Supporting Information

Marbach et al. 10.1073/pnas.0913357107

SI Text

SI Results and Discussion. ***Best performance methods did consistently well across sizes.*** We provided subchallenges with networks of size 10, 50, and 100 in order to compare the performance of inference methods on different network sizes. From the 29 methods that were applied to the networks of size 10, all but two were also applied to the networks of size 50. The two teams that participated only in the subchallenge with networks of size 10 did not benefit from specializing on small networks. They ranked 22nd and 27th, and their predictions were not significantly better than random predictions (overall p-values of 0.24 and 0.44, respectively). After removing these two methods from the ranking, we compared the relative performance of methods on networks of size 10 and size 50 (Fig. S2*A*). The best and the second-best methods were the same in both subchallenges. The methods from ranks 3 to 7 are also the same for both network sizes, though not in the same order. Thus, the methods that performed best on the small networks of size 10 also performed best on the intermediate networks of size 50.

Similar observations can be made when comparing the ranking of the 22 methods that were applied both to networks of size 50 and 100 (Fig. S2*B*). In this case, the correlation between the two rankings is even stronger. A notable exception is the method that ranked second on networks of size 10 and size 50, which ranked only 15th on networks of size 100. Even though the predictions of this method were of excellent quality for networks of size 50 (overall p-value $<10^{-31}$), they were barely significant for networks of size 100 (overall p-value $= 0.01$). In summary, best performance methods did consistently well across sizes, with few exceptions.

The top five methods are all based on a different approach. As mentioned in the main text, the most common types of approaches for network inference were all represented among the successful teams of the challenge. The identities of the top-five teams are given in Table S1. *B Team* relied strongly on a Gaussian model of the noise (2), which will be discussed more in detail in the following sections. The teams *Bonneau* and *Zuma* inferred different types of dynamical network models. Team *Bonneau* applied and extended a previously described algorithm, the Inferelator (3, 4), which uses regression and variable selection (5). Team *Zuma* introduced a nonparametric dynamical model that was inferred using a Bayesian approach (6). The teams *Usmtec347* and *Intigern* applied yet unpublished inference methods. The method of team *Usmtec347* was based on dynamic Bayesian networks. Team *Intigern* integrated predictions obtained using different correlation-based and information-theoretic measures.

Systematic prediction errors due to network motifs. We performed the network-motif analysis for all methods that inferred networks of size 50 and size 100 (networks of size 10 are too small for a statistically significant analysis). For the best five methods of the networks of size 100, the results are graphically represented and discussed in Fig. S3. A detailed description of the methodology is given in *SI Methods*. As discussed in the main text, we found that inference methods are affected by three types of systematic prediction errors. Fig. S3 shows that the inference methods have different strengths and weaknesses: whereas some are more robust to certain types of error, they are more strongly affected by other types of errors.

In Fig. 4 of the main text, we discussed the 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). In the more detailed analysis of Fig. S3*C*, two additional, minor effects can be observed.

First, in addition to the incorrectly predicted "shortcuts" ($1 \rightarrow 3$) in cascades, some inference methods have a slightly reduced prediction confidence for the true edges of cascades ($1 \rightarrow 2$ and $2 \rightarrow 3$). This may be due to: (1) incorrect prediction of the shortcut $1 \rightarrow 3$ *instead of* (and not *in addition to*) the true links $1 \rightarrow 2 \rightarrow 3$ and/or (2) a tendency for cascades to overlap with fan-ins.*

Second, the three links of feed-forward loops (FFLs) often have a reduced prediction confidence. This can be explained by the same types of systematic errors that occur in fan-ins and cascades. The links $1 \rightarrow 3$ and $2 \rightarrow 3$ of FFLs form a fan-in, and are thus affected by the fan-in error. The links $1 \rightarrow 2 \rightarrow 3$ form a cascade, thus, they have a reduced prediction confidence for the same reason as the corresponding links in the cascade motif. One minor effect remains to be explained: in FFLs, the prediction confidence was often slightly lower for edge $1 \rightarrow 3$ than for edge $2 \rightarrow 3$. For example, this is the case for all except the best-performer method in Fig. S3*C*. It seems that in addition to the fan-in error, which affects both these edges, the edge $1 \rightarrow 3$ is also affected by an additional systematic error, which could be called the FFL-error. This error occurs when a method interprets the variation of gene 3 to be due solely to the indirect regulation via gene $2 (1 \rightarrow 2 \rightarrow 3)$, instead of both the indirect and the direct regulation ($1 \rightarrow 2 \rightarrow 3$ and $1 \rightarrow 3$). However, since this effect was negligible compared to the fan-in error that affects these edges, we did not consider it as a fourth type of main prediction error.

We conclude that the fan-out, fan-in, and cascade errors are the three main types of systematic prediction errors observed on three-node motifs, and that inference methods are differentially affected by these errors. The network-motif analysis, demonstrated here using three-node motifs, could potentially be extended to colored motifs, higher-order motifs, or to the network dynamics by considering activity motifs (7), for instance.

Systematic prediction errors due to the indegree and outdegree of genes. Methods that are affected by the fan-in error have a reduced prediction confidence for edges that target genes with two or more inputs. Fig. 5 of the main text shows how the indegree of genes affects the prediction confidence for the best five methods on the networks of size 100. We have performed this analysis for all inference methods that were applied to networks of size 50 and 100, which confirms that the best-performer method had the most robust performance on high indegrees (Fig. S4).

As control, we did the same analysis also for the outdegree of genes. As expected, in contrast to the indegree, the increasing outdegree of genes does not affect the prediction confidence of links (Fig. S4).

