Eprint

Citation: Marbach D, Mattiussi C, and Floreano D (2009) Replaying the Evolutionary Tape: Biomimetic Reverse Engineering of Gene Networks. *Ann N Y Acad Sci*, 1158:234–245.

This eprint is identical in content to the postprint of this article, which is available at www.blackwell-synergy.com and annalsnyas.org. Related articles are available at: http://lis.epfl.ch/grn

Replaying the Evolutionary Tape: Biomimetic Reverse Engineering of Gene Networks

Daniel Marbach, Claudio Mattiussi, and Dario Floreano*

Laboratory of Intelligent Systems, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

In this paper, we suggest a new approach for reverse engineering gene regulatory networks, which consists of using a reconstruction process that is similar to the evolutionary process that created these networks. The aim is to integrate prior knowledge into the reverse engineering procedure, thus biasing the search towards biologically plausible solutions. To this end, we propose an evolutionary method that abstracts and mimics the natural evolution of gene regulatory networks. Our method can be used with a wide range of nonlinear dynamical models. This allows us to explore novel model types such as the log-sigmoid model introduced here. We apply the biomimetic method to a gold standard dataset from an in vivo gene network. The obtained results won a reverse engineering competition of the second DREAM conference.

DREAM challenge | gene regulatory networks | reverse engineering | prior knowledge

Introduction

The goal of reverse engineering is to unravel unknown cellular networks from quantitative experimental data. Conceptually, there are three basic entities involved in the reverse engineering procedure.[1] In the case of gene network reverse engineering, these entities are: 1) A dataset of gene expression measurements; 2) A mathematical model of gene regulation; 3) A search method that can find, within the framework of the model, the networks that are most probable given the dataset and possibly some prior knowledge. These three aspects must be balanced for effective reverse engineering.

The quantity and quality of available data strongly influences the choice of a suitable model type and reverse engineering method. For example, microarrays simultaneously assess the expression of thousands of genes. This results in an extremely high-dimensional search space. For such large-scale reverse engineering, statistical methods[2] or regression techniques relying on relatively simple dynamical models based on first-order approximations of gene expression dynamics[3,4] are typically used. Here, we are interested in inference methods that target smaller networks, but use more accurate and biologically plausible, nonlinear gene network models. This generally requires more selective and accurate measurement of the expression level of single genes, for example using quantitative Polymerase Chain Reaction (q-PCR) or fluorescent transcriptional reporters. These technologies are advancing at a fast pace. As the quality of available data improves, we believe it timely to explore the possibility of using more detailed phenomenological model types—enabling a more faithful reconstruction of the gene network—than commonly used models of gene regulation such as the linear model,[5–7] the log-linear model,[3,4,8] the sigmoid model,[9–13] or S-Systems.[14,15] As a first step in this direction, we introduce a log-sigmoid gene network model that can approximate a broader range of gene regulation functions than the standard sigmoid model.

More sophisticated, nonlinear model types require the conception of adequate reverse engineering algorithms that can navigate the more complex search space. An important issue in the design of such algorithms is the incorporation of prior knowledge in order to 'guide' the search towards biologically plausible solutions.[16] This is especially important when the reverse engineering problem is underdetermined by the available data.[17] Because biological networks are generally sparse, most state-of-the-art reverse engineering methods include an explicit bias towards sparse networks,[3,6,7,11,12] for example by limiting the maximum number of connections per gene.

Here, we advocate a new approach for embedding prior knowledge in a reverse engineering method. Instead of formulating *ad hoc* constraints, we use an algorithm that bears close similarity with the way in which gene regulatory networks are thought to evolve in nature. Standard genetic algorithms have been used previously for gene network inference.[15] Corne and Pridgeon have argued that genetic algorithms may be particularly well suited for reverse engineering biological networks, which are themselves a product of an evolutionary process.[18] The approach proposed here goes further by extending the evolutionary algorithm with a biomimetic artificial genome (Analog Genetic Encoding [AGE]), which mimics the encoding of biological gene networks. By reproducing—at a certain level of abstraction—the structure and evolutionary con-

