View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Infoscience - École polytechnique fédérale de Lausanne

Eprint

Citation: Marbach D, Mattiussi C, and Floreano D (2009) Combining Multiple Results of a Reverse Engineering Algorithm: Application to the DREAM Five Gene Network Challenge. *Ann N Y Acad Sci*, 1158:102–113.

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

Combining Multiple Results of a Reverse Engineering Algorithm: Application to the DREAM Five Gene Network Challenge

Daniel Marbach, Claudio Mattiussi, and Dario Floreano*

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

The output of reverse engineering methods for biological networks is often not a single network prediction, but an ensemble of networks that are consistent with the experimentally measured data. In this paper, we consider the problem of combining the information contained within such an ensemble in order to (1) make more accurate network predictions and (2) estimate the reliability of these predictions. We review existing methods, discuss their limitations, and point out possible research directions towards more advanced methods for this purpose. The potential of considering ensembles of networks, rather than individual inferred networks, is demonstrated by showing how an ensemble voting method achieved winning performance on the Five Gene Network Challenge of the second DREAM conference (Dialogue on Reverse Engineering Assessment and Methods 2007, New York, NY).

DREAM challenge | ensemble methods | gene regulatory networks | reverse engineering

Introduction

Many reverse engineering methods for biological networks are based on the fitting of a mathematical model to a dataset of experimentally observed activity levels of the network components. Focusing for the sake of concreteness on gene regulatory networks, the input for such a reverse engineering method is a dataset of gene expression measurements, and the output is a network that is consistent with the data and possibly some prior knowledge. However, it is clear that with the typically noisy and relatively small datasets available, there are in general many different networks that are consistent with the data. Some methods identify a unique "best" network from this ensemble according to some additional criteria,[1–3] for example by posing constraints on the connectivity of the network (Fig. 1A). Here, we focus on an alternative approach, which aims at integrating the information contained within ensembles of plausible networks that are consistent with the data and the prior knowledge (Fig. 1B). Even though many methods have been proposed to construct such ensembles of networks (e.g., Monte Carlo techniques,[4] simulated annealing,[5,6] or genetic algorithms[7]) the problem of how to optimally analyze the ensemble in order to estimate the "true" structure of the underlying gene network has received relatively little attention.

In the following, we first motivate the use of ensemble methods in gene network reverse engineering and then proceed by formalizing the problem from a probabilistic perspective. In Section 2, we review existing approaches and describe a simple voting method that we use to process ensembles generated with our biomimetic evolutionary reverse engineering algorithm.[8,9] Finally, we discuss the results of an *in silico* benchmark, and the DREAM *in vivo* Five-Gene-Net reverse engineering competition. Our results show that in the presence of noise, predictions obtained from ensembles of networks are more accurate than any of the individual networks taken alone.

Ensemble methods. The classic example of an ensemble based system in decision making is the popular game show "Who wants to be a millionaire?". When unsure about a question, the contestant has the possibility to either call a friend who he/she knows to be particularly knowledgeable (an "expert"), or to poll the studio audience, which immediately votes on the question. At first sight, one might think that the experts would offer better help than "random crowds of people with nothing better to do on a weekday afternoon than sit in a TV studio".[10] It turns out the opposite is true, the audience giving the correct answer with a surprisingly high accuracy of about 90%, as compared to only 65% for the experts.[10,11] This is just one of many examples where ensembles of diverse individuals outperform a single expert on average.

Consider an ensemble of inferred networks obtained by a gene network reverse engineering method from a dataset of gene expression measurements. Each of these networks is a hypothesis on the true network structure, giving a prediction on the presence or absence of a regulatory link for every pair of genes. Now assume that the prediction of links is correct with probability $p > 0.5$ (better than random guessing) and that the errors between the different networks of the ensemble are uncorrelated. In this case, the prediction obtained from the ensemble by voting (see next Section) is on average more accurate than any of the individual networks of the ensemble.[12]

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

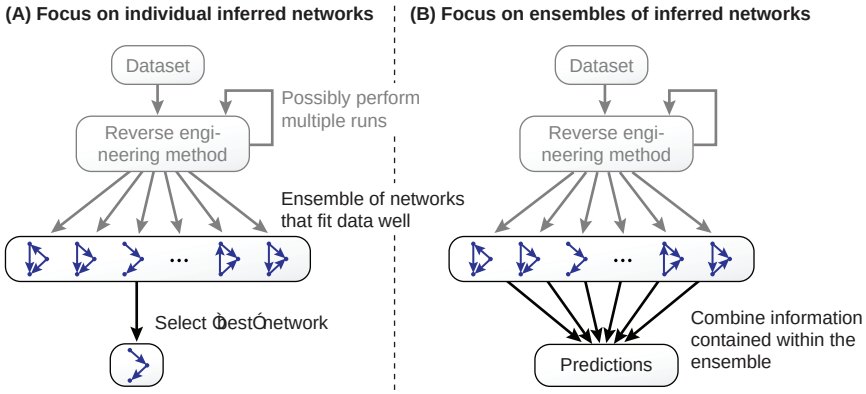

Fig. 1 (A) The most common approach in reverse engineering aims at identifying a single "best" network, e.g., the one with the best data fit and the fewest connections. **(B)** The approach considered here aims at integrating the information from ensembles of "plausible" networks in order to make one or several network predictions and estimate the reliability of these predictions

