Revealing strengths and weaknesses of methods for gene network inference

Daniel Marbach[a,b], Robert J. Prill[c], Thomas Schaffter[a], Claudio Mattiussi[a], Dario Floreano[a], and Gustavo Stolovitzky[c,1]

[a]Laboratory of Intelligent Systems, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland; [b]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139; and [c]IBM T. J. Watson Research Center, Yorktown Heights, New York, NY 10598

Edited by Charles R Cantor, Sequenom Inc., San Diego, CA, and approved February 4, 2010 (received for review November 18, 2009)

Numerous methods have been developed for inferring gene regulatory networks from expression data, however, both their absolute and comparative performance remain poorly understood. In this paper, we introduce a framework for critical performance assessment of methods for gene network inference. We present an *in silico* benchmark suite that we provided as a blinded, community-wide challenge within the context of the DREAM (Dialogue on Reverse Engineering Assessment and Methods) project. We assess the performance of 29 gene-network-inference methods, which have been applied independently by participating teams. Performance profiling reveals that current inference methods are affected, to various degrees, by different types of systematic prediction errors. In particular, all but the best-performing method failed to accurately infer multiple regulatory inputs (combinatorial regulation) of genes. The results of this community-wide experiment show that reliable network inference from gene expression data remains an unsolved problem, and they indicate potential ways of network reconstruction improvements.

DREAM challenge | community experiment | reverse engineering | transcriptional regulatory networks | performance assessment

Some of our best insights into biological processes originate in the elucidation of the interactions between molecular entities within cells. In the past, these molecular connections have been established at a rather slow pace. For example, it took more than a decade from the discovery of the well known tumor suppressor gene p53 to determine that it formed a regulatory feedback loop with the protein MDM2, its key regulator (1). Indeed, the mapping of biological interactions in the intracellular realm remains the bottleneck in the pipeline to produce biological knowledge from high-throughput data. One of the promises of computational systems biology are algorithms that feed in data and output interaction networks consistent with those input data. To accomplish this task, the importance of having accurate methods for network inference cannot be overestimated.

Spurred by advances in experimental technology, a plethora of network-inference methods (also called *reverse engineering* methods) has been developed (2–10), at a rate that has been doubling every two years (11). However, the problem of rigorously assessing the performance of these methods has received little attention until recently (11, 12). Even though several interesting and telling efforts to compare between different network-inference methods have been reported (13, 14, 15), these efforts typically compare a small number of algorithms that include methods developed by the same authors that do the comparisons. Consequently, there remains a void in understanding the comparative advantages of inference methods in the context of blind and impartial performance tests.

To foster a concerted effort to address this issue, some of us have initiated the DREAM (Dialogue on Reverse Engineering Assessment and Methods) project (11, 16). One of the key aims of DREAM is the development of community-wide challenges for objective assessment of reverse engineering methods for biological networks. Similar efforts have been highly successful in the field of protein structure prediction (17). However, the design of such benchmarks for biological network inference is problematic. On the one hand, well-known networks cannot be used because their identity is not easily hidden from the participants to create "blinded" challenges. On the other hand, there is not yet a gold-standard experiment for establishing the ground truth (the true network structure) for unknown in vivo networks. Consequently, *in silico* benchmarks (i.e., simulated networks and data) remain the predominant approach for performance assessment of reverse engineering methods: in simulation, the ground truth is known and predictions can be systematically evaluated (18, 19).

In this paper, we describe the results of a gene-network reverse engineering challenge, the so-called DREAM3 *in silico* challenge, which was one of the four DREAM3 challenges that we organized within the context of the DREAM project. The challenge is based on a series of *in silico* networks (Fig. 1), which we created using a unique approach for the generation of biologically plausible network structures and dynamics. The DREAM3 *in silico* challenge, with 29 participating teams from over ten countries, has become by far the most widely used benchmark for gene-network reverse engineering. The participants have submitted almost 400 network predictions, which we have evaluated in a double-blind manner (Fig. 1).

In what follows we dissect the predictions and analyze the performance of the 29 methods that inferred networks for the challenge. We developed unique methodologies to extract lessons from the ensemble of submissions based on the efficacy of the different predictions to learn local connectivity patterns (network motifs (20, 21)), and combinatorial regulation (in-degree distribution (22)). Our analyses clearly show that some network motifs (fan-in, fan-out, and cascade motifs) were poorly predicted even by high-rank performing submissions, indicating systematic errors in inference and potential ways of network reconstruction improvements. The set of submitted networks form a veritable dataset, contributed by the systems biology community and obtained by field experimentation (the DREAM challenges). The analysis of the results of this community-based experiment reveals the strengths and weaknesses of the state-of-the-art efforts on network inference.

