Comparing Mathematical Models on the Problem of Network Inference

Christian Spieth
Centre for Bioinformatics Tübingen (ZBIT)
72076 Tübingen, Germany
spieth@informatik.uni-tuebingen.de

Nadine Hassis
Centre for Bioinformatics Tübingen (ZBIT)
72076 Tübingen, Germany
n.hassis@web.de

Felix Streichert
Centre for Bioinformatics Tübingen (ZBIT)
72076 Tübingen, Germany
streiche@informatik.uni-tuebingen.de

ABSTRACT

In this paper we address the problem of finding gene regulatory networks from experimental DNA microarray data. We focus on the evaluation of the performance of different mathematical models on the inference problem. They are used to model the underlying dynamic system of artificial regulatory networks. The dynamics of the artificial systems represent different basic types of behavior, dimensionality and mathematical properties. They are all created with three commonly used approaches, namely linear weight matrices, H-systems, and S-systems. Due to the complexity of the inference problem, some researchers suggested evolutionary algorithms for this purpose. However, in many publications only one algorithm is used without any comparison to other optimization methods. Thus, we introduce a framework to systematically apply evolutionary algorithms for further comparative analysis.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Miscellaneous*; J.3 [**Computer Applications**]: LIFE AND MEDICAL SCIENCES—*Biology and genetics*

General Terms

ALGORITHMS, PERFORMANCE

Keywords

Evolutionary Computation, Inference, Systems Biology

1. INTRODUCTION

Gene regulatory networks (GRNs) represent the dependencies of the different actors in a cell operating at the genetic level. They dynamically determine the level of gene expression for each gene in the genome by controlling whether a gene will be transcribed into RNA or not. A simple GRN consists of one or more input signalling pathways, several target genes, and the RNA and proteins produced from those target genes. In addition, such networks often include dynamic feedback loops that provide further network regulation activities and output. In order to understand the underlying structures of activities and interactions of intracellular processes one has to understand the dependencies of gene products and their impact on the expression of other genes. Therefore, finding a GRN for a specific biological process explains this process from a logical point of view, thus explaining many diseases. Therefore, the model reconstruction of gene regulatory networks has become one of the major topics in bioinformatics. However, the huge number of system components requires a large amount of experimental data to infer genome-wide networks. Recently, DNA microarrays have become one of the major tools in the research area of microbiology. This technology enables researchers to monitor the activities 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. With this technique, cells can be studied under several conditions such as medical treatment or different environmental influences.

Microarray experiments often result in time series of measured values indicating the activation level of each tested gene in a genome. These data series can then be used to examine the reactions of the cell to external stimuli. A model would enable biologists to predict the reactions of intracellular signalling processes. To re-engineer or infer the regulatory processes computationally from these experimental data sets, one has to find a model that is able to produce the same time series data as the experiments. The idea is then that the model reflects the true system dependencies, i.e. the dependencies of the components of the regulatory system.

Several publications addressing the problem of inferring gene regulatory networks can be found in the literature. De Jong gives a good overview about related work in [1]. A major part of the work done in this field is using deterministic mathematical models to simulate regulatory networks. One kind of those deterministic models are linear models like the weighted matrix model [11, 12]. These models have only a small number of system parameters compared to S-systems but are often not flexible enough to model biological systems in detail, since they model the dependencies linearly. S-systems, on the other hand, model dynamic systems in a nonlinear manner. They consist of a set of differential

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

equations describing the changes in expression over time. However, they show a significant higher number of system parameters. S-systems have been recently examined in [4, 5]. Most applications of deterministic models use evolutionary algorithms (EA) to determine the correct parameters of the mathematical model. EAs have proven to be successful in finding parameters of mathematical models representing GRNs.

In this publication, different mathematical models and identification and optimization algorithms are applied to model a nonlinear dynamic system from experimental data. The main focus is the application of standard algorithms and extensions to find models of a high quality with respect to the similarity of the dynamic behavior of the experimental data. For this, standard algorithms taken from the literature are compared on the problem of inferring a set of artificial benchmark problems.

