In: Banzhaf, Wolfgang and Christaller, Thomas and Dittrich, Peter and Kim, Jan T. and Ziegler, Jens: *Advances in Artificial Life (ECAL 2003)*, 686-695. Springer Verlag, Berlin Heidelberg, 2003.

Developing and Testing Methods for Microarray Data Analysis Using an Artificial Life Framework

Dirk Repsilber[1] and Jan T. Kim[2]

[1] Institute of Medical Biometry and Statistics
Ratzeburger Allee 160, House 4, 23538 Lübeck, Germany,
dirk.repsilber@imbs.uni-luebeck.de
[2] Institute for Neuro- and Bioinformatics,
Seelandstraße 1a, 23569 Lübeck, Germany, kim@inb.uni-luebeck.de

Abstract. Microarray technology has resulted in large sets of gene expression data. Using these data to derive knowledge about the underlying mechanisms that control gene expression dynamics has become an important challenge. Adequate models of the fundamental principles of gene regulation, such as Artificial Life models of regulatory networks, are pivotal for progress in this area.

In this contribution, we present a framework for simulating microarray gene expression experiments. Within this framework, artificial regulatory networks with a simple regulon structure are generated. Simulated expression profiles are obtained from these networks under a series of different environmental conditions. The expression profiles show a complex diversity. Consequently, success in using hierarchical clustering to detect groups of genes which form a regulon proves to depend strongly on the method which is used to quantify similarity between expression profiles. When measurements are noisy, even clusters of identically regulated genes are surprisingly difficult to detect. Finally, we suggest cluster support, a method based on overlaying multiple clustering trees, to find out which clusters in a tree are biologically significant.

1 Introduction

High throughput technology for molecular analysis of biological systems has rapidly advanced during the last decade. However, the developments in theory and modelling the fundamental dynamical principles of living systems has not kept up with the massive increase in availability of biological data. More specifically, while the amount of gene expression data has explosively grown during the last few years, an integrated theory of gene expression and regulatory networks is not yet available, even though advances in theory of transcription factor bioinformatics [1,2], modelling regulatory networks [3,4,5] and their evolution [6,7] have been made.

As a consequence of the divergence between availability of data and theoretical foundations, the major bottleneck for making progress in understanding biological systems is not due to scarcity of data, but due to lack of ways for harnessing the available data for theory and modelling. This constitutes a major challenge for Artificial Life research. One of the obstacles for linking biological data and mathematical or computer based models is the definition of adequate criteria to evaluate the degree of consistency between model and data, and for providing directions for refinements in modelling as well as in data acquisition.

Microarray data analysis has become an important tool which has been used for classification of biological systems [8], for inferring regulatory patterns and structures, such as clusters of co-regulated genes [9,10]. One of the most important long-term goals of these efforts is inferring regulatory interactions and, ultimately, entire regulatory networks [11,12,13]. In this contribution, we address this challenge by subjecting artificial regulatory networks to the same analysis procedures which are used in molecular biology and asking how much information about the underlying regulatory network can be retrieved by these approaches.

2 Systems and Methods

The transsys framework [14] provides a computer language which allows to describe regulatory gene networks by comprehensive, object-oriented *programs*. A program consists of factor (i.e. protein) and gene specifications. An *instance* represents the concentrations of the factors in a program at a specific time t. The *update method* computes the expression levels at time $t+1$ based on the levels at time t and the information provided by the program.

A set of tools for microarray simulation and analysis with transsys has been implemented using the Python programming language [15] and the `R` statistics system [16]. The software is available on the website [17].

2.1 Regulatory Network Construction

Genes can often be categorized into regulatory genes, which encode products that control gene expression (e.g. transcription factors), and structural genes, which encode products that do not function as regulators, such as proteins that form body structures or enzymes. Our method for randomly constructing transsys programs was designed to reflect this property. An example network is shown in Fig. 1.