Results

The DREAM3 in-Silico Challenge. To assess the performance of gene network-inference methods *in silico*, it is essential that the benchmarks are biologically plausible. This involves generating realistic structures for the benchmark networks, generating the corresponding kinetic models, and using these models to

Author contributions: D.M., R.J.P., and G.S. designed research; D.M., R.J.P., T.S., and G.S. performed research; D.M., R.J.P., C.M., D.F., and G.S. analyzed data; and D.M., R.J.P., T.S., C.M., D.F., and G.S. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence should be addressed. E-mail: gustavo@us.ibm.com.

This article contains supporting information online at www.pnas.org/cgi/content/full/0913357107/DCSupplemental.

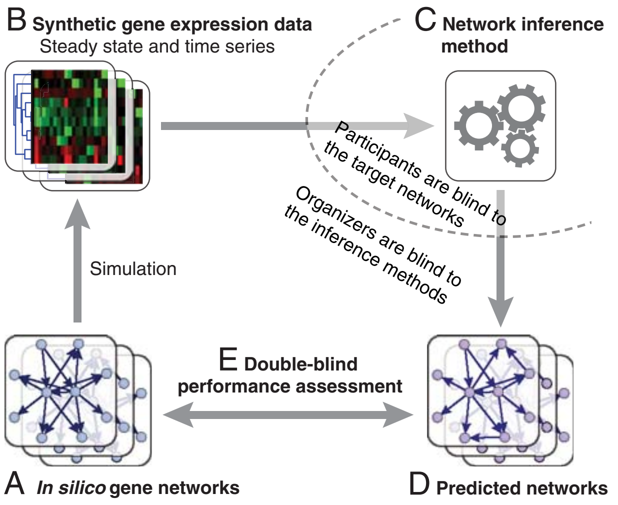

B Synthetic gene expression data
Steady state and time series
C Network inference method
Participants are blind to the target networks
Organizers are blind to the inference methods
Simulation
E Double-blind performance assessment
A In silico gene networks
D Predicted networks

Fig. 1. Double-blind performance assessment of network-inference methods. (*A*, *B*) From a set of *in silico* benchmark networks (the so-called *gold standards*), steady-state and time-series gene expression data was generated and provided as a community-wide reverse engineering challenge. (*C*, *D*) Participating teams were asked to predict the structure of the benchmark networks from this data. They were blind to the true structure of these networks. (*E*) We evaluated the submitted predictions, being blind to the inference methods that produced them. This allowed for a double-blind performance assessment.

produce synthetic gene expression data by simulating different biological experiments.

The challenge was structured as three separate subchallenges with networks of 10, 50, and 100 genes, respectively. For each size, we generated five *in silico* networks. We produced realistic network structures by extracting modules from known biological interaction networks (19). For each size, we extracted two structures from an *Escherichia coli* transcriptional regulatory network (21), and three structures from a yeast genetic interaction network (23). Examples of networks are shown in Fig. 2*A* and Fig. S1. We endowed these networks with dynamics using the models of transcription and translation described in *Methods*.

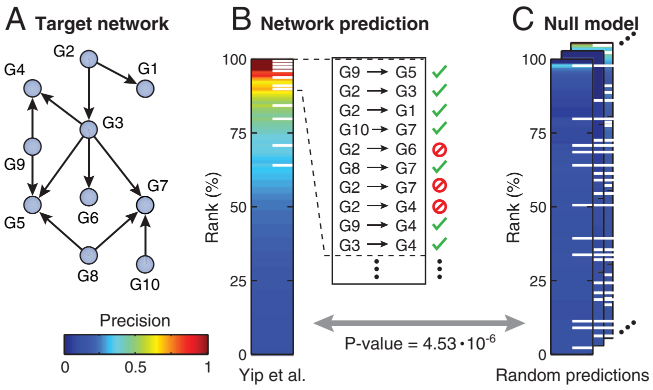


Fig. 2. Evaluation of network predictions. (*A*) The true connectivity of one of the benchmark networks of size 10. (*B*) Example of a submitted prediction (it is the prediction of Yip et al., the best-performer team). The format is a ranked list of predicted edges, represented here by the *vertical colored bar*. The *white stripes* indicate the true edges of the target network. A perfect prediction would have all *white stripes* at the top of the list. The *inset* shows the first ten predicted edges: the top four are correct, followed by an incorrect prediction, etc. The color indicates the *precision* at that point in the list. E.g., after the first ten predictions, the precision is 0.7 (7 correct predictions out of 10 predictions). (*C*) The network prediction is evaluated by computing a *P*-value that indicates its statistical significance compared to random network predictions.

Transcriptional regulation of genes was modeled using a standard approach based on thermodynamics (24). Both independent ("additive") and synergistic ("multiplicative") interactions occur in the networks.

We used these *in silico* gene networks to produce different types of steady-state and time-series gene expression data that are commonly used for gene network inference (8, 12): steady-state expression levels of the unperturbed network, steady-state levels of knockout and knockdown experiments for every gene, and time-series data showing how the network recovers from multifactorial perturbations (see *Methods*). The gene expression data were provided to the participants in the form of mRNA concentrations with moderate additive Gaussian noise. The protein concentrations were not provided, as would be the case with experiments based purely on transcriptional data. Participants were asked to predict the underlying networks from the given gene expression datasets. Our Java tool used to generate the benchmarks is available open-source (25).

