriptional regulatory networks from them. Furthermore, with the advent of diverse types of high-throughput data, the research in network inference has received a new impulse. The use of diverse types of data, together with the increasing tendency of building the inference on biologically plausible simplifications, allows a more reliable and more complete description of networks. Here, we discuss how the research focus in the field of network inference is increasingly shifting from methods trying to reconstruct networks from a single data type towards integrative approaches dealing with several data sources simultaneously to infer regulatory modules.

Keywords: Module network, transcriptional network, network inference, systems biology.

INTRODUCTION

Recent technological advances have dramatically changed our views on molecular biology. Whereas a few years ago each gene or protein was studied as a single genetic entity, new, so-called 'omics' technologies (transcriptomics, proteomics, metabolomics and interactomics) allow analyzing large numbers of genes or proteins simultaneously [1-4]. As a result, a gene is no longer studied in isolation but as being part of a complex regulatory network. In a systems biology approach, a cell is considered a system that continuously interacts with its environment. The cell receives dynamically changing environmental cues and transduces these signals into the observed behavior (i.e., change of phenotype or change of physiological response). This signal transduction is mediated by the regulatory network. Genetic entities located on top of a regulation cascade transduce the signal downstream in the cascade *via* protein-protein interactions and/or through chemical modifications of intermediate proteins. The combined interaction of these transcriptional regulators activates or inhibits transcription of genes and therefore the production of the corresponding functional proteins. A complete regulatory network thus consists of proteins interacting with each other, with DNA or with metabolites to constitute a complete signaling pathway [4]. Unraveling these regulatory networks has become one of the major challenges of the field of interdisciplinary biology, known as "network reconstruction" or "network inference" [4-6]. Although the final goal is the description of global regulatory networks at systems level, so far current data availability mainly allows for high level inference of transcriptional networks; i.e., reconstructing that part of the signal transduction network observable when measuring mRNA expression based on methods that use extremely simplified representations of biological reality [4,7].

In this review, we first give a short overview of studies that describe the reconstruction of transcriptional networks solely from mRNA expression data. Traditional methods for network inference from gene expression data consider every gene as an individual node in the network, and their goal is to infer all interactions between these nodes (Fig. **1A**). Because of the large search space when treating all genes as individual nodes, most of these methods have extensive data requirements obviating their practical usage. However, for a biologist, the primary interest does not lie so much in reconstructing interactions between all genes but in recovering the interactions between the main mediators of the signal transduction, being the regulators and their target genes. Conceptual simplifications that reduce the complexity of the inference problem are therefore possible. For many genes, those with regulatory functions can be distinguished from structural ones by sequence based features, such as DNA binding domains for instance [7]. Compiling a limited regulator list based on genome annotation before inferring the network can therefore drastically reduce the number of parameters to be estimated. Recently, there is also a growing interest in the modular description of regulatory networks [8]. Genes being coexpressed in a subset of conditions and undergoing similar interactions within the regulatory network can be grouped into modules [9]. Thus, when belonging to the same module, genes are assumed to share the same properties. Using this representation, all genes within a module can be described by the same parameter set instead of using a unique parameter set for each single gene (Fig. **1B**). We will discuss network reconstruction methods that are based on this simplified network representation.

*Address correspondence to this author at the Dept. Microbial and Molecular Systems, K.U.Leuven, Kasteelpark Arenberg 20, B-3001 Heverlee, Belgium; Tel: +32 (0)16 329685;
E-mail: kathleen.marchal@biw.kuleuven.be

1574-8936/06 $50.00+.00

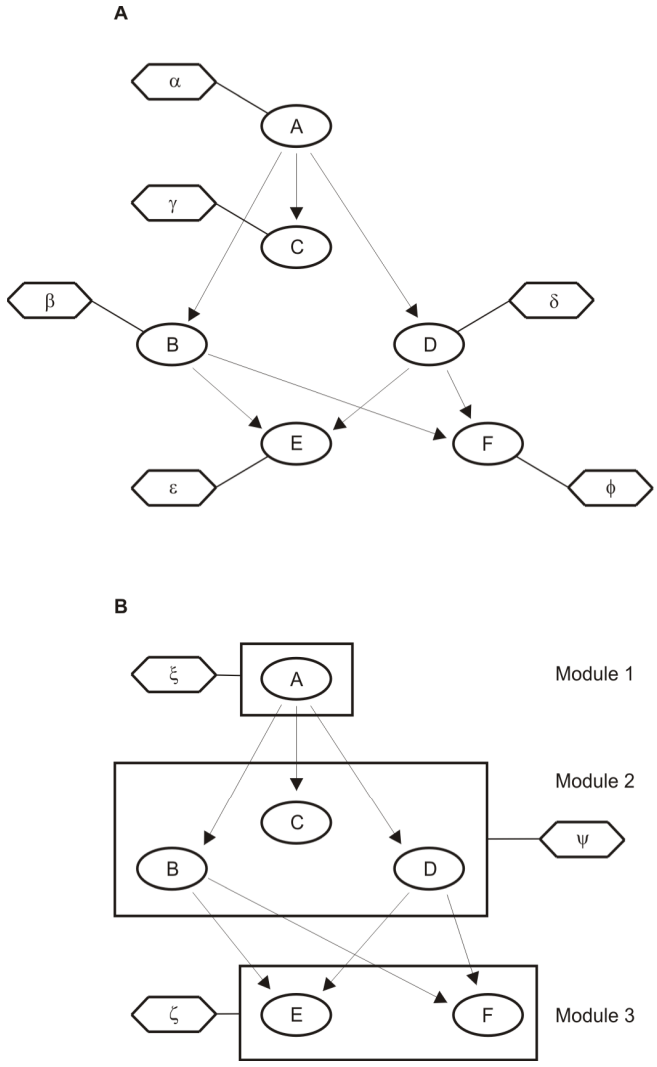

Fig. (1). Graphical representation of networks.

A) A complete network: each node corresponds to a single gene and is represented by an oval. The arrows correspond to the interactions between the genes. For each gene, a unique set of parameters (indicated by hexagons and a Greek letter) describes how the expression of that gene depends on the expression levels of its parents. B) A module network: each node corresponds to a single gene, denoted as oval. The arrows correspond to the interactions between the genes. Genes that depend on the same parents are grouped into modules. For each module, the module parameters (indicated by Greek letters) describe how the expression of all genes within the module depends on the module's parents. A single set of parameters is thus shared by all genes in the module (groups indicated by squares).

With the availability of heterogeneous 'omics' data, the problem of network/module inference becomes even more tractable. Different 'omics' data unveil distinct aspects of regulatory networks and integrating them allows a more complete insight into the regulatory networks. Here, we will focus on how well distinct computational methods for inference of transcriptional networks can deal with the specific biological features of relevant high-throughput data. It should be noted that the methods described throughout this review are not organism specific although most of them have been field-tested on *S. cerevisiae*, being the most extensively studied model organism [2].

INFERRING NETWORKS FROM GENE EXPRESSION DATA

Using Microarrays for Inferring Networks

Microarray experiments used for network inference can either be static or dynamic. Static experiments measure gene expression after the cell has adapted to its new environment, for instance if the cell or pathway under study has reached a steady state. Dynamic experiments on the other hand profile the changes in expression level during cellular adaptation. Although dynamic experiments inherently contain much more information on the causal interactions between the genes, most algorithms developed so far are not able to exploit this information.

An important issue that is often underestimated is the preprocessing of the gene expression data prior to the inference of networks. Although microarray technology produces continuous data, many methods require data discretization prior to further analysis. Data discretization implicitly assumes that in a large compendium of microarrays the complete dynamic range of expression values was observed for each gene. This complete range can then for example be subdivided into discrete levels such as high, basal, or low expression level. This discretization step is critical due to the potential loss of information. Also interpreting discretization levels as over-, basal and under-expression should be treated with caution, because observing the complete dynamic range of a gene can never be guaranteed unless a large compendium of data is used. This problem is exacerbated as expression values are often expressed relative to a reference (i.e., when using two-color based array techniques [10]). Large compendia consist of a concatenation of separately performed array experiments that rarely use the same reference [11]. Interpretation of what is over- or under-expressed should always be related to the proper reference.