Network construction involves two steps. Firstly, a core regulatory network with N genes is generated. Each gene encodes a unique product and has its expression controlled by K factors which are randomly chosen and randomly designated to be an activator or a repressor. The architecture of the core network is an NK network [3,6]. Secondly, structural genes are added. R genes of the core network are randomly chosen as regulators of a regulon. For each regulator, a number of structural genes, activated by the regulator, are generated. Each structural gene encodes a unique product. The entire regulon consists of the regulator and the structural genes which it controls. The structure of regulons

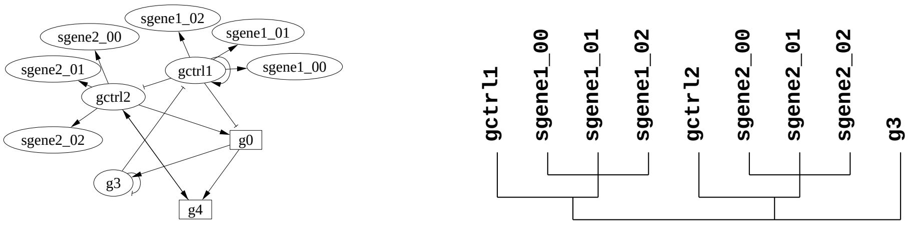


Fig. 1. Left: A regulatory network consisting of $N = 5$ regulatory genes, each of which receives $K = 2$ regulatory inputs. Edges ending in arrows depict activation, those ending with bars show repression. 2 genes, labelled `gctrl`, control a regulon. Structural genes are labelled `sgene`. Genes encoding products that are directly perturbed by the environment are boxed. Right: Regulon tree representing the regulons in the network. Genes which are directly perturbed are excluded from the regulon tree.

in a network can be represented by a hierarchical graph which we refer to as the *regulon tree*, as shown in Fig. 1.

The transsys program is constructed such that in each time step t, expression of a gene g_i results in synthesis of an amount of f_i, the product encoded by the gene, given by

$$\mathrm{synth}(f_i, t) = c + \max\{0, \sum_{k=1}^{K} \mathrm{reg}(r_k, \alpha, \beta)\}$$

where $C(f, t)$ denotes the concentration of factor f at time t and $r_1, \ldots r_K$ are the factors regulating g_i. The regulation terms are given by

$$\mathrm{reg}(r_k, \alpha, \beta) = \begin{cases} \beta C(r, t)/(\alpha + C(r, t)) & \text{if } r_k \text{ is an activator} \\ -\beta C(r, t)/(\alpha + C(r, t)) & \text{if } r_k \text{ is a repressor.} \end{cases}$$

The concentration of a gene product f at time $t + 1$ is given by $C(f, t + 1) = (1 - d)C(f, t) + \mathrm{synth}(f, t)$, where d is a decay rate.

The components of the core regulatory network are parameterized differently from those of the structural part. The parameters for the regulatory core are denoted by c_{reg}, α_{reg}, β_{reg} and d_{reg}. The parameters for the structural part are α_{struct}, β_{struct} and d_{struct}, $c = 0$ for all structural genes.

2.2 Simulation of Gene Expression Measurements

Comparative approaches have become a mainstream in gene expression analysis. In these approaches, an expression measurement performed under reference conditions is compared with measurements obtained with alternate environmental conditions (see e.g. [9,18]). Frequently, a time series of measurements is made after changing the environment. The entire set of expression changes for a gene g

is called the *expression profile* of g. In a subsequent step, distances are computed for all pairs of expression profiles. Finally, the resulting distance matrix used for hierarchical clustering. The result is a *clustering tree* in which leaves represent genes and the distance between the expression patterns of gene pairs is correlated to the length of the path connecting the genes. It is assumed that clusters in the tree (i.e. groups of genes which form subtrees) may reflect regulons. The simulation presented subsequently models this comparative approach.

For the reference measurement, a transsys instance is generated by initializing all expression levels with a random value drawn from a uniform distribution over the unit interval $[0,1]$. Starting with this initial instance, 5000 updates are simulated to allow the system to converge into an attractor. The resulting transsys instance is used as the reference state. The concentration of factor f in the reference state is denoted by $C_{\mathrm{ref}}(f)$.

