The NetGenerator Algorithm: Reconstruction of Gene Regulatory Networks

Susanne Toepfer[1], Reinhard Guthke[2] Dominik Driesch[1], Dirk Woetzel[1], and Michael Pfaff[1]

[1] BioControl Jena GmbH, Wildenbruchstr. 15, D-07745 Jena, Germany
susanne.toepfer@biocontrol-jena.com
[2] Leibniz Institute for Natural Product Research and Infection Biology - Hans Knoell Institute, Beutenbergstr. 11a, D-07745 Jena, Germany

Abstract. Mathematical models of gene regulatory networks aim to capture the causal regulatory relationships by fitting the network models to the monitored time courses of gene expression levels. In this paper, the NetGenerator algorithm is presented that generates mathematical models in form of linear or nonlinear differential equation systems. The problem of finding the most likely interactions between genes is solved by a combinatorial search strategy and can be efficiently supported by the incorporation of available expert knowledge. Using favorable parameter identification methods from a system identification point of view allows to fit accurate and sparsely connected models. By the inclusion of higher order submodels, the algorithm enables the identification of gene-gene interactions with significantly time delayed gene regulation.

1 Introduction

Gene regulatory networks control biological functions by regulating the level of gene expression. Discovering and understanding of the complex causal relationships within gene regulatory networks has become a major goal of systems biology, computational biology and bioinformatics. Today, large-scale measurement technologies are opening novel possibilities for covering information about the regulatory mechanisms underlying specific biological processes as e.g. reactions on different developmental and environmental conditions. One example of large-scale measurement technologies are DNA microarray experiments that allow to obtain the output of gene regulatory networks by measuring the gene expression levels of thousands of genes. Gene expression time courses describe the temporal changes of expression levels that are caused by the dynamic nature of regulatory interactions. Analyzing those gene expression time series data by reverse engineering techniques allows to provide insight into the dynamic processes and to generate hypotheses of the causal structure of specific functional modules of gene regulatory networks.

Using data-driven reverse engineering techniques, structural information is typically inferred by firstly fitting the parameters of a given mathematical model to the available time series data and subsequently interpreting the resulting

model structure. In order to infer biologically meaningful models at least the following conditions have to be met:

- The mathematical model has to provide an acceptable simplification that leads to an adequate description of the regulatory processes for a certain level of abstraction.
- An appropriate identification algorithm must allow to reverse engineer gene regulatory networks by fitting the model output to the time series observations.
- The time series data have to cover the main regulatory effects of a considered gene regulatory network function with respect to both the gene expression levels and the relevant external input signals.

There exists a huge amount of model architectures and corresponding identification schemes for the data based reconstruction of gene regulatory networks. Each modeling approach emphasizes another aspect of the biological mechanisms. Well known mathematical models are e.g. directed graphs, Bayesian networks, differential equation systems, stochastic models, Boolean networks and rule-based models [1]. All theses models can be interpreted as networks of interacting nodes. Each node possesses a corresponding node function (e.g. conditional probability distribution, Boolean function, weighted sum) that processes the information that comes from other nodes or external inputs. In the model, the gene-gene interactions are represented by model parameters that determine the information processing between the nodes. While in principle the models allow the nodes to interact with each other one, it is known that in regulatory networks the genes interact with only a small number of other genes. Therefore, it is the general goal of the identification algorithm to estimate the small subset of *relevant model parameters* from the set of possible ones. The relevant parameters are those that are required to generate an adequate fit of the model output to the measured time courses. It is assumed that these relevant model parameters coincide with the gene-gene interactions of the underlying gene regulatory network.

In general, the results of data-based modeling critically depend on the quality and the quantity of the given measurement data. The data from microarray experiments are corrupted by measurement noise with unknown characteristics and unfavorable signal-to-noise ratios. Furthermore, because of the high costs of microarray experiments, the number of consecutive time points is still strongly restricted. On the other hand, there are hundreds or thousands of genes that are considered simultaneously. Therefore, without the inclusion of additional mathematical or biological constraints the relevant model parameters can not be uniquely estimated from the available data [2].