From Inferring Complete Networks to Inferring Module Networks

The classical approaches for network reconstruction from gene expression data aimed at inferring the interactions between all genes. Methods based on Boolean models, Bayesian networks, differential equations and hybrids of those have been described (for exhaustive overviews we refer to D'Haeseleer *et al*. [12], van Someren *et al*. [13], and de Jong *et al*. [14]). Although some of these methods have lead to biologically relevant findings, in general the size of the currently available gene expression data sets does not meet the extensive data requirements for most of these methods. The number of experimental data points is still much smaller than the number of parameters to be estimated. This problem of under-determination is aggravated by the low signal to noise level of microarray data [15] and the inherent stochasticity of biological systems [16,17]. Therefore, inferring transcriptional networks using the methods described above is usually restricted to small networks or to situations where much data is available.

However, recently, there has been a major interest in the identification of module networks. Reformulating the

problem of inferring networks as a problem of inferring module networks can greatly simplify the complexity of the problem. For this review, we adopt the terminology introduced by Segal *et al*. [18] for modules and regulatory programs: a set of genes, coexpressed under all or under a particular set of conditions, is assumed to undergo similar interactions within the network. Such a gene set is called a *module*. A *regulatory program* is defined as the set of regulators of which the concerted action is responsible for the condition dependent expression of the genes in the corresponding module. The *module network inference* problem consists of two subtasks, namely identification of the modules and identification of the regulatory programs.

As module networks are a conceptual abstraction of the real networks some biological considerations have to be made. Module networks are condition dependent by definition, meaning that the genes of a module are only coregulated under a specific subset of conditions (i.e. tissues, time points, environmental conditions, etc.). In order to grasp this context specificity of a module, searching for modules in a large compendium of gene expression data not only implies identifying sets of coexpressed genes, but also selecting the conditions in which the genes exhibit a correlated behavior. This context specificity is reflected in the combinatorial composition of the regulatory program**.** Due to the combined action of regulators, genes of a module behave similar in a condition dependent way [19]. This is illustrated by a hypothetical example in Fig. **2** and a real example in Fig. **3**. The hypothetical example is a generalization of our own observations and those described by Ihmels *et al*. [20].

A gene expression dataset can therefore be subdivided in several overlapping context dependent modules. Modules comprising many conditions can be expected to contain few genes (called hereafter *seed genes*) with a potentially highly related function. Indeed, the more conditions genes appear to be coexpressed in, the more similar their regulatory program tends to be and the more connected their role in the pathway becomes. In a module, the number of genes will usually increase with a decreasing number of conditions. Obviously, there will be more genes that only share part of their regulatory program, i.e. the part that is active under the tested set of conditions. The fewer the number of conditions included in the module one considers, the less stringent the requirements on the overlap in the regulatory program becomes (Fig. **2**). Although these considerations seem trivial from a biological point of view, these properties of modules and programs make inferring modules and their corresponding regulatory program a non-straightforward task.

Identifying Modules Using Gene Expression Data

Biclustering algorithms are well suited to identify modules. They assign genes to condition dependent and potentially overlapping regulatory units, i.e. modules. In contrast to classical two-way clustering approaches [21], these biclustering algorithms do not group genes and conditions independently, but simultaneously, thereby identifying subsets of genes that are each correlated under a subset of conditions [9]. For the purpose of clustering,

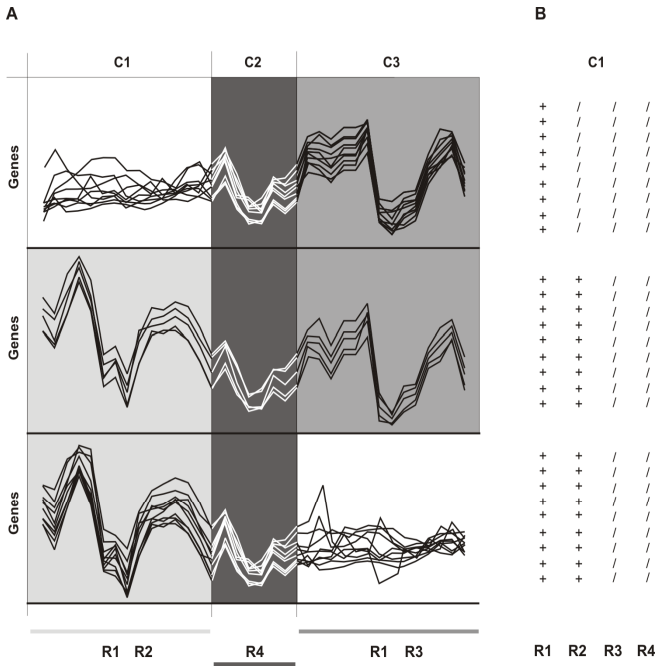

Fig. (2). Hypothetical example of higher order module organization.

C1, C2, C3: represent three unrelated conditions. R*i*: represent different regulators active in the respective condition dependent regulatory programs. R1, R2 are active in C1; R1, R3 are active in C3; R4 is active in C2. A) Distinct partially overlapping modules exist. Modules consisting of a few genes, tightly coexpressed in many conditions can be hypothesized to be associated with a highly specific function (horizontal middle panel). They consist of genes that respond to the same regulatory program and are coexpressed under all conditions. As modules are extended with more genes, the number of conditions can be expected to decrease. Genes within such extended modules only share part of the regulatory program, i.e. the one that is active under the selected conditions (top and bottom panel). Which of these overlapping modules will be detected by biclustering depends on the specificities of the algorithm and the parameter choices when applying the algorithm.

B) Hypothetical ChIP-chip result for the respective regulators obtained from a ChIP-chip compendium measured in C1 only. +: binding between target gene and regulator is observed; /: no binding is observed. Since only ChIP-chip data of condition C1 exists, the data contains much missing and conflicting information when extrapolating it to other conditions. Therefore, such information should be interpreted with caution for data-integration. See text for details.

microarray experiments are usually organized in an expression matrix, where the rows correspond to genes and the columns correspond to different conditions. Biclustering algorithms can be grouped according to different criteria such as whether they are based on probabilistic methods or not, whether they allow for overlapping modules or not, whether they search for all modules at once or try to identify the modules separately in subsequent runs, whether the obtained modules are self-consistent or not, and whether they use seeds as a starting point for module identification or not. From a biological perspective, having an algorithm that allows for overlapping modules is desirable (see Fig. **2**) and the considerations made above). From an algorithmic perspective, self-consistency of the module (a criterion

Table 1. Comparison of Module Inference Algorithms

Study	Method	Data types	Overlapping modules	Module network	Subset conditions	Integration of additional data	WWW
Eisen *et al.*[1]	Pairwise average-linkage clustering	gene expression	no	no	no	No	http://genome-www.stanford.edu/clustering/
Ihmels *et al.*[2]	Based on Signature algorithm	gene expression	yes	no	yes	yes	http://www.weizmann.ac.il/home/barkai/modules/
Tanay *et al.*[3]	Weighted bipartite graph	gene expression, protein interaction, growth phentotype, TF binding	yes	yes	yes	yes	http://www.cs.tau.ac.il/~rshamir/samba
Gasch and Eisen[4]	Modified k-means	gene expression	yes	no	no	no	http://rana.lbl.gov/FuzzyK
Sheng *et al.*[5]	Gibbs sampling, Bayesian network	gene expression	yes	no	yes	yes	http://www.esat.kuleuven.ac.be/~qsheng/query_driven.html
Segal *et al.*[6]	PRM-inspired	gene expression	no	yes	no	yes	http://ai.stanford.edu/~erans/module_nets/
Cheng and Church[7]	biclustering	gene expression	no	no	yes	no	http://arep.med.harvard.edu/biclustering/
Getz *et al.*[8]	CTWC	gene expression	no	no	yes	no	http://www.weizmann.ac.il/physics/complex/compphys/
Kluger *et al.*[9]	Spectral biclustering	gene expression	no	no	yes	no	
Lee *et al.*[10]	Bayesian network / information theory	gene expression, biological annotation	yes	yes	no	yes	http://biosoft.kaist.ac.kr/~dhlee/monet/index.html
De Bie *et al.*[11]	Apriori algorithm inspired, statistics	gene expression, ChIP-chip, sequence data	yes	no	no	yes	http://www.esat.kuleuven.ac.be/~kmarchal/Supplementary_Info_PSB2005/SuppWebsiteYeastPSB.html
Bar-Joseph *et al.*[12]	Statistics	gene expression, ChIP-chip	yes	no	no	yes	http://psrg.lcs.mit.edu/GRAM/Index.html

The table compares the different module inference algorithms that were discussed in the review. Columns in the table denote the different attributes for which the methods were compared. ***Method*****:** the basic methodology used, ***Data types***: the different data types which the algorithm combines in the study; ***Overlapping modules***: indicates whether the method is able to generate overlapping modules; ***Module network***: indicates if the method generates a module network (module and transcriptional program); ***Subset conditions***: indicates if the modules are defined for a subset of the conditions or for all conditions; ***Integration of additional data***: indicates whether it is possible to easily extend the algorithm to incorporate data sources other than the data sources described in the study; ***WWW***: a link to the online resources/software of the algorithm.

introduced by Ihmels *et al*. [20]) allows for generating optimal and potentially overlapping modules instead of optimizing the global data partitioning. The use of seeds, defined as initial sets of genes around which a module is formed, leads to a straightforward extension for data integration (see further). The different biclustering algorithms together with their most important properties are summarized in Table **1**.