Alterations of the environmental conditions result in changes of certain gene products, e.g. by photochemical transformations. In the simulation framework, a subset of genes (shown as boxes in Fig. 1) from the core network is designated to encode products that are subject to direct modification by the environment. An environmental impact is simulated by perturbing the concentration of each factor f encoded by this subset according to

$$C_e(f,0) = (C_{\mathrm{ref}}(f) + \epsilon) \cdot 2^{g\sigma_e} \tag{1}$$

where $C_e(f,0)$ denotes the initial concentration in the new environment, g is a random value drawn from a Gaussian normal distribution, σ_e is a control parameter that allows tuning the variance and $\epsilon = 0.01$ is a small offset which allows alteration of factor concentrations which are zero in the reference state.

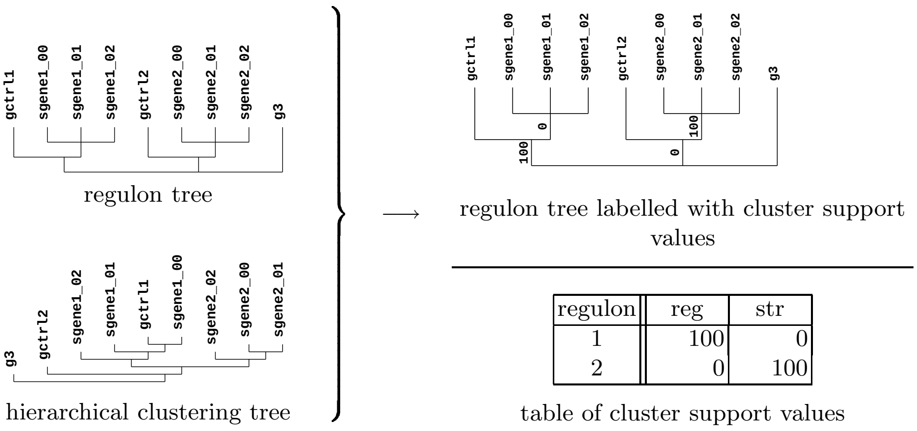
Starting with the transsys instance representing the initial conditions for an environment, network dynamics are simulated. Every $\Delta t = 5$ time steps, a sample is taken until 20 samples are collected. For each sample, intensity values are computed assuming the additive background model

$$a_e(f,t) = C_e(f,t) + b \cdot \exp(g\sigma_a) \tag{2}$$

where $b = 0.1$ reflects the median fluorescence of an array spot with no complementary labelled product bound to it, g is a random value from a Gaussian normal distribution and σ_a controls the variance of background fluorescence. Logarithmized intensity ratios $r_e(f,t) = \log_2(a_e(f,t)/a_{\mathrm{ref}}(f))$ are calculated for all factors not subject to direct environmental impact. Here, $a_{\mathrm{ref}}(f)$ denotes the intensity value calculated from $C_{\mathrm{ref}}(f)$ according to (2). For each gene g, the expression intensities measured in environment e are aggregated into a 20-dimensional vector $X_e(g)$ with components $x_{e,i}(g) = r_e(f, \Delta t \cdot i)$. $X_e(g)$ thus represents the expression profile of gene g in environment e.

2.3 Cluster Analysis and its Evaluation

Hierarchical clustering trees were constructed by common methods [9,10] from expression profiles of all genes that encode factors which are not directly perturbed. We used data sets of expression profiles from individual environments,



regulon	reg	str
1	100	0
2	0	100

Fig. 2. Finding consistent edges. The regulon tree and the hierarchical clustering tree both contain an edge separating regulon 1 (regulatory gene `gcntrl1` and structural genes `sgene1_01`, `sgene1_02` and `sgene1_03`) from the remaining genes. Likewise, the three structural genes of regulon 2 form a cluster in both the regulon tree and the clustering tree. Thus, the corresponding edges in the regulon tree are supported by the clustering tree, amounting to 100% because there is just one clustering tree in this example. A table shows these values more comprehensively.

and additionally a set of aggregated profiles obtained by concatenating the profiles from all environments.