*To whom correspondence may be addressed. E-mail: dario.floreano@epfl.ch

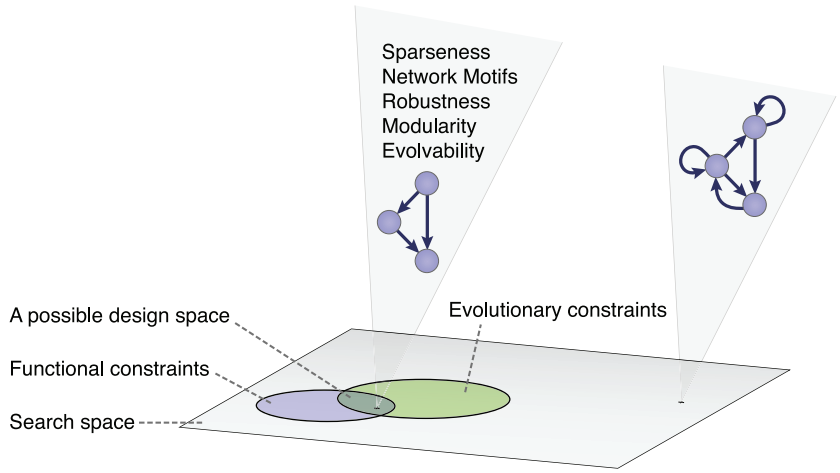

Fig. 1 Evolutionary and functional constraints shape the 'design space' of biological networks,[31] which often have 'design features' such as sparseness, network motifs, robustness, etc. For example, some motifs (left) may be encountered more frequently in a particular network than other topologies (right). By 'replaying the evolutionary tape', the biomimetic reverse engineering approach aims at reproducing the evolutionary constraints of biological networks, thereby partly biasing the search towards nature's design space

straints of biological gene networks, the reverse engineering process can be biased towards biologically plausible solutions, thereby improving the accuracy of predictions (Figure 1).

Sequence-based artificial genomes similar to AGE were originally proposed for modeling evolutionary dynamics of gene networks[19,20] and have been applied, for example, in artificial ontogeny[21] and artificial chemistries.[32] In contrast to these artificial genomes, AGE has been specifically designed for the evolutionary synthesis and reverse engineering of dynamical networks, and it permits the evolution of various dynamical gene network models with real-valued parameters. We have previously shown that AGE displays state-of-the-art optimization performance in other network synthesis problems than reverse engineering.[22,23] Preliminary results on gene network inference from simulated, noise-free steady-state data were reported in.[24]

Here, we use a more realistic gene model and assess our method on the DREAM five-gene-network challenge, a gold standard provided by Cantone et al. (unpublished data) as a reverse engineering competition for the second DREAM conference (New York, 2007). This gold standard consists of a time series dataset obtained from an *in vivo* gene network of known topology. The topology of the network was not disclosed by the organizers prior to submission of this paper. Our predictions, which won the challenge, were thus obtained blinded towards the true topology and assessed by an independent evaluation team of the DREAM challenge organizers.[25]

Results and discussion

Biomimetic reverse engineering method. We have developed a biomimetic reverse engineering algorithm that can be used with a large class of gene network models. This class of models abstracts gene networks as nonlinear dynamical systems described by a system of ordinary differential equations, where genes are characterized by a list of internal parameters $\mathbf{p}$ (e.g. decay rate, maximum transcription rate, etc.) and the interactions between pairs of genes are characterized by a single parameter called weight w. The topology of the network and the values of all numerical parameters are encoded in an artificial genome similarly to the way biological gene networks are encoded in the genome in nature. We would like to stress that the goal is not building a detailed model of the workings of gene networks, but abstracting some key features believed to be important in their evolution. These features are illustrated in Figure 2.

The biomimetic genome, complemented with a set of biologically inspired genetic mutation and crossover operators and a process of artificial evolution, allows us to evolve gene networks *in silico* according to a given fitness criterion (the *fitness function*) as described in Methods. The fitness function measures how well the experimental data is reproduced by an evolved network in simulation, using a sum of squares error.

The biomimetic genome implies a bias towards sparse networks because—as in biological gene networks—regulatory interactions need to be actively evolved through creation of appropriate motifs ('binding sites') in the *cis*- and *trans*-acting sequences. Links tend to be pruned by random mutations and only the links under selective pressure (i.e., those that contribute positively to the fitness) are maintained. Thus, given the choice between a highly connected and a sparse network that fit the data equally well, the biomimetic algorithm evolves the sparse network with a higher probability—consistent with our prior knowledge that biological networks are generally sparse.

The log-sigmoid gene network model. We have applied a principled approach to design a new type of gene network model (manuscript in preparation). In short, we have considered the complete set of biologically conceivable *cis*-regulatory input functions (the input function of a gene describes the combined effect of its regulators on the transcription rate) and compared how well different phenomenological models approximate these functions. We found that existing phenomenolog-

