GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods

Schaffter, Thomas; Marbach, Daniel; Floreano, Dario

Abstract: Motivation: Over the last decade, numerous methods have been developed for inference of regulatory networks from gene expression data. However, accurate and systematic evaluation of these methods is hampered by the difficulty of constructing adequate benchmarks and the lack of tools for a differentiated analysis of network predictions on such benchmarks.

Results: Here we describe a novel and comprehensive method for in silico benchmark generation and performance profiling of network inference methods available to the community as an open-source software called GeneNetWeaver (GNW). In addition to the generation of detailed dynamical models of gene regulatory networks to be used as benchmarks, GNW provides a network motif analysis that reveals systematic prediction errors, thereby indicating potential ways of improving inference methods. The accuracy of network inference methods is evaluated using standard metrics such as precision-recall and receiver operating characteristic (ROC) curves. We show how GNW can be used to assess the performance and identify the strengths and weaknesses of six inference methods. Furthermore, we used GNW to provide the international DREAM (Dialogue for Reverse Engineering Assessments and Methods) competition with three network inference challenges (DREAM3, DREAM4, and DREAM5).

Availability: GNW is available at http://gnw.sourceforge.net along with its Java source code, user manual, and supporting data.

DOI: https://doi.org/10.1093/bioinformatics/btr373

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: https://doi.org/10.5167/uzh-79529

Originally published at:

DOI: https://doi.org/10.1093/bioinformatics/btr373
GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods

Thomas Schaffer 1, Daniel Marbach 2,3, and Dario Floreano 1 *

1 Laboratory of Intelligent Systems, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
2 MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, Massachusetts, USA
3 Broad Institute of MIT and Harvard, Cambridge, Massachusetts, USA

ABSTRACT

Motivation: Over the last decade, numerous methods have been developed for inference of regulatory networks from gene expression data. However, accurate and systematic evaluation of these methods is hampered by the difficulty of constructing adequate benchmarks and the lack of tools for a differentiated analysis of network predictions on such benchmarks.

Results: Here we describe a novel and comprehensive method for in silico benchmark generation and performance profiling of network inference methods available to the community as an open-source software called GeneNetWeaver (GNW). In addition to the generation of detailed dynamical models of gene regulatory networks to be used as benchmarks, GNW provides a network motif analysis that reveals systematic prediction errors, thereby indicating potential ways of improving inference methods. The accuracy of network inference methods is evaluated using standard metrics such as precision-recall and receiver operating characteristic (ROC) curves. We show how GNW can be used to assess the performance and identify the strengths and weaknesses of six inference methods. Furthermore, we used GNW to provide the international DREAM (Dialogue for Reverse Engineering Assessments and Methods) competition with three network inference challenges (DREAM3, DREAM4, and DREAM5).

Availability: GNW is available at http://gnw.sourceforge.net along with its Java source code, user manual, and supporting data.

Contact: dario.floreano@epfl.ch

1 INTRODUCTION

A challenging issue in systems biology is the development of computational tools for the reverse engineering of gene regulatory networks from quantitative experimental data. Over the last decade, high-throughput assays for mRNA expression have opened the door to the inference of regulatory networks by allowing simultaneous measurements of the expression levels of thousands of genes. Technologies such as spotted microarrays (Davis et al., 1995) and oligonucleotide chips (Lockhart et al., 1996) have enabled genome-wide quantification of differential gene expression profiles and, more recently, short read sequencing technologies such as RNA-seq (Mortazavi et al., 2008) have provided more precise quantification of mRNA levels.

Researchers have proposed a plethora of methods for reverse engineering the complex network of interactions between the genes and their RNA and protein products (also called regulatory program) from spatial and temporal high-throughput gene expression data (Bansal et al., 2007). Regulatory networks are often represented as directed, signed graphs in which nodes represent genes or transcription factors (TF). In this context, edges correspond to enhancing or inhibitory regulations that affect gene transcription rates. Network inference methods rely on various computational approaches such as correlation (Rice et al., 2005), mutual information (MI) (Margolin et al., 2006; Faith et al., 2007), ordinary differential equations (ODE) models (Bonneau et al., 2006; Aijö and Lähdesmäki, 2009), Bayesian networks (Yu et al., 2004), or hybrid algorithms (Yip et al., 2010). Numerous methods have been developed for inference of gene regulatory networks, however relatively little effort has been put into evaluating the performance of those methods on adequate benchmarks. So far, three main strategies have been proposed to generate benchmark networks. A first strategy consists in evaluating network predictions made by reverse engineering algorithms on well studied in vivo pathways from model organisms (Kim et al., 2003; Gama-Castro et al., 2011). However, those networks are incomplete maps of the physical interactions in the cell that are responsible for cellular functions and using them as benchmarks imply making errors when evaluating network predictions. Another strategy consists of genetically engineering synthetic in vivo networks (Camacho and Collins, 2009; Cantone et al., 2009). The main drawback of this strategy is that only a few small networks are available. Yet another strategy consists in developing in silico gene regulatory networks that can be simulated to produce artificial gene expression data. The simulation of in silico networks has the advantages of being fast, easily reproducible, and less expensive than biological experiments. A few instances of small in silico networks with handcrafted topologies (Kremling et al., 2004) have been proposed as benchmarks for reverse engineering algorithms. More recently, several generators have been developed to automate the construction of in silico regulatory networks including up to thousands of genes to be used as benchmark networks for reverse engineering.