Performance Metrics. We evaluated the ability of inference methods to predict the presence of regulatory interactions between genes (some methods predict additional aspects, such as the kinetics parameters of the interactions, which were not considered here). Participants were asked to submit network predictions in the form of ranked lists of predicted edges (16). The lists had to be ordered according to the confidence of the predictions, so that the first entry corresponds to the edge predicted with the highest confidence. In other words, the edges at the top of the list were believed to be present in the network, and the edges at the bottom of the list were believed to be absent from the network. The number of possible edges in an N-gene network without autoregulatory interactions is $N(N-1)$. Autoregulatory edges were not expected in the predictions. Therefore, for networks of size 10, 50, and 100, the length of a complete list of predictions is 90, 2,450, and 9,900 edges. An example is shown in Fig. 2.

As mentioned above, each subchallenge had five networks. To participate, teams were required to submit a prediction for each of the five networks. We statistically evaluated predictions by computing P-values indicating the probability that random lists of edge predictions would be of the same or better quality (see Fig. 2 and *Methods*). The final score that we used for the ranking was a negative log-transformed P-value: for example, a P-value of 10^{-2} gives a score of 2, and a P-value of 10^{-3} gives a score of 3. Thus, larger scores indicate smaller P-values, hence better predictions.

Performance Assessment of Network-Inference Methods. In total, 29 teams participated in the challenges. The majority of teams submitted predictions for all three network sizes (10, 50, and 100 genes): the corresponding subchallenges had 29, 27, and 22 participants, respectively, totaling in 390 submitted network predictions (there are five networks of each size). The scores of the top ten teams for each subchallenge are shown in Fig. 3. The complete set of results is available on the DREAM website (28). Note that participants are anonymous, except for the best performers and teams who voluntarily disclose their identity.

The ranking is similar in the three subchallenges, i.e., the performance of most methods was consistent over different network sizes (Fig. S2). The method of Yip et al. (29) obtained the best performance on all three network sizes. A representative prediction of the best-performer method is shown in the example of Fig. 2: most links were correctly recovered, but there were also some incorrect predictions (false positives) among the high-confidence edges at the top of their prediction lists (the origin of these errors will become apparent in the network-motif analysis below). As can be seen in Fig. 3, several other methods achieved highly significant predictions. For example, on networks