For hierarchical clustering, we used Euclidean distance and the Pearson-type measure described in [9]. The latter measure $S(X, Y) \in [-1, 1]$ was devised to reflect similarity of expression profile shape regardless of absolute values. As a distance measure derived from this similarity measure, we used $1 - S(X, Y)$. For clustering, the average-linkage and the single-linkage algorithm were used. For each distance measure and clustering method, a consensus tree of the clustering trees obtained for the individual environments was computed with the `consense` program [19].

Clustering trees were evaluated by comparing them to the regulon tree. For each edge in the regulon tree, we determined whether a consistent edge exists in the clustering tree. Consistency of two edges in different trees means that both edges imply the same split of the set of leaf nodes. The result can be visualized and processed into a table as illustrated in Fig. 2. We refer to the percentage of clustering trees that contain a consistent edge as the *cluster support* of an edge in the regulon tree. Cluster support values can also be computed for clustering trees instead of regulon trees. We explore using cluster support to identify biologically significant clusters in the results presented below.

regu-	**Euclidean distance**								**Pearson-type distance**							
lon	**average linkage**				**single linkage**				**average linkage**				**single linkage**			
	$\sigma_a = 0$		$\sigma_a = 1$		$\sigma_a = 0$		$\sigma_a = 1$		$\sigma_a = 0$		$\sigma_a = 1$		$\sigma_a = 0$		$\sigma_a = 1$	
	reg	str	reg	str	reg	str	reg	str	reg	str	reg	str	reg	str	reg	str
26	10	100	0	10	10	100	0	10	30	100	10	10	10	100	10	10
03	30	100	10	90	70	100	10	90	0	100	0	80	0	100	0	60
01	0	100	0	10	0	100	0	10	0	100	0	10	0	100	0	10
49	0	100	0	10	0	100	0	10	0	100	0	10	0	100	0	10
29	0	100	0	10	0	100	0	10	0	100	0	10	0	100	0	10
22	0	100	0	10	0	100	0	10	0	100	0	10	0	100	0	10
43	0	100	0	60	0	100	0	50	0	100	0	50	0	100	0	50
16	0	100	0	10	0	100	0	10	0	100	0	10	0	100	0	10

Table 1. Cluster support values obtained with all combinations of Euclidean distance and Pearson-type distance, average and single linkage clustering and $\sigma_a = 0$ and $\sigma_a = 1$, as illustrated in Fig. 2.

3 Results and Discussion

3.1 Cluster Analysis

Networks of $N = 50$ regulatory genes, each having $K = 2$ regulatory inputs, were generated with the parameters $c_{\mathrm{reg}} = 0.2$, $\alpha_{\mathrm{reg}} = 1$, $\beta_{\mathrm{reg}} = 0.5$ and $d_{\mathrm{reg}} = 0.2$. $R = 8$ regulons, each consisting of 3 structural genes with $\alpha_{\mathrm{struct}} = 1$, $\beta_{\mathrm{struct}} = 3$ and $d_{\mathrm{struct}} = 0.1$, were attached to the regulatory core. 20 genes were chosen for direct perturbation. Expression dynamics were computed for 10 environments generated with $\sigma_e = 2$. Microarray simulations were performed with array measurement noise levels ranging from $\sigma_a = 0$ to $\sigma_a = 1$.

Table 1 shows the results of evaluating cluster analyses of expression profiles generated with such a network. With no noise in array measurement, all structural genes in a regulon have identical expression profiles with our current parameterization method. Thus, the groups of structural genes within all regulons receive 100% cluster support with all clustering methods. The expression profile of the regulatory gene in a regulon differs from the profiles of the structural genes. Even with no array noise, the association between the regulatory and the structural genes in a regulon is not detected by any clustering method for 6 out of 8 regulons. Only regulons 3 and 26 are recognized. Cluster support for regulon 3 is consistently higher with Euclidean distance whereas the Pearson-type distance detects regulon 26 more reliably.

When noise in array measurement is simulated, cluster support values for these groups become smaller than 100% and reveal a characteristic pattern. The cluster support for the structural gene groups in regulons 3 and 43 is at least 50% in all analyses conducted with $\sigma_a = 1$, while the structural gene groups in all other regulons receive only 10% cluster support.