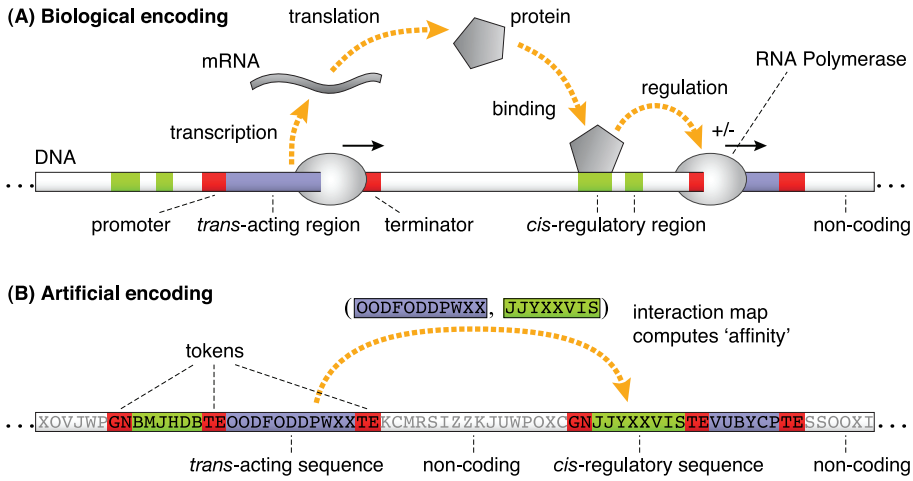

Fig. 2 Implicit encoding of genetic interactions in the biological and the artificial genome. **(A)** In a cell, the regulatory interaction between two genes is the result of a biochemical process that depends among other things on the coding region (*trans*-acting region) of the first gene, which encodes the characteristics of the regulatory protein, and the *cis*-regulatory region of the second gene, which contains the potential binding sites for the regulatory protein. **(B)** The artificial encoding abstracts the following aspects of the biological encoding: 1) *The genome is a sequence of nucleotides*. The artificial genome is constituted by one or more chromosomes, which are sequences of characters (A–Z). 2) *Genes can be located anywhere in the genome*. The beginning and the end of genes are marked by special motifs called 'tokens' (GN and TE) analogous to promoters and terminators of biological genes. 3) *Implicit encoding of regulatory interactions*. The potential regulatory interaction between two genes is not encoded *explicitly* in the genome. Instead, genes have a *cis*-regulatory sequence and a *trans*-acting sequence, which may interact via an interaction map that computes an 'affinity' (interaction strength) as described in Methods

ical models used for reverse engineering, such as the standard sigmoid model,[9–13] typically perform well only on input functions similar to a Boolean OR-gate. The log-sigmoid model introduced in Methods circumvents this limitation. In summary, the following points make the log-sigmoid model an interesting choice:

- The log-sigmoid model is identical to the standard sigmoid model, except that the logarithm of the inputs is used. This makes it compatible with many reverse engineering methods originally developed for the standard sigmoid model.
- In contrast to the standard sigmoid model, it approximates AND-type and OR-type gene regulation functions equally well (unpublished results).
- It is equivalent to a Hill-type model and the weights w_{ij} can be interpreted as the Hill coefficients of the regulators (see Methods).
- Some types of gene expression data are naturally treated in log space, as discussed in the next section.

Predictions and confidence levels. It is clear that with the noisy and relatively small datasets available, it is impossible to infer regulatory links with 100% certainty. Within this context, the goal of reverse engineering is not identifying a single 'true' network, but rather making a set of predictions of regulatory links, which can have different confidence levels assigned. Such a list of predictions is the official format in which reverse engineering results are to be submitted to the DREAM challenge.

If the reverse engineering problem is underdetermined by the available data, any run of a stochastic inference method (as the evolutionary method proposed here) generally converges to a different network. From N runs, we thus get a set of N different inferred networks. Analyzing such an ensemble of inferred networks to make predictions and assign confidence levels is not trivial. For now, we simply define the confidence level of a regulatory link as the fraction of times that it was present in the set of inferred networks. Enhancing and inhibitory links are counted separately, i.e., the predictions are signed. A detailed discussion of different methods to extract predictions and confidence levels from ensembles of inferred networks is the focus of our companion paper in this issue.[26]

In silico and in vivo benchmarks. To allow objective comparison with other methods, we have applied our algorithm to a gold standard dataset provided for the five-gene-network reverse engineering challenge of the second DREAM conference (Cantone et al., unpublished data). This dataset consists of two time series of 15 and 11 samples respectively, and was obtained from an *in vivo* gene network (henceforth referred to as *target network*) using q-PCR. The goal is to predict the topology of the target network from this data. The true topology of the target network was not yet disclosed by the organizers of the DREAM challenge at time of submission of this paper.

Here, we can not yet include a more detailed description of the *in vivo* gene network and the challenge because this information has not yet been published. For this reason, the q-PCR data and the true network structure are not shown in the discussion of the results below. We will supplement more information as soon as possible on our website (http://lis.epfl.ch/grn).

As a further test case we have constructed an *in silico* five-gene network, from which we generated two time series with the same number of samples as the DREAM challenge dataset described above. Different levels of log-normal noise were added to the simulated data. The topology of the *in silico* target network is a loop of inhibitory connections (Figure 3A). The *in silico* gene network in SBML format and the corresponding datasets are available from the authors upon

