INFERRING GENE REGULATORY NETWORKS BY MACHINE LEARNING METHODS

JOCHEN SUPPER, HOLGER FRÖHLICH, CHRISTIAN SPIETH, ANDREAS DRÄGER,
ANDREAS ZELL*

Centre for Bioinformatics Tübingen (ZBIT), Sand 1, 72076 Tübingen
jochen.supper@uni-tuebingen.de

The ability to measure the transcriptional response after a stimulus has drawn much attention to the underlying gene regulatory networks. Several machine learning related methods, such as Bayesian networks and decision trees, have been proposed to deal with this difficult problem, but rarely a systematic comparison between different algorithms has been performed. In this work, we critically evaluate the application of multiple linear regression, SVMs, decision trees and Bayesian networks to reconstruct the budding yeast cell cycle network. The performance of these methods is assessed by comparing the topology of the reconstructed models to a validation network. This validation network is defined *a priori* and each interaction is specified by at least one publication. We also investigate the quality of the network reconstruction if a varying amount of gene regulatory dependencies is provided *a priori*.

1. Introduction

Transcriptional data sets provide valuable insight to cellular processes under various conditions. These data sets can be analyzed by cluster analysis, thereby providing undirected gene relations. In order to model gene regulatory networks (GRN) directed relations between genes have to be considered. Most GRNs can be represented as interconnections between genes, each indicating that one gene influences the expression of another gene.

Today, modeling of GRNs is guided by a rich flow of experimental data. The stream is still widened by an increasing pool of measurement techniques. Despite of all this information, detailed knowledge regarding network models is still almost exclusively collected by biologists. They collect and integrate data, expand and refine their models and finally validate them. For our modeling efforts, we will concentrate on the regulatory information that can be extracted solely from transcriptional response data. The restriction to transcriptional response data provides us with a large number of measured genes along with a small sampling rate. This, of course, leads to a high level of ambiguity for every GRN reconstruction method.

Several approaches for GRN reverse engineering have emerged during the last years. These approaches include analytical methods such as Boolean networks[12], (non)-linear

*This work was supported by the National Genome Research Network (NGFN II) of the Federal Ministry of Education and Research in Germany under contract number 0313323.

networks[19] and differential equations[3] but also machine learning methods such as decision trees[16] and Bayesian networks[9].

To reconstruct a GRN, a set of transcriptional response measurements has to be available. Given this data, one of the above mentioned reconstruction methods may be employed to untangle the underlying topological structure of the interaction network. One problem thereby is that it is very hard to validate the performance of the proposed approaches. This makes it difficult to compare methods and even more to judge, if certain approaches are helpful at all. In previous publications GRN models have been validated mostly by co-citation[16] or on artificial data[18]. Despite these efforts no general validation method has emerged.

In this work, we present one network that has been investigated thoroughly and where the interactions are known in many cases. This network is a subset containing 20 genes involved in the budding yeast cell cycle defined by Chen *et al.*[2] for which Spellman *et al.*[17] and Cho *et al.*[4] publicly provide time-series measurement data. This enables us to build a validation network, for which the interactions can be specified. Additionally, it allows us to systematically investigate, how prior knowledge on parts of the networks changes the validity of results obtained by an automatic GRN reconstruction. Thereby, we concentrate on machine learning methods, such as Bayesian networks[8], multiple linear regression, CART decision trees[1] and SVMs[5]. For the last three we closely follow the framework proposed by Soinov *et al.*[16]. They used a so-called wrapper approach[11] in combination with decision trees to learn the minimal subsets of genes, which best predict the up/down regulation of a considered gene. By comparing these different approaches we build upon the work of Husmeier *et al.*[9], who performed a sensitivity and specificity analysis of GRN reconstruction for Bayesian networks.