[1] Eisen *et al*. [76]
[2] Ihmels *et al*. [20]
[3] Tanay *et al*. [60]

Simultaneous Inference of Modules and Regulatory Programs

Biologists are not only interested in inferring the module composition but also in reconstructing the regulatory program. Because the regulatory program determines the behavior of the genes in the module and the presence of a module reveals which programs are active, it does make sense to simultaneously infer the active programs and to partition the genes into modules. A first step into this direction is the *module networks method* developed by Segal *et al*. [18], which is inspired by the probabilistic relational model (PRM) framework [22-25] (see Fig. **2**).

While Friedman *et al*. [26], in the initial applications of probabilistic models for network inference, assigned each gene as a separate node with its own parameters to the Bayesian network, Segal *et al*. [18] grouped genes into modules, where genes belonging to the same module share the same parameters and have the same set of regulators. This considerably reduces the number of model parameters to be estimated and at the same time increases the number of data points available for estimating each parameter. For each module, the effect of the set of regulators on the expression profile of the genes in the module is modeled as a transcriptional program by using a regression tree. The iterative procedure uses an *Expectation-Maximization* (EM) algorithm to search the optimal regulatory program for each module (*M-step*, using a regression tree for the regulatory program) and to subsequently reassign each gene to the module of which the program best predicts its behavior (*E-step*, using the model score). Although very innovative, the approach still has some shortcomings, mainly since it is based on gene expression data only.

In the original setup of Segal *et al*. [18], a large set of candidate regulators is selected based on their annotation, while the regulatory program of each module is composed of the subset of candidate regulators for which the expression profiles best explains the expression profile of the genes in the module. This criterion complicates distinguishing between regulators that are actually causing the modules behavior (and thus belong to the regulatory program) and those for which the observed expression behavior is a consequence of the action of the program (and thus belongs to the module). Moreover, constitutively expressed regulators activated post-transcriptionally will never be part of the regulatory program because their expression profile will not correlate with the genes in the module. The number of regulators for which the expression profile correlates well with the profiles of its target genes, is limited anyhow as Herrgard *et al*. [27] showed that in yeast over 80% of the tested pairs of expression profiles between regulators and targets were not significantly correlated. In the method of Segal *et al*. [18], context specificity of the modules is not explicitly taken into account (no conceptual biclustering) because genes belonging to a module are required to be coexpressed over all conditions tested. By definition, a gene can only belong to a single module and overlapping modules are therefore not possible. Segal *et al*. [18] applied their method to the Gasch *et al*. dataset [11] (a large scale microarray experiment (173 arrays) assessing expression changes under various stress conditions in the yeast *S. cerevisiae*) and identified 50 modules involved in various processes. They proved the biological potential of their method by experimentally validating three of the hypotheses that followed from their predictions.

NETWORK INFERENCE BY DATA INTEGRATION

Importance of Additional Data

Since gene expression data does not contain sufficient information to fully reconstruct transcriptional programs, an increasing number of inference methods exploit the use of heterogeneous data. The most frequently used data types are motif, chromatin immunoprecipitation DNA microarray (ChIP-chip), and protein interaction data (for more elaborated reviews on these data sources see Blais and Dynlacht [3], Wei *et al*. [7], Bader *et al*. [2]). Each of these additional data sources describes the molecular biological networks from a different perspective and combining them allows a more detailed representation of the networks.

ChIP-chip data (or location data) measures *in vivo* the direct interaction between a transcriptional regulator and its target genes [28,29] and contains information both for the identification of the regulatory programs as for the inference of the modules. The regulatory program of a module simply consists of the combined set of regulators binding to genes within the module under a subset of conditions. On the other hand, genes bound by the same regulators are more likely to belong to the same module. ChIP-chip data is based on the direct physical interaction between a regulator and its target genes. Unlike expression data, the use of ChIP-chip data allows for the identification of constitutively expressed regulators activated by posttranslational modifications as members of the regulatory program. Like gene expression data, ChIP-chip data is condition dependent and some interactions of a regulator with its target genes only occur in very specific conditions (for an example of such condition enabled regulators (see Fig. **3**)). Moreover, the binding of a regulator in a specific condition does not necessarily imply that the regulator actually regulates the gene under the prevailing conditions, for instance in case of combinatorial control when the presence of an additional regulator or coactivator is required [3]. Hence, a ChIP-chip based network of interactions is by definition static, i.e. it shows interactions but not the context dependency of these interactions [30]. Being tedious to generate, as it requires a separate set of microarray experiments per tested regulator and per tested condition, it is unlikely that a separate ChIP-chip compendium will be available for each condition and for each transcription factor in the short term. Extrapolating the already measured static interactions to infer regulatory programs for conditions not primarily tested in the ChIP-chip assay, is thus required. However, one should bear in mind that a static network of ChIP-chip data will have both

[4] Gasch and Eisen [77]
[5] Sheng *et al*., Query-driven biclustering of microarray data by Gibbs sampling. *Internal Report 05-33, ESAT-SISTA, K.U.Leuven (Leuven, Belgium)*
[6] Segal *et al*. [18]
[7] Cheng and Church [78]
[8] Getz *et al*. [79]
[9] Kluger *et al*. [80]
[10] Lee *et al*. [46]
[11] De Bie *et al*. [58]
[12] Bar-Joseph *et al*. [57]

missing interactions and false positive interactions that are not supported by other data sources obtained in different experimental conditions (see Fig. **2**). For simple organisms such as yeast and prokaryotes, promoter arrays with a cDNA probe for each intergenic sequence will be sufficient to identify most of the transcription factor binding sites in a ChIP-chip experiment. For higher eukaryotes such as human, with long intergenic sequences in which the motifs can lie far from the genes they regulate (and even in the introns), arrays that tile all non-repetitive sequences of the chromosome will be needed (Buck and Lieb [29]). Although for several model organisms arrays usable for ChIP-chip experiments [31,32] are becoming available and cover either the whole genome or regions of interest, data are publicly available only for yeast so far.

A *DNA-motif* is a short conserved DNA-sequence located in the promoter region of a gene that serves as the recognition site for a transcriptional regulator. Compared to ChIP-chip, motif data contains less information. Motifs can help identifying modules (genes with similar motifs should belong to the same module) and can give an idea on the number of regulators belonging to the regulatory program of the corresponding module but contain no information on the identity of these regulators. By using sequence based *in silico* techniques to identify motifs, a set of possible motifs can be compiled independent of the experimental conditions [33-35]. Although the binding of a regulator to a specific motif is condition dependent [36], the compendium itself can contain all motifs and thus is more complete than a compendium of ChIP-chip data (less missing data). The