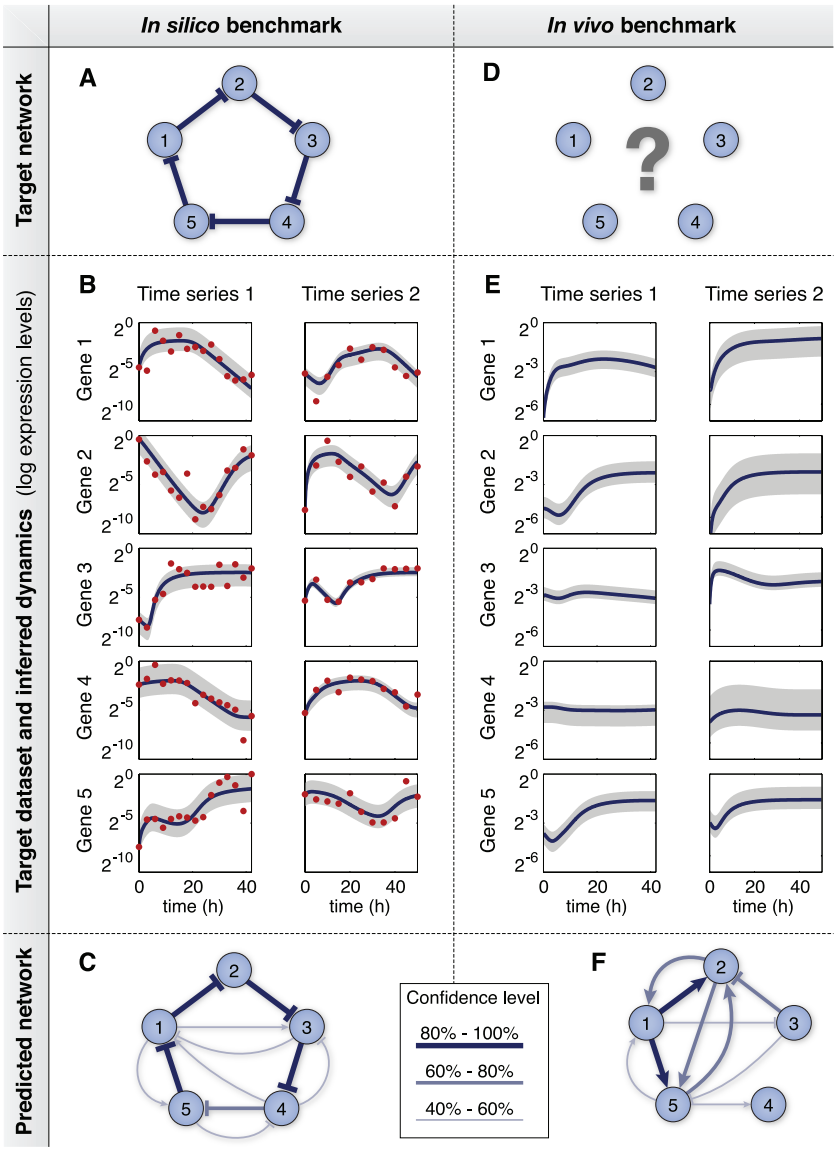

Fig. 3 (A-C) In silico benchmark. **(A)** The *in silico* target network is a loop of five inhibitory interactions. **(B)** Normalized gene expression levels—plotted on a logarithmic scale—for the two time series. Noisy data, generated from the in silico target network, is used as input for the reverse engineering method (points). The time course of the inferred network fits the input data without overfitting to noise. **(C)** The regulatory links of the target network were correctly predicted with high confidence levels. Arrows are enhancing, T-ends denote inhibitory interactions.
(D-F) In vivo DREAM challenge. **(D)** The topology of the *in vivo* target network has not yet been disclosed prior to submission of this paper. **(E)** The inferred network dynamics plotted as normalized, negative q-PCR log expression ratios. The DREAM challenge dataset has not yet been released for publication. For this reason, the accuracy of the data fit is qualitatively shown with the shaded areas. For any time series of a gene, over 90% of all data points fall within this area. **(F)** Predicted network topology

request.

We assume log-normal noise on the data because microarrays and q-PCR assess gene expression on a logarithmic scale. Hence, the measurement error is expected to be approximately log-normal. Furthermore, there is experimental evidence that biological noise in gene regulation also has a log-normal distribution.[27] For this reason, the reverse engineering algorithm fits the models to the original q-PCR log-expression ratios, without first transforming the data to a linear scale (see Methods).

In silico benchmark results. We first applied the reverse engineering method to the *in silico* benchmark at different levels of noise. The results obtained from a batch of 25 runs on the dataset with log-normal noise of standard deviation 1.0 are shown in Figure 3. All runs seemed to fit the noisy data reasonably—the best run with a mean square error of 0.76 (Figure 3B) and the worst run with a mean square error of 1.04. Despite the strong noise in the dataset, the inferred network dynamics are close to the noise-free network dynamics of the target network (result not shown). Four out of the five inhibitory links of the target network were correctly inferred in over 95% of the runs. The fifth link was also correctly pre-

	Directed pos.	
	Team	AUC
1	Team 58	0.72
2	**AGE**	**0.43**
3	Team 110	0.41
4	Team 40	0.41
5	Team 107	0.35
6	Team 60	0.17
7	Team 119	0.17

Directed neg.	
Team	AUC
AGE	**0.14**
Team 107	0.13
Team 58	0.12
Team 110	0.11
Team 40	0.06
Team 60	0.06
Team 119	0.06

Undirected Unsigned	
Team	AUC
AGE	**0.87**
Team 40	0.79
Team 80	0.78
Team 110	0.48

Table 1 DREAM five-gene-network challenge results for directed excitatory, directed inhibitory, and undirected unsigned link predictions.