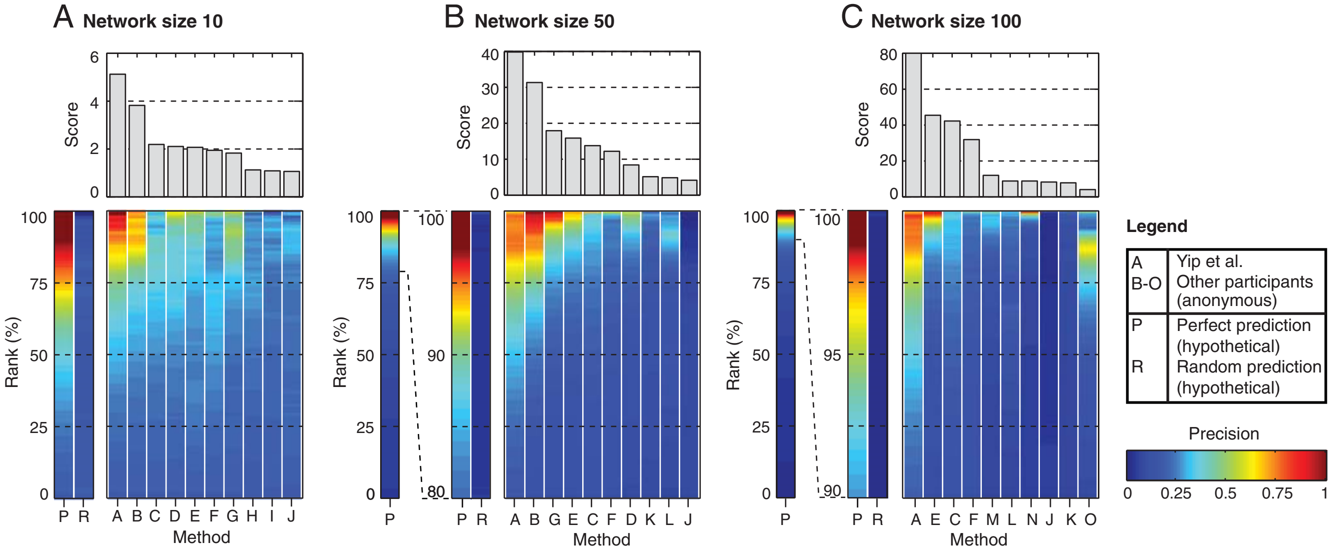


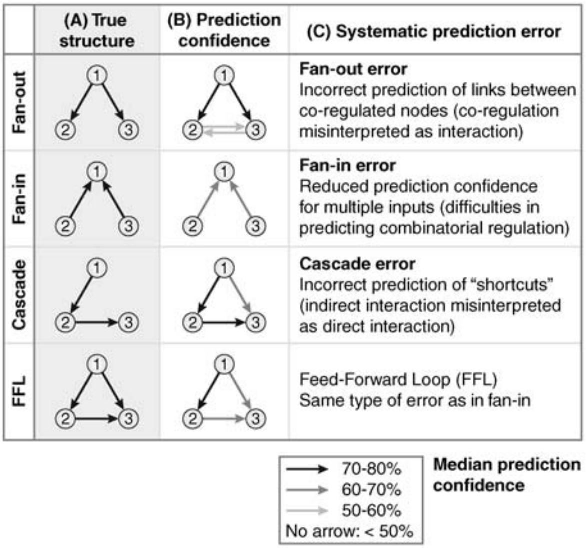
Fig. 3. Average performance of the best ten teams for each of the three subchallenges. The *bar plots* on top show the overall scores, and the *color bars below* show the precision of the corresponding lists of predictions, as explained in Fig. 2 (since each subchallenge has five networks, this is the average precision of the five lists). In addition to the submitted network predictions (methods *A*–*O*), we always show the plots for a hypothetical perfect prediction *P* (all true edges at the top of the list) and a randomly generated prediction *R*, which allows to visually appreciate the quality of the submitted predictions. Remember that for networks of size 10, 50, and 100, the length of the lists is 90, 2,450, and 9,900 edges. Note that for networks of size 50 and size 100, we have zoomed in to the top 20% and 10% of the lists, respectively.

of size 100, the top four teams all had scores above 30 (i.e., P-values smaller than 10^{-30}).

However, for the majority of inference methods the precision of the predictions was rather low (<0.5, blue tones in Fig. 3). In addition, a surprisingly large number of methods (11 out of the 29) produced network predictions that were, on average, not significantly better than random guessing (P-values >0.01). This is a sobering result for the efficacy of the network-inference community. It should be kept in mind that some participants may not be experienced in network inference, which could explain the low performance of some teams. However, many well-known practitioners in the field were spread over all ranks.

According to a survey that we conducted among the participants, the applied inference methods span a wide range of approaches commonly used to reverse engineer gene networks, including correlation-based methods (6), information-theoretic methods (9, 27), Bayesian network predictions (4), and methods based on dynamical models (2, 3, 8). There seems to be no correlation between the general type of inference method used and the scores. Indeed, all four approaches mentioned above are represented among the top five inference methods of the challenge (see *SI Text* and Table S1). At the same time, all of these approaches were also used by teams that didn't produce significant predictions, implying that success is more related to the details of implementation than the choice of general methodology. Concerning the type of data used to infer the networks, the top five teams all integrated both steady-state and time-series data, i.e., they took advantage of all provided data. In retrospect, the steady-state levels of the gene knockout experiments seemed to have been the most informative: the score of the best-performer team was mainly due to predictions derived from the knockout datasets and not those from the time-series (see *SI Text*).

Network-Motif Analysis Reveals Three Types of Systematic Prediction Errors. In order to understand the differences in performance of inference methods, we need to know what types of prediction errors they make. To this end, we have analyzed the inference methods performance on the basic building blocks of networks, the network motifs (20, 21). More precisely, we have analyzed how well the inference methods predict edges pertaining to different network motifs. The first column of Fig. 4 shows the four types of motifs that occur in the benchmark networks of the challenge (fan-in, fan-out, cascade, and feed-forward loop). As an illustrative example, the second column shows how well their links were predicted, on average, by the method that ranked second on the networks of size 100 (8). It can be seen that not all links of the motifs were predicted with the same median *prediction confidence*—some were predicted less reliably (at lower confidence) than others (the prediction confidence of edges was defined as their rank in the list of edge predictions, scaled such that the first edge in the list has confidence 100%, and the last edge in the list has confidence 0%).



	(A) True structure	(B) Prediction confidence	(C) Systematic prediction error
Fan-out			**Fan-out error** Incorrect prediction of links between co-regulated nodes (co-regulation misinterpreted as interaction)
Fan-in			**Fan-in error** Reduced prediction confidence for multiple inputs (difficulties in predicting combinatorial regulation)
Cascade			**Cascade error** Incorrect prediction of "shortcuts" (indirect interaction misinterpreted as direct interaction)
FFL			Feed-Forward Loop (FFL) Same type of error as in fan-in

Fig. 4. Systematic errors in the prediction of motifs. (*A*) The true connectivity of the motifs. (*B*) As an example, we show how the motifs were predicted *on average* by the inference method that ranked second on the networks of size 100 (8). The darkness of the links indicates their median prediction confidence. (*C*) We can identify three types of systematic prediction errors: the *fan-out error*, the *fan-in error*, and the *cascade error*.