Possibilities to cope with this problem by the inclusion of mathematical constraints are first the resampling of time courses based on an interpolation of time series data [3] and second singular value decomposition based methods [4]. Biological constraints can be taken into account by clustering of co-expressed genes and the subsequent generation of network models based on clustered time

courses. Here, co-expressed genes are assumed to be co-regulated by the same processes [5–7]. Another biologically inspired approach is the application of combinatorial search strategies that directly base on the general knowledge of limited connectivity between genes [8, 9].

In this paper the NetGenerator algorithm is described that infers gene regulatory networks from gene expression time series data. The method uses a combinatorial search strategy to identify parsimonious models in form of linear or nonlinear differential equation systems (Sec. 2.1). Model structure (Sec. 2.2) and parameter identification (Sec. 2.3) are performed under special consideration of methods from system identification theory in order to reduce undesired effects from measurement noise. Searching for an appropriate model structure can be supported by the integration of available expert knowledge (Sec. 2.4). The identification algorithm allows the generation of models with a vary accurate fitting to the observed time courses while the models remain simple and interpretable. Section 3 gives an overview of some NetGenerator applications presented in former publications. The focus of this paper is a detailed description of the NetGenerator algorithm itself.

2 The NetGenerator Algorithm

2.1 Modeling Approach

The NetGenerator modeling approach bases on systems of either linear differential equations

$$\dot{x_i}(t) = \sum_{j=1}^{q_s} w_{i,j} x_j(t) + \sum_{l=1}^{p} b_{i,l} u_l(t), \tag{1}$$

or nonlinear differential equations

$$\dot{x_i}(t) = a_i g \left(\sum_{j=1, j\neq i}^{q_s} w_{i,j} x_j(t) + \sum_{l=1}^{p} b_{i,l} u_l(t) + c_i \right) + w_{i,i} x_i(t). \tag{2}$$

Here, the continuous-valued state variable x_i describes the expression level of gene i. The parameters $w_{i,j}$ are the elements of the gene-gene interaction matrix $\boldsymbol{W}$ that possesses positive entries for inducers, negative entries for repressors and zero entries if there is no influence from gene j to gene i. The input variable u_l represents the $l$te environmental factor. Then, the parameters $b_{i,l}$ of the input matrix $\boldsymbol{B}$ determine how the environmental factor u_l influences the expression level x_i. The change in expression level x_i at each point in time depends on a weighted sum of influential factors. In case of nonlinear differential equations (2) the nonlinear function g realizes a nonlinear monotonic sigmoidal activation function. a_i and c_i are additional parameters of the nonlinear model.

The overall model consists of a set of q_s coupled equations of the mathematical form (1) or (2). Such a system models the regulatory interactions between

q genes with $q \leq q_s$. While each gene expression time series is typically simulated by a single equation according to (1) or (2), the NetGenerator algorithm also allows to model a time series by more than one differential equation. Those correlated equations increase the dynamic order of the submodel and allow to identify more complex time courses. The overall model can be subdivided into q submodels or nodes each consisting of those equations that model a single time series (Fig. 1).

Given the model architecture and suitable directly measured or preprocessed time series data for the gene expressions and the external inputs, the NetGenerator algorithm constructs sparse interaction and input matrices $\boldsymbol{W}$ and $\boldsymbol{B}$. The identification of the relevant non-zero model parameters bases on a combinatorial search strategy that separates the model structure identification problem from the model parameter identification problem.

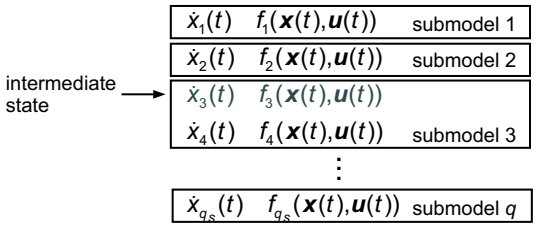

Fig. 1. Organisation of the NetGenerator model: The q submodels that model the q gene expression time series can consist of one or more differential equations. In case of higher order submodels, intermediate states are included.

2.2 Model Structure Identification

The model structure is determined by the connectivity between the network nodes being equivalent to the information about the non-zero parameters in the matrices $\boldsymbol{W}$ and $\boldsymbol{B}$. The NetGenerator structure identification method aims to detect suitable model structures as well as to allow the simultaneous application of favorable parameter identification methods from a system identification point of view.