1.1 Regulatory Systems

The goal of this paper is to compare different models and inference algorithms on the problem of system modeling. For a comparison, a set of experimental data has to be established to measure the performance of each model and algorithm. The main focus of the research for this publication was the identification of biological processes and within the collaboration with biological and medical research institutes, several data sets have been collected. However, the structure of these biological processes and their kinetics are yet totally unknown. This raises the problem of verification, because the inferred model cannot be validated against a true system known in biology. Therefore, beside the biological data sets, artificial data was used for evaluation of the algorithms. This artificial data has the advantage that structure, parameters and mathematical properties are known and thus models found during the optimization can be directly compared.

For the evaluation of the algorithms and methods to infer dynamic systems, a set of benchmark systems was created. The benchmark systems differ in size and the underlying mathematical model that was used for simulation. All of the systems were initialized randomly and represent three classes of size: **2-dimensional**, **5-dimensional**, and **10-dimensional**. The three dimensions represent small and medium-sized systems and the results of all dimensions can be used to evaluate the scalability of each algorithm.

The main focus while generating the artificial data sets was on creating sparsely connected structures to have resemblance to biological systems which are sparse themselves by the very nature of genetic regulation. The maximum connectivity of the five-dimensional systems is 3, i.e. each component depends on at most 3 other components. The maximum connectivity of the ten-dimensional systems is 5 and in the two-dimensional systems, the components are depending on 2 genes. The distribution of interactions followed the scale-free-network approach, where the majority of the components show low connectivity, whereas some components, so called hubs, are connected with a maximum number of others.

Beside the distinctive feature size, four types of different dynamic properties were defined to cover a large part of possible system behaviors: **converging**, **oscillating**, **discontinuous**, and **diverging** dynamics. These four classes represent the basic types of dynamic behavior of real-world systems and thus they can be used to evaluate the performance of the examined algorithms and methods on these basic properties. Due to the diversity of the benchmark set differentiated comparison experiments are possible and detailed conclusions can be drawn from them.

For the actual comparison experiments, an instance of each behavior class was selected from each size, forming three groups with four elements each and the three mentioned examples taken from the literature, thus a total number of 39 data sets. In the following, the performance of the algorithms using different models is averaged within each dimension group, resulting in an average performance depending on the size of the target system. This simplifies the evaluation and shows the scalability of the proposed methods. Interesting details to selected systems are discussed in the corresponding section. The general details and the time dynamics of the model that are used in the following can be found with further information about the systems in the database, which is accessible through the JCell project website[1] [8]. Figure 5 gives an example for the discontinuous benchmark dynamics for each model type that was used in this publication.

1.2 Models

Model building is central to the understanding of real-world phenomena. In modeling, researchers such as engineers, biologists and social scientists mimic their observations in a formal way. But to understand an unknown system without being overwhelmed by the amount of data that can be collected by modern experimental techniques, models have to abstract reality. Those models are the result of structural and dynamical assumptions about the unknown underlying process. The key for obtaining a good model is to find a trade off between simplifying the original system and loosing essential regulatory mechanisms and dependencies. In most real-world applications, where dynamic systems are to be modeled, no structural information of the mathematical equations that represent the system is available. Thus, generic models have to be used to approach the modeling problem. These parameterized models can be inferred by means of parameter optimization. Here, the main task is to adjust the parameters of the model in such a way that the simulated dynamics of the model fits the experimental data of the unknown system. Several algorithms exists that have proven to be very successful for parameter optimization, however, it cannot be guaranteed that a model is found that sufficiently explains the unknown dynamic processes. This is commonly a problem, where an adequate number of experiments is not available, for example in cases where the experiments are very expensive or time consuming.

