Iteratively Inferring Gene Regulatory Networks with Virtual Knockout Experiments

Christian Spieth, Felix Streichert, Nora Speer, and Andreas Zell

Centre for Bioinformatics Tübingen (ZBIT), University of Tübingen,
Sand 1, D-72076 Tübingen, Germany,
spieth@informatik.uni-tuebingen.de,
http://www-ra.informatik.uni-tuebingen.de

Abstract. In this paper we address the problem of finding gene regulatory networks from experimental DNA microarray data. We introduce enhancements to an Evolutionary Algorithm optimization process to infer the parameters of the non-linear system given by the observed data more reliably and precisely. Due to the limited number of available data the inferring problem is under-determined and ambiguous. Further on, the problem often is multi-modal and therefore appropriate optimization strategies become necessary. Therefore, we propose a new method, which will suggest necessary additional biological experiments to remove the ambiguities.

1 INTRODUCTION

In the past few years, DNA microarrays have become one of the key techniques in the area of gene expression analysis. This technology enables the monitoring of thousands of genes in parallel and can therefore be used as a powerful tool to understand the regulatory mechanisms of gene expression in a cell.

However, due to the huge number of components within the regulatory system, a large amount of experimental data is needed to infer genome-wide networks. This requirement is almost impracticable to meet today, because of the high costs of these experiments and due to the fact that the investigated processes are too short and do not allow for more sampling points in time. To bypass this problem, additional data has to be acquired like knock-out, over-expression experiment data or data sets with different starting conditions that decrease the uncertainties in the system.

In this paper we propose a methodology for reverse engineering large sets of time series data obtained by expression analysis. This is successively done by optimizing the parameters of systems of differential equations modelling the interactions in the network for the given data followed by a second phase, aimed to reduce the ambiguities by suggesting subsequent knock-out experiments. Information gained by these follow-up experiments are incorporated into the first phase to increase the probability of finding the correct network model. And although traditional knock-out experiments are expensive and time consuming,

techniques like chemical knock-outs are subject of recent research and will become more flexible in future. Further on, time series in which single gene products are over-expressed can be accomplished comparably easily and result in information that can be used in our approach as well. Our approach is also able to use data sets with different starting concentrations of the relevant gene products, i.e. examining the genes of interest under different environment conditions.

Section 2 of this paper presents an overview over related work and lists associated publications. Detailed description of our proposed method will be given in section 3 and example applications will be shown in section 4. Finally, conclusions and an outlook on future research will be covered by section 5.

2 RELATED WORK

Researchers are interested in understanding the mechanisms of gene regulatory processes and therefore in inferring the underlying networks. This has recently become one of the major topics in bioinformatics due to the increased amount of data available. The following section briefly describes the work that has been done in this area.

One kind of model to simulate regulatory systems found in the literature are Boolean or Random Boolean Networks (RBN) [10, 19]. In Boolean Networks gene expression levels can be in one of two states: either 1 (on) or 0 (off). The quantitative level of expression is not considered. Two examples for inferring Boolean Networks are given by Akutsu [1] and the REVEAL algorithm [12].

In contrast to discrete methods like RBNs, qualitative network models allow for multiple levels of gene regulation. Two examples for this kind of approach are given by Thieffry and Thomas in [16]. Akutsu et al. suggest a heuristic for inferring such models in [2].

Quantitative models like the weighted matrix model by Weaver et al. [18] consider the continuous level of gene expression. The topology and the parameters of this model have been successfully inferred by the use of Genetic Algorithms in [3] and [4]. Inference methods based on linear models for gene regulatory networks are given for example in [5] and [6]. An example for mathematical models using S-Systems to infer regulatory mechanisms has been examined by Tominaga et al. [17].

3 MODELLING

On an abstract level, the behavior of a cell is represented by a gene regulatory network of N genes. Each gene g_i produces a certain amount of RNA x_i when expressed and therefore changes the concentration of this RNA level over time: $\boldsymbol{x}(t+1) = h(\boldsymbol{x}(t))\,,\ \boldsymbol{x}(t) = (x_1, \cdots, x_n)$.

To model and to simulate regulatory networks we decided to use S-Systems since they are well-documented and examined. But there are alternatives as listed in section 2, which will be the subject of research in future applications.