The NetGenerator algorithm is characterized by the separate identification of the q submodels. In an outer optimization loop, the overall model is extended by a newly optimized submodel in each iteration step. The given gene expression time series are not identified in an arbitrary order, instead the order is optimized within an inner optimization loop. Thus, in each iteration step of the inner loop, the identification of all time series (that have not been identified up to this point) is tested and finally the best one is selected. Within a single iteration step of the inner optimization loop, the true submodel structure optimization is performed utilizing a combinatorial search strategy.

Consequently, the NetGenerator structure identification method consists of three interlocking parts. First, the outer optimization loop realizes the separate identification of the time series. Second, the inner optimization loop is responsible for the optimization of the order of the time series identification. Third, the combinatorial search strategy performs the optimization of all submodel structures.

The aim of the optimization of the time series order in the second part is to obtain suitable conditions for the model parameter identification. The optimization of this order causes that simple models that need only few influential factors to be adequately modeled are selected first. In contrast, more complex time series that require many influential factors are identified later. Understanding the advantages of this effect requires the introduction of three different types of connections within the interaction matrix $\boldsymbol{W}$ (Fig. 2). Forward connections are positioned in the lower triangular part of the square matrix $\boldsymbol{W}$. The local feedbacks form the main diagonal and the backward connections or global feedbacks are the elements of the upper triangular matrix.

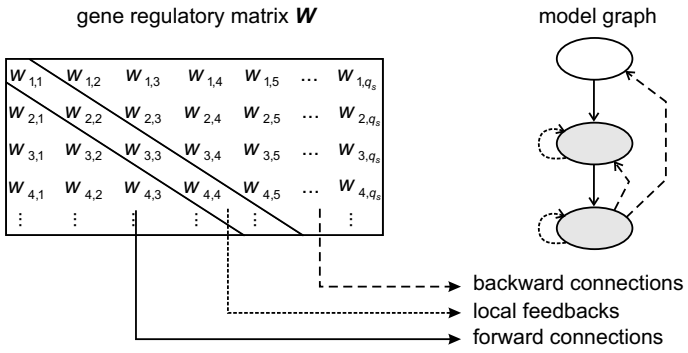

Fig. 2. Forward connections, local feedbacks, backward connections/global feedbacks with respect to their positions in the interaction matrix as well as their directions in the model graph

Now, it is assumed that the submodel for gene i has to be identified and that the gene is regulated by another gene j. The following both situations can be distinguished:

- The time series of gene j has not yet been identified. The corresponding interaction parameter $w_{i,j}$ is an element of the upper triangular matrix. That means, a backward connection is included. In this situation, the time series of gene j is only known at the measurement time points. Thus, the simulation of the submodel that is required for parameter identification (Sec. 2.3) has to use interpolated measurement data. Two facts are disadvantageous. The measured expression levels are corrupted by noise and the measured time

course can significantly differ from the later estimated time course. In this situation, the parameters estimated and possibly the submodel structure will not be optimal.
- The time series of gene j has already been identified. The parameter $w_{i,j}$ is an element of the lower triangular matrix. That parameter corresponds to a forward connection. For the simulation of the submodel of gene i instead of the measured expression levels of gene j the simulated expression levels can be used.

Since the measurement data from microarray experiments include considerable noise levels and the model architecture is a rough simplification of biological regulatory networks, the use of interpolated measurement data for model simulation and parameter identification has clear disadvantages. Interpolated measurement data are required if backward connections are included into the submodel. Fitting simple time series with few gene-gene interactions first minimizes the number of the critical backward connections during the identification process. Forward connections allow to take the real modeled time courses and their associated modeling errors into account. The situations described above correspond to the prediction error and the output error method known from system identification theory [10].

Of course, in the final model, forward and backward connections possess no biological interpretation. The order of the final submodels can be arbitrarily permuted. It should be mentioned that the order of the time series during identification is only relevant with respect to parameter identification.