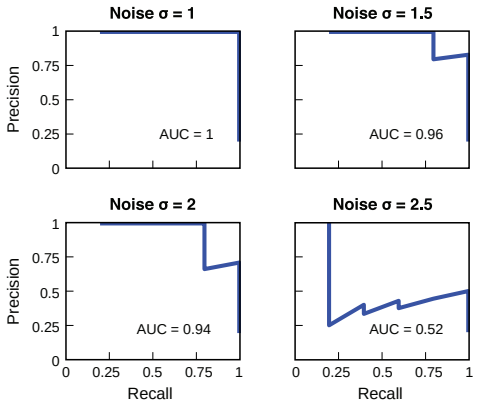

Fig. 4 Precision versus recall curves of the predictions for the *in silico* benchmark at different levels of noise (standard deviation $\sigma = 1 \ldots 2.5$). Accuracy of predictions is measured by the area under the curve (AUC)

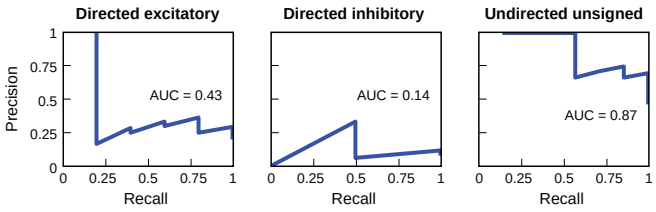

Fig. 5 Precision versus recall curves of our results for the DREAM challenge five-gene-network in the categories of directed excitatory, directed inhibitory, and undirected unsigned link predictions

dicted, though with a lower confidence level of 76%. All other links had confidence levels below 60% (Figure 3C).

When adding more noise to the dataset (standard deviation 1.5), four of the correct links were still identified in over 90% of the runs, but two links were now incorrectly predicted with the same confidence level as the fifth link of the target network. With noise of standard deviation above 2.0, the predictions were not accurate anymore. Precision versus recall curves (see Methods) for inhibitory link predictions (there are no excitatory links in this benchmark) are given in Figure 4.

In vivo DREAM challenge results. We launched 50 runs of our reverse engineering algorithm on the DREAM five-gene-network dataset. The inferred networks fit the data with a mean square error between 0.36 (best run, Figure 3E) and 0.50 (worst run). The network topology prediction is given in Figure 3F. The predictions have significantly lower confidence levels than those of the *in silico* benchmark.

Predictions for excitatory and inhibitory connections were evaluated separately by the DREAM organizers. The corresponding precision versus recall curves are shown in Figure 5. In addition, we derived undirected unsigned predictions from the directed signed predictions as described in our companion paper.[26] The accuracy of predictions of all participating teams was ranked according to the area under the precision versus recall curve (AUC, see Methods). As shown in Table 1, the biomimetic method based on AGE has a very competitive performance overall (the identities of other teams have not yet been disclosed). The result of highest statistical significance (according to the test mentioned below) of the challenge was obtained in the undirected unsigned category by our method. Here, we focus our discussion on the directed signed predictions, which are the primary output of our method. The other categories are discussed in our companion paper.[26] Further results are available on the DREAM website (http://wiki.c2b2.columbia.edu/dream, we are team 55).

Despite the competitiveness of our results, it has to be stressed that the accuracy of the predictions of all participating teams (including us) is low. The DREAM organizers have analyzed results based on a null hypothesis of a randomized gold standard and found that predictions of inhibitory links are not statistically significant at a level of 5%. Even though the results of the other categories are significant according to this test, the inferred networks are not close to the true topology.

The fact that several state-of-the-art reverse engineering methods, applied by different participating teams, have failed to predict the true topology with reasonable accuracy gives strong reason to believe that the network is not identifiable from the provided dataset. Even though the two time series of the dataset were obtained after application of the same initial perturbation, the dynamics are very different in the two cases (data not shown). Details on the experiments have not yet been disclosed. If the measures were done on single cells, these differences may be due to intrinsic noise (stochasticity in gene expression) and/or extrinsic noise (the aggregate effect of variations in other cellular components).[27] In this case, a stochastic model may be more suitable than the determinis-

tic model used here. However, if the measures were obtained from samples containing many cells, the noise in gene regulation is averaged out and the variations in the two time series may be due to different experimental conditions. In this case, the deterministic model may be adequate, but the experimental condition that underlies the variation would have to be included in the model. This ambiguity underlines the importance of considering the nature of the experimental data in the choice of the model and the reverse engineering method.

Conclusions

We have presented a biomimetic approach for the design of gene network reverse engineering methods. By taking inspiration from the mechanisms that enable the evolution of complex gene regulatory networks in nature, we have designed an artificial genome (AGE) that permits simultaneous inference of network structure and numerical parameter values with different types of nonlinear models.

