Optimizing Topology and Parameters of Gene Regulatory Network Models from Time-Series 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. Different approaches to infer the dependencies of gene regulatory networks by identifying parameters of mathematical models like complex S-systems or simple Random Boolean Networks can be found in literature. Due to the complexity of the inference problem some researchers suggested Evolutionary Algorithms for this purpose. We introduce enhancements to the 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. We propose a new method, which evolves the topology as well as the parameters of the mathematical model to find the correct network.

1 INTRODUCTION

The inference of regulatory dependencies between genes from time series data has become one of the most challenging tasks in the field of functional genomics. With new experimental methods like DNA microarrays, which have become one of the key techniques in the area of gene expression analysis in the past few years, it is possible today to monitor thousands of genes in parallel. Therefore, these techniques can be used as a powerful tool to explore 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 impracticable to meet today, because of the high costs of these experiments and due to the combinatorial nature of gene interaction.

The earliest models to simulate regulatory systems found in the literature are Boolean or Random Boolean Networks (RBN) [6]. 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 *et al.* [1] and the REVEAL algorithm [10] by Liang *et al.* These models have the advantage that they can be solved with only small computational effort. But they suffer from the disadvantage of being tied to discrete system states. In contrast, qualitative network models allow for multiple levels of gene regulation. An example for this kind of approach is given by Thieffry and Thomas in [16]. But these models use only qualitative dependencies and therefore only a small part of the information hidden in the time series data. Quantitative models based on linear models for gene regulatory networks like the weighted matrix model by Weaver *et al.* [18] or the singular value decomposition method by Yeung *et al.* [19] consider the continuous level of gene expression. Other approaches to infer regulatory systems from time series data by using Artificial Neural Networks [7] or Bayesian Networks [4] have been recently published, but face some drawbacks as well. Bayesian networks, for example, do not allow for cyclic networks. More general examples for mathematical non-linear models like S-Systems to infer regulatory mechanisms have been examined by Maki *et al.* [11] or Kiguchi *et al.* [8].

In our method we try to use the advantages of flexible mathematical models like S-Systems. We introduce a method, which separates the inference problem into two subproblems. The first task is to find the topology or structure of the network with a Genetic Algorithm. In the second task the parameters of a mathematical model are optimized for the given topology with an Evolution Strategy. The second problem can be seen as a local search phase of a Memetic Algorithm (MA).

The remainder of this paper is structured as follows. Section 2 describes the proposed algorithm and the mathematical model used in the optimization process. Applications and results are listed in section 3 and the conclusions and an outlook are given in section 4.

2 INFERENCE METHOD

The following section gives an overview over the proposed algorithm.

2.1 Memetic Algorithm

Evolutionary Algorithms have proven 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 [3], Evolution Strategies (ES), developed by I. Rechenberg [12] and H.-P. Schwefel [14] and Genetic Programming (GP) by J.R. Koza [9]. Each of these uses different representations of the data and different main 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 of repeated mutation and selection as proposed by Charles Darwin.

In the current implementation we used a combination of a Genetic Algorithm for optimizing the topology together with an Evolution Strategy to locally find the best parameters for the given topology. The general principle is outlined in Fig. 1.

```
begin
  initGApop()
  eval(GApop)
  while (termination criteria not met)
    selectGAparentPop()
    createGAoffsprings()
    eval(GApop)
    selectNewGApop()
  do
end
```

```
eval(GApop) {
  for each topology
    initESpop()
    eval(ESpop)
    while (termination criteria not met)
      selectESparentPop()
      createESoffsprings()
      eval(ESpop)
      selectNewESpop()
    do
    setFitness(GAind, bestESfitness)
  do
}
```

Fig. 1. Pseudo-code describing the general principle of the Memetic Algorithm

2.2 Global Genetic Algorithm

In our implementation the Genetic Algorithm evolves populations of structures of possible networks. These structures are encoded as bitsets where each bit represents the existence or absence of an interaction between genes and therefore of non-zero parameters in the mathematical model. The evaluation of the fitness of each individual within the GA population uses a local search described below.

2.3 Local Evolution Strategy

For evaluation of each structure suggested by the global optimizer an Evolution Strategy is used, which is suited for optimizing problems based on real values. The ES optimizes the parameters of the mathematical model used for representation of the regulatory network.

Fitness. For assessing the quality of the locally obtained results we used the following equation for calculation of the fitness values for the ES optimization process:

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

where N is the total number of genes in the regulatory system, T is the number of sampling points taken from the time series and $\hat{x}$ and x distinguish between estimated data and experimental data. The overall problem is to minimize the fitness value f.