S-Systems are a type of power-law formalism which has been suggested by Irvine and Savageau [9, 14] and can be described by a set of nonlinear differential equations:

$$\frac{dx_i(t)}{dt} = \alpha_i \prod_{j=1}^{N} x_j(t)^{\mathcal{G}_{i,j}} - \beta_i \prod_{j=1}^{N} x_j(t)^{\mathcal{H}_{i,j}} \tag{1}$$

where $\mathcal{G}_{i,j}$ and $\mathcal{H}_{i,j}$ are kinetic exponents, α_i and β_i are positive rate constants and N is the number of equations in the system. The equations in (1) can be seen as divided into two components: an excitatory and an inhibitory component. The equation system is integrated using a fourth-order Runge-Kutta algorithm (with adaptive step size controlling). The parameters of the S-System $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathcal{G}$, and $\mathcal{H}$ are optimized with an enhanced Evolutionary Algorithm described in the following.

Evolutionary Algorithms have proved to be a powerful tool for solving complex optimization problems. Three main types of evolutionary algorithms have evolved during the last 30 years: Genetic Algorithms (GA), mainly developed by J.H. Holland [8], Evolutionary Strategies (ES), developed by I. Rechenberg [13] and H.-P. Schwefel [15] and Genetic Programming (GP) by J.R. Koza [11]. Each of these uses different representations of the data and different operators working on them. They are, however, inspired by the same principles of natural evolution. Evolutionary Algorithms are a member of a family of stochastic search techniques that mimic the natural evolution as proposed by Charles Darwin of mutation and selection.

Because ES are suited for optimizing problems based on real values, they meet our requirement best. The following listing describes the general principle of Evolutionary Strategies:

1. Create an initial set (population) $P_{t=0}$ of λ solutions (individuals).
2. Evaluate all individuals of this population P_t according to a given fitness function.
3. Select the μ best individuals of the population with respect to the calculated fitness value as the population of parents P'_t.
4. Mutate/recombine individuals of the parent generation to create a new population of λ offsprings P''_t .
5. Replace the initial population by the new population of offsprings $P_{t+1} = P''_t$ (eventually merged with P_t).

Repeat steps 2 to 5 until a termination criterion is met.

In our application an ES individual encodes the parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathcal{G}$ and $\mathcal{H}$ and represents a possible solution of the model identification problem.

For evaluating the fitness of the individuals we used the following equation for calculation of the fitness values:

$$f = \sum_{i=1}^{N} \sum_{k=1}^{T} \left\{ \left(\frac{\hat{x}_i(t_k) - x_i(t_k)}{x_i(t_k)} \right)^2 \right\} \tag{2}$$

where N is the total number of genes in the system, T is the number of sampling points taken from the time series and $\hat{x}$ and x distinguish between estimated data and sampled data. The overall problem is to minimize the fitness value f. In theory, the solution, i.e. the inferred GRN, should be the best individual found by the ES after termination.

Due to the small number of data the system is highly under-determined and therefore finding the biologically correct model is very difficult. A large number of different sets of model parameters fit the given data with comparably good fitness values (in respect to the fitness function mentioned above) but with only small resemblance to the true system.

To cope with this issue, our proposed method consists of a framework holding m ES populations, which will be optimized separately to gain different models satisfying the constrains given by the fitness function. This framework combines the best individuals from each population to form a population of best-suited models, which will have comparably good fitness values due to the ambiguity of the data but different parameters. To choose the very best model, i.e. the model representing the real biological dependencies, each of the combined models is further examined by simulating virtual knock-out, over-expression or changed start condition experiments. These virtual experiments are performed *in silico* for every gene in each model, i.e. every gene in a model is knocked out to gather information about the impact of that gene on the network represented by the current model. After the genes are ranked for each model, a committee decision is made to determine which gene has to be knocked out in real world experiments to gain the strongest information benefit for the inference process. The rankings is presented to biological researchers to actually perform the corresponding experiments.

The whole process is repeated with incorporation of the new data until a minimum quality level of the resulting models is reached. These models can then be verified by biologists to find the overall best network model. The details of this iterative process are described in the following work flow.