Embedded in the outer and inner optimization loop, the NetGenerator algorithm performs the structure identification of the submodels. Starting with an initial submodel structure the algorithm executes the following steps in an iterative procedure:

1. Modification of the submodel structure by the heuristic search strategy
2. Fitting of the relevant submodel parameters to the data
3. Simulation of the resulting model to obtain the submodel output
4. Determination of the modeling error

The search strategy applied in the first step suggests combinations of influential genes and external inputs to be examined and compared. All $q+p$ potential influential factors[3] are summarized in the set Z

$$Z = [x_1, \ldots, x_{q_s}, u_1, \ldots, u_p]. \tag{3}$$

Testing all possible combinations of influential factors or non-zero parameters is even for very small networks an impractical task. Therefore, the NetGenerator algorithm employs a search strategy that makes reasonable restrictions of the search space. Possible solutions are directed towards simple, plausible and

[3] Note that the $q_s - q$ intermediate states that result from submodels with more than one equation are not allowed to influence other genes.

interpretable model structures. The search is performed by applying a number of growing and pruning steps that modify a given e.g. initial submodel structure. The growing and pruning steps vary the submodel complexity with respect to the number of gene-gene interactions, the number of influences from external inputs and the dynamic order of the submodel. The model selection is controlled by a number of stopping criterions.

Initial submodel. Each submodel structure optimization starts with a simple initial submodel that represents a first order lag element. The initial submodel of gene i possesses two non-zero parameters; the local feedback parameter $w_{i,i}$ that realizes the self-regulation effect and the parameter $b_{i,1}$ that assumes an influence from the first external input on the expression of gene i.

Modification of the submodel complexity. The NetGenerator algorithm selects subsets of relevant model parameters by searching in two directions: model growing (forward selection) and model pruning (backward elimination). Forward selection bases on the assumption that the best intermediate solution is part of the best final solution. Since this assumption does not have to be true, backward elimination is applied in order to remove unimportant interactions.

1. *Model growing (forward selection):* A forward selection of the most likely interactions is performed by adding new gene-gene interactions or interactions from environmental factors. Starting from a given submodel structure with n non-zero parameters, all possible solutions with $n+1$ non-zero parameters are examined. The best solution with respect to the model fit is retained and further expanded in the next iteration until a stopping criterion is met.
   Selecting gene-gene interactions, forward connections are preferred, while backward connections are only included if other connectivities could not provide acceptable solutions.
2. *Model pruning (backward elimination):* Backward elimination removes gene-gene interactions and external inputs from the submodel structure. In order to decrease the model complexity, all possible solutions that result from the removal of one interaction are considered. Again, the best solution is retained and tested for possible further removals until a stopping criterion is met. If interactions are removed, the algorithm guarantees that the decreased model structures remain biologically plausible. For example, structures with only one local feedback parameter are meaningless and are generally excluded.
3. *Inclusion or removal of additional time lag elements:* The third possibility to obtain improved model fits is to adapt the type of dynamic dependency between the interacting genes. The general model structure involves first order dynamics for all submodels. In order to overcome this limitation, the NetGenerator algorithm allows to include submodels that consist of R differential equations and that represent lag elements of order R. The search strategy tests different dynamic orders up to a predefined maximum dynamic order and selects the best fitting one. Although, the dynamic behavior of the included higher order submodels changes significantly, their allowed parameterization is strongly restricted to transfer functions with R equal poles and

no or only one zero. Higher order submodels are well suited to identify regulatory interactions that are characterized by significant time delays. They preserve the connectivity of the network model. It should be mentioned that oscillating submodels are excluded since the associated submodel complexity would allow them to adapt to highly complex time courses solely based on submodel dynamics instead of submodel connectivity.

Stopping criterion. In order to avoid overfitting and to reach some predefined model characteristics, the inclusion or the removal of interactions is tied to a number of conditions e.g. an increase in submodel complexity must lead to a considerably improved model fit, a decreased submodel complexity only leads to a marginally worsened model fit, the number of relevant submodel parameters is smaller than the number of data points in the corresponding time series and the number of submodel interactions does not exceed a predefined limit.