Inaccurate prior assumptions induce systematic prediction errors. 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

*Note that motifs do not occur in isolation in networks and edges usually pertain to several "overlapping" motif instances. We intend to investigate potential effects due to overlapping motifs in future work.

mechanisms and kinetics of gene regulation underlying the expression data are more complicated, and also not known in advance. Thus, inference methods are bound to make simplifying prior assumptions, e.g., by adopting a linear gene network model, to name just one example.

However, if the prior assumptions are inaccurate, they may lead to systematic prediction errors. For example, the best-performing team strongly relied on an inference method based on a very simple principle: it predicts an interaction from gene A to gene B if, after a knockout of A, there is a significant change in the expression level of B. The significance of a change in the expression level was estimated using a Gaussian model of the noise in the gene expression data (2). This method is based on the inaccurate prior assumption that the change in the expression level of gene B is always due to a direct regulatory interaction $A \rightarrow B$ (in reality, it may also be due to an indirect interaction via other genes, e.g. $A \rightarrow C \rightarrow B$). This inaccurate prior assumption induced systematic cascade errors (see Fig. S3).

However, the best-performer method was also the most robust of all applied methods to the fan-in error (Fig. S3), i.e., it had better performance than other methods on genes that are combinatorially controlled by multiple regulatory inputs. Interestingly, the best-performer method makes a strong prior assumption on the type of noise in the gene expression data (Gaussian), but remains uncommitted to the type of regulatory dynamics in the networks. In contrast, according to our survey among the participants, other inference methods tend to make strong assumptions on the regulatory dynamics (e.g., by adopting simple, often linear, phenomenological functions to approximate combinatorial regulation of genes by multiple regulators). These assumptions are partly inaccurate compared to the more detailed, kinetic model of the benchmarks (they may be even less accurate compared to the complicated mechanisms and kinetics of biological gene networks, as discussed in the following sections). Consequently, these methods have a strongly reduced performance on genes with multiple regulatory inputs (the fan-in error), where their prior assumptions are inaccurate.

Is simpler better? The relatively low performance of the majority of methods might suggest that very sophisticated techniques were required to reliably infer the networks. This was not the case.

The best performer-team used four different models to infer the networks, including the Gaussian model of the noise mentioned in the previous section and three Ordinary Differential Equation (ODE) models (2). They intended to combine the predictions from the four models, however, finally they mainly relied on the predictions derived from the simple model of the noise and added only few of the predictions from the ODE models at lower ranks. Thus, the excellent quality of their predictions is mainly due to the performance of the simple method based on the model of the noise. We have confirmed this by showing that a very similar, but even simpler method, would have obtained approximately the same performance. We used only the knockout dataset. For every gene i, we computed the mean expression level μ_i and the standard deviation σ_i. Next, we evaluated how much the expression level x_{ij} of gene i in the knockout experiment of gene j changed by the Z-score $z_{ij} = (x_{ij} - \mu_i)/\sigma_i$. The absolute value of z_{ij} was used as a measure of confidence for the edge from gene j to gene i, i.e., the lists of predictions were ordered according to the absolute values of the Z-scores. This method would have ranked second on the networks of size 10, first on the networks of size 50, and it would have tied for the first place with the best-performer method on the networks of size 100.

The Z-score method outlined in the previous paragraph is arguably one of the simplest conceivable network inference approaches. As the best-performer method based on the Gaussian model of the noise, it is based on the inaccurate assumption that any change in the expression level of a gene is due to a direct regulatory interaction, which induces systematic cascade errors (see previous section). Still, this "naive" approach consistently outperformed more sophisticated, state-of-the-art methods in this benchmark.

The fact that simple, but effective methods can often outperform more advanced, theoretically motivated approaches has already been observed in several in vivo challenges of the previous edition of DREAM. For example, Baralla et al. used both a simple approach based on Pearson or Spearman correlation, and a more advanced approach based on partial correlation analysis, in challenges 1 and 3 of DREAM2 (9). The goal in DREAM2 challenge 1 was to predict the direct regulatory targets of the human transcription-factor BCL6 from a collection of microarray experiments, and the goal of Challenge 3 was to infer the structure of an in vivo synthetic-biology network in yeast from two time-series datasets (10, 11). As Baralla et al. note: "[the] simpler approaches completely outperformed the more elegant partial correlations approach, presumably because the latter approach relies more strongly on the correctness of underlying assumptions" (9).

The results reported here support this conclusion: sophisticated methods that would in theory be expected to perform better than the "naive" approach described above, were more strongly affected by inaccurate prior assumptions in practice, as discussed in the previous section. Note that these observations do not imply that *always* simpler methods perform better. If their underlying assumptions are correct, advanced methods based on more accurate models would be expected to outperform the simple approaches described above. In other words, the method and its assumptions has to adapt to the type of data at hand.

Reliable gene network inference from gene expression data remains an unsolved problem. The performance of most reverse engineering methods was unsatisfactory (see *Discussion* of the main text). Although the measured performance is in principle specific to the *in silico* networks used here, it is unlikely that a method would perform better in an equivalent in vivo application. First, we provided much more and better quality data than is typically available in biological applications (knockouts and knockdowns for every gene, plus dozens of time series). Secondly, the prior assumptions of inference methods are in general more accurate, and would thus be expected to lead to better predictions, for the *in silico* networks than for biological gene regulatory networks. For example, biological gene networks have additional layers of control, such as posttranscriptional regulation, which are not modeled by prevalent gene network inference methods. We could list many more points in which the prior assumptions of current network inference methods are more accurate, and would thus be expected to lead to better predictions, for the *in silico* networks than for biological gene regulatory networks.

It should be noted that some methods may perform better using other types of experimental data. For example, linear models only provide a good approximation of the nonlinear dynamics for weak perturbations, as Tegner et al. note: "[…] if the perturbation is too large, the gene network will be pushed outside of its linear response regime and this will weaken the algorithms assumption of linearity. As a result, the error in the predicted network will increase" (8). For this reason, we provided in addition to the knockout data, where the expression of genes is completely suppressed, also knockdown data, where the expression of genes is only reduced by half. However, it may be that the knockdowns were still too strong of a perturbation for the linear assumption to hold, and that the performance of the methods based on linear models would be better when using weaker perturbations.