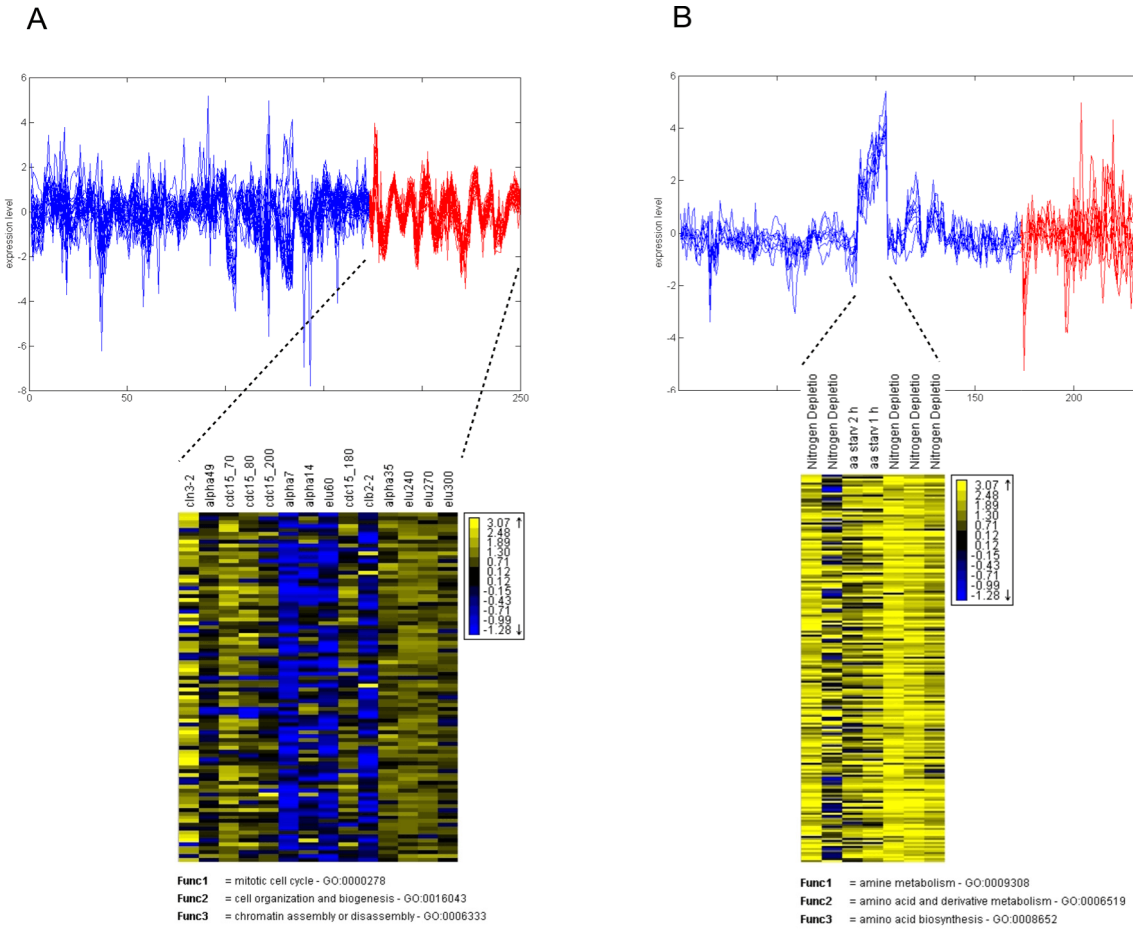

Fig. (3). Biological example illustrating the context dependency of a module.

The expression datasets used in this example consists of the Gasch *et al*. [11] and the Spellman *et al*. [53] datasets. Upper panels show the expression profiles of two seeds in the Gasch *et al*. dataset [11] (blue) and the Spellman *et al*. dataset [53] (red). The respective module seeds were identified by applying *ReMoDiscovery* [58] on the expression data and on the ChIP-chip dataset of Harbison *et al*. [55]. Lower panels show biclusters extracted from the same expression dataset using the biclustering tool *SAMBA* [61] *via* the *EXPANDER* software [60,81] after concatenating both datasets. A) The seed module's regulatory program consisted of the following regulators, known to be involved in cell cycle: NDD1, FKH2, MCM1 [82]. The regulatory module clearly shows a tighter coregulation in the cell cycle related Spellman *et al*. dataset [53] than in the stress related Gasch *et al*. dataset [11]. Indeed, the bicluster obtained when applying the *SAMBA* biclustering to the dataset, only contained cell cycle related conditions. B) The seed module's regulatory program consisted of the regulators DAL81, DAL82, GAT1 and GLN3, which have been described previously to be involved in nutrient-sensing signaling pathways [83]. Binding of these regulators to their target genes was only observed under nutrient deprived conditions (and not in ChIP-chip performed, in rich medium, for instance). This module is clearly active in the stress related Gasch *et al*. dataset [11] but genes showed a non-correlated profile in the Spellman *et al*. dataset [53]. Applying the *SAMBA* biclustering algorithm resulted in a bicluster containing the conditions of nitrogen depletion of the Gasch *et al*. dataset [11] and selecting genes involved in amine and amino acid metabolism.

drawback is that these compendia usually contain many false positives (motifs predicted by *in silico* analysis that are not biologically relevant). Although in theory motif compendia can be constructed for any organism for which a sufficient number of sequence data is available, applying the motif detection procedures is far from trivial in organisms with long intergenic sequences such as vertebrates. Specialized databases exist (such as TRANSFAC [37], regulonDB [38], …) that contain information on regulatory motifs of diverse organisms.

Protein interaction data provide experimental information on direct interactions between proteins [39,40]. Similar to ChIP-chip data, they are condition dependent and thus contain missing data. As with ChIP-chip data they also provide information on both the level of module detection and regulatory program inference. Mutually interacting proteins have a higher chance of being coexpressed, and regulators interacting at protein level reveal information on the part of the regulatory program that is invisible at transcriptional level. Protein interaction data for several organisms can be found in databases such as DIP [41] and BIND [42].

ChIP-chip, gene expression, motif and protein interaction data are all based on high throughput technologies and are therefore prone to noise. Measurement and biological noise in the individual data sources can be so prohibitive that only by combining them, reliable predictions can be made. Especially when combining distinct data sources that have missing values and false positives, the issue of specificity and sensitivity of the different methods becomes of major importance. Depending on the biological question (e.g. when the prediction is needed for the planning of expensive and tedious downstream experiments), one can choose for a more or less conservative approach, on the one extreme requiring that all data sources confirm each other or on the other extreme also allowing making predictions based on partially contradicting data (e.g. to discover potentially new relationships between genes). Recently, different methods have been developed to integrate these heterogeneous data. Here we mainly discuss some recent methods for integration that have module network identification (emphasizing modules and regulatory programs characterized by condition dependency) as their main focus. Most of the methods to generate the high throughput data described above have originally been developed for yeast, partly explaining why data availability for other model organisms, even simpler ones such as *Escherichia coli,* is lacking behind. Although some of the public data are made available in specialized databases, much of the (raw) information is only accessible *via* the supplementary files of the corresponding publications.

Methods for Data Integration

Among the approaches developed to infer complete networks, in particular probabilistic approaches have been extended to integrate heterogeneous data sources. Additional data is used to supplement the expression data for example by using ChIP-chip data or protein interaction data as priors for Bayesian networks [43-48]. Biclustering algorithms that start from a set of seed genes to define biclusters (see Table **1**) can, to some extent, integrate heterogeneous data: the seeds can be defined by using other data sources and can thus be considered as prior information to the biclustering algorithm [9].

An example of a deterministic method for data-integration is the *MA-networker* algorithm for integrative modeling of expression and ChIP-chip data [49]. This algorithm uses multivariate regression of mRNA expression levels on the genome-wide binding profiles of a large number of transcriptional factors (ChIP-chip data) to explain to what extent each transcription factor is responsible for the observed changes in mRNA expression in a single microarray experiment. When performing the regression procedure in parallel on a compendium of multiple microarray experiments, a transcriptional factor activity profile is obtained that implicitly expresses the conditional dependence of a specific regulator on the tested experimental conditions, indicating whether the regulator is responsible for the expression changes per experimental condition. For the identification of additional target genes of a specific regulator, the method of Gao *et al*. [49] is original in that, in contrast to most other studies, it does not search for genes of which the expression profile is highly correlated with the one of the regulator but for genes with an expression profile highly correlated with the activity profile of the regulator (defined as the coupling strength). Although the method searches for the condition dependent activation of a gene by one regulator it does not yet use this information to determine the concerted action of more regulators, i.e. to compile complete regulatory programs [49]. MA-networker was applied to the yeast ChIP-chip data of Lee *et al*. [50] and a compendium of 750 microarray experiments covering different physiological conditions. They found that 58% of the genes whose promoter was bound by a regulator were true targets and that a set of target genes of which the expression profile exhibits a large coupling strength with the activity profile of a specific regulator was significantly enriched for specific functional categories.

The following methods aim at identifying modules and regulatory programs. Similar to Gao *et al*. [49], they treat additional data sources with equal importance compared to gene expression data (contrary to other approaches where additional data sources are treated as prior).