2.3 Model Parameter Identification

Model parameter identification for a given submodel structure is a repeatedly executed operation. In this approach, the parameter identification is performed by a constrained nonlinear optimization procedure that minimizes the mean square error between the model fit and the expression data. The self-regulation parameters $w_{i,i}$ are constrained by the condition $w_{i,i} < 0$, i.e. the generated submodels are locally stable. It should be mentioned that even for linear differential equation systems the nonlinear optimization is preferred. A linear regression method requires the information about the time derivatives. However, estimating the time derivatives from sparsely sampled and noisy time courses is extremely unreliable. That problem agrees with the unfavorable optimization of submodels including backward connections. Nevertheless, the time derivatives are used for parameter initialization, since linear least squares regression is applied to obtain the initial parameters for the nonlinear optimization. Here, time derivatives are calculated based on a Hermite interpolation. These time derivatives are exclusively used in order to find initial parameter values. The parameters of the nonlinear differential equation system according to (2) are initialized in the same way. Their additional parameters are initialized in such a manner that they provide a linear output for a wide operating range. The initial conditions $\boldsymbol{x}(0)$ are not optimized, they are derived directly from the measured time courses.

2.4 Integration of Expert Knowledge

Because of the high complexity of gene regulatory networks as well as the serious limitations of the measurement data, it is very advantageous to incorporate as much biological knowledge as possible into the network model. The combinatorial search strategy bases on the general knowledge that each gene interacts with only a limited number of other genes. However, the structure identification method also allows to introduce specific expert knowledge about the existence or

the absence of gene-gene interactions or input signaling pathways. Figure 3 illustrates such information and shows the corresponding model graph. The structure identification algorithm ensures the consistency of all examined submodel structures with the previously known relationships. The possibility to constrain the given model structure can also be utilized to test different hypotheses extracted by former network reconstruction and to assess their effects on the remaining model structure.

Genes	1	2	3	4	5	6	7	8
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	1	-	-	-	-	1	-
4	-	-	-	-	-	-	-	-
5	-	-	0	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	0	-	1	-

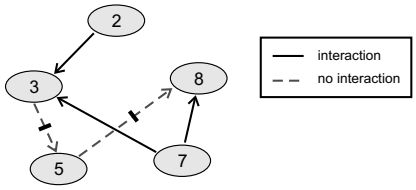

Fig. 3. Prior knowledge about interactions between eight genes and corresponding model graph; (- no prior knowledge, 1 interaction exists, 0 interaction exists not)

3 Applications

The NetGenerator algorithm has been applied to generate hypotheses about gene regulatory interactions within different biological networks from gene expression time series data. For all applications, an exhaustive clustering analysis was performed. The main kinetics were extracted by primarily fuzzy clustering of differentially expressed genes. Clustering analyses included the optimization of the number of clusters by evaluating cluster validity indices and the utilization of process-specific knowledge from databases. Network reconstruction was performed by detecting the regulatory interactions between cluster-representative genes. The selection of these representative genes was based on their fuzzy membership degree to clusters, on biological expert knowledge or also on methods as e.g. gene description text mining. As a result of clustering, the NetGenerator algorithm had to discover the causal relationships between 4 and 10 main kinetics characterizing the biological processes. The available time series contained between 5 and 9 measurement points. For several applications, alternative network models were generated and investigated based on different initializations of the algorithm or the integration of different expected prior knowledge. For some applications, the robustness of network reconstruction was analyzed by performing a huge number of identification runs with artificially perturbed data simulating the effects of measurement noise. The resulting models included between 8 and 22 interaction parameters that were compared with knowledge not included into the network reconstruction. The following applications have been published:

- Immune response of peripheral blood mononuclear cells to bacterial infection with heat-killed pathogenic *E. coli* [2]; (Fig. 4)
- Stress response during recombinant protein expression in *E. coli* [11]
- Effect of LiCl stimulation on hepatocytes [12]
- Stress response to a temperature shift in *A. fumigatus* [13]
- Effect of culture media on primary mouse hepatocytes [14]

In [15], the application of the NetGenerator algorithm for network model based analysis of a bioartificial liver cell system is reported. In contrast to the applications listed above, relationships between the kinetics of biochemical variables and amino acids were analyzed.