3.1 Inference work flow

The following work flow illustrates the interactive process of computer scientists and biologists to infer a GRN from expression data:

Phase $i = 1a$ The first optimization phase is started with m different initial populations to reach a diverse set of individuals. After the first optimization, the algorithm collects the best l individuals of the m ES populations and evaluates each of the $l * m$ models by finding the gene having the strongest impact on the dynamics if knocked out *in silico*. This is done by simulating the network without the corresponding gene and evaluating the differences of the calculated time course to the dynamics of the complete network. In this first implementation, we use a simple relative squared error (Eq. 2) summed up at each sampling point over time. After this, the resulting list of genes is ranked and the top candidate gene is suggested for further investigation.

Phase $i = 1b$ Additional microarray experiments based on the knock-out proposals have to be accomplished yielding another set of expression data. These experiments can either be carried out using techniques like knock outs or by inhibiting single gene products chemically.

Phase $i = 2a$ The set of new data is incorporated in the next optimization step. The whole process is then repeated iteratively until a termination criterion is met.

4 APPLICATIONS

To illustrate our method, we established two regulatory network systems, which were simulated to gain sets of expression data. After creating the data sets, we used our proposed algorithm to reverse engineer the correct model parameters. The following sections show this for a 5-dimensional and a 10-dimensional example, respectively.

4.1 Gene Regulatory Network with $N = 5$ genes

Due to the fact that GRNs in nature are sparse systems, we created regulatory networks randomly with a maximum cardinality of $k <= 3$, i.e. each of the $N = 5$ genes depends on three or less other genes within the network. The dynamics of the example can be seen in Fig. 1.

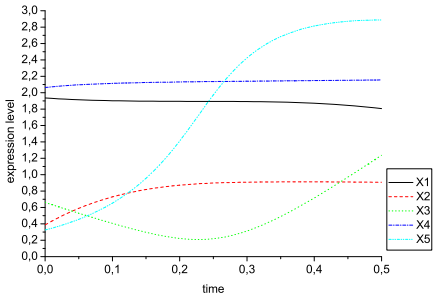

Fig. 1. Artificial 5-dimensional gene regulatory network

parameter	value	parameter	value
α_1	0.233	G_{11}	-2.000
α_2	2.330	G_{25}	-0.788
α_3	1.217	G_{31}	-0.496
α_4	1.602	G_{35}	2.072
α_5	3.153	G_{42}	-0.473
		G_{53}	-0.958
β_1	0.703	H_{12}	-0.591
β_2	2.012	H_{13}	1.462
β_3	2.737	H_{21}	-1.025
β_4	1.597	H_{22}	0.112
β_5	2.573	H_{53}	-0.023

Fig. 2. Model parameter of the target S-System

In Fig. 1, each x_i represents the RNA level of a certain gene. At this point, we do not differentiate between closely related molecules like mRNA and distantly related like proteins.

Inference This time course data was then subject to our inference method as described in Section 3. In the following subsections each phase of the algorithm is explained using the 5-dimensional example. The results are then compared to a standard ES with identical optimization settings but without incorporating additional information.

Phase 1a The optimization process was performed using a (μ,λ)-ES with $\mu = 5$ and $\lambda = 30$ together with a Covariance Matrix Adaptation (CMA) mutation operator [7] and no recombination to evolve individuals. This optimization was repeated $m = 20$ times with different starting populations to calculate 20 different populations, i.e. 20 different models. After evolving the models for $10,000$ generations (total number of $300,000$ fitness evaluations), the best individual of each population was taken to form a population of best individuals. For each of these individuals, virtual knock-out experiments were simulated and the top candidate genes were ranked. Tables 1 - 3 list the ranking of each gene for the corresponding algorithm phase, i.e. the number of votes in each network.

Table 1. Ranking of the genes (phase 1)

Gene	Votes
1	**9**
2	3
3	4
4	1
5	3
	20

Table 2. Ranking of the genes (phase 2)

Gene	Votes
1	-
2	1
3	4
4	4
5	**11**
	20

Table 3. Ranking of the genes (phase 3)

Gene	Votes
1	-
2	1
3	**16**
4	3
5	-
	20

Phase 1b After ranking the importance of the gene within the network the biological knock-out experiments were performed *in silico* resulting in an additional set of expression data.