Mathematical Model 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 mRNA x_i, when expressed, and therefore changes the concentration of this mRNA over time: $x_i(t+1) = h(\boldsymbol{x}(t))$ with $\boldsymbol{x}(t) = (x_1, \cdots, x_n)$, where h describes the changing of each RNA level depending on all or only on some RNA concentrations at the previous time step.

To model and to simulate regulatory networks we decided to use S-Systems since we think they are flexible enough to model important gene regulatory dependencies like feed back loops, etc. But there are alternatives as listed in section 1, which will be the subject of research in future applications.

S-Systems. S-Systems are a type of power-law formalism, which has been suggested by Irvine and Savageau [5, 13] and can be described by a set of nonlinear differential equations:

$$\frac{dx_i(t)}{dt} = \alpha_i \prod_{j=1}^{N} x_j(t)^{\mathcal{G}_{i,j}} - \beta_i \prod_{j=1}^{N} x_j(t)^{\mathcal{H}_{i,j}} \tag{2}$$

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 (2) can be seen as divided into two components: an excitatory and an inhibitory component.

The kinetic exponents $\mathcal{G}_{i,j}$ and $\mathcal{H}_{i,j}$ determine the structure of the regulatory network. In the case $\mathcal{G}_{i,j} > 0$ gene g_j induces the synthesis of gene g_i. If $\mathcal{G}_{i,j} < 0$ gene g_j inhibits the synthesis of gene g_i. Analogously, a positive (negative) value of $\mathcal{H}_{i,j}$ indicates that gene g_j induces (suppresses) the degradation of the mRNA level of gene g_i.

The S-System formalism has a major disadvantage in that it includes a large number of parameters that have to be estimated. The total number of parameters in S-Systems is $2N(N+1)$, with N the number of state variables x_i (genes). This causes problems with increasing number of participating genes due to the quadratically increasing number of parameters to infer. The parameters of the S-System $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathcal{G}$, and $\mathcal{H}$ are optimized with Evolutionary Algorithms described in the previous paragraphs.

3 RESULTS

To verify the concepts of our idea we first compare two network inference examples where the first inference process is initialized without any prior knowledge of the network structure. In the second case we incorporate the correct topology of the dependencies of each gene together with the experimental data to validate the theoretical ability of our approach to find the correct model. After this verification step, we use the proposed method to infer gene regulatory systems from artificial microarray expression data.

3.1 Preliminary Experiments

For validation purposes we examined a small example of gene regulatory networks described in the literature, which has been studied by a variety of researchers in the past. It was first introduced by Savageau [13] and was subject of several attempts to re-engineer networks: Tominaga *et al.* [17] tried to infer only selected genes in their work and Kiguchi *et al.* [8] and Maki *et al.* [11] proposed new methods to reverse engineer the complete system but changed the parameters of the original system for unknown reasons.

Fig. 2 shows the dependencies of the regulatory network as used in all of the publications listed above.

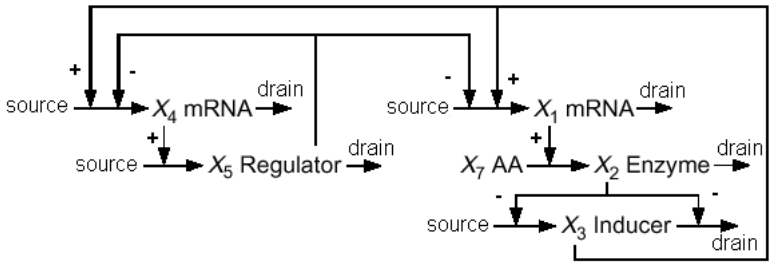

Fig. 2. 5-dimensional gene regulatory network [Savageau [13]]

The total number of parameters to be optimized with the Evolution Strategy in this example was $N = 60$ if modelled with an S-System. Fig. 3 shows the time courses for each mRNA level of the regulatory system. The optimization process was repeated $m = 20$ times to gain averaged fitness courses.

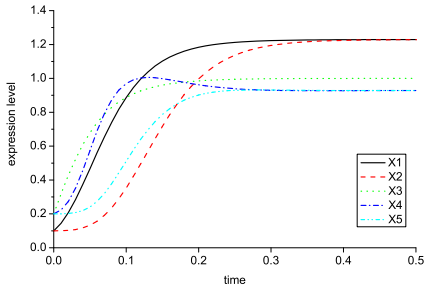

Fig. 3. Time dynamics of the 5-dim regulatory system

Without topological information In the case of the inference without any additional knowledge a (μ,λ)-ES with $\mu = 5$ parents and $\lambda = 35$ offspring is used together with a Covariance Matrix Adaptation (CMA) mutation operator and no recombination to evolve individuals in the optimization process. The CMA operator is one of the most powerful self adaption mechanisms today available for ES. For further details see [2].