2. Materials and Methods

2.1. *Data*

2.1.1. *Budding Yeast Cell Cycle*

The biological model used for this research is the budding yeast cell cycle, which has been thoroughly investigated over many years. Cho *et al.*[4] and Spellman *et al.*[17] contributed to these investigations by publicly providing a large transcriptional data set. They measured the progression of the cell cycle with different synchronization techniques. Altogether this results in 73 time point measurements. These measurements were performed on microarrays[6], each consisting of 6178 data points, from which we select a subset containing 20 genes. This is done according to Chen *et al.*[2], who did an extensive literature search to set up a system of differential equations to define the topology of the GRN. In addition to the interactions provided by the differential equations we searched TRANSFAC[20], Entrez Gene[13] and the *Saccharomyces* Genome Database (SGD[10]) for known dependencies between a pair of genes. The entire network contains 56 interactions and is depicted in Figure 1. It will serve as our validation network for the studies performed in this paper. Although this validation network might contain some false interactions or others, which were not active at the time the measurements were taken, we can nevertheless rank our reconstructions with regard to their closeness to this network. That means, the closeness of an inferred GRN to the validation network should not be understood in an absolute, but in a relative sense.

2.1.2. *Preprocessing and Availability of the Data*

The data set is normalized by the average $\log_2$ ratio, which implicitly describes a non-linear relationship between the genes. We also performed pre-experiments without normalization and with normalization through a sigmoidal function, but found the results to be inferior.

The data set as well as the described models are all available from public data sources. An SBML version of the topological validation network is available on our homepage[a].

2.2. ***Machine Learning Methods for GRN Reconstruction***

Our starting point is a gene-expression matrix $\mathbf{X} \in \mathbb{R}^{20\times 73}$, where each row represents a gene and each column represents a sample taken at a specific time step. That means, an element X_{ij} of $\mathbf{X}$ indicates the expression level of gene i in sample j. We consider Bayesian networks (BN), multiple linear regression (MLR), CART decision trees (CART) and Support Vector Machines (SVMs) for GRN reconstruction from this data.

2.2.1. *Bayesian Networks*

For learning the GRN with a BN we discretized the data in the following way: For each gene i we distinguish only the two states "expressed above average" and "expressed below average". That means we transform each entry X_{ij} to an entry Y_{ij} , which is defined as:

$$Y_{ij} := \begin{cases} 1 \; X_{ij} \geq \bar{X}_i, \text{ where } \bar{X}_i \text{ is the average} \\ \quad \text{expression level of gene } i \\ 0 \text{ otherwise} \end{cases}$$

This is done in accordance to Soinov *et al.*[16]. We then learn the structure of a dynamic BN using a MCMC search in the structure space as proposed by Husmeier *et al.*[9]. Thereby we use the MATLAB$^{\text{TM}}$ code provided on his homepage[b]. After training the dynamic BN we construct a GRN by only considering those dependencies, for which the expected posterior probability is above average.

2.2.2. *Multiple Linear Regression*

The GRN reconstruction by means of MLR resembles the framework by Soinov *et al.*[16]: For each gene i we identify two prediction problems:

[a] www-ra.informatik.uni-tuebingen.de/mitarb/supper/ml/
[b] http://www.bioss.sari.ac.uk/∼dirk

- The prediction of the expression of gene i in sample j from all other genes in sample j.
- The prediction of the expression of gene i in sample j from all other genes in sample $j-1$.

In both cases we search for a minimal combination of genes that allows to predict the expression of gene i reliably. This is achieved by considering only those genes, for which the Pearson correlation of the expression level with gene i is at least 60 %. These genes are subsequently selected to train a MLR model that predicts the expression level of gene i. If the 10-fold cross-validated mean correlation of the model output with the true expression level of gene i is above 60 %, then the MLR model is considered as reliable and the selected genes are considered as probable regulators for gene i in the GRN reconstruction. The bottom line is that the MLR reconstruction can be viewed as a correlation network with directed edges.

2.2.3. *Decision Trees*

In case of the GRN inference by means of CART we formulate the two prediction tasks from the last paragraph as classification rather than regression tasks. This allows to follow the framework by Soinov *et al.* directly. More specifically, we now have three prediction problems instead of the two stated above:

- the prediction of the state Y_{ij} of gene i in sample j from the expression levels of all other genes in sample j.
- the prediction of the state Y_{ij} of gene i in sample j from the expression levels of all other genes in sample $j-1$.
- the prediction of the *change* of state Y_{ij} of gene i in sample j from the state changes of all other genes in sample j.

The change of state Y_{ij} is either "equal", "regulated up" or "regulated down". That means in the first two cases we have a binary and in the last one a three-class classification problem. We use the CART implementation provided in the MATLAB$^{\text{TM}}$ 7.0 statistics toolbox with pruning turned on and the Gini diversity index as node split criterion. This way selecting a good combination of genes, which allow forecasting the state of gene i reliably, is embedded into the learning of the decision tree. Similar to above, we set an accuracy threshold of 75% beyond which we consider the predictions made by the CART model as acceptable.