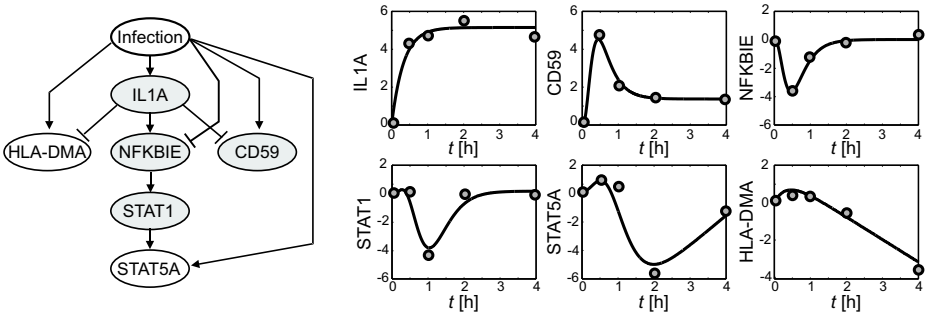

Fig. 4. Graph and simulated output of the inferred network model in [2]

4 Discussion

The NetGenerator algorithm presented in this paper has been devised to identify gene regulatory network models in form of linear or nonlinear differential equation systems using gene expression time series data. The advantages of differential equation systems are their capability to present dynamic system behavior explicitly and to model dynamics at continuous-valued expression levels, rather than just the two levels on and off. There are a number of identification methods available for inferring those dynamic systems. Well known examples are least squares methods [3], singular value decomposition based methods [4], genetic algorithms [7], simulated annealing [5] and combinatorial search strategies [16, 8]. Most of these approaches incorporate linear regression methods. They suffer from the drawback of requiring the time derivatives that have to be calculated from sparsely sampled and noisy measurement data. The NetGenerator algorithm proposed uses nonlinear optimization to estimate the parameters of a given submodel structure. Beside the accurate fitting that can be obtained due to the favorable system theoretic conditions, it also allows the optimization of higher order submodels. Those submodels enable the identification of significantly time delayed gene regulation. This capability is very important, since it

is known that there can be a considerable time delay between the expression of one gene and the observation of its effects [17]. With the inclusion of intermediate states, the possibility is given to consider a broader class of biologically meaningful dynamic dependencies.

Due to the limitations of available data, it is extremely important to find an adequate subset of genes, gene clusters or cluster-representative genes for network reconstruction. This selection has to include as much expert knowledge as possible. The applications presented in Sec. 3 perform clustering in order to preprocess the given time courses. If the quality and the quantity of data is seriously limited and if no further knowledge is available, for two very similar time courses, it makes no sense to derive different connectivities for the corresponding genes. Even though that gene-specific information can be lost and that co-expressed genes do not have to be regulated by the same biological function, similar expression patterns should be clustered. Advantages of clustering are the inherent reduction of dimensionality and noise.

The state variables of the differential equation systems are not restricted to model levels of gene expressions. In general, they can correspond to measured or preprocessed data of any type (e.g. proteins, metabolites). Also, the input variables are not restricted to directly measured environmental factors. These can be combinations or nonlinear transformations of those measured factors combined or transformed based on given knowledge. If no measurements for external factors are available, their qualitative behavior valid under the experimental conditions has to be assessed and a temporal behavior has to be assumed (e.g. Heaviside step function for sudden and ongoing changes of environmental conditions).

In reverse engineering, in general, a small modeling error gives no guarantee that the model obtained will show structural equivalence to the gene regulatory network analyzed. Differential equation systems are rough representations of biological mechanisms, since the complex regulatory effects of intermediate products are simplified to linear or specific nonlinear relationships between genes [18]. Otherwise, data delivered by DNA microarray experiments mostly do not contain enough information to reconstruct more complex models. Therefore, as long as data availability is not improved, the models that result will not be adequate to make correct predictions for the gene regulatory response under changed experimental conditions.

For these reasons, the NetGenerator algorithm allows to express hypotheses on the most likely activating or repressing interactions of the underlying network. Providing further support for these hypotheses, additional specifically designed experiments are necessary. Integrating diverse biological knowledge can dramatically improve the results of network reconstruction. A limitation of the presented algorithm is its relatively long calculation time for larger networks (e.g. more than 15 genes). This effect is caused by the exhaustive search strategy that includes a huge number of nonlinear parameter optimizations. However, the NetGenerator algorithm allows to reconstruct very sparsely connected network models with a high accuracy of model fitting.