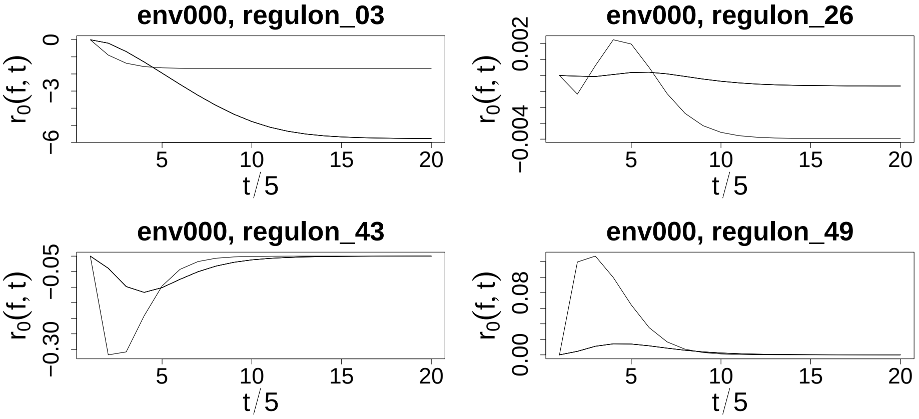


Fig. 3. Expression profiles for regulons 3, 26, 43 and 49, measured for environment 0 Graphs show the logratio profiles of the regulator and the structural genes of a regulon. The response of the structural genes lags behind the regulator, but is more pronounced.

3.2 Individual Expression Profiles

Fig. 3 shows expression profiles for regulons 3, 26, 43 and 49. Regulon 3 strongly responds to the impact of environment 0, the regulon, both the regulatory gene and the structural ones, are effecively switched off. The resulting similarity in expression profiles is captured by Euclidean distance.

Regulon 26 exhibits a complex expression profile in which an initial phase of downregulation is followed by transient upregulation until the system finally settles to a new state which is downregulated with respect to the reference state. This characteristic down-up-down shape is reflected in the expression profiles of both the regulator and the structural factors. Although the response of the structural genes lags behind that of the regulator, the Pearson-type distance proves to be suitable for detecting the characteristic similarity in shape.

Regulons 43 and 49 respond to the environmental impact by transient regulatory changes. The amplitude of the change in regulon 43 is substantially greater than in regulon 49. Therefore, the group of structural genes of regulon 43 can be detected with noisy measurements, whereas the structural gene groups of regulons 49, 26 and others with a small response amplitude become indistignuishable when noise is present.

3.3 Detecting Significant Clusters

For empirical data sets, the regulon tree is not known. A clustering tree can be constructed based on the full data set. Cluster support values for this tree can be determined by using trees computed from data subsets corresponding to individual environments. These cluster support values may be useful to indicate