In the literature, several mathematical models can be found that address the problem of analyzing dynamic systems. In general, the models used to simulate regulatory systems are divided into three major classes, *discrete*, *continuous*, and *probabilistic* models. The first two model classes are also referred to as deterministic models whereas the latter model type is stochastic. Deterministic models assume that every event or action is causally determined by an unbroken chain of prior occurrences. In stochastic models, the dependencies between the components of a system are modeled by probabilistic transition values. The main focus of this work are deterministic models.

[1] http://www.jcell.de

The following sections describe some of the mathematical models that can be used to model dynamic relationships. Common to the mentioned model types is that they have been evaluated in this publication. But the list of models is by far not complete.

1.3 Linear Weight Matrices

Linear weight matrices (WM) have been originally introduced in [11]. In this approach, the regulative interactions between the genes are represented by a weight matrix, $\mathcal{W}$, where each row of $\mathcal{W}$ represents all the regulatory inputs for a specific gene. The regulatory effect of gene g_j on gene g_i at time t is simply the expression level of g_j, x_j, multiplied by its regulatory influence on g_i, w_{ij}. The total regulatory input to g_i is derived by summing across all the genes in the system and in the following referred to as $r_i(t)$:

$$r_i(t) = \sum_j w_{ij} x_j(t) \tag{1}$$

Here, a positive value for w_{ij} indicates that gene g_j is stimulating the expression of gene g_i, x_i. Similarly, a negative value indicates repression, while a value of zero indicates that gene g_j does not influence the transcription of gene g_i. By modeling regulatory interactions with a weight matrix, we can use mathematical matrix approaches found in the field of neural networks for subsequent analyses of the resultant models. With the regulatory state of each gene, we are now able to model the response of each gene to the given input. The impact of $r_i(t)$ on gene g_i is calculated using a so called "squashing" function (Eqn. 2).

$$x_i(t+1) = \frac{m_i}{1 + e^{-(\alpha_i r_i(t) + \beta_i)}} \tag{2}$$

where $r_i(t)$ is the mentioned regulatory state of gene g_i, and α_i and β_i are gene specific constants that define the shape of the squashing function for gene g_i. The resulting expression level is only a relative value between 0 and 1, with 0 representing complete repression and 1 representing maximal expression. Thus, these relative levels have to be converted into the real expression space. In addition, the genes can have different levels of maximal expression. Hence, we multiply the calculated relative gene expression level x_i by the maximal expression level for each gene m_i, to get the final expression level for g_i $x_i(t+1)$ as shown in equation (2).

1.4 S-systems

Another, more flexible type of model, are S-systems (SS). They employ a general formalism, which allow for capturing the nonlinearity and general dynamics of the gene regulation. S-systems are a type of power-law formalism, which have been suggested by [7] and can be described by a set of nonlinear differential equations:

$$\frac{dx_i(t)}{dt} = \alpha_i \prod_{j=1}^{N} x_j(t)^{\mathcal{G}_{i,j}} - \beta_i \prod_{j=1}^{N} x_j(t)^{\mathcal{H}_{i,j}} \tag{3}$$

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 genes in the system. The equations in (3) 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.

1.5 H-systems

H-systems are another type of a set of parameterized differential equations. They originate from the idea to enhance weight matrices with an additional term to ensures nonlinearity in the model. They have been suggested by Hadeler and Spieth [2] and have the form of:

$$\frac{dx_i(t)}{dt} = c_i + \sum_k b_{ik} x_k(t) + x_i(t) \sum_k a_{ik} x_k(t) \tag{4}$$

Although H-systems have the same order of complexity $O(N^2)$ as S-systems, they have a significant advantage. For S-systems, the equilibrium positions are too fixed, which becomes obvious even for the one-dimensional case $\dot{x} = x^{\alpha} - x^{\beta}$. Furthermore, H-systems can be clearly motivated by dividing the equations into a constant rate and a linear term as in case of the weight matrices and extending this with a nonlinear term.