In order to evaluate whether some edges of motifs were systematically predicted less reliably than others, we compared their median prediction confidence to the *background prediction confidence*, which is the median prediction confidence of all links of the network (independently of which motifs they belong to, see *SI Methods*). If the motifs had no effect on the prediction confidence, the edges pertaining to different motifs would all be inferred, on average, with the background prediction confidence. This was not the case: The median prediction confidence of some motif edges diverged significantly from the background prediction confidence (evaluated at a level of 0.01 using a two-sided Wilcoxon-Mann-Whitney rank-sum test and Bonferroni correction for multiple hypothesis testing).

We found three different types of significant, systematic errors in the prediction of motifs (cf. Fig. 4*C*):

(*i*) **The fan-out error** corresponds to a tendency to incorrectly predict edges between coregulated nodes (2 → 3 and 3 → 2). The expression levels of coregulated genes are often correlated. The fan-out error occurs when this correlation is wrongly interpreted as an interaction between the two genes;

(*ii*) **The fan-in error** is a reduced prediction confidence for multiple inputs. In other words, fan-in links (2 → 1 and 3 → 1) are predicted less reliably than other links of the target network. This error is due to difficulties in accurately modeling and inferring *combinatorial regulation* of genes (regulation of genes by several inputs);

(*iii*) **The cascade error** is a tendency for incorrectly predicted "shortcuts" in cascades. This error occurs when indirect regulation (1 → 2 → 3) is misinterpreted as direct regulation (1 → 3); and

(*iv*) The links 1 → 3 and 2 → 3 of **feed-forward loops** (FFLs) often have a reduced prediction confidence. These links form a fan-in, and their reduced prediction confidence can thus be explained in terms of the fan-in error (hence, we did not consider this as an additional type of systematic prediction error).

Note that we analyze the different motif types independently from each other. Possible effects due to overlapping motifs will be considered elsewhere.

We performed the network-motif analysis for all inference methods that were applied to the networks of size 50 and 100 (networks of size 10 are too small for a statistically significant analysis). We did not observe other types of systematic errors than the three discussed above. However, we found that inference methods are affected to various degrees by these errors—they have different *error profiles*. Whereas some inference methods are more robust to certain types of error, they are more strongly affected by other types of errors, i.e., they have different strengths and weaknesses (Fig. S3). For example, the best-performer method was the most robust to the fan-in error, but it was more strongly affected by the cascade error than other inference methods.

Most Inference Methods Fail to Accurately Predict Combinatorial Regulation. The network-motif analysis has shown that all inference methods had a reduced prediction confidence for multiple inputs (combinatorial regulation) of genes (fan-in error), here, we analyze this type of error in more detail. Specifically, we compare how well, on average, inference methods predict the regulatory input(s) of genes with a single input (in-degree 1), two inputs (in-degree 2), three inputs (in-degree 3), etc. The results of this analysis are shown in Fig. 5 for the best five inference methods on networks of size 100 (as mentioned above, the prediction confidence of edges was defined as their rank in the list of edge predictions, scaled such that the first edge in the list has confidence 100%, and the last edge in the list has confidence 0%). These data show that several methods predict single inputs of genes with high confidence. However, for all but the best-performer method, the prediction confidence degrades drastically as the number of inputs increases. For example, the fourth place method reliably identified links that are the only input of their targets (median prediction confidence 97%), but did no better than random guessing in predicting inputs of genes with in-degree nine (median prediction confidence 46%). We have performed this analysis for all methods that inferred networks of size 50 and 100, which confirmed that only the best-performer method had a robust performance on high indegrees (Fig. S4).

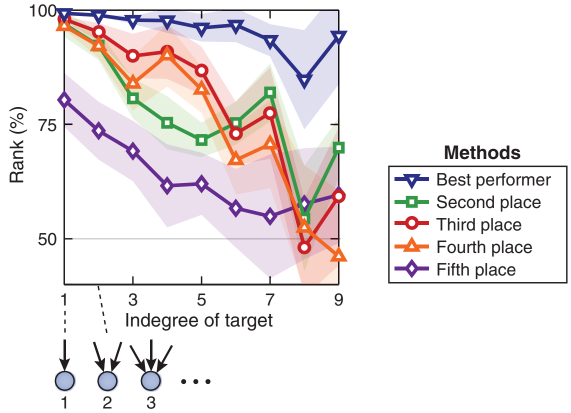


Fig. 5. How the indegree of genes affects the prediction confidence. The plots show, for the best five methods on networks of size 100, the median prediction confidence for links that target genes of increasing indegree. The shaded areas indicate 95% confidence intervals for the medians. Single-input links were reliably predicted with a similar, high prediction confidence by the best four methods (*points in the top left corner*). However, for all but the best-performer method, the performance drops drastically for higher indegrees.