In practice, the picture is more complex. First, since the different networks of the ensemble are inferred from the same dataset, the error of a given link may be correlated *between* the networks (e.g., all networks have a tendency to wrongly predict a given link). Second, there may be a correlation between the different links *within* the networks (e.g., in a given network, there is either link A or link B, but not both). The simple voting methods typically used in gene network reverse engineering ignore these correlations. Despite these limitations, we will see that in practice even simple ensemble methods are sometimes useful to process the output of reverse engineering methods and often allow to improve the accuracy compared to individual inferred networks of the ensemble.

A probabilistic formalization. The aim of the somewhat simplistic description above was to give an intuitive understanding of the potential advantages of ensemble methods. We now proceed with a more rigorous probabilistic formalization. Assume the reverse engineering target is a gene regulatory network of N genes (henceforth called *target network*). The majority of data-fitting reverse engineering algorithms represents this network by an NxN weight matrix W. The entries w_{ij} of this matrix give the strength of the regulatory effect of gene j on gene i (positive for enhancers, negative for repressors, and zero for no interaction). For simplicity, let's assume that we want to determine just the weight matrix—additional parameters of the genes could be treated in an analogous way. We possess a collection D of noisy observations of the activity of the network, from which a reverse engineering algorithm infers (possibly using multiple runs) an ensemble of tentative networks. Each network has an associated score s_k that indicates how well it fits the data. The ensemble is thus a collection $E = \{(\mathrm{W}_k, s_k)\}$. The problem we consider is how to process the ensemble E to obtain an estimate of the "true" weight matrix W.

From a probabilistic perspective, the aim is to estimate the posterior probability $p(\mathrm{W}|D, I)$ for W, given the dataset D and the prior knowledge I. The ensemble is a collection of samples of this distribution[4,13] (Fig. 2). From this perspective, the goal of reverse engineering is not only to find the solution that maximizes the posterior probability, but rather to integrate the information contained within the complete ensemble to make predictions on the target network.

Methods for combining ensembles of inferred networks

Selecting the "best" network from the ensemble. If the quantity and quality of the data is sufficient to uniquely identify the target network (Fig. 2A), it can be sufficient to simply select the network W_k with the highest score s_k as the most plausible network prediction and discard the information contained within the rest of the ensemble.[5,7,9,14] For example, this is usually done when several independent runs of a stochastic search algorithm converge to this same optimal network, which is then assumed to represent the global optimum and most plausible network prediction.[5]

Analysis of the posterior weight distributions. Another popular approach is to analyze the posterior distribution weight by weight, i.e., without considering possible correlations between the weights. The goal is to qualitatively judge how reliable the different weights are determined by the ensemble. The more closely the collection of inferred values for a specific weight w_{ij} are clustered together, the more reliably it is assumed to be predicted. Whether the ensemble of inferred values for w_{ij} indicates a reliable prediction or not is often judged qualitatively by considering the standard deviation and plotting the distribution.[6,15–17]

In general, this type of qualitative analysis is done in combination with the strategy described above: the network with the best score is chosen as the most plausible network prediction, and the posterior weight distributions are used only to indicate which of the weights are predicted reliably. Instead of taking the weight values of the best network from the ensemble, one may also consider using the average of the predicted values[4] (possibly weighted using the scores s_k).

As discussed in the introduction, using the ensemble average instead of taking simply the best network of the ensemble often gives a more robust prediction. However, averaging multiple networks only makes sense if they agree more or less on a similar network prediction. If the networks of the ensemble are very different, e.g., they fall within two categories as in Fig. 2C, averaging leads to a meaningless "blur" of alternative structures. In this case, a more sophisticated analysis taking into account the joint probability distributions would be required.