Fig. 4 shows the averaged fitness course of the gene regulatory network (GRN) model optimized with a standard ES with no topological information provided. As can be seen in this graph, the ES converges prematurely after approximately $2,000$ generations into a local optimum and the fitness remains static until the end of the optimization process.

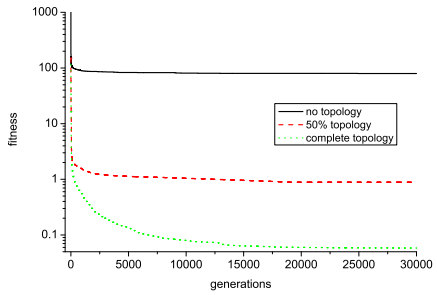

Fig. 4. Fitness of an ES optimization with and without topological information

With partial topological information The second trial incorporated partial information about the topology. In this test case, 50% of the correct interactions of the regulatory network graph was used during optimizing the mathematical model. To use the correct topological information 50% of the values of $\mathcal{G}$ and $\mathcal{H}$ representing no dependencies in the network graph, i.e. $\mathcal{G}_{ij} = 0.0$ or $\mathcal{H}_{ij} = 0.0$, were excluded from the optimization process and therefore fixed to 0.0. This was implemented by a reduced vector of decision variables for the ES. The fitness of this test case is given Fig. 4.

With complete topological information As a third test case we incorporated the correct information about the topology to verify the idea of our method, i.e. solving the overall problem by first finding the correct topology and then identifying the corresponding parameters. Fig. 4 shows the fitness of the second case, optimized by an ES with the optimization settings as given in the previous section. The results of the third test case with the additional information about the structure of the network yields far better fitness values compared to the first test case and better results than the second case as can be seen in Fig. 4.

3.2 Artificial Regulatory Network

To test the method on larger systems we created two artificial microarray data sets, which were to be reverse engineered by our algorithm. The first data set represented the time dynamics of an artificial 10-dimensional regulatory system, i.e. the relationships between 10 genes, which were randomly assigned and simulated. The second example was an artificially created 20-dimensional GRN, which consists of 20 components. All settings for the Evolutionary Algorithms were determined in preliminary experiments.

10-dimensional network Due to the fact that GRNs in nature are sparse systems, we created regulatory networks randomly with a maximum cardinality of $k \leq 3$, i.e. each of the $N = 10$ genes depends on three or less other genes within the network. The dynamics of the example can be seen in Fig. 5.

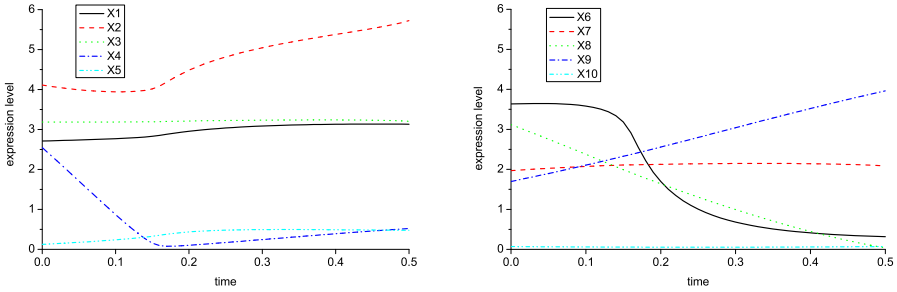

Fig. 5. Time dynamics of the 10-dim regulatory system

The optimization process was performed using a (μ,λ)-ES with $\mu = 10$ parents and $\lambda = 50$ offsprings together with a Covariance Matrix Adaptation (CMA) mutation operator without recombination. This optimization was repeated $m = 20$ times with different starting populations. After evolving the models for $200,000$ generations (total number of $1,000,000$ fitness evaluations), the m best fitness values found were averaged, as shown in Fig. 6.

As illustrated by the fitness plot the standard ES was not able to find a solution for the optimization problem, because it got stuck in local optima. The proposed method on the other hand found solutions with very good fitness values.

Unfortunately, our MA found only twice the correct target system with respect to the topology and parameter values. In the remaining 18 optimization runs, systems were found, which fitted the experimental data, but showed different relationships between the component genes. We address this problem in the discussion in sect. 4.