1.6 Model Quality

For evaluating the quality or fitness of the inferred models, i.e. the similarity of the time dynamics between the experimental and the simulated data, the following equation can be used, referred to as the standard squared error (SSE):

$$f_{SSE} = \sum_{i=1}^{N} \sum_{k=1}^{T} \left(\hat{x}(t_k)_i - x(t_k)_i\right)^2 \tag{5}$$

Here, N is the total number of components of the system, T is the number of sampling points taken from the experimental time series and $\hat{x}$ and x distinguish between estimated data of the simulated model and data sampled in the experiment. This is the most straight-forward way of comparing the system's output to the output of the simulated model. However, in this publication, the relative squared error or relative standard error (RSE) was used, to adapt the error to the maximum amplitude of the system's dynamics:

$$f_{RSE} = \sum_{i=1}^{N} \sum_{k=1}^{T} \left\{ \left(\frac{\hat{x}(t_k)_i - x(t_k)_i}{x(t_k)_i} \right)^2 \right\} \tag{6}$$

In the current implementation, a Runge-Kutta method of fourth order is used to integrate the differential equation systems to simulate the mathematical model [6]. The overall optimization problem is then to minimize the fitness values of objective function f_{RSE}. This fitness function has already been used by several publications on this problem and was thus selected to evaluate the model quality.

2. COMPARISON OF MODELS

The first step to infer a dynamical system is the appropriate choice of mathematical model that is used to represent the underlying dependencies. This publication gives an overview of the performance of those models on the inference problem.

To examine the behavior of the different model types, the benchmark systems described above were used to evaluate, whether the models are able to model time dynamics created not only with the identical type of model but also with

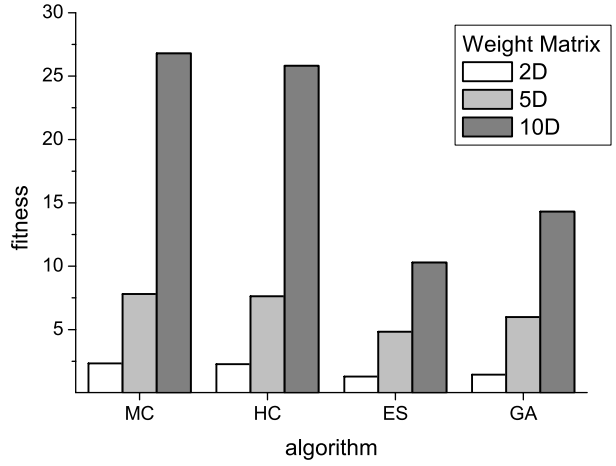

Figure 1: Performance of the standard optimization algorithms on weight matrices.

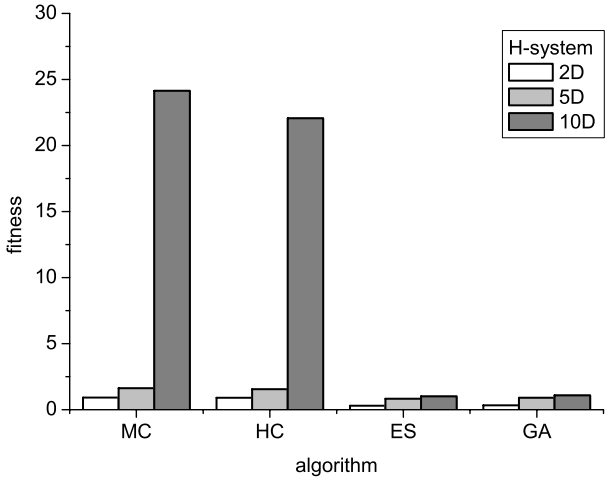

Figure 2: Performance of the standard optimization algorithms on H-systems.

different types. The following comparison results are separately evaluated for the different levels of dimensionality. In the first test case, the models were used to redisplay the dynamic systems created with the same type of mathematical formula. In the second test case, each type was evaluated to represent the dissimilar models.