2.2.4. *Support Vector Machines*

SVMs have attracted a high interest within the bioinformatics community during the last years due to their good prediction performance for various tasks. They rely on principles from statistical learning theory[15]. The idea is to construct an optimal hyperplane between two classes +1 and -1 such that the margin, i.e. the distance of the hyperplane to the point closest to it, is maximized. To allow for nonlinear classification, so-called kernel functions are employed, which can be thought of as special similarity measures. They implicitly map the original data into some high dimensional feature space, in which the optimal hyperplane can be found.

In our case we consider linear kernels $k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$ as well as polynomial kernels of degree 2 $k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^2$, where $\mathbf{x}$ and $\mathbf{x}'$ are the expression levels of all genes except of gene i in sample j. The polynomial kernel implicitly computes all pairwise products between expression levels of two genes. This way not only linear, but also nonlinear dependencies between gene expressions can be captured.

In addition to a kernel function, a soft margin parameter C has to be fixed. In our case we choose C from the grid $2^{-2}...2^{14}$ by means of 5-fold cross-validation. To determine for each gene i, which genes are suited best to predict its state, we employ the RFE[7]. This algorithm successively eliminates that gene, which influences the size of the margin least. The termination of this procedure is determined by an additional 10-fold cross-validation.

Of course, a direct comparison of the different methods for GRN reconstruction introduced is not unproblematic, because each algorithm depends on certain parameter settings and different data formats are in use. Nevertheless, we think that a comparative study, even if we should not forget about its limitations, might be useful to gain some insights.

2.3. *Network Validation*

2.3.1. *Statistical Stability of The Solution*

For validating all of the above approaches we are interested in those parts of the true network, which are reconstructed in a *statistically stable* way by each single method. That means, we are interested in those inferred gene regulatory dependencies, which are not sensitive to the respective training data, but to the underlying biological process. For this purpose we use 10-fold cross-validation: We randomly split the measurement data into 10 parts, train our model on 9 parts and then test the model on the remaining part. This procedure is iterated until each part is left out exactly once for testing. At the end we only use those connections consistently inferred during the 10-fold cross-validation. Hence, the resulting network can be seen as a consensus model, integrating results from different data splits.

2.3.2. *Validating the Topology*

We validate the network topologies obtained from the different consensus models for each algorithm by calculating the following statistics:

(1) the fraction of correctly identified edges in the validation network (*recovered* connections)
(2) the fraction of correctly constructed edges in relationship to all constructed edges (*direct* connections)
(3) the fraction of constructed edges connecting genes with topological distance 2 in relationship to all constructed edges (*indirect* connections): If in the validation

network we have $a \to b \to c$ and we reconstruct $a \to c$, then the included edge directly models an indirect regulatory influence.

(4) the same as 3. with distances > 2 (*spurious* connections).

(5) the graph edit distance (*GED*) between the constructed and the correct GRN: The graph edit distance describes the minimal number of edit operations (edge insertion/edge removal) that transform one graph into another one. In our case we use the algorithm by Kelly *et al.*[14] to calculate this distance.

The first statistic can be viewed as a sensitivity measure for the GRN reconstruction algorithm, whereas 2. - 4. describe the specificity. Discriminating between different types of inferred edges here seems beneficial, because it allows a better insight into the quality of the reconstruction. The graph edit distance, on the other hand, is a combined statistic capturing both, the number of correctly recovered dependencies and the number of inferred edges, which do not exist in the validation network. We think that the discrimination between sensitivity and specificity of the GRN reconstrucion is necessary, because, there is a trade-off between the fraction of correctly identified relations in the validation network and the fraction of all inferred connections, which are correct. A maximization of the first goal could be achieved trivially by connecting every gene in the network, which would be rather naive. In contrast, a pure maximization of the second goal would lead to an edge free graph, obviously containing no false connection. A good GRN reconstruction should therefore find a fair balance between a high number of inferred edges and a low number of spurious connections.