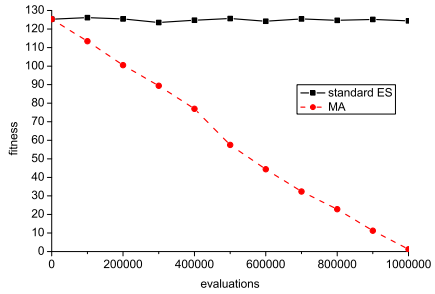

Fig. 6. Fitness graph of the 10-dim regulatory system (standard ES vs. MA)

20-dimensional network The second GRN inferred with the proposed method is an artificial 20-dimensional system. As in the example before, we created the dependencies of the network randomly with a cardinality $k \leq 3$. The simulated time courses are not given here because the number of components of the system make the graph unclear.

The optimization was performed with the same parameter settings as described in sect. 3.2. Due to the larger number of system components, we increased the total number of fitness evaluations to $1,500,000$, thus increasing computing time as well.

Fig. 7 shows the fitness course averaged over the 20 repeated optimization runs. Again, the standard ES did not find a solution whereas the MA converged to optima with good fitness values.

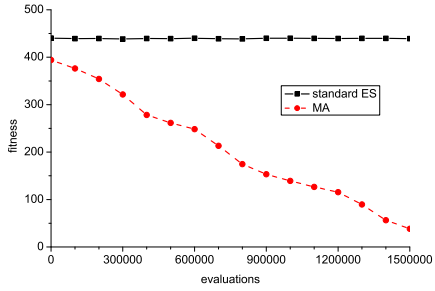

Fig. 7. Fitness graph of the 20-dim regulatory system (standard ES vs. MA)

As before in the 10-dimensional example the problem of finding the correct target system emerged again. The resulting time courses fitted the experiment

data (yielding good fitness values) but showed different dependencies and interactions between the participating genes of the regulatory network.

4 DISCUSSION

In comparison to a standard ES with CMA the proposed method yielded far better fitness values. In both test cases the ES was not even able to find models that fit the given data at all.

Additionally, our algorithm proved to work even for middle-sized examples. Most examples found in literature are artificial and very small, i.e. with a total number of ten genes or lower, while in biological networks even small systems have at least 50–100 components. We showed that our method is able to handle sparse systems ($k \leq 3$) with 20 genes, restricted currently only by computational performance. Because we use a bitset representation of the topology the algorithm reduces the total number of parameters and makes it therefore possible to infer larger systems. Future experiments on high performance computers will address large-scale systems with a minimum number of 100 genes.

Further on, the solutions found by the MA are sparse due to the preceding structure optimization. Because in nature GRNs are sparse systems the solutions of the MA represent better resemblance to biological systems than the standard ES, which resulted always in complete and thus dense matrices. Therefore, the proposed method shows promising results to be suitable to infer gene regulatory systems.

Due to the large number of model parameters and the small number of data sets available, the system of equations is highly under-determined. Therefore, multiple solutions exist, which fit the given data, but show only little resemblance with the original target system. This problem is known in literature but there are currently only few publications reflecting on this issue. Recently, Spieth *et al.* published a new method to incorporate data sets obtained by additional experiments [15]. In future enhancements of our algorithm we plan to incorporate additional methods to identify the correct network.

In future work we also 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 plan to 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 sect. 1 to overcome the problems with a quadratic number of model parameters of the S-System.

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

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. 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.
3. 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.
4. S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano. Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. In *Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB 03)*, pages 104 –113. IEEE, 2003.
5. 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.
6. S. A. Kauffman. *The Origins of Order*. Oxford University Press, New York, 1993.
7. E. Keedwell, A. Narayanan, and D. Savic. Modelling gene regulatory data using artificial neural networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 02)*, volume 1, pages 183–188, 2002.
8. S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita. Dynamic modeling of genetic netowrks using genetic algorithm and s-sytem. *Bioinformatics*, 19(5):643–650, 2003.
9. J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
10. 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.
11. Y. Maki, D. Tominaga, M. Okamoto, S. Watanabe, and Y. Eguchi. Development of a system for the inference of large scale genetic networks. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 6, pages 446–458, 2001.
12. I. Rechenberg. *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.
13. 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.
14. H.-P. Schwefel. *Numerical optimization of computer models*. John Wiley and Sons Ltd, 1981.
15. C. Spieth, F. Streichert, N. Speer, and A. Zell. Iteratively inferring gene regulatory networks with virtual knockout experiments. In R. et al., editor, *Proceedings of the 2nd European Workshop on Evolutionary Bioinformatics (EvoWorkshops 2004)*, volume 3005 of *LNCS*, pages 102–111, 2004.
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. M. K. S. Yeung, J. Tegner, and J. J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. In *Proceedings of the National Academy of Science USA*, volume 99, pages 6163–6168, 2002.