Community predictions are more reliable than individual inference methods. In the main text, we have discussed the performance of community predictions on the networks of size 10. Here, we first describe in more detail how we combined predictions of participating teams, and then we analyze the performance of the resulting community predictions for all network sizes.

Remember that the submission format of the DREAM3 *in silico* challenge was a ranked list of edge predictions. The question is thus how to optimally combine an ensemble of such lists, submitted by different participating teams, to form a community prediction.

We use a straightforward approach to combine the ensemble of edge-prediction lists, which consists of simply taking the average rank for each edge. Thus, the list of edge predictions of the community is ordered according to the average ranks of the edges in the ensemble, as illustrated in Fig. S5.

As described in the main text, 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 (there are three subchallenges corresponding to networks of size 10, 50, and 100). Here, we formed in addition community predictions without including the best-performers, i.e., we removed the best-performing team and then formed communities composed of the top-two methods, top-three methods, etc. For each of these communities, we derived community predictions for the five networks of the subchallenge.

The scores of both the community predictions and the individual teams are shown in Fig. S6. Some of the community predictions outperform the best-performing team on networks of size 10 and size 50. For example, the community of the top-five teams would have won these two subchallenges. As more and more teams with a bad performance are added to the community, the accuracy of the community prediction decreases. However, the performance is remarkably robust. Even when combining all methods, the majority of which have low scores (remember that about a third of the methods did not perform better than random guessing), the community prediction still ranks second on networks of size 10 and 100, and third on networks of size 50.

The robustness of the community predictions is even more evident in the communities that don't include the best-performer (the squares in Fig. S6). For example, the community composed of all teams (second-place to last-place) outperforms the second-place team on networks of size 100. In summary, the community predictions are consistently as good or better than the best methods of the ensemble, and this even when the majority of the methods of the ensemble have a low performance.

SI Methods. ***Gene network model.*** We model transcriptional regulatory networks consisting of genes, mRNA, and proteins. The state of the network is given by the vector of mRNA concentrations $\mathbf{x}$ and protein concentrations $\mathbf{y}$. We model only transcriptional regulation, where regulatory proteins (transcription factors) control the transcription rate (activation) of genes. The gene network is modeled by the system of differential equations

$$\frac{dx_i}{dt} = m_i \cdot f_i(\mathbf{y}) - \lambda_i^{\mathrm{RNA}} \cdot x_i \qquad \textbf{[S1]}$$

$$\frac{dy_i}{dt} = r_i \cdot x_i - \lambda_i^{\mathrm{Prot}} \cdot y_i, \qquad \textbf{[S2]}$$

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 $\mathbf{y}$.

Gene regulation is modeled using a standard approach based on thermodynamics (12, 13). Good introductory texts are references (14, 15). The basic assumption of this approach is that binding of TFs to cis-regulatory sites on the DNA is in quasi-equilibrium, since it is orders of magnitudes faster than transcription and translation. In the most simple case, a gene i is regulated by a single TF j. In this case, its promoter has only two states: either the TF is bound (state S_1) or it is not bound (state S_0). The probability $P\{S_1\}$ that the gene i is in state S_1 at an instant in time is given by the *fractional saturation*, which depends on the TF concentration y_j

$$P\{S_1\} = \frac{\chi_j}{1+\chi_j} \quad \text{with} \quad \chi_j = \left(\frac{y_j}{k_{ij}}\right)^{n_{ij}}, \qquad \textbf{[S3]}$$

where k_{ij} is the dissociation constant and n_{ij} the Hill coefficient. At concentration $y_j = k_{ij}$ the saturation is half-maximal, i.e., the promoter is bound by the TF 50% of the time. Many TFs bind DNA as homodimers or higher homooligomers. This and other mechanisms that affect the effective cooperativity of promoter binding are approximated by the Hill coefficient n_{ij}, which determines the "steepness" of the sigmoid described by Eq. **3**.

The bound TF activates or represses the expression of the gene. In state S_0 the relative activation is α_0 and in state S_1 it is α_1. Given $P\{S_1\}$ and its complement $P\{S_0\}$, it is straightforward to derive the input function $f_i(y_j)$, which computes the mean activation of the gene as a function of the TF concentration y_j

$$f(y_j) = \alpha_0 P\{S_0\} + \alpha_1 P\{S_1\} = \frac{\alpha_0 + \alpha_1 \chi_j}{1+\chi_j}. \qquad \textbf{[S4]}$$

This approach can be used for an arbitrary number of regulatory inputs. A gene that is controlled by N TFs has 2^N states: each of the TFs can be bound or not bound. Thus, the input function for N regulators would be

$$f(\mathbf{y}) = \sum_{m=0}^{2^N-1} \alpha_m P\{S_m\}. \qquad \textbf{[S5]}$$

Using thermodynamics, it is straightforward to compute the probability $P\{S_m\}$ for every state m. For example, the resulting expression for a gene with two inputs is

$$f(y_1, y_2) = \frac{\alpha_0 + \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 \rho v_1 v_2}{1 + v_1 + v_2 + \rho v_1 v_2} \qquad \textbf{[S6]}$$

with $v_j = (y_j/k_j)^{n_j}$, where n_j is the Hill coefficient, k_j the dissociation constant, ρ the cooperativity factor, and α_s are the relative activations when none of the TFs (α_0), only the first (α_1), only the second (α_2), or both TFs are bound (α_3).