The reverse engineering benchmarks considered here are underdetermined by the available data. Hence, individual runs of the evolutionary algorithm converge to different networks that fit the data approximately equally well. Yet, we have shown that by considering the set of inferred networks obtained from multiple runs, our method successfully predicts the network topology of the *in silico* benchmark in the presence of realistic levels of noise.

The biomimetic method based on AGE was the best performer of the *in vivo* five-gene-network DREAM challenge, obtaining the result of highest statistical significance and performing competitive in all categories. However, the accuracy of the predictions submitted by the participants in this challenge (including us) is not satisfactory. We believe that this is due to high levels of noise in the dataset. A more in depth analysis will be possible after the target network and details of the time series experiments will be disclosed.

Someren et al. have proposed to incorporate prior knowledge by formulating criteria for various features of biological gene networks—e.g. sparseness, stability, modularity, and robustness—thus leading to a multi-criterion optimization problem.[16] However, the formulation and weighting of *ad hoc* criteria is difficult in practice. The biomimetic approach circumvents this problem by exploiting a more fundamental prior, namely the fact that biological gene networks originate from an evolutionary process. However, it remains to be shown to what extent mimicking evolutionary constraints of gene networks actually 'guides' the reverse engineering process towards biologically plausible solutions. Here, we have only discussed the resulting bias towards sparse networks. We are currently studying other possible implications of the biomimetic algorithm, for example evolution of robustness, using the *Drosophila* gap gene network[10] as a biological reverse engineering target.

Methods

Artificial genome. The artificial genome is based on Analog Genetic Encoding (AGE). We give here only a brief description, for details and justifications of the different design choices, refer to.[23, 24]

The AGE genome is constituted by one or several sequences of characters (chromosomes) drawn from a genetic alphabet. Here, the genetic alphabet contains 26 letters (`A`–`Z`). As shown in Figure 2, the start of a gene is indicated by the motif `GN`. A gene is composed of sub-sequences, which are delimited by the motif `TE`. Each sub-sequence encodes a feature of the gene model. In the experiments reported here, the log-sigmoid gene model was used. For this specific model, valid genes have the following sub-sequences s, which can be character strings of arbitrary length

$$\text{gene} \;=\; \texttt{GN}\; s_{\text{cis}}\; \texttt{TE}\; s_{\text{trans}}\; \texttt{TE}\; s_m\; \texttt{TE}\; s_b\; \texttt{TE}\; s_\lambda\; \texttt{TE}\;. \qquad (1)$$

The sequences s_{cis} and s_{trans} implicitly encode the regulatory interactions and their strength (see Figure 2). The weight w_{ij}, which measures how strong gene j regulates gene i, is decoded by an interaction map that computes an 'affinity' w_{ij} between the respective sequences, where $w_{ij} = I(s_{\text{trans},j}\,, s_{\text{cis},i})$. The interaction map is based on the local alignment score of the two sequences. Figuratively speaking, the closer the match between two subsequences ('binding sites') of s_{cis} and s_{trans}, the stronger the interaction. The affinity between two sequences may be zero, in which case there is no regulatory link between the two genes. For details, refer to.[24] The exact implementation of the interaction map is not critical. What matters is the implicit nature of the encoding, i.e., the fact that the N^2 possible connections (where N is the network size) are encoded implicitly in only N *cis*- and *trans*-acting elements.

The sequences s_m, s_b, and s_λ encode the gene parameters m, b, and λ of the log-sigmoid model (5). The numerical value is decoded from the character strings using Center of Mass Encoding (CoME), which is a self-adaptive, variable length encoding for real-valued parameters.[28]

Genetic operators and evolutionary algorithm. Apart from the biomimetic genotype and the genetic mutation and crossover operators described below, the evolutionary algorithm is similar to a standard generational genetic algorithm.[29] Starting from a population of randomly initialized genomes,[24] the evolutionary algorithm is run for 50'000 generations, which takes approximately 5 hours on a standard desktop PC (Intel Pentium 4 processor, 1GB memory). The population size is 100. We use elitism, i.e., the best individual is protected from replacement. At each generation, 50 parents are chosen using tournament selection.[29] From the 50 parents, 100 new individuals are created and the following genetic operators are applied probabilistically to randomly chosen parts of the genome:

- *Nucleotide deletion, insertion, and substitution*: A character is removed, inserted, or substituted in the genome with probability 0.001 per character. Random characters from the genetic alphabet are used for insertions and substitutions.
- *Chromosome fragment deletion, transposition, and duplication*: Two points are chosen in a chromosome and the intervening genome fragment is deleted, transferred or copied to another point of the genome with probability 0.01 per chromosome.
- *Crossover*: Chromosomes of parents are recombined with probability 0.5, provided that a homologous crossover point is found.[23]

The choice of the parameters listed above is not critical. They were chosen heuristically based on a series of test runs and the experiences reported in.[23, 24]