Majority voting on the network structure. The quantity and quality of available data is often not sufficient to precisely in-

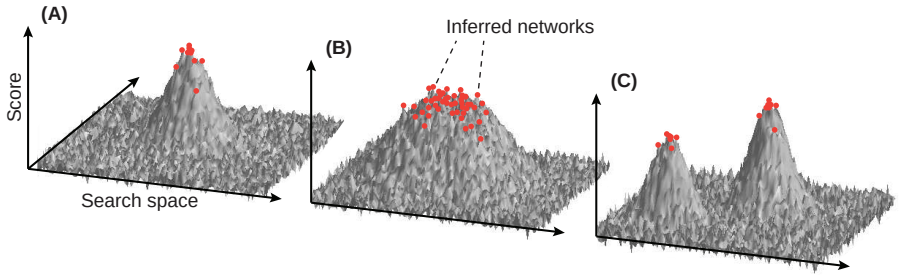

Fig. 2 Schematic representation of possible posterior distributions in a reverse engineering problem. The horizontal plane represents the search space of all possible networks and the vertical axis corresponds to the score (e.g., the posterior probability). The dots are tentative networks inferred by a reverse engineering algorithm. **(A)** The data is sufficient to identify a unique, distinctive global optimum. **(B)** The problem is underdetermined by the available data—there are many different networks that score approximately equally well. **(C)** There are several distinctive classes of networks that fit the data well

fer numerical values for the weights. In this case, one may be satisfied with predicting only the network structure from the ensemble and disregard the numerical values of parameters. A straightforward approach to do so is majority voting (the same method as the audience polling in the game show mentioned above). Every network of the ensemble votes on the classification of a given link as excitatory ($w_{ij} > 0$), inhibitory ($w_{ij} < 0$), or absent ($w_{ij} = 0$ or smaller than a certain threshold). The type of the link is defined by the majority of the votes. In addition, the votes could also be weighted by the scores of the networks.

Unsigned predictions can be treated analogously. For example, Hartemink et al. use weighted voting with Bayesian scores to estimate the probability that a given link is present in the target network.[13] As for the averaging of the weights described in the previous section, the underlying assumption is that regulatory links are predicted independently from each other.

For signed predictions, the basic voting scheme described above may not be optimal because it treats the three possible types of a link (excitatory, inhibitory, and zero) all equal. For example, assume that two links A and B are predicted to be excitatory by 80% of all networks of the ensemble. However, link A is predicted to be zero by the remaining 20%, and link B is predicted to be inhibitory by the remaining 20%. The basic voting would predict both links to be excitatory with equal probability of 0.8. However, one may argue that in this situation 20% of inhibitory votes should be weighted stronger than 20% of zero votes because they directly oppose the excitatory predictions. The voting scheme introduced in the next section addresses this issue.

Signed voting on the network structure. We have devised a simple voting scheme, which we call signed voting, that is suitable for predicting signed regulatory links from an ensemble of inferred networks. In addition, signed voting estimates a confidence level (reliability) for these predictions. In contrast to majority voting, excitatory and inhibitory votes cancel each other out, whereas votes for the absence of a link are neutral.

Assume that network structures are represented by a matrix A, where $a_{ij} = 1$ if the link is excitatory ($w_{ij} > 0$), $a_{ij} = -1$ if the link is inhibitory ($w_{ij} < 0$), and $a_{ij} = 0$ if the link is absent ($w_{ij} = 0$). Suppose we have an ensemble of K networks, and the structure of the k'th network is defined by the matrix A^k (entries a_{ij}^k). We define the signed vote v_{ij} for link a_{ij} as

$$v_{ij} = \frac{\sum_{k=1}^{K} a_{ij}^k}{K} \tag{1}$$

The vote v_{ij} equals 1 if the corresponding link is excitatory in all networks of the ensemble, and -1 if it is inhibitory in all networks. We now define a confidence level l that a given link a_{ij} is excitatory or inhibitory

$$l\{a_{ij} = +1\} \quad := \quad v_{ij} \tag{2}$$
$$l\{a_{ij} = -1\} \quad := \quad -v_{ij} \tag{3}$$

Thus, the confidence level that a link a_{ij} is excitatory is 1 if there is strong supporting evidence (all networks agree on the excitatory connection), it is 0 if there is no supporting evidence (e.g., half of the networks vote inhibitory and half excitatory, or all vote for a zero connection), and it is -1 if there is strong evidence to the contrary (all networks vote for an inhibitory connection).

Note that in contrast to majority voting, the absence of links is not explicitly predicted. Instead, one assumes that a connection is zero if there is no strong evidence for an excitatory or an inhibitory link, i.e., if the absolute value of the signed vote is smaller than some threshold $|v_{ij}| < c$. The smaller c, the more (uncertain) links are included in the final network prediction. As in any classification problem, the choice of the threshold is a tradeoff between the number of false positives (links that are predicted present, but are absent in the target network) and false negatives (links that are predicted zero, but are present in the target network).