Phase 2-4 These phases were repeated until the correct model was found. Fig. 3 shows the averaged fitness values for each repetition phase, i.e. for the degree of additional knock-out information in comparison with the fitness values for a standard ES optimizer.

As can be seen in the figure, the fitness converges quickly to 0.0 for both algorithms, which corresponds to a very good model quality with respect to the fitness function. Unfortunately, the models found by the standard ES resemble the original parameters only little. This is illustrated by Fig. 4, where the euclidian distance between the inferred parameters and the parameters of the original system is shown. The standard ES converges to a local optimum, which has a comparably good fitness value but represents completely different dependencies.

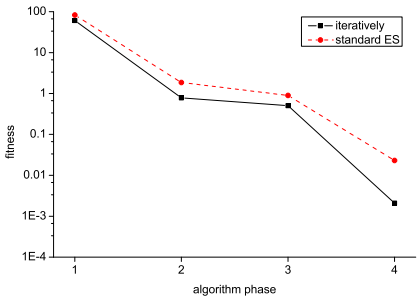

Fig. 3. Best fitness values for each phase

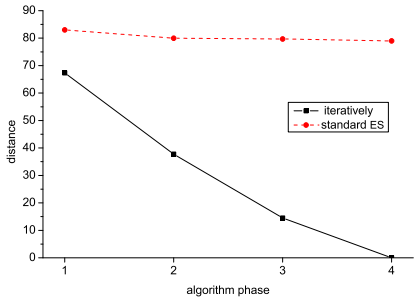

Fig. 4. Distance values for each phase

The proposed method on the other hand leads directly to the global optimum, i.e. the correct network by successively removing ambiguities.

4.2 Gene Regulatory Network with $N = 10$ genes

As a second and due to the increased number of participating genes more difficult test case, we created another regulatory network randomly with a maximum cardinality of $k <= 3$. The dynamics of the example can be seen in Fig. 5, where each gene expression level is again represented by the corresponding x_i.

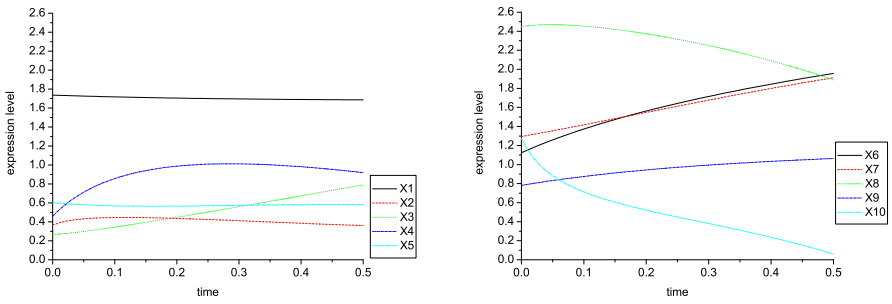

Fig. 5. Artificial 10-dimensional gene regulatory network

Inference The given time course data was then again inferred by our algorithm. The optimization process was performed using the same settings for the ES as in example 1 (see Section 4.1). The optimization was repeated $m = 20$ times with different starting populations to calculate 20 different populations, i.e. 20 different models. The resulting ranking tables are not shown here due to the limited space available.

The different phases of our algorithm were repeated until the termination criterium was reached, i.e. a total number of 500,000 fitness evaluations per algorithm phase. Fig. 6 shows the averaged fitness values for each phase.

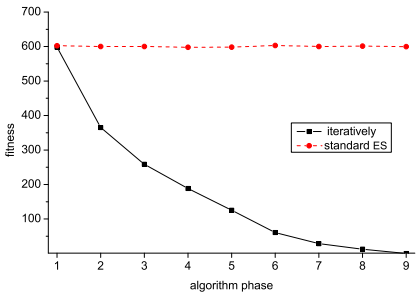

Fig. 6. Best fitness values for each phase

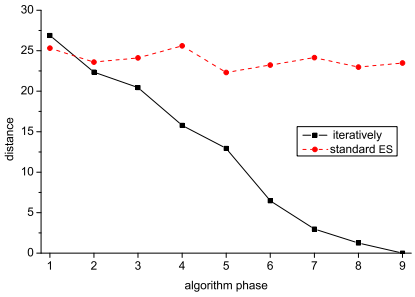

Fig. 7. Distance values for each phase

In this example, a standard ES was not able to find a solution for the optimization problem. Only the enhanced algorithm, which included additional information, found the correct system, as illustrated in Fig. 7.