For the evaluation of the different model types, four standard algorithms were chosen to form a set of representative optimization methods:

- Monte-Carlo search (MC): standard real-value encoding Monte-Carlo search.
- Hill climbing (HC): single-start hill climber with real-value variable encoding and a fixed mutation step size of 0.2.
- Evolution strategy (ES): (μ,λ)-ES with $\mu = 5$ parents and $\lambda = 10$ offspring, global mutation ($p_m = 0.8$), discrete uniform crossover ($p_c = 0.2$), and best selection.
- Genetic Algorithm (GA): binary representation with a population size of 250, one-point crossover ($p_c = 0.8$), invert-bit mutation ($p_m = 0.2$), and tournament selection with a tournament group size of $t_{group} = 8$.

Beside these settings, the evolution strategies used an extension, where the initial population is significantly larger than the standard λ-sized population. In the current implementation, 250 individuals were created to form the initial population to be better comparable to the other algorithm with much larger populations. This extension is crucial, because it increases the chance of finding stable models in the primordial population. The evolution strategies then decrease the size of the population by the standard best selection method and continue with the number of individuals as given above.

For these preliminary comparison experiments, each algorithm used the default settings suggested in the literature. These standard algorithms may not perform as best as they can, but they are sufficient for a comparison of the different model types. Further on, the algorithms terminated after 50,000 fitness evaluations and each experiment was repeated 20 times to gain a sound statistical interpretation.

The fitness values listed in the following are averaged over the experiments of one dimensionality group. The dimension groups consist of the four behavior classes of one level of dimension. For evaluating the different groups, the best-of-run results of each experiment of the multi-runs was used to gain the averaged resulting fitness value.

2.1 Identical Mapping

The first test case aims to evaluate the inference performance of the three major model types, namely weight matrices, H-systems, and S-systems. For this test case, the benchmark data was divided into three different sets, grouping the data according to the type of model that was used to create them. The groups consisted of the corresponding data sets of each model class. Thus, each group contained 12 data sets of three dimensions and four basic behaviors. The different test groups were then inferred with the same type of model that was used to generate the data.

Figures 1 - 3 give the averaged results of the standard optimization algorithms on the inference problem. As can be seen, the Monte-Carlo search and the hill climber did not perform as well as the GA or the ES. This obviously results from the multi-modal nature of the search space, where the MC failed to converge and the single-start HC almost always converged prematurely to local optima. Both, ES and GA, found satisfying solutions to the inference problem with respect to the given fitness function. Furthermore, the ES converged to slightly better results than the GA, possibly due to the ability to search greedily for a very good solution.

Overall, the different mathematical approaches seem able to model the time dynamics of data sets that were created with the same type of model. However, the algorithms faced some problems to find good solutions to the weight matrix data sets. Even the optimization with more sophisticated ES and GA yielded only solutions with a moderate level of

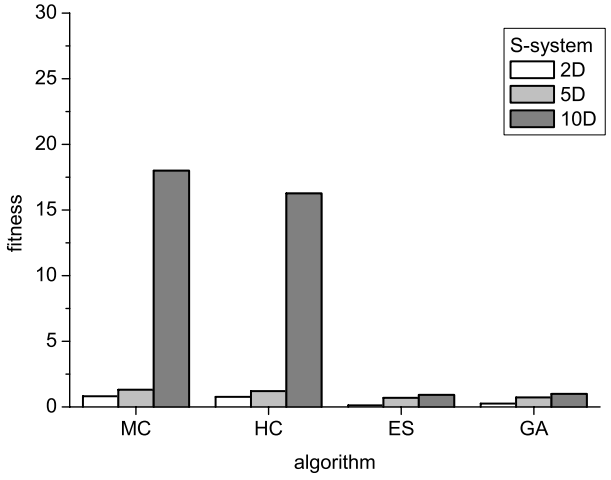

Figure 3: Performance of the standard optimization algorithms on S-systems.

quality. Both other model types, namely the H-system and S-system, were more effective while inferring the dynamic systems, with a slight advantage for the S-systems, which yielded better fitness values; both outperformed the weight matrices in this test scenario especially in combination with the two evolutionary algorithms.