Results

We use the same benchmark networks and the same reverse engineering method as described in our companion paper in this volume[8] as an example for demonstrating the potential of ensemble approaches in gene network reverse engineering. Note that the ensemble voting methods used here can in principle be applied to ensembles generated by any other suitable reverse engineering method.

Constructing the ensembles. For generating the ensembles of tentative networks, we use our biomimetic evolutionary reverse engineering method. This method is based on an evolutionary process that bears close similarity with the way in which gene regulatory networks are thought to evolve in nature.[8,9] Traditional genetic algorithms use *direct encodings*,

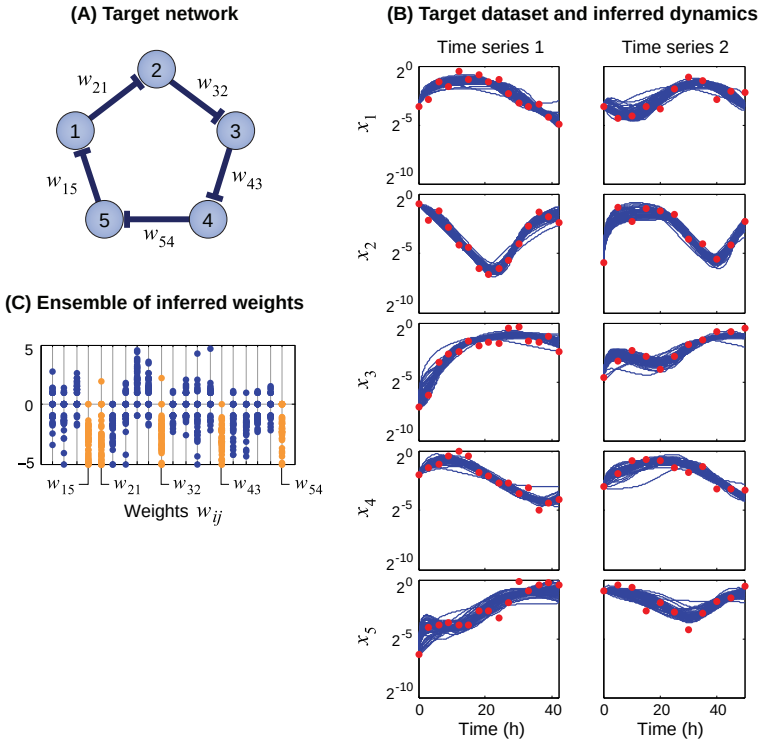

Fig. 3 (A) *In silico* target network structure. **(B)** Normalized gene expression levels—plotted on a logarithmic scale—for the two time series. The points are the input dataset with log-normal noise of standard deviation 0.5. The lines show the data fit by the inferred networks of the ensemble. **(C)** The inferred weights by the networks of the ensemble (the nonzero weights of the target network are highlighted). Despite a good data fit by the majority of networks, the numerical values of their weights vary a lot, which indicates that the problem is underdetermined

where the genome is a sequence of discrete or real-valued numbers. Our approach employs a biomimetic artificial genome (Analog Genetic Encoding [AGE]), which encodes the topology of the networks using a potentially more evolvable *implicit encoding*.[18,19] AGE abstracts and mimics the biological encoding of gene networks in nature, thus allowing for the application of genetic mutation and crossover operators that are functionally equivalent to their biological counterparts. The AGE genome, complemented with a process of artificial evolution, allows us to evolve gene networks *in silico* according to a given fitness criterion. The fitness measures how well the experimental data is reproduced by an evolved network in simulation, using a sum of squares error.[8]

AGE is compatible with a wide range of dynamical gene network models. Here, we use a log-sigmoid model,[8] which describes the expression level x_i of gene i by

$$\frac{dx_i}{dt} = m_i \cdot \sigma\Big(\sum_{j\in R_i} w_{ij} z_j + b_i\Big) - \lambda_i x_i\,, \qquad (4)$$
$$\text{with } z_k = \log(x_k)\,,$$

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. The activation function is a sigmoid $\sigma(y) = 1/(1+e^{-y})$.

The topology and all numerical parameters of the log-sigmoid model are encoded in the AGE genome and evolved with an evolutionary algorithm for 50'000 generations using the same setup as in Ref. [8]. Each run evolves a population of networks that fit the data well. We found that an evolutionary run typically converges to a single network structure, i.e., in the final population the structures of all networks are identical and only the numerical parameter values vary slightly. This is expected, because we do not use techniques that enforce diversity in the population after convergence. Thus, a single population is not well suited to construct the ensemble in our case. Instead, we construct the ensemble from multiple runs of the evolutionary algorithm. From each run, only the network with the best fitness is included in the ensemble. In the experiments reported here, we did 50 runs for every dataset.