5 DISCUSSION

The problem of inferring GRNs is a very difficult process due to the limited data available and the large number of unknown variables in the system. Most examples found in literature are artificial and very small, i.e. with a total number of ten genes or lower. And although the dimensionality of these examples is by far not relevant to biological processes, they show the first attempts of modelling regulatory networks from high-throughput experimental techniques.

In this paper we have shown a method to infer gene regulatory systems even in cases where standard approaches were not able to cope with the problem of under-determination. Our method yields promising results by incorporating additional knowledge into the inference procedure. The necessary information can be gathered by additional biological experiments like (chemical) knock-out and over-expression experiments or by altering environmental conditions to change the initial concentrations of the relevant gene products.

In future work we plan to include a-priori information into the inference process like partially known pathways or information about co-regulated genes, which can be found in literature. For better coverage of the solution space of the optimizer we will use a cluster-based niching algorithm which was developed in our group. Additional models for gene regulatory networks will be examined for simulation of the non-linear interaction system as listed in Section 3 to overcome

the problems with those gene regulatory networks which cannot be modelled by S-Systems.

Further on, we will continue to test our method with real microarray data in close collaboration with biological researchers at our facility.

References

1. T. Akutsu, S. Miyano, and S. Kuhura. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 17–28, 1999.
2. T. Akutsu, S. Miyano, and S. Kuhura. Algorithms for identifying boolean networks and related biological networks based on matrix multiplication and fingerprint function. In *Proceedings of the fourth annual international conference on Computational molecular biology*, pages 8 – 14, Tokyo, Japan, 2000. ACM Press New York, NY, USA.
3. S. Ando and H. Iba. Quantitative modeling of gene regulatory network - identifying the network by means of genetic algorithms. In *Poster Session of Genome Informatics Workshop 2000*, pages 278–280, 2000.
4. S. Ando and H. Iba. Inference of gene regulatory model by genetic algorithms. In *Proceedings of the 2001 Congress on Evolutionary Computation*, pages 712–719. IEEE Press, 2001.
5. T. Chen, H. L. He, and G. M. Church. Modeling gene expression with differential equations. In *Proceedings of the Pacific Symposium on Biocomputing*, 1999.
6. P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi. Linear modeling of mRNA expression levels during CNS development and injury. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 4, pages 41–52, 1999.
7. N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of the 1996 IEEE Int. Conf. on Evolutionary Computation*, pages 312–317, Piscataway, NJ, 1996. IEEE Service Center.
8. J. H. Holland. *Adaption in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Systems*. The University Press of Michigan Press, Ann Arbor, 1975.
9. D. H. Irvine and M. A. Savageau. Efficient solution of nonlinear ordinary differential equations expressed in S-systems canonical form. *SIAM Journal of Numerical Analysis*, 27(3):704–735, 1990.
10. S. A. Kauffman. *The Origins of Order*. Oxford University Press, New York, 1993.
11. J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
12. S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 3, pages 18–29, 1998.
13. I. Rechenberg. *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.
14. M. A. Savageau. 20 years of S-systems. In E. Voit, editor, *Canonical Nonlinear Modeling. S-systems Approach to Understand Complexity*, pages 1–44, New York, 1991. Van Nostrand Reinhold.
15. H.-P. Schwefel. *Numerical optimization of computer models*. John Wiley and Sons Ltd, 1981.

16. D. Thieffry and R. Thomas. Qualitative analysis of gene networks. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 77–87, 1998.
17. D. Tominaga, N. Kog, and M. Okamoto. Efficient numeral optimization technique based on genetic algorithm for inverse problem. In *Proceedings of German Conference on Bioinformatics*, pages 127–140, 1999.
18. D. Weaver, C. Workman, and G. Stormo. Modeling regulatory networks with weight matrices. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 4, pages 112–123, 1999.
19. A. Wuensche. Genomic regulation modeled as a network with basins of attraction. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 3, pages 89–102, 1998.