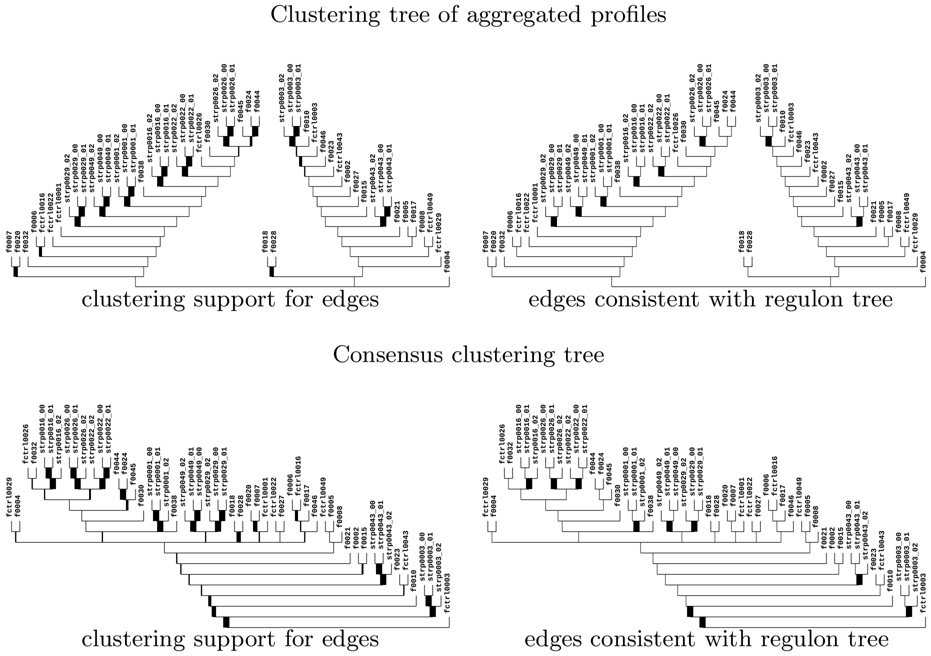

Clustering tree of aggregated profiles
clustering support for edges
edges consistent with regulon tree
Consensus clustering tree
clustering support for edges
edges consistent with regulon tree

Fig. 4. Clustering tree of aggregated profiles and consensus tree, computed with $\sigma_a = 0$, Euclidean distance and single linkage clustering. Edges in trees on the left side are displayed by lines with thickness proportional to their clustering support. In trees on right side, edges which are consistent with the regulon tree are highlighted by thick lines. Clustering support and consistency with the regulon tree are clearly correlated.

which edges are biologically significant in the sense that they correspond to regulons.

To explore this approach, we have computed a clustering tree from the aggregated expression profiles and the consensus tree of the 10 clustering trees corresponding to the 10 environments, and assigned cluster support values to the edges in both trees. The results are shown in Fig. 4. In both trees, clustering support is distributed in a characteristic pattern. Elevated cluster support is strongly correlated to consistency with the regulon tree. Interestingly, the full regulon 3 is captured by the consensus tree but not by the tree of aggregated expression profiles.

4 Conclusion and Outlook

Detection of a majority of regulons on the basis of noisy expression measurements proved to be unexpectedly difficult. To a substantial extent, this difficulty is due to the fact that an environmental impact does not necessarily elicit a pronounced response of a regulon. Regulons with a small and transient response cannot be distinguished on the basis of noisy expression profiles. This finding highlights the fact that coregulation may strongly depend on the conditions under which it is observed. Genes may appear unregulated or coregulated under many conditions, and yet, under other conditions, their expression profiles may strongly differ. Even the detection of simple regulons therefore critically depends on the number and choice of conditions under which gene expression is sampled.

It is possible that small and transient responses are overrepresented in the networks which we constructed. For example, one may assume that selection induces a bias towards regulons which respond with pronounced changes to many environmental challenges. This issue is most adequately addressed by using an evolutionary algorithm for regulatory network generation, which we plan to do in future analyses.

The correlation between the expression patterns of the regulator and the structural genes in a regulon proved to be difficult to detect, even with our simplistic model in which structural genes are controlled directly and exclusively by one regulator. Depending on the particular properties of the expression profiles, different distance measures may be useful to detect the corresponding regulons. Our results indicate that, as also suggested in [20], there is no universally adequate, straightforward method to classify genes by their expression profiles. Advances in understanding the fundamental principles of regulatory networks may eventually result in the development of more generic methods of classification, and we plan to use our framework as a point of departure for research in this field. At this time, however, the choice of classification approaches should be based on the specific biological subject of research. Therefore, we also plan to extend our analysis to additional distance measures, such as mutual information [10] or jackknife correlation [21].

Cluster support can be applied to assess biological significance of clusters. We demonstrated that elevated levels of cluster support and consistency with the regulon tree are strongly correlated. This applies for clustering trees constructed from aggregated expression profiles as well as for consensus trees. It will be interesting to explore whether this correlation extends to other methods of clustering and data analysis, and whether other generic approaches such as bootstrapping [22] can be applied to assess biological significance. Performing multiple cluster analyses and selecting the most significant clusters from each analysis may provide a component of a more generic classification method.