Combining the ensembles and evaluating the predictions. We used the evaluation protocol of the DREAM2 challenges to assess the accuracy of network predictions. Inhibitory and excitatory links were predicted separately in DREAM2. A confidence level has to be assigned to each of the N^2 possible links of the network, indicating the degree of belief that this link is excitatory/inhibitory. The network prediction is given by a list of links, ranked according to the confidence levels, and the performance is measured by the area under the precision versus recall curve[20] (AUC). To achieve a maximum AUC score of 1 it is sufficient that the true links of the target network are ranked first in the list. Note that the confidence levels are exclusively used to order the ranked list of link predictions and are not taken into account by the evaluation otherwise.

We compared three strategies to predict the network structure from the ensemble of inferred networks: (1) simply take

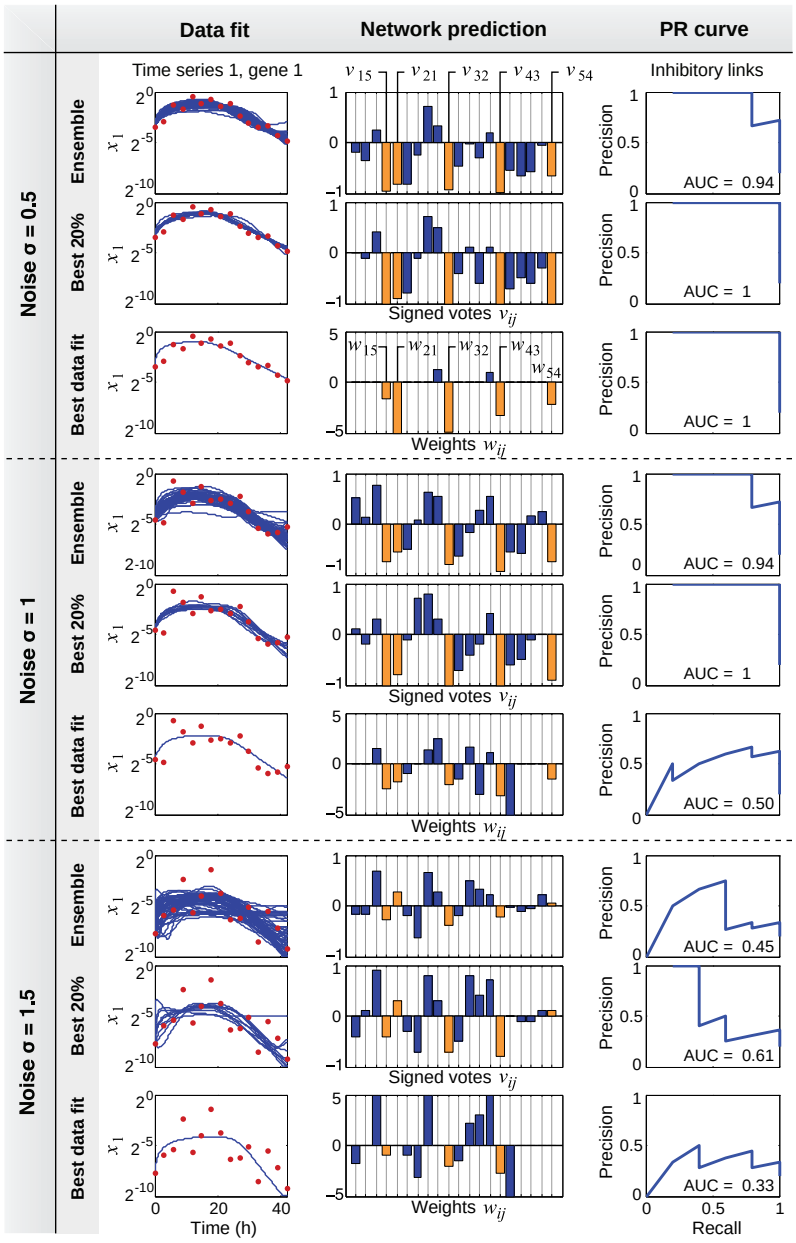

Fig. 4 Comparison of predictions obtained from the network with the best data fit, signed voting by the complete ensemble, and signed voting by the top 20% of the ensemble, on three datasets with different levels of noise (standard deviations 0.5, 1, and 1.5). From the precision versus recall curves and the AUC scores in the third column, it can be seen that at intermediate and strong levels of noise, ensemble voting (especially by the top 20%) predicts the network structure much more accurately than the network with the best data fit

the network with the best data fit from the ensemble and use the strength of the connections (the weights w_{ij}) as confidence levels, (2) use signed voting, and (3) use signed voting, but allow only the M highest scoring network to vote. For the results reported here we used $M = 10$ networks (the top 20% of the ensemble).