Wang *et al*. [51] propose a heuristic semi-integrative, semi-sequential method to combine motif- and gene expression data in order to search for target genes of a particular transcription factor, the context specificity of the transcription factor, and the combinatorial control between different regulators. Their method is based on the assumption that if a transcription factor is activated under a particular condition, its target genes should have similar responses as those observed in a perturbation experiment of that transcription factor. Regulatory motifs recognized by a particular transcription factor, and their corresponding targets, are identified with the *REDUCER* algorithm [52] in an experiment where the particular transcription factor is perturbed (e.g. overexpressed, mutated). Based on this information, a score vector for the perturbation experiment is constructed which consists, for each potential target gene, of a value that increases with the ratio of overexpression of that gene in the prevailing experiment and with the number of motifs for the transcription factor of interest. Besides for the

perturbation experiment, this vector is also calculated for different other microarray experiments. From the correlation between these calculated score vectors, the conditional activity of the transcription factor is derived. Since in some microarray experiments more regulators can be active simultaneously, the overlap in their target genes is used to derive combinatorial action of different transcription factors. Based on a microarray compendium (which comprised the Gasch *et al*. [11] and Spellman *et al*. [53] datasets), for 28 transcription factors of which a perturbation experiment was available (from the Rosetta compendium [54]), Wang *et al*. [51] identified the corresponding target genes, motifs and relevant conditions.

Harbison *et al*. [55] and Kato *et al*. [56] use a heuristic approach that is partially integrative. For each regulator, they first compile reliable lists of target genes, based on the integrated knowledge from literature [50,55], ChIP-chip [55,56], and comparative genomics data [55]. Subsequently, the search for statistically overrepresented motifs in the promoter regions of these target genes results in the identification of the motif tags characteristic for each of the regulators. Kato *et al*. [56] go one step further in reconstructing the modules and programs by searching for statistically overrepresented motif combinations. Genes of which the promoters contain a particular motif combination and that share a similar expression profile over time comprise a module. In a final step, regulatory programs are identified based on the ChIP-chip data by determining the identities of the regulators that are statistically overrepresented in the genes of the respective modules. When applying their method to the ChIP-chip data of Lee *et al*. [50], they specifically focused on the cell cycle and could identify most of the previously described cell cycle transcriptional complexes.

A conceptual extension to the previously mentioned heuristic methods is proposed by Bar-Joseph *et al*. [57] and De Bie *et al*. [58]. Regulatory programs and module seeds are defined in a joint learning step based on ChIP-chip and gene expression data [57], or based on ChIP-chip data, gene expression data, and motif data [58]. In the former approach (*GRAM*), seeds are defined by identifying sets of genes with a common set of transcription factors and having a highly correlated expression profile (determined by microarray analysis). In the latter approach (*ReMoDiscovery*), module seeds are maximal gene sets of which the expression profiles are highly similar and that have a minimal set of regulators and motifs in common [58]. The shared set of seed regulators and motifs corresponds to the regulatory program determining the observed coexpressed behavior of the module seed genes. Searching for maximal gene sets that meet these requirements on all three datasets translates into a combinatorial problem, which is solved by a modification of the *APriori* algorithm [58]. Because the initial seed discovery in both approaches relies on stringent criteria (information in all datasets has to be mutually consistent), the seed modules are likely to underestimate the true module size. For this reason, both algorithms use a second module extension step. Bar-Joseph *et al*. [57] extend the module seeds by first identifying candidate genes with an expression profile sufficiently similar to the seed profile and with a sufficiently low *p*-value for the binding of each of the individual regulators of the module seed. A combined *P*-value based on the individual p-values for all module regulators is calculated for each of the candidate genes passing these requirements and the gene is added to the module if the combined *P*-value is sufficiently low. *ReMoDiscovery* contains a second module extension step where additional genes are identified for which the expression profile is highly correlated with that of the seed genes. The optimal size in number of genes of the module is determined by the correlation coefficient resulting in a module with the largest enrichment in seed motifs and regulators. The regulatory modules detected by both approaches can be used as input sets for motif detection tools. Note that both approaches [57,58] yield few false positive modules, but they fail to identify modules if not all data sources separately confirm the presence of a seed module. Neither Bar-Joseph *et al*. [57] nor De Bie *et al*. [58] in its original implementation explicitly take into account the conditional nature of the regulatory program. De Bie *et al*. [58] solve the problem by grouping microarray experiments performed in the same experimental condition and applying the algorithm to each group of microarrays separately. Only when the regulatory program is active in the specific dataset, the seed module can be extended. Both methods were applied to the yeast ChIP-chip compendia and various microarray experiments in yeast. Although using slightly different datasets, both groups identified a similar number of modules, involving a comparable number of regulators. By performing gene specific ChIP-chip experiments, Bar-Joseph *et al*. [57] experimentally validated a random selection of predictions proving the potential of their approach.

Xu *et al*. [59] extended the module networks framework of Segal *et al*. [18] (see higher), by incorporating ChIP-chip data. For identifying the most likely candidates of a regulatory program, they select regulators for which the expression profile shows a high mutual information with the one of the module genes (comparable to the approach of Segal *et al*. [18]) and regulators with high binding probabilities based on ChIP-chip data. The binding probability between a regulator and its target genes is also regarded as a structure prior to the Bayesian score, which scores the inferred module networks. As a result, the score of the resulting network is both increased when there is a high correlation between the expression profile of the regulator and the module genes (when calculating the regression tree that derives the regulatory program) and when the binding probability between a regulator and its targets is high (in the form of a structure prior). This joint scoring allows different weaker indications from separate data sources to be joined to a significant indication, to indicate for example that a gene is part of a module. It also allows constitutively expressed regulators with low location probability to be part of the regulatory program. As with the method of Segal *et al*. [18], conditional dependence of the regulatory programs is not taken into account. Using the ChIP-chip compendium of Lee *et al*. [50] and the microarray experiments of Gasch *et al*. [11] and Spellman *et al*. [53], Xu *et al*. [59] identified 50 modules involving 86 regulators covering a wide range of cellular/physiological processes.

Another method for module detection is *SAMBA*, developed by Tanay *et al*. [60], which uses a bipartite graph based representation of the data where one subset of nodes represents genes and the other subset represents the

properties derived from the distinct heterogeneous data. An edge represents the assignment of a property to a gene with the weight of the edge being indicative of the statistical strength of the assignment. The problem is then reduced to finding 'heavy' subgraphs in a weighted bipartite graph. A graph-based biclustering algorithm [61] is used to identify modules (i.e. a set of genes that show similarities only in a subset of properties). Like for other biclustering algorithms, overlapping modules are allowed. This method thus fully exploits all data sources, allows dependency of the program, not only conditioned on the expression data but also on the other data sources. This is important, considering that ChIP-chip or protein interaction data, assessed under specific conditions might not be supported by, for instance, expression data measured under different conditions. One drawback of this method, from a biological point of view, is that while the uniform representation of the heterogeneous data allows the automatic identification of modules, the compilation of the regulatory programs is not automatically derived from the analyses. It is also unclear how the different sets of properties of the genes should be weighted compared to one another, while this will have a profound impact on the identified modules. Integration of additional data sources is straightforward as long as the data can be described as gene properties. Tanay *et al*. [62] identified 1200 significant modules in a large yeast compendium of heterogeneous datasets. Eighty six percent of the modules were based on more than one dataset and for the construction of 68% of the modules at least three different data sources were used, indicating the importance of using complementary information.

Besides the data integration methods mentioned above, there are many other methods that focus on different aspects of regulation, such as the combined identification of regulatory motifs and coexpressed genes (e.g. [52]).

Comparison Between Representative Tools

Regarding their biological relevance, most methods described above have been applied to public data sets on yeast, integrating, depending on the specificities of the study, the ChIP-chip compendium data of Lee *et al*. [50] and Harbison *et al*. [50,55], motif data [33], protein interaction data [63], and various public microarray experiments (most often including the study of Gasch *et al*. [11], which profiles gene expression in various stress conditions and the study of Spellman *et al*. [53], describing the dynamic changes in gene expression during yeast cell cycle; see Table **1**). It is heartening to see that each of these studies is able to retrieve many of the known modules and programs in yeast. The fact that each study uses different data sets as input obviates a nonbiased comparison between them. However, as each of these methods aims at detecting global regulatory modules and was applied to the same biological system, some overlap in the results is to be expected. Indeed, among all modules discovered by the distinct methods, those involved in processes related to ribosome biogenesis, cell cycle, stress response, amino acid metabolism/biosynthesis, and energy/carbohydrate metabolism seem to frequently reoccur (see Table **2**). This overlap in detected modules is partially due to the fact that most studies used cell cycle and stress related datasets, but also because some processes seem to act globally and appear active under very diverse conditions (for instance, ribosome biogenesis).