We nondimensionalize the gene network models as described in the Supplementary Information of reference (16). As von Dassow et al. note: "This entails replacing every occurrence of dimensioned state variables (concentrations of molecular species and time) with scaled products yielding new state variables free of units. […] The dimensionless model is identical to the dimensional one since it is merely an algebraic transformation." (16). One of several advantages of nondimensionalization is that it is much more easier to generate biologically plausible instances of the dimensionless model. We will provide a more detailed description of the gene network model and the nondimensionalization elsewhere.

Clearly, the model outlined above is still extremely simplied compared to the real biological mechanisms. However, two of the key simplifying assumptions of most reverse engineering

methods based on dynamical models are not made. First, mRNA and proteins are not lumped into a single state variable. Second, the gene regulation function is not phenomenologically approximated by additive or multiplicative terms, but is derived using the thermodynamic approach. One consequence of the thermodynamical model is that it can express all types of regulatory logic (AND, OR, etc). We have initialized the networks of the DREAM3 *in silico* challenges such that there is approximately equal amounts of independent and synergistic gene regulation. Thus, a priori neither the additive nor the multiplicative phenomenological modeling approach used in gene network reverse engineering should be favored in these benchmarks.

Simulation of gene expression data. Gene knockouts were simulated by setting the maximum transcription rate m_i of the deleted gene to zero, knockdowns by dividing it by two. 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.[†]

For each time series, we used a different random initial condition for the mRNA and protein concentrations. The initial mRNA concentrations $x_i(0)$ were obtained by adding a random number from a Gaussian distribution with mean zero and standard deviation 0.5 to the wild-type steady-state level of every gene. We assume that the multifactorial perturbation that led to the perturbed state $x_i(0)$ was applied for sufficient time for the protein levels to stabilize at their new steady state. Thus, the initial protein concentrations were obtained by setting $dy_i/dt = 0$ in Eq. **2**

$$y_i(0) = \frac{r_i}{\lambda_i^{\text{Prot}}} \cdot x_i(0). \qquad \textbf{[S7]}$$

Each time series consists of 21 time points (from $t = 0$ until $t = 200$). Trajectories were obtained by integrating the networks from the given initial conditions using a Runge-Kutta 4/5 solver with variable step size of the Open Source Physics library (17).

White noise with a standard deviation of 0.05 was added *after* the simulation to the generated gene expression data. Concentrations that became negative due to the addition of noise were set to zero.

Network-motif analysis. The goal of the network-motif analysis is to evaluate, for a given network inference method, whether some types of edges of motifs are systematically predicted less (or more) reliably than expected. This involves: (1) determining the prediction confidence of edges pertaining to different motifs, (2) determining the expected prediction confidence independently of motifs (the *background prediction confidence*), and (3) evaluating whether divergences of the prediction confidence of motif edges from the expected prediction confidence are statistically significant (cf. Fig. S3).

Definition of prediction confidence. Given a network prediction in the format described in the main text, we define the prediction confidence of edges as their rank in the list of edge predictions. We scale the prediction confidence such that the first edge in the list has confidence 100%, and the last edge in the list has confidence 0%.

[†]For networks of size 50, we provided the same number of time series (23) as Pedro Mendes did for his networks of size 50 in the DREAM2 *in silico* challenges, to allow comparison of results between the two challenges. For networks of size 10 and 100, we scaled the number of provided time series with the network size (two times more for size 100, five times less for size 10).

Determining prediction confidence of motif edges. First we need to identify all three-node motif instances in the target network (the same approach could be used for higher-order motifs). We use the efficient algorithm of Wernicke for this purpose (18). Next, for every type of motif edge, we determine the prediction confidence of all its instances. For example, we identify all links of type $1 \to 2$ of cascades in the target network (the numbering of the nodes of the motifs is defined in Fig. S3), and we record the prediction confidences that were assigned to these links by the given network inference method. More formally, we construct the set $C^{1\to2}_{\text{cascade}} = \{c_k\}$, where c_k is the prediction confidence of the link $1 \to 2$ in the k'th cascade of the target network. Note that we also record the prediction confidences of "absent edges" of the motifs, for example, $C^{1\to3}_{\text{cascade}}$ is the set of prediction confidences of the "shortcuts".

For every motif type m, we determine the set of prediction confidences assigned by the inference method to each of its six possible edges ($C^{1\to2}_m$, $C^{1\to3}_m$, $C^{2\to1}_m$, $C^{2\to3}_m$, $C^{3\to1}_m$, and $C^{3\to2}_m$). Note that fan-in as well as fan-out motifs are symmetric: nodes 2 and 3 can't be distinguished (see Fig. S3). Thus, both fan-ins and fan-outs have only three types of edges: $1 \to 2$, $2 \to 1$, and $2 \to 3$. The edges $1 \to 3$, $3 \to 1$ and $3 \to 2$ are equivalent to $1 \to 2$, $2 \to 1$, and $2 \to 3$, respectively. The prediction confidence of equivalent edges is recorded in the same set. For example, the set $C^{1\to2}_{\text{fanout}}$ contains all prediction confidences assigned by the inference method to outgoing edges of fan-outs ($1 \to 2$ or $1 \to 3$).

Determining the background prediction confidence. Before we can analyze the effect of motifs on the edge-prediction confidence, we first need to determine the expected edge-prediction confidence independently of motifs. There are three types of predicted edges:

1. Predicted edges that are *true edges* of the target network. Their background prediction confidence is given by $C_{\text{true\_edge}}$, which is the set of prediction confidences that were assigned by the inference method to the edges that are part of the target network.
2. Predicted edges for which the directionality is incorrect. We call these *back edges*. Their background prediction confidence is given by $C_{\text{back\_edge}}$, which is the set of prediction confidences assigned to edges $B \to A$ that are *not* part of the target network, but for which the edge in the opposite direction $A \to B$ *is* part of the target network.
3. Predicted edges between two nodes that are *not* connected by an edge in the target network. We call these *absent edges*. Their background prediction confidence is given by $C_{\text{absent\_edge}}$, which is the set of confidences of predicted edges between nodes that are not directly connected in the target network.