Ensemble voting outperforms individual networks on an in silico test case. We first tested the ensemble approach on an *in silico* five gene network. The structure of this network is a loop of inhibitory connections (Fig. 3A) and the dynamics are simulated with the log-sigmoid model (Eqn. 4). We generated two time series of 15 and 11 samples respectively (same number of time points as in the quantitative PCR [q-PCR] dataset described in the next section) and added different levels of log-normal noise (we assume log-normal noise because q-PCR assesses gene expression on a logarithmic scale[8]).

Fig. 3B shows a dataset with log-normal noise of standard deviation 0.5, and the fit by an ensemble of networks, which were inferred with the biomimetic method described above. Besides from few outliners that converged prematurely to local optima, the majority of the networks fits the data reasonably, without overfitting to noise. However, even though

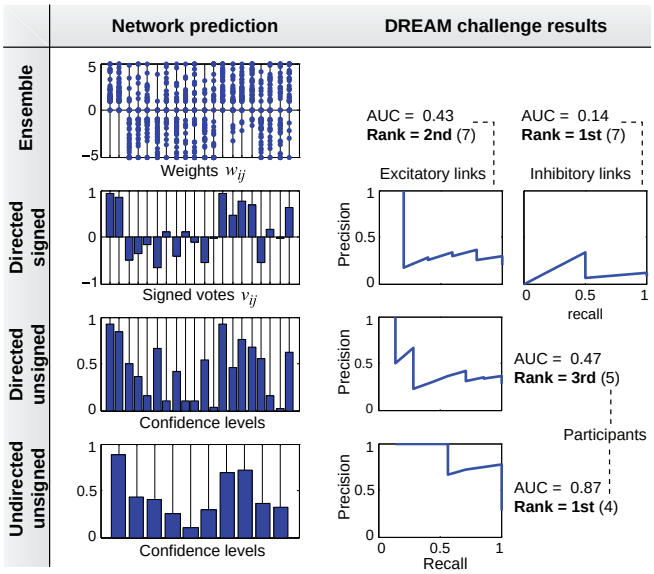

Fig. 5 Performance of the biomimetic method, coupled with ensemble voting, in the DREAM Five-Gene-Net Challenge. The ensemble of inferred weights is very diverse (first row). Our ranking in the competition is good, but the accuracy of the predictions of all participants (including us) is not satisfactory. See main text for discussion

most networks fit the data well, they have very different numerical values for the weights (Fig. 3C). The same is true for the gene parameters m_i, b_i, and λ_i of the model (data not shown). This indicates that for the biomimetic method used here, the reverse engineering problem is underdetermined by this relatively small and noisy dataset.

Still, the inhibitory links of the target network are correctly predicted (i.e., are put on top of the ranked list) both by the best network of the ensemble and by signed voting of the top 20%. Signed voting by the complete ensemble performs slightly worse (Fig. 4, top three rows). Probably, the information contained in this dataset is sufficient to constrain the network structures that can fit the data to a relatively narrow peak in the fitness landscape (Fig. 2A), and this peak seems to coincide with the true network structure. In this situation, it is not surprising that the best scoring network performs as good or better than ensemble voting.

As we add more noise to the data (standard deviation 1.0), the information content is reduced and the distribution of network structures that can fit the data broadens. Consequently, the individual networks of the ensemble are expected to be more diverse and predict the target network less accurately. Indeed, the network that fits the data best now performs poorly in predicting the network structure and has a low AUC score of 0.5. In contrast, signed voting of the top 20% still correctly predicts the network structure with a perfect AUC score of 1. Again, signed voting by the complete ensemble performs slightly worse (Fig. 4, middle three rows).

The vastly superior performance of the ensemble as compared to the network with the best data fit can be explained as follows. The individual networks of the ensemble consistently include the five inhibitory links correctly (the signed vote is close to -1 for these links), but in addition also have many false positives. However, it seems that the false positives are sufficiently uncorrelated between the networks of the ensemble to partly "even out" and obtain a lower confidence level than the true positives.

When adding excessive noise (standard deviation 1.5), the network structure is not predicted accurately anymore. Still, the AUC score is doubled by signed voting of the top 20% compared to the network with the best data fit (Fig. 4, last three rows).

The same quality of results was obtained on four different datasets with log-normal noise of standard deviations 0.5, 1.0, 1.5, and 2.0 (results not shown).

Ensemble voting achieves winning performance in an in vivo reverse engineering challenge. We have tested ensemble voting on a real dataset provided for the Five-Gene-Net 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 using q-PCR. The goal of the challenge was to predict the structure of the network from this dataset. The true network structure was not disclosed to the participants prior to the submission of the predictions.