Despite this overlap, the final number of detected regulatory modules, the number of genes contained within them and the size of the corresponding programs varies significantly between the different methods. Much of what is detected depends on the characteristics of the implementation and the specific choice of the parameter settings. In general, solutions are most similar between Bar-Jospeh *et al*. [57] and De Bie *et al*. [58], while these are moderately different from Tanay *et al*. [60] and most different from the solutions obtained by Xu *et al*. [59] (see Table **2** for a few examples). However, similarity between solutions is not a proof of biological truth, but rather a reflection of whether similar aspects of the data are uncovered by the approaches compared. For instance, the fact that Xu *et al*. [59] cannot detect cell cycle modules indicates that regulators of cell cycle, which are prominently posttranscriptionally regulated, are harder to detect using their approach. However, in contrast to the approaches of De Bie *et al*. [58] and Bar-Joseph *et al*. [57] that rely heavily on ChIP-chip data, Xu *et al*. [59] are able to identify the involvement of many regulators in the regulatory program for which no ChIP-chip data are available yet. Table **2** compares the composition of similar global regulatory modules across different studies. It illustrates well that no single best definition of a biological module exists. Each method detects a slightly different instance of a module involved in the same cellular process. Usually, a larger regulatory program implies a module with more conditions and less genes, and vice versa. What is still lacking in each of these methods is a comprehensive overview that gives an abstract representation of how modules are contained within larger ones and how this relates to the changing complexity of the regulatory program.

A method's applicability to some extent also depends on its user-friendliness. Some methods might require extensive parameter fine-tuning. This fine-tuning is facilitated if parameters have a well-defined biological meaning or if changing them changes the outcome in a predictable way (see Table **3**). Table **3** shows the different user-definable parameters of each algorithm, a short description, the default value, and a robustness range around the default values. The robustness range defines the range of parameter values where the resulting set of modules is still sufficiently similar to the set of modules for the default parameters. Besides the robustness analysis, we performed for different algorithms mentioned in the text a minimal benchmark of running times by using gene expression data of Gasch *et al*. [11] and ChIP-chip data of Harbison *et al*. [55] of which for both datasets, respectively the first 50, 70 and 100 columns were used for benchmarking. Running times for *GRAM* were 20-40 sec for datasets with over 6000 genes and up to 70 regulators. However, running time became prohibitively slow when more regulators were included in the ChIP-chip dataset: over 20 min (terminated) for a dataset with 100 regulators. *SAMBA* and *ReMoDiscovery* performed well on these datasets (less than 1 minute). The experiments were conducted on a laptop with a Pentium 4 Mobile 1.8GHz processor and 768MB RAM memory.

Table 2. Comparison of the Composition of Global Regulatory Modules, Identified by Different Methods

Study	Module ID	Module size	Most enriched functional category	Regulatory program
Cell cycle				
De Bie *et al.*[13]	6 (Spellman *et al.*[14])	113 genes	DNA synthesis and replication	Swi4; Swi6; Mbp1; Ash1
	7 (Spellman *et al.*[14])	186 genes	DNA processing	Swi4; Swi6; Mbp1; Stb1
Bar-Joseph *et al.*[15]	78	11 genes	Cell cycle and DNA processing	Swi6; Mbp1
	102	8 genes	Cell cycle and DNA processing	Swi4; Swi6; Mbp1
Tanay *et al.*[16]	619	29 genes (69 conditions)	Mitotic cell cycle	Mbp1; Swi4
	419	12 genes (135 conditions)	Mitotic cell cycle	Swi4; Swi6; Mbp1; Fkh2; Ste12
Xu *et al.*[17]	21	40 genes	Cell cycle (G1/S) and DNA replication	Cln2; Clb5; Zds2; Swe1; Clb6
	30	12 genes	Cell cycle (G1/S) and signaling II	Met18; Mad1; Hir2; Yjl206C; Syg1; Sum1
Amino acid metabolism / biosynthesis				
De Bie *et al.*[13]	48 (Gasch *et al.*[18])	2 genes	Amino acid metabolism	Gcn4; Leu3
	5 (Gasch *et al.*[18])	100 genes	Nitrogen and sulfur utilization	Gcn4; Cbf1; Met32
Bar-Joseph *et al.* [15]	13	14 genes	Amino acid biosynthesis	Gcn4
	86	5 genes	Amino acid biosynthesis	Gcn4; Arg80; Arg81
Tanay *et al.* [16]	848	112 genes (93 conditions)	Amino acid metabolism	Gcn4
	3057	16 genes (148 conditions)	Amino acid biosynthesis	Gcn4; Arg80; Arg81
Xu *et al.* [17]	20	36 genes	Sulfate amino acid, and purine metabolism and Ty ORFs	Hap1; Pdr3; Met32
	22	29 genes	Amino acid and purine metabolism	Gat1; Xbp1
Respiration				
De Bie *et al.*[13]	2 (Gasch *et al.*[18])	2 genes	Respiration	Hap1; Hap2;Hap4;Hap5;Gln3
	2E (Gasch *et al.*[18])	30 genes	Energy	Hap2; Hap4; Hap5
Bar-Joseph *et al.* [15]	18	15 genes	Mitochondrion	Hap4
	57	7 genes	Respiration	Hap3; Hap4
Tanay *et al.* [16]	1103	76 genes (83 conditions)	Energy derivation by oxidation of organic compounds	Hap4
	3103	18 genes (112 conditions)	Aerobic respiration	Hap2; Hap4
Xu *et al.* [17]	Not detected	-	-	-

Per study, two representative modules are shown, enriched for the indicated functional class. ***Study*****:** the original study in which the modules were detected, ***Module ID*****:** ID referring to the module in the original studies (from the respective websites with supplementary information). ***Module size*****:** the number of genes in the regulatory module. ***Most enriched functional category*****:** indicates for each regulatory module the most enriched functional category (as found on the respective websites). ***Regulatory program*****:** set of the regulators corresponding to the modules identified by each of the studies.

NECESSITY FOR PROPER BENCHMARKING

Solving network inference problems usually requires complicated algorithms with many tunable parameters. Moreover, due to the computational complexity of reconstructing regulatory networks, many algorithms only find local optimal solutions instead of the global optimum. Extensive validation is needed to test the influence of parameters on the results, and to assess the extent to which a found solution approximates the global optimal solution.

Testing the performance of network inference algorithms requires repeatedly applying them to large data sets, obtained from many experimental conditions and derived from different well-known networks. Unfortunately, experimental data sets of the appropriate size and design are usually not

[13] De Bie *et al.* [58]
[14] Spellman *et al.* [53]
[15] Bar-Joseph *et al.* [57]
[16] Tanay *et al.* [60]
[17] Xu *et al.* [59]
[18] Gasch *et al.* [11]

Table 3. User-Definable Parameters for Module Inference Algorithms that Use Heterogeneous Data Sources

Algorithm	Parameters	Description	Default Value	Robustness Boundary
SAMBA[19]	Overlap prior factor	The desired amount of overlap between the modules	0.1	0.01 - 0.12
GRAM	Core profile p-value	Threshold on the significance value of the core set of genes: the core set contains only genes for which all the transcription factors bind with at least the given significance value.	0.001	0.00090 - 0.00115
	Module p-value cutoff	The cutoff for the combined P-value for a module	0.01	0.0088 - 0.0115
	Number in core cutoff	The minimum number of genes required to extend the core profile into a module	5	1 - 7
ReMoDiscovery	Expression correlation threshold	The minimum correlation required between the expression profiles for all the genes in a seed module	0.75	0.72 - 0.78
	Motif/regulator threshold	Motifs/regulators with a p-value above this threshold are considered significant	0.99	0.989 - 0.991 (regulator threshold)
	Motif/regulator support	The minimum number of motifs/regulators required to identify a seed module	2	2-2[see 20] (regulator threshold)

The table describes the properties of user-definable parameters for different module inference algorithms mentioned in the text. Remark that Xu *et al*. [59] was not included as the software was not available. *Algorithm*: name of the algorithm; *Parameters*: list of different user-definable parameters; *Description*: a short description of each user definable parameter; *Default value*: the parameter setting as used in the original study; *Robustness boundary*: the parameter range for which at least 80% of the generated modules are sufficiently similar to at least one of the modules generated by the default settings. Sufficiently similar in this context is defined as follows: at least 80% of the genes of both modules overlap and the regulatory program maximally differ in 20% of the regulators (so modules with less than 5 regulators must have the same set of regulators to be considered similar). Datasets used for benchmarking tested were derived from gene expression data of Gasch *et al*. [11] and ChIP-chip data of Harbison *et al*. [55] of which for both datasets only the first 50 columns were used. Only genes without missing values for the expression- and ChIP-chip data were retained. *ReMoDiscovery* was used without motif data.

available and knowledge about the true underlying biological networks is often incomplete. As a consequence, algorithm validation is often limited to confirming previously known interactions in the reconstructed network when performed on experimentally obtained data. However, estimating the number of false positive interactions predicted by the algorithm is very difficult, since there are no databases in which *absent* interactions are described. The only proper validation strategy is to perform wet-lab experiments on a sufficiently large set of predicted interactions and predicted absent interactions to confirm or deny the presumed interaction. It is clear that such an approach is impractical, time-consuming and sometimes infeasible. Moreover, it is highly desirable to repeatedly test an algorithm with different parameter settings and on different datasets. Usually only data about a few networks are available, which might bias algorithm design and validation towards the specificities of these datasets and networks.