It is not unexpected that edges that are the sole input of their target gene are easier to infer than edges towards genes with many inputs. If a gene has only one regulator, and this regulator is being perturbed, the gene would show a clear response. In contrast, if a regulator of a gene with other regulatory inputs is being perturbed, the effect may be partially buffered or even completely masked by the other inputs, which would make this edge more difficult to infer. Thus, what was surprising to us was not that all methods are affected by the fan-in error, but that they are so to very different degrees.

Community Predictions Are More Reliable than Individual Inference Methods. In the previous sections, we have shown that inference methods have different strengths and weaknesses. A natural corollary of this observation is that the combination of network reconstruction methods could be a good strategy for network inference. We have formed "community predictions" by combining the prediction lists of multiple inference methods. We combined the edge-prediction lists simply by reranking the edge list according to the average rank for each edge (Fig. S5).

To gain a sense of the performance of community predictions in the DREAM3 *in silico* challenge, we systematically formed communities composed of the top two methods, the top three methods, the top four methods, etc., until the last community, which contains all applied methods of a particular subchallenge. In Fig. 6, we compare the scores of the community predictions with those of the individual teams for the networks of size 10. Some of the community predictions outperform the best-performer team (e.g., the community of the top five teams). More importantly, the performance of the community is robust to inclusion of methods with very low scores. Even when combining all methods, the community prediction still ranks second. Similar observations can be made on networks of size 50 and 100 (Fig. S6). Note that in a real application to an unknown biological network, it's impossible to know in advance which inference method would

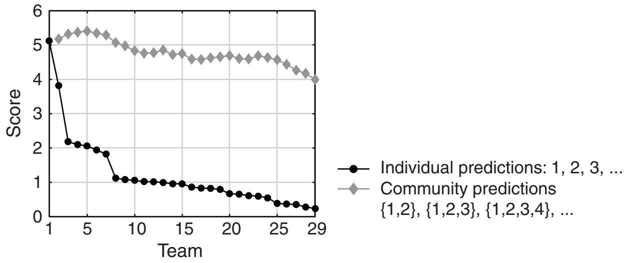


Fig. 6. Performance of community predictions for the networks of size 10. The *circles* are the scores of the individual teams. The *diamonds* correspond to the scores of the different community predictions, obtained by combining the two best teams, the three best teams, the four best teams, etc.

perform best. Our results show that, instead of choosing a single method to trust, a more reliable strategy is to apply all methods at hand and form a community prediction.

Discussion

A Word of Caution. The *in silico* benchmarks presented here are based on networks with similar types of structural properties and regulatory dynamics as occur in biological gene networks. In particular, the network structures correspond to modules of known gene networks, and the kinetic model is based on a thermodynamic approach (24), which has been shown to provide a good approximation to different types of transcriptional regulation (30). However, this representation is a simplified model of real biological mechanisms. Furthermore, additional layers of control, such as posttranscriptional regulation and chromatin states, are not modeled. Even though these *in silico* benchmarks by no means replace the need for careful characterization of performance in vivo (12), they remain an important tool to systematically and efficiently validate the performance of inference methods over many networks. Furthermore, it is likely that methods that don't fare well in these benchmarks, might fare even worse with real biological networks, as discussed in *SI Text*.

A general issue of benchmarks, be it *in silico* or in vivo, is that the measured performance of methods is always specific to the particular networks that were being used, and does not necessarily generalize to other, unknown networks, which may have different properties. Indeed, one of the main conclusions of this study is that the performance of current network-inference methods is strongly dependent on the properties of the network that is being inferred. For example, since methods were found to have very different network-motif error profiles, their performance depends on how many instances of each motif type are present in the network.

Thus, the overall performance (score) of the inference methods should be considered with caution, as it may vary on networks with different properties. However, the systematic errors identified with the network-motif analysis are expected to be less variant on different networks. For example, a method that failed to distinguish direct from indirect regulation (cascade error), would be expected to have similar difficulties also on biological gene networks.

Reliable Gene Network Inference from Gene Expression Data Remains an Unsolved Problem. The two major difficulties in gene-network reverse engineering are often considered to be the limited data, which may leave the inference problem underdetermined (10), and the difficulty of distinguishing direct from indirect regulation (the cascade error) (26). A number of approaches have been developed to overcome these difficulties, for example, partial correlation (26) and other methods (6, 27) have been proposed to encounter potential cascade errors. However, the results of the community experiment reported here call attention to additional difficulties that have to be overcome for reliable inference of gene networks.

We provided more and better data than is typically available in real biological experiments, yet the overall performance of the 29 applied network-inference methods was not satisfactory. The best-performer method (29) worked remarkably well, given that it is based on possibly the simplest model of all applied inference methods (see *SI Text*). However, it could not distinguish direct from indirect regulation. Despite this increased error rate in the cascade motif, it had significantly better overall performance than other state-of-the-art methods based on more advanced, probabilistic, or dynamical models. Even though some were more robust to the cascade error, they were strongly affected by other systematic errors, in particular the fan-in error. We expect these errors to be even more pronounced in real biological applications, suggesting that the performance of gene-network-inference methods may previously have been overestimated due to the lack of rigorous, blinded benchmarks.