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).

We predicted excitatory and inhibitory links using our biomimetic reverse engineering method coupled with signed voting by the complete ensemble of inferred networks. As for the *in silico* test case, the majority of the runs fits the data well (data fit not shown, see previous paragraph). The values of the inferred weights are very scattered (Fig. 5, first row) and the reverse engineering problem seems to be largely underdetermined. In contrast to the *in silico* test case, the network structure is not accurately predicted and the corresponding AUC scores are low (Fig. 5, second row). Note, however, that we compared well to other participating teams (2nd and 1st rank for excitatory and inhibitory predictions respectively). These results, and in particular possible explanations for the low AUC scores, are discussed in detail in our companion paper in this volume.[8] Here, we focus on the other categories of the challenge and the performance of ensemble voting.

Undirected and unsigned link predictions. In order to compare our approach with reverse engineering methods that produce undirected and unsigned network predictions, we have participated also in these categories of the DREAM challenge. Confidence levels for directed-unsigned links and undirected-unsigned links were derived from the signed votes of the ensemble [a]

$$l\{|a_{ij}|=1\} := |v_{ij}| \quad \text{(dir.-unsig.)}$$

$$l\{(|a_{ij}|=1) \text{ or } (|a_{ji}|=1)\} := \frac{|v_{ij}|+|v_{ji}|}{2} \quad \text{(undir.-unsig.)}$$

Our predictions derived in this way are competitive with methods that directly produce undirected and unsigned pre-

[a] Note that ensemble voting is done first, and the sign is removed afterwards. Thus, if there are inconsistent signed predictions for a link (e.g, 50% excitatory, 50% inhbitory), the corresponding unsigned prediction is still zero. This would not be the case if signs were removed first, and ensemble voting done afterwards.

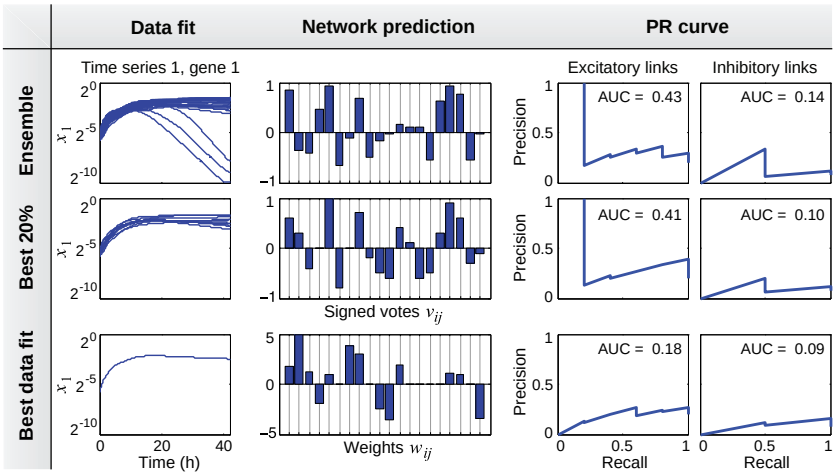

Fig. 6 Same analysis as Fig. 4, but for the *in vivo* dataset. The first column shows the data fit by the inferred networks (normalized, negative q-PCR log expression ratios). The target dataset provided by Cantone *et al.* is not shown because it is not yet published. As in the *in silico* test case, the accuracy is significantly improved by ensemble voting

dictions (Fig. 5). At first sight, the AUC score seems to be very high for the undirected-unsigned predictions as compared to the directed-signed predictions that they were derived from. Is water (inaccurate directed-signed predictions) made into wine (accurate undirected-unsigned predictions)? Certainly not. The undirected target network has the same number of true links, but only half as many possible links as the directed version. This makes it much easier to obtain a high AUC score.

Ensemble voting outperforms the network with the best data fit on the DREAM challenge. After the true structure of the target network was published on the DREAM website (http://wiki.c2b2.columbia.edu/dream), we analyzed the performance of ensemble voting on this benchmark. The observations on the *in silico* test case with intermediate and strong levels of noise are confirmed on the real DREAM challenge dataset. The AUC score is roughly doubled by ensemble voting compared to the network with the best data fit. Signed voting by the complete ensemble and by the top 20% perform approximately equally well (Fig. 6).

Conclusion

As discussed in the introduction, ensemble voting boosts the performance compared to individual members of the ensemble if the prediction errors are uncorrelated. This seemed unlikely for an ensemble of networks that are inferred from the same dataset. Yet, our results show that in practice, the prediction errors in ensembles of reverse engineered networks are sufficiently uncorrelated for ensemble voting to drastically improve the accuracy of predictions from noisy datasets. This was confirmed both on *in silico* and real q-PCR data from an *in vivo* gene network.