3. Results

3.1. *Comparison of Different Methods*

We validated the machine learning methods introduced in the last section as described above. The results in Table 1 show a relatively low statistical stability for all methods. The Bayesian network reconstruction (Fig. 1) leads to the lowest number of inferred edges (only 7) and hence to a very low sensitivity (only 1 connection of the true network was recovered). At the same time the fraction of indirect and spurious dependencies among these 7 was relatively high. Also the graph edit distance was the highest among all methods.

MLR, CART, linear and polynomial SVMs all recovered a substantially higher number of relations of the validation network. Thereby the recovery rate of the validation network was clearly highest for the linear SVM reconstruction and second highest for the MLR reconstruction. At the same time the fraction of direct connections in the inferred GRN was highest for the MLR reconstruction. The fraction of indirect connections was highest in the CART model. The linear SVM had the highest fraction of spurious edges. However, at the same time the graph edit distance was lowest for this model and second lowest for the polynomial SVM model.

All in all we observe that the dynamic BN is outperformed by the other methods. Among these we favor the linear SVM model, since it has the lowest graph edit distance, indicating a fair trade-off between the number of recovered edges from the true validation

Table 1. Validation of the GRN reconstruction with different methods. All statistics in % (see subsection 2.3 for explanation).

	BN	**MLR**	**CART**	**lin. SVM**	**poly. SVM**
recov.	1.79	7.14	3.57	10.71	5.36
direct	14.29	25.0	18.18	16.67	15.79
indir.	42.86	43.75	54.55	38.89	31.58
spur.	42.86	31.25	27.27	44.44	36.84
GED	35	32	32	**30**	31

network and spurious or false inferred dependencies.