Fitness function. The evolved gene networks are evaluated according to how well they reproduce the measured data. Let $\hat{x}_i^{tk}$ denote the estimated gene expression level of gene i at time point t of the k'th time series, obtained by simulating the evolved gene network,[a] and z_i^{tk} the corresponding logarithmic expression level of the measured target dataset (i.e., the negative q-PCR log-expression ratio). The fitness f of the evolved network is then given by the sum of squares error

$$f = \sum_i \sum_t \sum_k (z_i^{tk} - \hat{z}_i^{tk})^2 \,, \quad \text{with } \hat{z}_i^{tk} = \log(\hat{x}_i^{tk}) \,. \qquad (2)$$

I.e., the square error is taken on a logarithmic scale, consistent with our assumption of log-normal noise (see Results and Discussion).

Evaluating the accuracy of predictions. We use the scoring metrics proposed by the organizers of the DREAM challenge[25] to measure the accuracy of a set of predictions. The predictions are ranked according to their confidence level and their accuracy is defined as the area under the precision versus recall (PR) curve. The precision and the recall of the first k predictions are defined as

$$\text{precision}_k = \text{TP}_k/k \qquad (3)$$

$$\text{recall}_k = \text{TP}_k/\text{P} \,, \qquad (4)$$

where TP_k is the number of correct predictions (true positives) up to prediction k, and P is the total number of true links (positives) in the target network. PR curves are drawn by incrementing k from the first until the last element of the ranked list of predictions. The area under the PR curve is computed as described in ref. [30].

The log-sigmoid gene network model. The log-sigmoid model describes the expression level x_i of gene i by

$$\frac{dx_i}{dt} = m_i \cdot \sigma\Big(\sum_{j \in R_i} w_{ij} z_j + b_i \Big) - \lambda_i x_i \,, \qquad (5)$$

$$\text{with } z_k = \log(x_k) \,,$$

where m_i is the maximum transcription rate, b_i is a bias that relates to the basal transcription rate, and λ_i is the degradation rate. R_i is the set of regulators of gene i and w_{ij} represents the regulatory influence of gene j on gene i (positive for enhancers, negative for repressors). The activation function is a sigmoid $\sigma(y) = 1/(1+e^{-y})$. Underlying assumptions of this type of phenomenological modeling approach have been well described, for example, in ref. [10].

The standard sigmoid model[9–13] and the log-sigmoid model are based on the same gene regulation function, except that the latter one takes the inputs on a logarithmic scale. This leads to a Hill-type function

$$\begin{aligned} \sigma\Big(\sum_{j \in R_i} w_{ij} z_j + b_i \Big) &= \sigma\Big(\sum_{j \in R_i} w_{ij} \log(x_j) + b_i \Big) \\ &= \sigma\Big(\log\Big(\prod_{j \in R_i} x_j^{w_{ij}} \Big) + b_i \Big) \\ &= \frac{c_i \prod_{j \in R_i} x_j^{w_{ij}}}{1 + c_i \prod_{j \in R_i} x_j^{w_{ij}}} \,, \end{aligned} \qquad (6)$$

with $c_i \equiv e^{b_i}$. Thus, in the log-sigmoid model the weights w_{ij} may be loosely interpreted as Hill coefficients.

Acknowledgements

We thank Sven Bergmann, Peter Dürr, and Fred Marbach for helpful discussions. This work was supported by the Swiss National Science Foundation, grant no. 200021–112060.