References

1. Schneider, T.D., Stormo, G.D., Gold, L.: Information content of binding sites on nucleotide sequences. J.Mol.Biol. **188** (1986) 415–431
2. Kim, J.T., Martinetz, T., Polani, D.: Bioinformatic principles underlying the information content of transcription factor binding sites. Journal of Theoretical Biology **220** (2003) 529–544
3. Kauffman, S.A., Weinberger, E.W.: The NK model of rugged fitness landscapes and its application to maturation of the immune response. J. Theor. Biol. **141** (1989) 211–245

4. von Dassow, G., Meir, E., Munro, E.M., Odell, G.M.: The segment polarity network is a robust developmental module. Nature **406** (2000) 188–192
5. Reil, T.: Dynamics of gene expression in an artificial genome – implications for biological and artificial ontogeny. In Floreano, D., Nicoud, J.D., Mondada, F., eds.: Advances in Artificial Life. Lecture Notes in Artificial Intelligence, Berlin Heidelberg, Springer-Verlag (1999) 457–466
6. Kauffman, S.A.: Requirements for evolvability in complex systems: Orderly dynamics and frozen components. Physica D **42** (1990) 135–152
7. Bornholdt, S., Sneppen, K.: Neutral mutations and punctuated equilibrium in evolving genetic networks. Physical Review Letters **81** (1998) 236–239
8. Golub, T., Stonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science **286** (1999) 531–536
9. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA **95** (1998) 14863–14868
10. Michaels, G.S., Carr, D.B., Askenazi, M., Fuhrmann, S., Wen, X., Somogyi, R.: Cluster analysis and data visualization of large-scale gene expression data. In Altman, R.B., Dunker, A.K., Hunter, L., Klein, T.E., eds.: Biocomputing '98, Singapore, World Scientific (1998) 42–53
11. Akutsu, T., Miyano, S., Kuhara, S.: Inferring qualitative relations in genetic networks and metabolic pathways. Bioinformatics **16** (2000) 727–734
12. Morohashi, M., Kitano, H.: Identifying gene regulatory networks from time series expression data by *in silicio* screening and sampling. In Floreano, D., Nicoud, J.D., Mondada, F., eds.: Advances in Artificial Life. Lecture Notes in Artificial Intelligence, Berlin Heidelberg, Springer-Verlag (1999) 477–486
13. Repsilber, D., Liljenström, H., Andersson, S.G.: Reverse engineering of regulatory networks: Simulation studies on a genetic algorithm approach for ranking hypotheses. BioSystems **66** (2002) 31–41
14. Kim, J.T.: `transsys`: A generic formalism for modelling regulatory networks in morphogenesis. In Kelemen, J., Sosík, P., eds.: Advances in Artificial Life (Proceedings of the 6th European Conference on Artificial Life). Volume 2159 of Lecture Notes in Artificial Intelligence., Berlin Heidelberg, Springer Verlag (2001) 242–251
15. van Rossum, G., Drake, F.L.: Python reference manual (2002) `http://www.python.org/`.
16. Ihaka, R., Gentleman, R.: R: A language for data analysis and graphics. Journal of Computational and Graphical Statistics **5** (1996) 299–314
17. Kim, J.T.: The `transsys` home page (2003) `http://www.inb.uni-luebeck.de/transsys/`.
18. Gasch, A.P., Spellmann, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M., Storz, G., Botstein, D., Brown, P.O.: Genomic expression programs in the response of yeast cells to environmental changes. Molecular Biology of the Cell **11** (2000) 4241–4257
19. Felsenstein, J.: PHYLIP – phylogeny inference package, version 3.5c. (1993) `http://evolution.genetics.washington.edu/phylip.html`.
20. Quackenbush, J.: Computational analysis of microarray data. Nature Reviews Genetics **2** (2001) 418–426
21. Heyer, L., Kruglyak, S., Yooseph, S.: Exploring expression data: identification and analysis of coexpressed genes. Genome Res. **9** (1999) 1106–1115
22. Jain, A., Moreau, J.: Bootstrap techniques in cluster analysis. Pattern Recognition **20** (1987) 547–568