Note that in Fig. S3*B*, we have only shown the median prediction confidence of *true edges* ($1 \to 2$) and *back edges* ($2 \to 1$)—the median prediction confidence of *absent edges* was not shown.

Evaluating the divergence of the prediction confidence of motif edges from the background prediction confidence. We use the Wilcoxon-Mann-Whitney rank-sum test to compare the motif edge-prediction confidences with their corresponding background prediction confidence. The prediction confidence of true edges of motifs (e.g., $C^{1\to2}_{\text{cascade}}$ and $C^{2\to3}_{\text{cascade}}$) are compared with $C_{\text{true\_edge}}$, the prediction confidence of back edges of motifs (e.g., $C^{2\to1}_{\text{cascade}}$ and $C^{3\to2}_{\text{cascade}}$) are compared with $C_{\text{back\_edge}}$, and absent edges of motifs (e.g., $C^{1\to3}_{\text{cascade}}$ and $C^{3\to1}_{\text{cascade}}$) are compared with $C_{\text{absent\_edge}}$. We use Bonferroni correction for multiple hypothesis testing.

1. GeneNetWeaver project website: http://gnw.sourceforge.net
2. 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*, Jan 26;5(1):e8121.
3. Bonneau R, et al. (2006) The Inferelator: An algorithm for learning parsimonious regulatory networks from systems-biology datasets de novo. *Genome Biol*, 7:R36.
4. Bonneau R, et al. (2007) A predictive model for transcriptional control of physiology in a free living cell. *Cell*, 131:354–1365.
5. Madar A, Greenfield A, Vanden-Eijnden E, Bonneau R (2010) Network inference using dynamic context likelihood of relatedness and inferelator 1.0. *PLoS One*, in press.
6. Äijö T, Lähdesmäki H (2009) Learning gene regulatory networks from gene expression measurements using nonparametric molecular kinetics. *Bioinformatics*, 22:937–2944.
7. Chechik G, et al. (2008) Activity motifs reveal principles of timing in transcriptional control of the yeast metabolic network. *Nat Biotechnol*, 26:1251–1259.
8. Tegner J, Yeung MKS, Hasty J, Collins JJ (2003) Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proc Natl Acad Sci USA*, 100:5944–949.
9. Baralla A, Mentzen WI, de la Fuente A (2009) Inferring gene networks: dream or nightmare? *Annals of the New York Academy of Sciences*, 1158:246–256.
10. Stolovitzky G, Prill RJ, Califano A (2009) Lessons from the DREAM2 challenges. *Ann N Y Acad Sci*, 1158:159–195.
11. Cantone I, et al. (2009) A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell*, 137:72–181.
12. Ackers GK, Johnson AD, Shea MA (1982) Quantitative model for gene regulation by lambda phage repressor. *Proc Natl Acad Sci U S A*, 79:1129–33.
13. Shea MA, Ackers GK (1985) The or control system of bacteriophage lambda. a physical-chemical model for gene regulation. *J Mol Biol*, 181:211–230.
14. Bower JM, Bolouri H, editors (2004) Computational modeling of genetic and biochemical networks. *MIT Press*.
15. Bintu L, et al. (2005) Transcriptional regulation by the numbers: models. *Curr Opin Genet Dev*, 15:116–124.
16. Von Dassow G, Meir E, Munro EM, Odell GM (2000) The segment polarity network is a robust developmental module. *Nature*, 406:188–192.
17. Open Source Physics website: http://www.opensourcephysics.org
18. Wernicke S (2006) Efficient detection of network motifs. *IEEE/ACM Trans Comput Biol Bioinform*, 3:347–59.

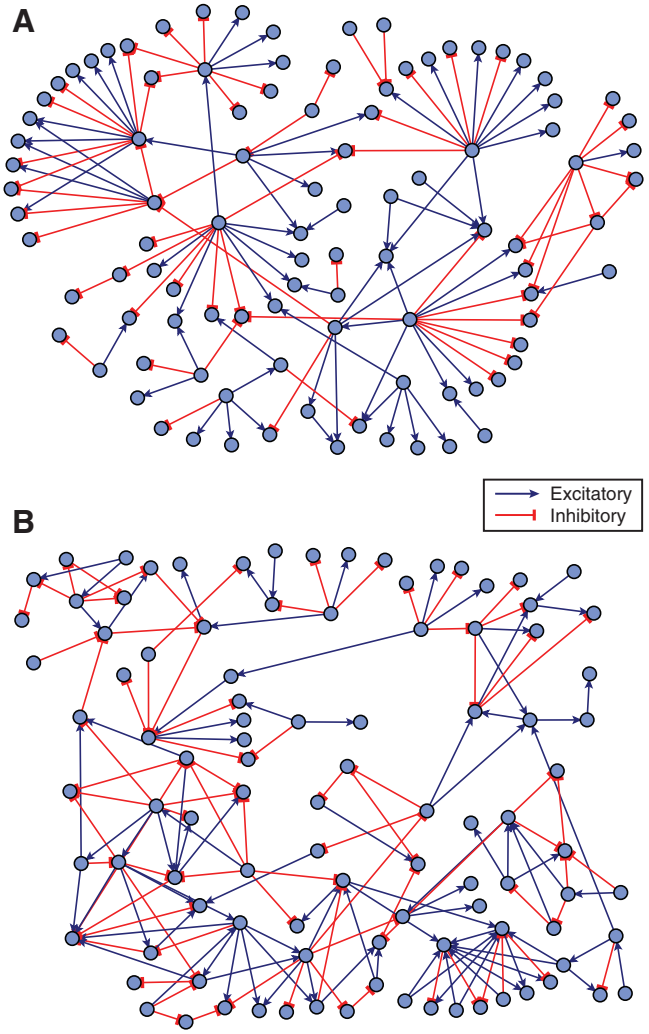

Fig. S1. Two target (gold standard) networks of size 100. (*A*) was extracted from an *E.coli* and (*B*) from a yeast gene network (see *Methods* of the main text). The graphs of all target networks of the challenge are available in the file *DREAM3 In Silico Challenges.zip* on the GeneNetWeaver website (1).