References

1. De Jong, H.: Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. Report (2000)
2. Guthke, R., Moeller, U., Hoffmann, M., Thies, F., Toepfer, S.: Dynamic Network Reconstruction from Gene Expression Data applied to Immune Resonse during Bacterial Infection. Bioinformatics **21** (2005) 1626–34
3. D'haeseleer, P., Wen, X., Fuhrmann, S., Somogyi, R.: Linear Modeling of mRNA Expression Levels During CNS Development and Injury. Pacific Symposium on Biocomputing **4** (1999) 41–52
4. Yeung, M. K. S., Tegne, J., Collins, J. J.: Reverse Engineering Gene Networks using Singular Value Decomposition and Robust Regression. Proc. Natl. Acad. Sci. **99** (2002) 6163–68
5. Mjolsness, E., Mann, T., Castano, R., Wold, B.: From Coexpression to Coregulation: An Approach to Inferring Transcriptional Regulation among Gene Classes from Large-Scale Expression Data. Advances in Neural Information Processing Systems **12** (2000) 928–34
6. D'haeseleer, P., Liang, S.,Somogyi, R.: Genetic Network Inference: from Co-Expression Clustering to Reverse Engineering. Bioinformatics **16** (2000) 707–26
7. Wahde, M., Hertz, J.: Coarse-Grained Reverse Engineering of Genetic Regulatory Networks. Biosystems **55** (2000) 129–36
8. van Someren, E.P., Wessels, L.F.A., Reinders, M.J.T., Backer, E.: Searching for Limited Connectivity in Genetic Network Models. International Conference on Systems Biology (2001)
9. Bonneau, R., Reiss, D.J., Shannon, P., Facciotti, M., Hood, L., Baliga, N.S., Thorsson, V.: The Inferelator: An Algorithm for Learning Parsimonious Regulatory Networks from Systems-Biology Data Sets *de novo*. Genome Biology **7** (2006)
10. Nelles, O.: Nonlinear System Identification. Springer (2001)
11. Schmidt-Heck, W., Guthke, R., Toepfer, S., Reischer, H., Duerrschmid, K., Bayer, K.: Reverse Engineering of the Stress Response during Expression of a Recombinant Protein. Proc. EUNITE Symp. (2004) 407–12
12. Zellmer, S., Schmidt-Heck, W., Gaunitz, F., Baldysiak-Figiel, A., Guthke, R., Gebhardt, R.: Dynamic Network Reconstruction from Gene Expression Data Describing the Effect of LiCl Stimulation on Hepatocytes. Journal of Integrative Bioinformatics (2005)
13. Guthke, R., Kniemeyer, O., Albrecht, D., Brakhage, A.A., Moeller, U.: Discovery of Gene Regulatory Networks in *Aspergillus fumigatus*. Lecture Notes in Bioinformatics (2006) (submitted)
14. Schmidt-Heck, W., Zellmer, S., Gebhardt, R., Guthke, R.: Effect of Culture Media on Primary Mouse Hepatocytes Identified by Dynamic Network Reconstruction from Gene Expression Data. Lecture Notes in Bioinformatics (2006) (submitted)
15. Schmidt-Heck, W., Zellmer, S., Gaunitz, F., Gebhardt, R., Guthke, R.: Data-based Extraction of Hypotheses about Gene Regulatory Networks in Liver Cells. European Symposium on Nature-inspired Smart Information Systems (2005)
16. Chen, T., He, H.L., Church, G.M.: Modeling Gene Expression with Differential Equations. Pacific Symposium on Biocomputing **4** (1999) 29–40
17. Li, X.,Rao, S., Jiang, W., Li, C., Xiao, Y., Guo, Z., Zhang, Q., Wang, L., Du, L., Li, J., Li, L., Zhang, T., Wang, Q.K.: Discovery of Time-Delayed Gene Regulatory Networks based on Temporal Gene Expression Profiling. Bioinformatics **7** (2006)
18. Wessels, L., van Someren, E., Reinders, M.: A Comparison of Genetic Network Models. Proc. of Pac. Symp. on Biocomputing (2001)