2.2 Cross-Model Mapping

The second test case is more important than the proof of modeling ability described in the identical mapping section. This test scenario aims to evaluate the performance of the three model types to infer data sets that were created with different models, i.e. infer dynamics with different mathematical properties. Thus, this test case evaluates the ability to generalize.

For the actual comparison experiments, the four mentioned algorithms were used to infer the data sets that do not correspond to the current model type. Again, the best-of-run results were averaged over the four basic behavior classes and the three alternative models for each dimension. And for statistical reasons, 20 multi-runs were performed. As the Monte-Carlo search (MC) and the hill climber (HC) always yielded significantly worse results than the evolution strategy (ES) or the genetic algorithm (GA), only the ES and the GA results are displayed in the results.

Figure 4 shows the matrix of results of the cross-model experiments. Each row corresponds to the inference model type, i.e. the model type that was used to infer the data sets. Each column refers to the target model, i.e. the type of model that was used to create the data sets. The main diagonal contains the results of the identical mapping.

The most obvious conclusion that can be drawn from the plots is that S-systems perform best in cross-model environments. They are flexible enough to model time dynamics that were simulated with a different type of underlying mathematical formulation and thus with different mathematical properties. Further on, H-systems are also able to model the time dynamics of other model types but not as well as S-systems. And finally, weight matrices are performing worst in this test case. Due to their quasi-linear nature, they fail to catch the highly non-linear dynamics especially in the oscillating benchmarks.

2.3 Conclusions

In this paper, we introduced the framework JCell that was developed to allow users to evaluate different algorithms on a set of well-defined benchmark systems to obtain comparable results. Several optimization algorithms together with a variety of mathematical models are implemented to study the performance on the inference problem. We systematically examined the performance of standard evolutionary algorithms on the problem of inferring gene regulatory networks from microarray data with different mathematical formulations. The comprehensive study was performed on an Opteron cluster with 16 dualcore CPUs with 2,2GHz and 2GB RAM per node. The overall computation time amounted approximately 320h.

Three observations can be made from the results of the previous test cases, namely the identical mapping and the cross-model test:

1. The results of the Monte-Carlo search showed that the problem is not trivial and cannot be solved by randomly choosing a solution.

2. The solutions space of the inference problem is highly multi-modal. The hill climber was not able to find good solutions with respect to the given optimization target; probably, it prematurely converged to local optima. The genetic algorithm and the evolution strategy on the other hand solved the optimization problem comparably well, resulting in models that fit the time course dynamics of the test data sets well.

3. All model types were able to infer data sets that were simulated with the same model type except for the weight matrices, which resulted in models that showed lower levels of quality than the other two model types. This is most probably to the highly non-linear dynamics of the benchmark systems, which can hardly be represented with a linear model. The evolutionary algorithms (ES and GA) found almost always very good solutions for both models, S-systems and H-systems. S-systems performed best in the identical mapping test case, closely followed by the H-systems.

4. The cross-model validation experiment showed clearly, that the parameter optimization algorithms using weight matrices as the underlying mathematical model failed to find good solutions with respect to the fitness function. Whereas EAs using H-systems and S-systems almost always found good solutions. S-systems were again slightly superior to H-systems.

An important conclusion that can be drawn from these three observations is that due to their generic nature, S-systems and H-systems are well suited to model the dynamic behavior of systems with different mathematical properties. It can be deduced that they are a type of model that is most likely better suited to represent biological and biochemical systems than any other model type that was examined, such as weight matrices.