Conclusion. We have presented a framework for critical performance assessment of gene-network-inference methods. This framework has allowed us to compare a large number of inference methods—applied independently by different teams—on multiple benchmark networks. In addition to assessment of the overall accuracy, we have evaluated the performance of inference methods on individual network motifs. This analysis revealed that current inference methods are affected, to various degrees, by three types of systematic prediction errors: the fan-out error (incorrect prediction of interactions between coregulated genes), the fan-in error (inaccurate prediction of combinatorial regulation), and the cascade error (failure to distinguish direct from indirect regulation). Distinguishing between direct and indirect regulation is a well-known difficulty in network inference (6, 26, 27), but was never quantitatively assessed. The network-motif analysis makes it possible to quantify how well this difficulty is resolved by different methods. Furthermore, it revealed two other types of systematic errors, the fan-in error and the fan-out error, which are equally important for the overall quality of predictions.

One of the difficulties that participants of the DREAM challenge had to face was that they did not know details of the kinetic model that was used to generate the gene expression data. This difficulty is even more pronounced in biological applications, where the mechanisms and kinetics of gene regulation underlying the expression data are more complicated, and also not known in advance. Consequently, inference methods are bound to make simplifying assumptions. However, inaccurate assumptions were shown to induce systematic prediction errors, which profoundly affected the performance of the applied network-inference methods (see also *SI Text*). There is thus a need for the development of inference methods that have a more robust performance despite uncertainty about the type of mechanisms (the "model") underlying the data. We have shown that one possible approach to achieve this goal is to combine the predictions from complementary inference methods to form more robust and accurate "community predictions". We are convinced that a better understanding of the capabilities and limitations of existing inference methods will enable the development of unique approaches that will ultimately make the DREAM of accurate, high-throughput inference of gene networks come true.

Materials and Methods

Simulation of Expression Data. Gene networks were modeled by a system of ordinary differential equations describing the dynamics of the mRNA concentration x_i and the protein concentration y_i of every gene

$$\frac{dx_i}{dt} = m_i \cdot f_i(\mathbf{y}) - \lambda_i^{\mathrm{RNA}} \cdot x_i \qquad [1]$$

$$\frac{dy_i}{dt} = r_i \cdot x_i - \lambda_i^{\mathrm{Prot}} \cdot y_i, \qquad [2]$$

where m_i is the maximum transcription rate, r_i the translation rate, λ_i^{RNA} and $\lambda_i^{\mathrm{Prot}}$ are the mRNA and protein degradation rates, and $f_i(\cdot)$ is the so-called input function of gene i. The input function computes the *relative activation* of the gene, which is between 0 (the gene is shut off) and 1 (the gene is maximally activated), given the transcription-factor (TF) concentrations $\mathbf{y}$. The input function is derived using a standard thermodynamic approach (24), where binding of TFs to cis-regulatory sites is approximated using Hill-type kinetics (see *SI Methods*). Knockouts were simulated by setting the maximum transcription rate m_i of the deleted gene to zero, knockdowns by dividing it by 2. Time-series experiments were simulated by integrating the networks using different initial conditions. For the networks of size 10, 50, and 100, we provided 4, 23, and 46 different time-series, respectively, with 21 time points each. Gaussian noise was added to the data after the simulation. See *SI Methods* for details.

Evaluation of Predictions. As described in *Results*, the submission format of predictions was a list of predicted edges with their assigned confidence measures, constructed in decreasing order of confidence from the most reliable to the least reliable prediction. The quality of the predictions was measured by the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPR) (16). In addition to the AUPR and AUROC values, we statistically evaluated predictions by computing corresponding *P*-values (p_{AUROC} and p_{AUPR}), which are the probability that a random list of edge predictions would obtain the same or better AUROC and AUPR than a given network prediction. Distributions for AUROC and AUPR were estimated from 100,000 instances of random lists of edge predictions. The *overall P*-value of the five networks of a subchallenge was defined as the geometric mean of the individual *P*-values: $(p_1 \cdot p_2 \ldots p_5)^{1/5}$. The final score of a method is the log-transformed geometric mean of the overall AUROC *P*-value ($\bar{p}_{\mathrm{AUROC}}$) and the overall AUPR *P*-value ($\bar{p}_{\mathrm{AUPR}}$): score $= -0.5 \cdot \log 10(\bar{p}_{\mathrm{AUROC}} \cdot \bar{p}_{\mathrm{AUPR}})$.