Due to these limitations of real experimental data, the use of simulated data for validating inference algorithms has gained much interest. Simulated data will never cover all the intricacies of real experimental data, but at least such data can be used to unveil some qualitative properties of the algorithm under test (e.g. noise robustness, sensitivity, optimality of the proposed solution) and to tune the parameter settings to some extent. Several efforts have been made to generate data that mimic true experimental data. Simulating data implies creating realistic network topologies consisting of nodes and edges that represent the genetic entities and interactions among them. For producing topologies or network structures of large networks comprising thousands of nodes, approaches using random graph models or based on sampling substructures from previously described networks have been used [64-66]. Each of these models create graph structures that share at least one topological property with known regulatory networks, like scale-free [67] and small-world properties [68].

Secondly, transition functions need to be defined that model the interactions between the nodes. Boolean [69,70], continuous [66,71,72], and probabilistic [66] models have been proposed. Most current network simulators [64,66,71-74] use a set of ordinary differential equations (ODE's) based on Michaelis-Menten or Hill-like reaction kinetics [75]. Good generators are able to generate networks with differing structures and interaction types to prevent overfitting of the algorithms towards specific properties of the simulated data (e.g. exclusively linear interactions).

CONCLUSIONS

Network inference has been the subject of intensive research during the last five years. Only recently, with the advent of high-throughput data other than microarrays and with the increasing interest in designing biologically relevant rather than mathematically innovative solutions, computational methods emerge that allow tackling realistic situations. The focus is increasingly shifting towards integrative approaches dealing with several data sources simultaneously and based on biologically realistic simplifications. However, the problem is still far from being

[19] For *SAMBA* only the parameters accessible from the user interface were discussed.

[20] Running time for regulator support = 1 was >10h, the algorithm was terminated before finishing. For regulator support (Rs) = 3, only 37% of the modules had a similar module under the default parameter settings (Rs=2). However, the remaining 63% of the modules was related to a module in the reference set for which the set of genes was a superset of the genes in the module for Rs=3 and for which the set of regulators was a subset of the regulators of the regulators genes in the module for Rs=3.

solved. The framework of most of the methods developed so far is rather specifically designed for the problem to be solved and is difficult to extend with other types of data sources and with data relating to aspects of the regulatory network other than the transcriptional aspects. Also relatively little effort has been put in developing methods for experiment design. Considering the tedious and expensive nature of high-throughput data, methods that predict the next most informative experiments based on a previous set of data would be very valuable to molecular biologists. Last but not least, benchmarking will become increasingly important to assess and tune these algorithms before they can be used in daily practice. From this perspective, the use of simulated data, although only roughly approximating biological reality, will be essential. Considering the importance of data integration in systems biology, there is a need for a simulator that produces biologically plausible, heterogeneous data sets. Taken together, in view of the recent breakthroughs and the open research challenges, network or module inference offers an intriguing research field and holds much promise for many novel and exciting discoveries in biology.

ACKNOWLEDGEMENTS

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, G.0241.04 and G.0413.03; 4. IUAP V-22 (2002-2006), 5. K.U.Leuven, COE EF/05/007 SymBioSys.

REFERENCES

[1] Greenbaum D, Luscombe NM, Jansen R, Qian J, Gerstein M. Interrelating different types of genomic data, from proteome to secretome: 'oming in on function. *Genome Res* **2001**; 11: 1463-1468.

[2] Bader GD, Heilbut A, Andrews B, Tyers M, Hughes T, Boone C. Functional genomics and proteomics: charting a multidimensional map of the yeast cell. *Trends Cell Biol* **2003**; 13: 344-356.

[3] Blais A, Dynlacht BD. Constructing transcriptional regulatory networks. *Genes Dev* **2005**; 19: 1499-1511.

[4] Papin JA, Hunter T, Palsson BO, Subramaniam S. Reconstruction of cellular signalling networks and analysis of their properties. *Nat Rev Mol Cell Biol* **2005**; 6: 99-111.

[5] Chua G, Robinson MD, Morris Q, Hughes TR. Transcriptional networks: reverse-engineering gene regulation on a global scale. *Curr Opin Microbiol* **2004**; 7: 638-646.

[6] Ehrenberg M, Elf J, Aurell E, Sandberg R, Tegner J. Systems biology is taking off. *Genome Res* **2003**; 13: 2377-2380.

[7] Wei GH, Liu DP, Liang CC. Charting gene regulatory networks: strategies, challenges and perspectives. *Biochem J* **2004**; 381: 1-12.

[8] Hartwell LH, Hopfield JJ, Leibler S, Murray AW. From molecular to modular cell biology. *Nature* **1999**; 402: C47-C52.

[9] Ihmels J, Friedlander G, Bergmann S, Sarig O, Ziv Y, Barkai N. Revealing modular organization in the yeast transcriptional network. *Nat Genet* **2002**; 31: 370-377.

[10] Leung YF, Cavalieri D. Fundamentals of cDNA microarray data analysis. *Trends Genet* **2003**; 19: 649-659.

[11] Gasch AP, Spellman PT, Kao CM, *et al*. Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell* **2000**; 11: 4241-4257.

[12] D'haeseleer P, Liang S, Somogyi R. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* **2000**; 16: 707-726.

[13] van Someren EP, Wessels LF, Backer E, Reinders MJ. Genetic network modeling. *Pharmacogenomics* **2002**; 3: 507-525.

[14] de Jong H. Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol* **2002**; 9: 67-103.

[15] Wei C, Li J, Bumgarner RE. Sample size for detecting differentially expressed genes in microarray experiments. *BMC Genomics* **2004**; 5: 87.

[16] Elowitz MB, Levine AJ, Siggia ED, Swain PS. Stochastic gene expression in a single cell. *Science* **2002**; 297: 1183-1186.

[17] Rao CV, Wolf DM, Arkin AP. Control, exploitation and tolerance of intracellular noise. *Nature* **2002**; 420: 231-237.

[18] Segal E, Shapira M, Regev A, *et al*. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet* **2003**; 34: 166-176.

[19] Li H, Wang W. Dissecting the transcription networks of a cell using computational genomics. *Curr Opin Genet Dev* **2003**; 13: 611-616.

[20] Ihmels J, Bergmann S, Barkai N. Defining transcription modules using large-scale gene expression data. *Bioinformatics* **2004**; 20: 1993-2003.

[21] Brazma A, Vilo J. Gene expression data analysis. *Microbes Infect* **2001**; 3: 823-829.

[22] Koller D, Pfeffer A, Probabilistic Frame-Based Systems, In: Proc AAAI, AAAI Press, Hadison, WI 1998; 580-587.

[23] Friedman N, Getoor L, Koller D, Pfeffer A. Learning Probabilistic Relational Models. **1999**; 1300-1309.

[24] Getoor L, Friedman N, Koller D, Taskar B. Learning probabilistic models of link structure. *J Mach Learn Res* **2002**; 3: 679-707.

[25] Getoor L, Friedman N, Koller D, Taskar B, Learning Probabilistic Models of Relational Structure, In: Proc. ICML, Morgan Kaufmann, San Francisco, CA, 2001; 170-177.

[26] Friedman N. Inferring cellular networks using probabilistic graphical models. *Science* **2004**; 303: 799-805.

[27] Herrgard MJ, Covert MW, Palsson BO. Reconciling gene expression data with known genome-scale regulatory network structures. *Genome Res* **2003**; 13: 2423-2434.

[28] Pollack JR, Iyer VR. Characterizing the physical genome. *Nat Genet* **2002**; 32 Suppl: 515-521.

[29] Buck MJ, Lieb JD. ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics* **2004**; 83: 349-360.

[30] Luscombe NM, Babu MM, Yu H, Snyder M, Teichmann SA, Gerstein M. Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature* **2004**; 431: 308-312.

[31] Cawley S, Bekiranov S, Ng HH, *et al*. Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding RNAs. *Cell* **2004**; 116: 499-509.