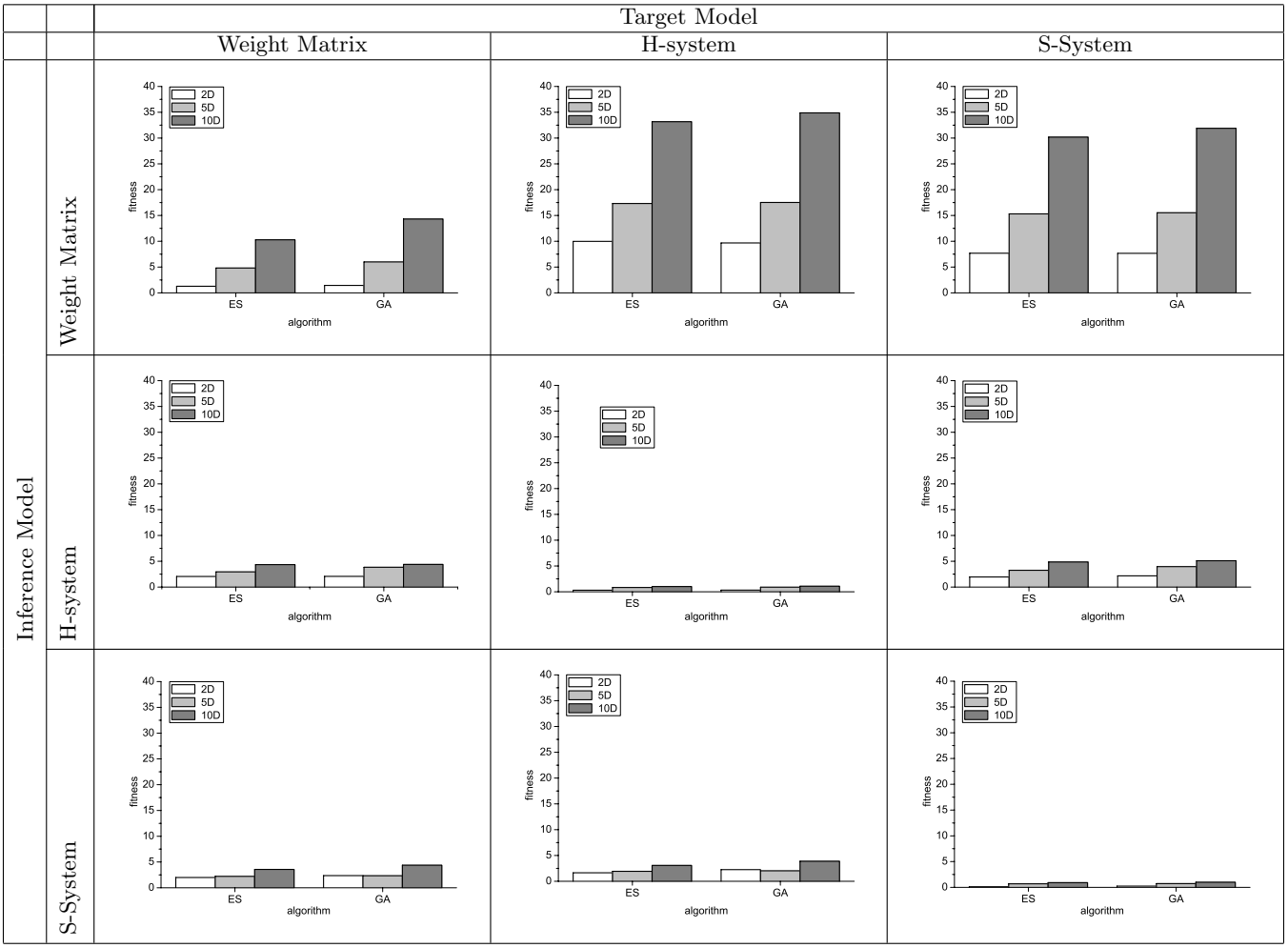

Figure 4: Matrix of the cross-model mapping experiments. The columns correspond to the target model, i.e. the type of model that was used to create the data sets, whereas the rows, corresponds to the inference model type, i.e. the model type that was used to infer the data sets.