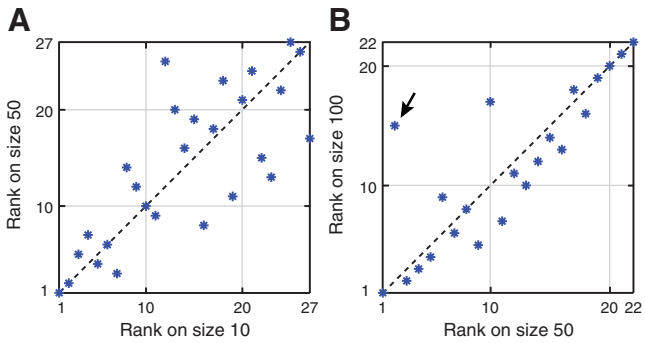

Fig. S2. Best performance methods did consistently well across sizes. (*A*) The rankings of the 27 teams that participated both in the subchallenges with networks of size 10 and 50. The methods with a good rank on networks of size 10 also performed well on networks of size 50 (points in the bottom-left corner of the plot). (*B*) The rankings of the 22 teams that participated both in the subchallenges with networks of size 50 and 100. The ranking is very similar in the two subchallenges. A notable exception is the second-place method on networks of size 50, which had a poor performance on networks of size 100 (indicated by the arrow).