ACKNOWLEDGMENTS. We thank all participants of the challenge for their contribution. Three anonymous reviewers provided valuable feedback on the original manuscript. G.S. and R.P. acknowledge support of the NIH Roadmap Initiative, the Columbia University Center for Multiscale Analysis Genomic and Cellular Networks (MAGNet), and the IBM Computational Biology Center. The work was supported through the Swiss National Science Foundation Grant no. 200021–112060 (to D.F., T.S., C.M., and D.M), the SystemsX.ch initiative (WingX project), and a postdoctoral fellowship for D.M.

1. Levine AJ, Oren M (2009) The first 30 years of p53: growing ever more complex. *Nat Rev Cancer* 9:749–758.
2. De la Fuente A, Brazhnik P, Mendes P (2002) Linking the genes: Inferring quantitative gene networks from microarray data. *Trends Genet* 18:395–98.
3. Gardner TS, di Bernardo D, Lorenz D, Collins JJ (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 301:102–105.
4. Friedman N (2004) Inferring cellular networks using probabilistic graphical models. *Science* 303:799–805.
5. Di Bernardo D, et al. (2005) Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nat Biotechnol* 23:377–83.
6. Rice JJ, Tu Y, Stolovitzky G (2005) Reconstructing biological networks using conditional correlation analysis. *Bioinformatics* 21:765–73.
7. De la Fuente A, Makhecha DP (2006) Unravelling gene networks from noisy underdetermined experimental perturbation data. *Systematic Biol (Stevenage)* 153:257–262.
8. Bonneau R, et al. (2006) The Inferelator: An algorithm for learning parsimonious regulatory networks from systems-biology datasets de novo. *Genome Biol* 7:R36 Available at http://www.genomebiology.com/2006/7/5/R36.
9. Faith JJ, et al. (2007) Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol* 5:e8 Available at http://www.plosbiology.org/article/info%3Adoi%2F10.1371%2Fjournal.pbio.0050008.
10. Marbach D, Mattiussi C, Floreano D (2009) Replaying the evolutionary tape: Biomimetic reverse engineering of gene networks. *Annals of the New York Academy of Sciences* 1158:234–245.
11. Stolovitzky G, Monroe D, Califano A (2007) Dialogue on reverse-engineering assessment and methods: The dream of high-throughput pathway inference. *Annals of the New York Academy of Sciences* 1115:1–22.
12. Cantone I, et al. (2009) A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell* 137:172–181.
13. Michoel T, de Smet R, Joshi A, van de Peer Y, Marchal K (2009) Comparative analysis of module-based versus direct methods for reverse-engineering transcriptional regulatory networks. *BMC Syst Biol* 3:49 Available at http://www.biomedcentral.com/1752-0509/3/49.
14. Bansal M, Belcastro V, Ambesi-Impiombato A, di Bernardo D (2007) How to infer gene networks from expression profiles. *Mol Syst Biol* 3:78 Available at http://www.nature.com/msb/journal/v3/n1/full/msb4100120.html.
15. Margolin AA, et al. (2006) ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 7(Suppl 1):S7 Available at http://www.biomedcentral.com/1471-2105/7/S1/S7.
16. Stolovitzky G, Prill RJ, Califano A (2009) Lessons from the DREAM2 challenges. *Annals of the New York Academy of Sciences* 1158:159–195.
17. Moult J, et al. (2007) Critical assessment of methods of protein structure prediction-round vii. *Proteins* 69(Suppl 8):3–9.
18. Mendes P, Sha W, Ye K (2003) Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics* 19(Suppl 2):ii122–129.
19. Marbach D, Schaffter T, Mattiussi C, Floreano D (2009) Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *J Comput Biol* 16(2):229–239.
20. Milo R, et al. (2002) Network motifs: simple building blocks of complex networks. *Science* 298:824–827.
21. Shen-Orr SS, Milo R, Mangan S, Alon U (2002) Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat Genet* 31:64–68.
22. Barabasi AL, Albert R (1999) Emergence of scaling in random networks. *Science* 286:509–512.
23. Reguly T, et al. (2006) Comprehensive curation and analysis of global interaction networks in saccharomyces cerevisiae. *J Biol* 5:11.
24. Ackers GK, Johnson AD, Shea MA (1982) Quantitative model for gene regulation by lambda phage repressor. *Proc Natl Acad Sci USA* 79:1129–1133.
25. GeneNetWeaver project website: http://gnw.sourceforge.net.
26. De la Fuente A, Bing N, Hoeschele I, Mendes P (2004) Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics* 20:3565–3574.
27. Basso K, et al. (2005) Reverse engineering of regulatory networks in human B cells. *Nat Genet* 37:382–390.
28. DREAM project website: http://wiki.c2b2.columbia.edu/dream/results.
29. Yip KY, Alexander RP, Yan KK, Gerstein M (2010) Improved reconstruction of *in silico* gene regulatory networks by integrating knockout and perturbation data. *PLoS One* 5(1):e8121 Jan. 26.
30. Setty Y, Mayo AE, Surette MG, Alon U (2003) Detailed map of a cis-regulatory input function. *Proc Natl Acad Sci USA* 100:7702–7707.