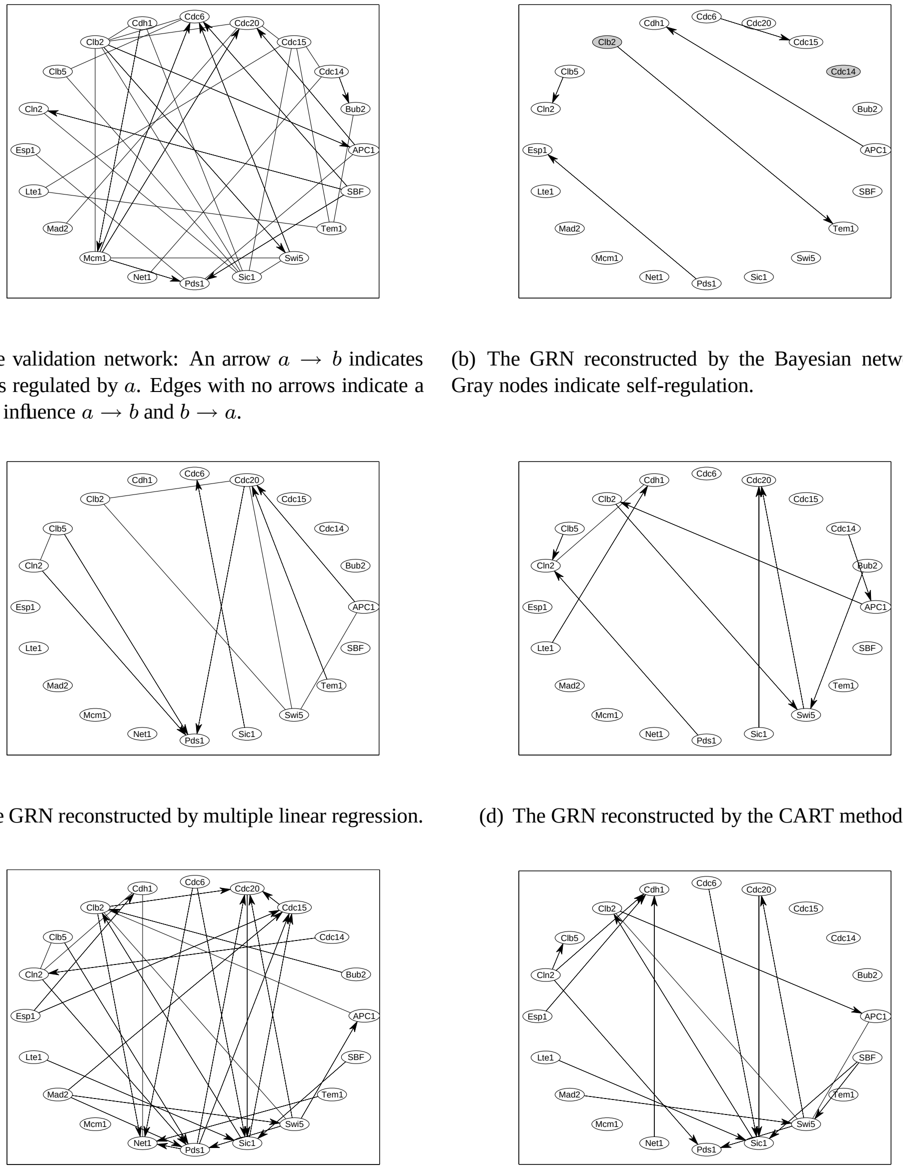
3.2. *Effect of Prior Knowledge*

In a second study we concentrated on the linear SVM reconstruction method and investigated the influence of incorporating prior knowledge of certain relations in the GRN. For this purpose we modified the procedure described in the last section such that for a gene k, which is known to influence gene i, the influence on the margin is explicitly set to ∞. This way the RFE algorithm is forced to rank such a gene highest. Furthermore, known relations are drawn in the GRN even if the classification accuracies for gene i are below the prescribed threshold of 75%.

In Figure 2a we depict the influence of prior knowledge on the sensitivity and specificity statistics, if 10, 20, ..., 50% randomly selected relations of the validation network are known. The results are averaged over 10 trials. As one can see the number of recovered edges increases in a piecewise linear fashion with the increase of the prior knowledge. The increase from no prior knowledge to 10% known edges is higher than e.g. from 10% to 20%. While at the beginning we have a gain of almost 15%, thereafter the gain decreases to around 10% only and is hence at the same level as the number of edges additionally provided by prior knowledge. The fraction of direct edges increases parallel to the fraction of recovered edges. The fraction of indirect and spurious connections decreases in a roughly linear fashion with the fraction of known relations. While at the beginning the largest fraction of all constructed edges is spurious, with 20% knowledge it is roughly at same level and with 30% below the fraction of direct relations.

As expected, the total number of relations in the inferred GRN, which also includes the connections drawn by prior knowledge, increases with the fraction of known edges (Fig. 2b). In contrast, the number of newly inferred edges decreases with the increase of prior knowledge, which seems surprising at the first glance. However, this phenomenon might be due to a lower number of missing edges with an increasing number of already known ones. In the graphs from Figure 2a we see the same piecewise linear behavior as in Figure 2b. Again, the increase from 0 to 10% prior knowledge has a higher impact than

Figure 1. Literature network and reconstructions by different methods.



(a) The validation network: An arrow $a \longrightarrow b$ indicates that b is regulated by a. Edges with no arrows indicate a mutual influence $a \longrightarrow b$ and $b \longrightarrow a$.

(b) The GRN reconstructed by the Bayesian network. Gray nodes indicate self-regulation.

(c) The GRN reconstructed by multiple linear regression.

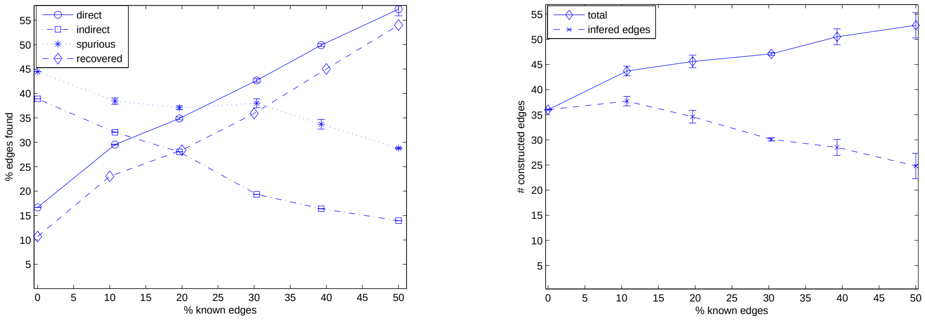
(d) The GRN reconstructed by the CART method.

(e) The GRN reconstructed by the linear SVM method.

(f) The GRN reconstructed by the polynomial SVM method.

e.g. from 20 to 30%.

Figure 2. Effect of incorporating prior knowledge on randomly selected edges



(a) Effect of incorporating prior knowledge on randomly selected edges: sensitivity and specificity statistics. Average values over 10 trials (± std. dev.).