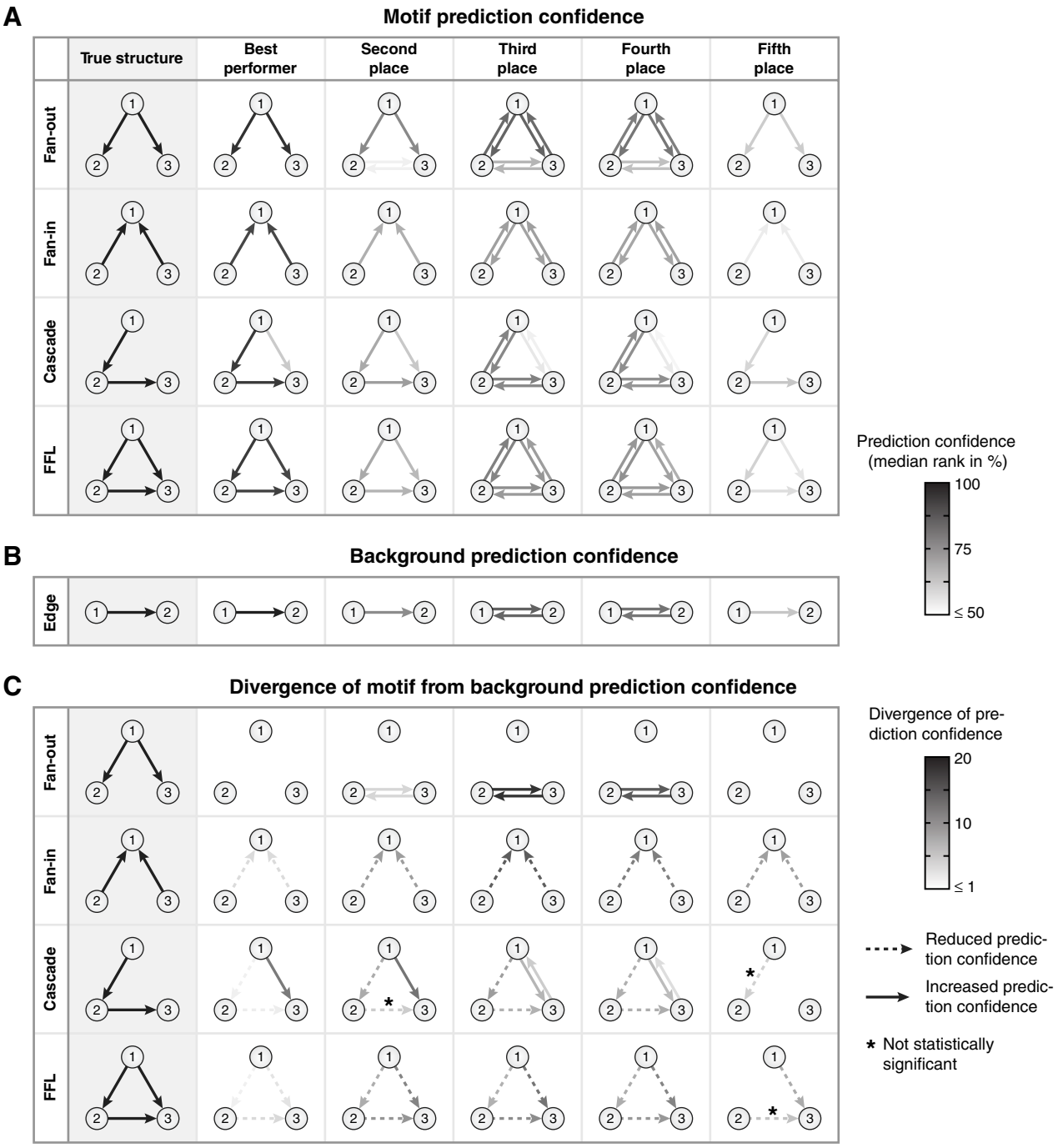

Fig. S3. Inference methods are differentially affected by systematic prediction errors. The first column shows the true connectivity of the motifs, the following columns show how they were predicted by the best five methods of the networks of size 100. (*A*) Median *prediction confidence* (see *SI Methods* for a definition) of the different edges (same as Fig. 4 of main text). Note that the third and fourth-place methods apparently can't distinguish the directionality of edges. (*B*) The background prediction confidence, which is the median prediction confidence of all links of the network, independently of which motifs they belong to. If the motifs had no effect on the prediction confidence, all edges of the motifs would be inferred with this background prediction confidence. (*C*) The divergence of the prediction confidence of the different motif edges from the background prediction confidence. The darkness of the links is proportional to the difference of the motif prediction confidence (*A*) from the background prediction confidence (*B*). Dashed arrows indicate a reduced, solid arrows an increased median prediction confidence. All differences are statistically significant at a level of 0.01 using a two-sided Wilcoxon-Mann-Whitney rank-sum test and Bonferroni correction for multiple hypothesis testing, except for the marked edges (*). The five methods are affected to various degrees by the three systematic prediction errors discussed in Fig. 4 of the main text. The second, third, and fourth-place methods have a tendency for false positives between coregulated genes (fan-out error). All methods have a reduced prediction confidence for edges that target genes with multiple inputs (fan-in error), but the best-performer method to a lesser degree than the others. However, the best-performer method has one of the strongest cascade errors (shortcuts 1 → 3). Since the third and fourth-place methods can't infer the directionality of interactions, the wrongly predicted "shortcut" appears in both directions.