The goal of a reverse engineering algorithm is often seen to consist in reliably finding the global optimum, i.e., the best scoring network. Here, we advocate for a different view, where the goal of the reverse engineering algorithm is to construct an ensemble of good scoring networks (repeatedly recovering the global optimum is contrary to this aim). Our results show that it is possible to make accurate predictions from such ensembles even if the problem is underdetermined and many different networks fit the noisy data equally well.

Compared to generating the ensemble of network predictions in the first place, the complexity of ensemble voting is negligible. Thus, the scalability to larger networks depends mainly on the reverse engineering method used to make the network predictions. In this paper, we have only considered small networks. Studying the performance of ensemble voting on different network sizes is a topic of future work.

The ensemble approach holds the promise to improve the accuracy of any reverse engineering method that can produce sufficiently diverse network predictions. However, the biomimetic evolutionary method used here is particularly well suited for this purpose because it reproduces, at a certain level of abstraction, the structure and evolutionary constraints of the biological genome. We hypothesize that this is an effective approach to incorporate prior knowledge and bias the search towards biologically plausible solutions.[8] Ensembles generated by "replaying the evolutionary tape" may thus provide a better sampling of the posterior distribution than ensembles generated with other optimization methods, because the fundamental prior that biological gene networks originate from an evolutionary process is taken into account.

The signed voting method has an intuitive appeal, and our results show that it works well in practice. However, we believe that there is a need for more sophisticated tools for a rational, probabilistic analysis of ensembles of inferred cellular networks, and we hope that the encouraging results presented here will stimulate further research in this direction.

Acknowledgements

We thank Gustavo Stolovitzky, Peter Dürr, Sara Mitri, and Fred Marbach for their helpful comments. This work was supported by the Swiss National Science Foundation, grant no. 200021–112060.

1. 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.
2. 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.
3. Tegner, J., M. K. S. Yeung, J. Hasty & J. J. Collins. 2003. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. Proc. Nat. Acad. Sci. U.S.A. 100: 5944–5949.
4. Battogtokh, D., D. K. Asch, M. E. Case, J. Arnold & H-B. Schuttler. 2002. An ensemble method for identifying regulatory circuits with special reference to the qa gene cluster of neurospora crassa. Proc. Nat. Acad. Sci. U.S.A. 99: 16904–16909.
5. Reinitz, J. & D. H. Sharp. 1995. Mechanism of eve stripe formation. Mech. Dev. 49: 133–158.
6. Jaeger, J., M. Blagov, D. Kosman, K. N. Kozlov, Manu, E. Myasnikova, S. Surkova, C. E. Vanario-Alonso, M. Samsonova, D. H. Sharp & J. Reinitz. 2004. Dynamical analysis of regulatory interactions in the gap gene system of drosophila melanogaster. Genetics 167: 1721–1737.
7. 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.
8. Marbach, D., C. Mattiussi & D. Floreano. 2009. Replaying the evolutionary tape: Biomimetic reverse engineering of gene networks. Ann. N.Y. Acad. Sci.: in this issue.
9. 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.
10. Surowiecki, J. 2004. The Wisdom of Crowds. Random House. New York, NY.
11. Polikar, R. 2006. Ensemble based systems in decision making. IEEE Circuits Syst. Mag. 6: 21–46.
12. Dietterich, T.G. 2000. Ensemble methods in machine learning. *In* Multiple classifier systems. J. Kittler & F. Roli, Eds.: 1–15. Springer. Cagliari, Italy.
13. Hartemink, A. J., D. K. Gifford, T. S. Jaakkola & R. A. Young. 2002. Combining location and expression data for principled discovery of genetic regulatory network models. Pac. Symp. Biocomput. 2002: 437–449.
14. Moles, C. G., P. Mendes & J. R. Banga. 2003. Parameter estimation in biochemical pathways: A comparison of global optimization methods. Genome Research 13: 2467–2474.
15. 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.
16. Wahde, M. & J. Hertz. 2001. Modeling genetic regulatory dynamics in neural development. J. Comput. Biol. 8: 429–442.
17. Deng, X., H. Geng & H. Ali. 2005. EXAMINE: A computational approach to reconstructing gene regulatory networks. Biosystems 81: 125–136.
18. Mattiussi, C. & D. Floreano. 2007. Analog Genetic Encoding for the evolution of circuits and networks. IEEE Trans. Evol. Comput. 11: 596–607.
19. Mattiussi, C., D. Marbach, P. Dürr & D. Floreano. 2008. The age of analog networks. AI Mag: to appear.
20. 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.