[32] Herring CD, Raffaelle M, Allen TE, *et al*. Immobilization of Escherichia coli RNA polymerase and location of binding sites by use of chromatin immunoprecipitation and microarrays. *J Bacteriol* **2005**; 187: 6166-6174.

[33] Kellis M, Patterson N, Endrizzi M, Birren B, Lander ES. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature* **2003**; 423: 241-254.

[34] Cliften P, Sudarsanam P, Desikan A, *et al*. Finding functional features in *Saccharomyces* genomes by phylogenetic footprinting. *Science* **2003**; 301: 71-76.

[35] Rombauts S, Florquin K, Lescot M, Marchal K, Rouze P, Van de PY. Computational approaches to identify promoters and cis-regulatory elements in plant genomes. *Plant Physiol* **2003**; 132: 1162-1176.

[36] Beer MA, Tavazoie S. Predicting gene expression from sequence. *Cell* **2004**; 117: 185-198.

[37] Wingender E, Chen X, Hehl R, *et al*. TRANSFAC: an integrated system for gene expression regulation. *Nucleic Acids Res* **2000**; 28: 316-319.

[38] Salgado H, Gama-Castro S, Martinez-Antonio A, *et al*. RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12. *Nucleic Acids Res* **2004**; 32: D303-D306.

[39] Aebersold R, Mann M. Mass spectrometry-based proteomics. *Nature* **2003**; 422: 198-207.

[40] Phizicky E, Bastiaens PI, Zhu H, Snyder M, Fields S. Protein analysis on a proteomic scale. *Nature* **2003**; 422: 208-215.

[41] Salwinski L, Miller CS, Smith AJ, Pettit FK, Bowie JU, Eisenberg D. The Database of Interacting Proteins: 2004 update. *Nucleic Acids Res* **2004**; 32: D449-D451.

[42] Alfarano C, Andrade CE, Anthony K, *et al*. The Biomolecular Interaction Network Database and related tools 2005 update. *Nucleic Acids Res* **2005**; 33: D418-D424.

[43] Hartemink AJ, Gifford DK, Jaakkola TS, Young RA. Combining location and expression data for principled discovery of genetic regulatory network models. *Pac Symp Biocomput* **2002**; 437-449.

[44] Imoto S, Higuchi T, Goto T, Tashiro K, Kuhara S, Miyano S. Combining microarrays and biological knowledge for estimating gene networks *via* bayesian networks. *J Bioinform Comput Biol* **2004**; 2: 77-98.

[45] Hartemink AJ, Segal E. Joint learning from multiple types of genomic data. *Pac Symp Biocomput* **2005**; 445-446.

[46] Lee PH, Lee D. Modularized learning of genetic interaction networks from biological annotations and mRNA expression data. *Bioinformatics* **2005**; 21: 2739-2747.

[47] Tamada Y, Kim S, Bannai H, *et al*. Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics* **2003**; 19 Suppl 2: II227-II236.

[48] Bernard A, Hartemink AJ. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. *Pac Symp Biocomput* **2005**; 459-470.

[49] Gao F, Foat BC, Bussemaker HJ. Defining transcriptional networks through integrative modeling of mRNA expression and transcription factor binding data. *BMC Bioinformatics* **2004**; 5: 31.

[50] Lee TI, Rinaldi NJ, Robert F, *et al*. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* **2002**; 298: 799-804.

[51] Wang W, Cherry JM, Botstein D, Li H. A systematic approach to reconstructing transcription networks in *Saccharomyces cerevisiae*. *Proc Natl Acad Sci USA* **2002**; 99: 16893-16898.

[52] Bussemaker HJ, Li H, Siggia ED. Regulatory element detection using correlation with expression. *Nat Genet* **2001**; 27: 167-171.

[53] Spellman PT, Sherlock G, Zhang MQ, *et al*. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell* **1998**; 9: 3273-3297.

[54] Hughes TR, Marton MJ, Jones AR, *et al*. Functional discovery *via* a compendium of expression profiles. *Cell* **2000**; 102: 109-126.

[55] Harbison CT, Gordon DB, Lee TI, *et al*. Transcriptional regulatory code of a eukaryotic genome. *Nature* **2004**; 431: 99-104.

[56] Kato M, Hata N, Banerjee N, Futcher B, Zhang MQ. Identifying combinatorial regulation of transcription factors and binding motifs. *Genome Biol* **2004**; 5: R56.

[57] Bar-Joseph Z, Gerber GK, Lee TI, *et al*. Computational discovery of gene modules and regulatory networks. *Nat Biotechnol* **2003**; 21: 1337-1342.

[58] De Bie T, Monsieurs P, Engelen K, De Moor B, Cristianini N, Marchal K. Discovering regulatory modules from heterogeneous information sources. *Pac Symp Biocomput* **2005**; 483-94.

[59] Xu X, Wang L, Ding D. Learning module networks from genome-wide location and expression data. *FEBS Lett* **2004**; 578: 297-304.

[60] Tanay A, Sharan R, Kupiec M, Shamir R. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc Natl Acad Sci USA* **2004**; 101: 2981-2986.

[61] Tanay A, Sharan R, Shamir R. Discovering statistically significant biclusters in gene expression data. *Bioinformatics* **2002**; 18 Suppl 1: S136-S144.

[62] Tanay A, Steinfeld I, Kupiec M, Shamir R. Integrative analysis of genome-wide experiments in the context of a large high-throughput data compendium. *Mol Sys Biol* **2005**; msb4100005: E1-E10.

[63] Xenarios I, Eisenberg D. Protein interaction databases. *Curr Opin Biotechnol* **2001**; 12: 334-339.

[64] Knüpfer C, Dittrich P, Beckstein C. Artificial Gene Regulation: A Data Source for Validation of Reverse Bioengineering. In: Proc. GWAL6, Akademische Verlagsgesellschaft Aka, Berlin **2004**; 66-75.

[65] Van den Bulcke T, Van Leemput K, Naudts B, van Remortel P, Ma H, Verschoren A, De Moor B, Marchal K. SynTReN: a generator of synthetic expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics* **2006**; 7:43.

[66] Mendes P, Sha W, Ye K. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics* **2003**; 19 Suppl 2: II122-II129.

[67] Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. *Nature* **1998**; 393: 440-442.

[68] Albert R, Barabasi AL. Topology of evolving networks: local events and universality. *Phys Rev Lett* **2000**; 85: 5234-5237.

[69] Akutsu T, Miyano S, Kuhara S. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pac Symp Biocomput* **1999**; 17-28.

[70] Reil T. Dynamics of Gene Expression in an Artificial Genome - Implications for Biological and Artificial Ontogeny. *Eur Conf Artific Life* **1999**; 457-466.

[71] Husmeier D. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics* **2003**; 19: 2271-2282.

[72] Zak DE, Doyle FJ, Schwaber JS. Simulation Studies for the Identification of Genetic Networks from cDNA Array and Regulatory Activity Data. *Proc Sec Int Conf Sys Biol* **2001**; 231-238.

[73] Smith VA, Jarvis ED, Hartemink AJ. Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics* **2002**; 18 Suppl 1: S216-S224.

[74] Bono H, Okazaki Y. Functional transcriptomes: comparative analysis of biological pathways and processes in eukaryotes to infer genetic networks among transcripts. *Curr Opin Struct Biol* **2002**; 12: 355-361.

[75] Mendes P, Kell DB. MEG (Model Extender for Gepasi): a program for the modelling of complex, heterogeneous, cellular systems. *Bioinformatics* **2001**; 17: 288-289.

[76] Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA* **1998**; 95: 14863-14868.

[77] Gasch AP, Eisen MB. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biol* **2002**; 3: 1-22.

[78] Cheng Y, Church GM. Biclustering of Expression Data. *Proceedings ISMB*, AAAI Press **2000**; 93-103.

[79] Getz G, Levine E, Domany E. Coupled two-way clustering analysis of gene microarray data. *Proc Natl Acad Sci USA* **2000**; 97: 12079-12084.

[80] Kluger Y, Basri R, Chang JT, Gerstein M. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res* **2003**; 13: 703-716.

[81] Sharan R, Maron-Katz A, Shamir R. CLICK and EXPANDER: a system for clustering and visualizing gene expression data. *Bioinformatics* **2003**; 19: 1787-1799.

[82] Winderickx J, Taylor PM. Nutrient responses in eukaryotic cells, Springer, Berlin Heidelberg **2004**.

[83] Simon I, Barnett J, Hannett N, *et al*. Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell* **2001**; 106: 697-708.

Received: August 19, 2005 Revised: November 4, 2005 Accepted: November 16, 2005