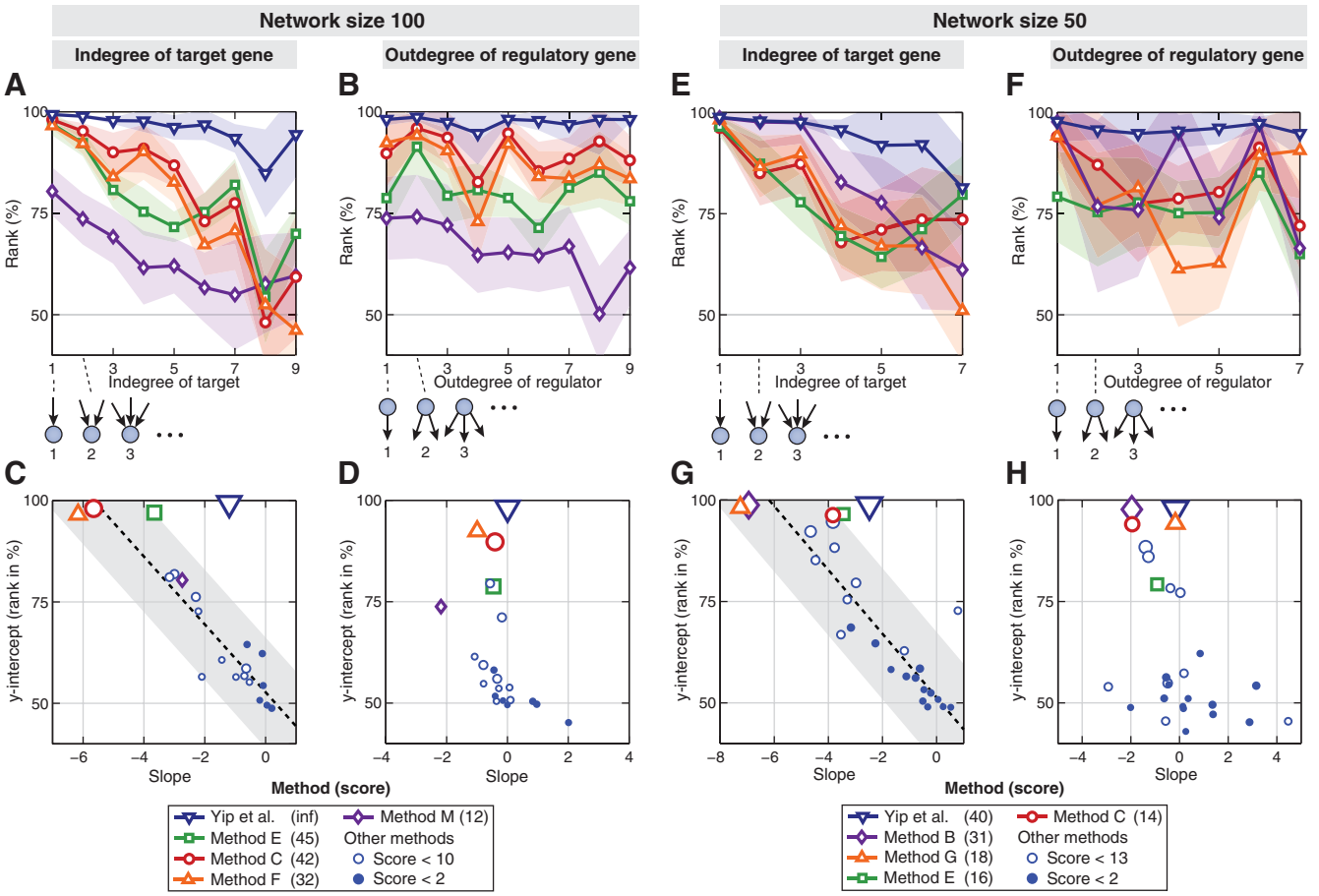

Fig. S4. How the indegree and outdegree of genes affects the prediction confidence. The first row shows, for the best five methods on the networks of size 100 and size 50, the median prediction confidence assigned to edges of varying indegree of target genes and outdegree of regulators (the prediction confidence of an edge is its rank in the list of edge predictions, as defined above for the network-motif analysis). The shaded areas indicate 95% confidence intervals for the medians. Note that the top-five methods (first row) are not all the same for networks of size 100 and size 50 (see legends). (*A*, *E*) Single-input links were reliably predicted with a similar, high prediction confidence by several inference methods (points in the top left corner). However, for all but the best-performer method, the performance drops drastically for higher indegrees. Note that the results for the smaller networks of size 50 are noisier, and the confidence intervals wider, due to the smaller sample sizes. We summarized these plots by fitting for every method a line through the curve: (*C*, *G*) show the y-intercept and the slope of the fitted lines for all methods, which summarize the performance on single inputs and the robustness to the fan-in error, respectively. A linear regression analysis (dashed line, the shaded area is the 95% prediction interval) confirms that the best-performer method (Yip et al.) is an outlier. It has the most robust performance on high indegrees (flattest slope) of all methods that have a reasonably high score (y-intercept >75%). The results for networks of size 50 are noisier, but consistent with the observations on the networks of size 100. (*B*, *F*) In contrast to the indegree of target genes, the outdegree of regulators does not affect the prediction confidence of their links. (*D*, *H*) Consequently, the slopes of the fitted lines are close to zero. There is no significant correlation between the y-intercept and the slope. Thus, the regression done in the corresponding plots for the indegrees is not applicable here.

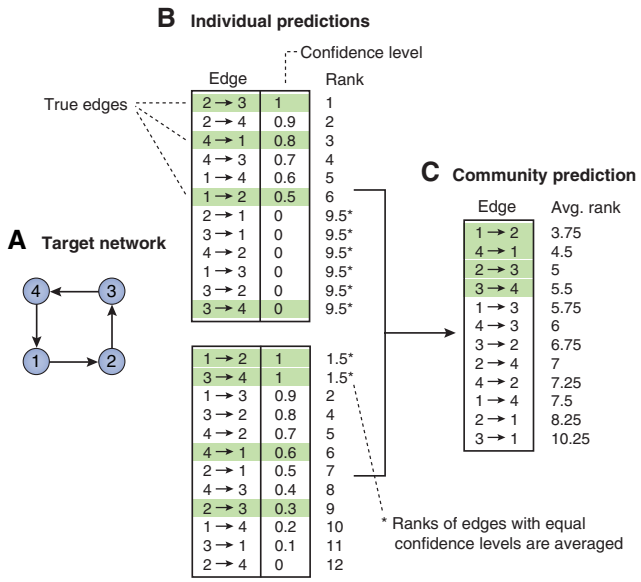

Fig. S5. Example of a community prediction formed from two individual network predictions. (*A*) The target network of this example is a loop of four genes. (*B*) Two possible lists of edge predictions. The lists are ranked according to the confidence levels of the edges, the most confident prediction at the top of the list has rank 1. The true edges of the target network are highlighted. (*C*) The community prediction is obtained by taking for each edge the average of its rank in the two individual predictions. Here, the community prediction is perfect (all true edges are at the top of the list). This example illustrates how a community prediction can be more accurate than the individual predictions that it is composed of.

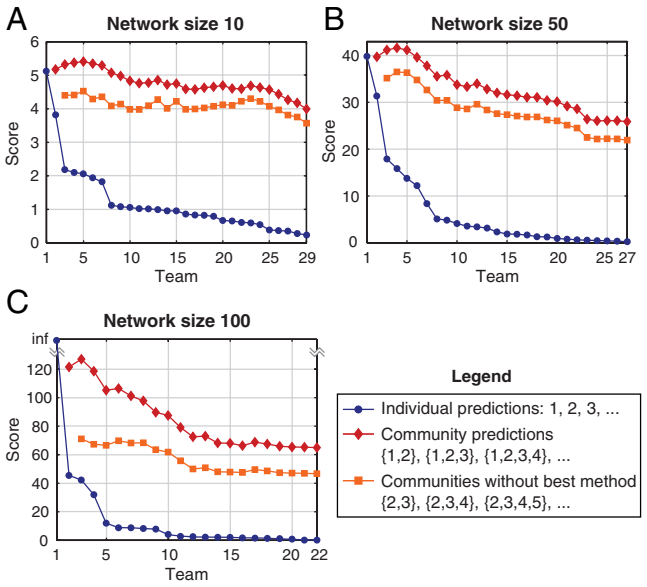

Fig. S6. Community predictions in the DREAM3 *in silico* challenge. The circles are the scores of the individual teams. The diamonds correspond to the scores of the different community predictions, obtained by combining the two best teams, the three best teams, the four best teams, etc. The squares correspond to the community predictions obtained without including the best-performer. The performance of the community is remarkably robust. Even when including all teams (rightmost diamonds), the score of the community is better than the second-best method on networks of size 10 and 100, and better than the third-best method on networks of size 50.

Table S1. The identities and methods of the five top-performing teams in the DREAM3 *in silico* network inference challenge

Team name	Members	Affiliation	Rank	Type of method	References
B Team	Kevin Y. Yip, Roger P. Alexander, Koon-Kiu Yan, and Mark Gerstein	Yale University, USA	1st on all sizes	Statistical method	(2)
USMtec347	Peng Li and Chaoyang Zhang	University of Southern Mississippi, USA	2nd on size 10, 2nd on size 50	Bayesian network	-
Bonneau	Aviv Madar, Alex Greenfield, Eric Vanden-Eijnden, and Richard Bonneau	New York University, USA	2nd on size 100	Dynamical model	(3, 4, 5)
Intigern	Xuebing Wu, Feng Zeng, and Rui Jiang	Tsinghua University, China	3rd on size 10, 3rd on size 100	Statistical method	-
Zuma	Tarmo Äijö and Harri Lähdesmäki	Tampere University of Technology, Finland	3rd on size 50	Dynamical model	(6)