(b) Effect of incorporating prior knowledge on randomly selected edges: total number and number of newly inferred edges. Average values over 10 trials (± std. dev.).

4. Conclusion

In this paper we systematically compared different machine learning methods for GRN reconstruction. We considered Bayesian networks, multiple linear regression, CART decision trees, linear and nonlinear SVMs. We developed a framework for evaluating the inference methods with regard to their statistical stability by using 10-fold cross-validation. A well investigated biological data set from the literature served as our basis. This enabled us construct a validation network against which we could compare our results by means of sensitivity and specificity analysis. We found linear SVMs to produce slightly superior reconstructions compared to the other methods, especially to dynamic Bayesian networks. Thereby the inference scheme closely followed the method proposed by Soinov *et al.*[16] for decision trees. However, it has to be remarked that our results depend on the specific parameter and data format settings for the individual algorithms.

We additionally investigated the influence of prior knowledge on the quality of the learned network topology. We found that adding known connections has a relatively large positive impact, when no prior knowledge existed before, whereas if further increasing the prior knowledge the gain becomes smaller.

We think that the main benefit of this work lies in a first trial to compare different machine learning methods for GRN reconstruction in a systematic manner, which to our best knowledge has rarely been done before. In our future work we will extend our research on different data sets from the one used here and evaluate the results in a similar manner.

References

1. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1984.
2. Katherine C Chen, Laurence Calzone, Attila Csikasz-Nagy, Frederick R Cross, Bela Novak, and John J Tyson. Integrative analysis of cell cycle control in budding yeast. *Mol Biol Cell*, 15(8):3841–3862, Aug 2004.
3. Kuang-Chi Chen, Tse-Yi Wang, Huei-Hun Tseng, Chi-Ying F Huang, and Cheng-Yan Kao. A stochastic differential equation model for quantifying transcriptional regulatory network in Saccharomyces cerevisiae. *Bioinformatics*, 21(12):2883–90, Jun 2005.
4. R. J. Cho, M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart, and R. W. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell*, 2(1):65–73, Jul 1998.
5. C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273 – 297, 1995.
6. M. B. Eisen and P. O. Brown. DNA arrays for analysis of gene expression. *Methods Enzymol*, 303:179–205, 1999.
7. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46:389 – 422, 2002.
8. D. Heckerman. A tutorial on learning with bayesian networks. *Data Mining and Knowledge Discovery*, 1:79 – 119, 1997.
9. Dirk Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19(17):2271–2282, Nov 2003.
10. Laurie Issel-Tarver, Karen R Christie, Kara Dolinski, Rey Andrada, Rama Balakrishnan, Catherine A Ball, Gail Binkley, Stan Dong, Selina S Dwight, Dianna G Fisk, Midori Harris, Mark Schroeder, Anand Sethuraman, Kane Tse, Shuai Weng, David Botstein, and J. Michael Cherry. Saccharomyces Genome Database. *Methods Enzymol*, 350:329–346, 2002.
11. R. Kohavi and G. John. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97(12):273 – 324, 1997.
12. S. Liang, S. Fuhrman, and R. Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac Symp Biocomput*, 1:18–29, 1998.
13. Donna Maglott, Jim Ostell, Kim D Pruitt, and Tatiana Tatusova. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res*, 33(Database issue):D54–D58, Jan 2005.
14. A. Robes-Kelly and E. Hancock. Edit distance from graph spectra. In *Proc. 9th IEEE Int. Conf. Comp. Vis.*, volume 1, pages 234 – 241, 2003.
15. B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
16. Lev A Soinov, Maria A Krestyaninova, and Alvis Brazma. Towards reconstruction of gene networks from expression data by supervised learning. *Genome Biol*, 4(1):R6, 2003.
17. P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, 9(12):3273–3297, Dec 1998.
18. Jochen Supper, Christian Spieth, and Andreas Zell. Reverse engineering non-linear gene regulatory networks based on the bacteriophage lambda ci circuit. In *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 2005.
19. D. C. Weaver, C. T. Workman, and G. D. Stormo. Modeling regulatory networks with weight matrices. *Pac Symp Biocomput*, 1:112–23, 1999.
20. Edgar Wingender. TRANSFAC, TRANSPATH and CYTOMER as starting points for an ontology of regulatory networks. *In Silico Biol*, 4(1):55–61, 2004.