1. Ljung, L. 1999. System identification: Theory for the user. Prentice Hall. Upper Saddle River, NJ.
2. Basso, K., A. A. Margolin, G. Stolovitzky, U. Klein, R. Dalla-Favera & A. Califano. 2005. Reverse engineering of regulatory networks in human B cells. Nat. Genet. 37: 382–390.
3. Gupta, A., J. D. Varner & C. D. Maranas. 2005. Large-scale inference of the transcriptional regulation of bacillus subtilis. Comput. Chem. Eng. 29: 565–576.
4. Di Bernardo, D., M. J. Thompson, T. S. Gardner, S. E. Chobot, E. L. Eastwood, A. P. Wojtovich, S. J. Elliott, S. E. Schaus & J. J. Collins. 2005. Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. Nat. Biotechnol. 23: 377–383.
5. D'Haeseleer, P., X. Wen, S. Fuhrman & R. Somogyi. 1999. Linear modeling of mRNA expression levels during CNS development and injury. Pac. Symp. Biocomput. 1999: 41–52.
6. Gardner, T. S., D. di Bernardo, D. Lorenz & J. J. Collins. 2003. Inferring genetic networks and identifying compound mode of action via expression profiling. Science 301: 102–105.
7. Tegner, J., M. K. S. Yeung, J. Hasty & J. J. Collins. 2003 Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. Proc. Natl. Acad. Sci. U.S.A. 100: 5944–5949.
8. Liao, J. C., R. Boscolo, Y.-L. Yang, L. My Tran, C. Sabatti & V. P. Roychowdhury. 2003. Network component analysis: reconstruction of regulatory signals in biological systems. Proc. Natl. Acad. Sci. U.S.A. 100: 15522–15527.
9. Mjolsness, E., D. H. Sharp & J. Reinitz. 1991. A connectionist model of development. J. Theor. Biol. 152: 429–453.
10. Reinitz, J. & D. H. Sharp. 1995. Mechanism of eve stripe formation. Mech. Dev. 49: 133–158.
11. Weaver, D. C. Modeling regulatory networks with weight matrices. 1999. Pac. Symp. Biocomput. 1999: 112–123.
12. Wahde, M., J. A. Hertz & M. L. Andersson. 2001. Reverse engineering of sparsely connected genetic regulatory networks. *In* 2nd Workshop on Computation of Biochemical Pathways and Genetic Networks. R. Gauges *et al.*, Eds. Logos Verlag. Berlin, Germany.
13. Perkins, T. J., J. Jaeger, J. Reinitz & L. Glass. 2006. Reverse engineering the gap gene network of Drosophila melanogaster. PLoS Comput. Biol. 2: e51.
14. Voit, E. O. & M. A. Savageau. 1987. Accuracy of alternative representations for integrated biochemical systems. Biochemistry 26: 6869–6880.
15. Kimura, S., K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu & A. Konagaya. 2005. Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. Bioinformatics 21: 1154–1163.
16. Van Someren, E. P., L. F. A. Wessels, M. J. T. Reinders & E. Backer. 2003. Multi-criterion optimization for genetic network modeling. Signal Processing 83: 763–775.
17. Jaynes, E. T. 1984. Prior information and ambiguity in inverse problems. SIAM–AMS Proc. 14: 151–166.
18. Corne, D. & C. Pridgeon. 2004. Investigating issues in the reconstructability of genetic regulatory networks. *In* Proc. of the 2004 Congress on Evolutionary Computation. 1: 582–589. IEEE. Piscataway, NJ.
19. Reil, T. 1999. Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. *In* Advances in Artificial Life: Proc. 5th European Conf. (ECAL '99). D. Floreano *et al.*, Eds.: 457–466. Springer. London, UK.

[a] The gene networks are simulated by integrating the system of differential equations (5). The measured expression levels at the first time point are used as initial conditions. Numerical integration is done using the Runge-Kutta-Fehlberg (4,5) method of the GNU Scientific Library (GSL, http://www.gnu.org/software/gsl).

20. Watson, J., N. Geard & J. Wiles. 2004. Towards more biological mutation operators in gene regulation studies. Biosystems 76: 239–248.

21. Bongard, J. 2002. Evolving modular genetic regulatory networks. *In* Proc. of the 2002 Congress on Evolutionary Computation. 2: 1872–1877. IEEE. Piscataway, NJ.

22. Mattiussi, C., D. Marbach, P. Dürr & D. Floreano. 2008. The age of analog networks. AI Mag: to appear.

23. Mattiussi, C. & D. Floreano. 2007. Analog Genetic Encoding for the evolution of circuits and networks. IEEE Trans. Evol. Comput. 11: 596–607.

24. Marbach, D., C. Mattiussi & D. Floreano. 2007. Bio-mimetic evolutionary reverse engineering of genetic regulatory networks. *In* Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics. E. Marchiori *et al.*, Eds.: 155–165. Springer. Berlin, Germany.

25. Stolovitzky, G., D. Monroe & A. Califano. 2007. Dialogue on reverse-engineering assessment and methods: The dream of high-throughput pathway inference. Ann. N.Y. Acad. Sci. 1115: 1–22.

26. Marbach, D., C. Mattiussi & D. Floreano. 2009. Combining Multiple Results of a Reverse Engineering Algorithm: Application to the DREAM Five Gene Network Challenge. Ann. N.Y. Acad. Sci.: in this issue.

27. Rosenfeld, N., J. W. Young, U. Alon, P. S. Swain, and M. B. Elowitz. 2005. Gene regulation at the single-cell level. Science 307: 1962–1965.

28. Mattiussi, C., P. Dürr, and D. Floreano. 2007. Center of mass encoding: A self-adaptive representation with adjustable redundancy for real-valued parameters. *In* Proc. 9th annual Conf. on Genetic and Evolutionary Computation. 1304–1311. ACM. New York, NY.

29. Bäck, T., D. B. Fogel & Z. Michalewicz, Eds. 2000. Evolutionary Computation 1: Basic Algorithms and Operators. Institute of Physics, Bristol.

30. Davis, J. & M. Goadrich. 2006. The relationship between precision-recall and roc curves. *In* ICML '06: Proc. 23rd Intl. Conf. on Machine learning. C.W. Cohen & A. Moore, Eds.: 233–240. ACM. New York, NY.

31. Kitano, H. 2007. Towards a theory of biological robustness. Mol. Syst. Biol. 3: 137.

32. Hintze, A. & C. Adami. 2008. Evolution of complex modular biological networks. PLoS Comput. Biol. 4: e23.