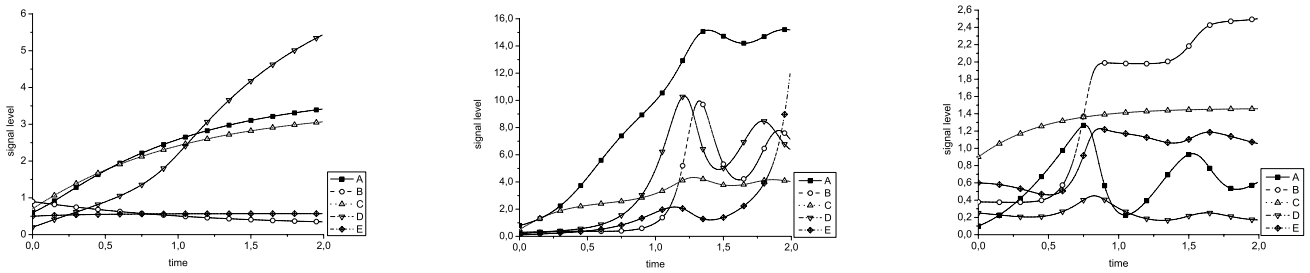

Figure 5: Examples of the discontinuous dynamics of the three mathematical models, weight matrix, H-system, and S-system, from left to right.

However, the problem of ambiguity still remains. Multiple sets of model parameters fit the given data with very good confidence, but with totally different system structures. Without any additional data, this problem can only be addressed by incorporating biological knowledge as described for example in [3, 9].

In a related publication [10], we present a comprehensive study of evolutionary algorithms and carefully tuning different evolutionary algorithms, namely Monte-Carlo search,(multi-start) hill climber (MS-HC), binary genetic algorithm (binGA), real-valued genetic algorithm (realGA), standard evolution strategy (stdES), evolution strategy with CMA mutation (cmaES), differential evolution (DE), and particle swarm optimization on the problem of inferring S-systems of different mathematical properties and size.

3. ACKNOWLEDGMENTS

This work was supported by the National Genome Research Network (NGFN-II) in Germany under contract number 0313323.

4. ADDITIONAL AUTHORS

Additional authors: Jochen Supper (Centre for Bioinformatics Tübingen, Nora Speer (Centre for Bioinformatics Tübingen, Klaus Beyreuther (Centre for Bioinformatics Tübingen, and Andreas Zell (Centre for Bioinformatics Tübingen).

5. REFERENCES

[1] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, January 2002.

[2] K. Hadeler. Gedanken zur Parameteridentifikation. Personal Communication, 2003.

[3] 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*, pages 104 –113, 2003.

[4] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita. Dynamic modeling of genetic networks using genetic algorithm and S-system. *Bioinformatics*, 19(5):643–650, 2003.

[5] S. Kimura, K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya. Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. *Bioinformatics*, 21(7):1154–1163, 2005.

[6] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C: The Art of scientific computing*. Cambridge University Press, 1992.

[7] M. A. Savageau. 20 years of S-systems. In *Canonical Nonlinear Modeling. S-systems Approach to Understand Complexity*, pages 1–44, 1991.

[8] C. Spieth. JCell : A java framework for inferring genetic networks. Technical Report WSI-2005-07, Centre for Bioinformatics Tübingen, University of Tübingen, 2005.

[9] C. Spieth, F. Streichert, N. Speer, and A. Zell. Inferring regulatory systems with noisy pathway information. In *Proceedings of the German Conference on Bioinformatics*, 2005.

[10] C. Spieth, R. Worzischek, F. Streichert, J. Supper, N. Speer, and A. Zell. Comparing evolutionary algorithms on the problem of network inference. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2006.

[11] 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.

[12] 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*, volume 99, pages 6163–6168, 2002.