* to whom correspondence should be addressed
algorithms (Mendes et al., 2003; Van den Bulcke et al., 2006; Di Camillo et al., 2009).

Benchmark generators such as AGN (Mendes et al., 2003) aim to produce in silico gene networks exhibiting topological properties observed in biological networks using Erdös-Rényi, Watts-Strogatz (small-world), or Albert-Barabási (scale-free) random graph models. However, the structures generated using random graphs capture only few of the structural properties of gene regulatory networks (Van den Bulcke et al., 2006) and do generally not display important properties such as modularity (Ravasz et al., 2002) or occurrences of network motifs, which are statistically over-represented regulatory patterns in biological networks (Shen-Orr et al., 2002). Instead of constructing more complex random structures based on graph theory, which may be difficult to justify (Mendes et al., 2003), SynTReN (Van den Bulcke et al., 2006) and ReTRN (Li et al., 2009) chose to generate network structures by extracting parts of known in vivo regulatory network structures. This approach has the advantage of capturing several structural properties observed in in vivo network structures (Van den Bulcke et al., 2006).

In order to produce gene expression data, the generated structures must be endowed with dynamical models of gene regulation. Systems of non-linear ordinary differential equations (ODE) are widely used (Roy et al., 2008; Hache et al., 2009), but other approaches exist (Di Camillo et al., 2009). ODE systems allow to continuously describe levels of gene products and rates of reactions taking place in the network models where biological processes that have not been fully characterized yet are abstracted. Because current high-throughput technologies do not allow the monitoring of protein expression as microarrays do for RNA (Di Camillo et al., 2009), some benchmark generators consider mRNA as a proxy for protein expression and thus do not model translation independently of transcription (Van den Bulcke et al., 2006; Li et al., 2009). Protein expression, however, does not correlate perfectly with mRNA expression in real biological systems due in part to different degradation rates of mRNA and protein products (Belle et al., 2006). RENCO (Roy et al., 2008), GeNGe (Hache et al., 2009), and GRENDel (Haynes and Brent, 2009) are examples of available benchmark generators considering both transcription and translation processes in their respective dynamical models.

Here we describe a method for in silico benchmark generation and performance profiling of network inference methods available to the community as an open-source software called GeneNetWeaver (see Figure 1). GNW has an intuitive graphical user interface that makes the generation and simulation of gene network models as simple as a few clicks. Network topologies are generated by extracting modules from known in vivo gene regulatory network structures such as those of E. coli (Gama-Castro et al., 2011) and S. cerevisiae (Kim et al., 2003). These structures are then endowed with detailed dynamical models of gene regulation including both transcription and translation processes using a thermodynamic approach accounting for both independent and synergistic interactions (Ackers et al., 1982). Expression data can be generated either deterministically or stochastically to model molecular noise in the dynamics of the networks, and experimental noise can be added using a model of noise observed in microarrays (Stolovitzky et al., 2005). Different types of in vivo
experimental procedures, such as wild type, knockout (null-mutant), knockdown (heterozygous), and multifactorial perturbations, can be reproduced by the software. In addition, a unique feature of GNW is the systematic and comparative evaluation of predictions by different inference methods, which none of the existing benchmark generators provide. GNW performs an exhaustive network motif analysis for a set of network predictions, which often reveals systematic prediction errors, thereby indicating potential ways of network reconstruction improvements. The accuracy of network inference is also assessed using standard metrics such as precision-recall and receiver operating characteristic (ROC) curves.

Furthermore, we show how GNW can be used to generate in silico benchmark suites to assess the performance and identify strengths and weaknesses of six network inference methods. We also show how the performance of those inference methods are affected by the structural properties and the size of the gene regulatory networks to infer, and how GNW can help to identify the most informative type of gene expression data to provide to a given inference method. Finally, we assess the performance of those six inference methods on the network inference challenge that we provided to the international DREAM4 competition (Dialogue for Reverse Engineering Assessments and Methods) (manuscript in preparation).

2 METHODS

2.1 Topology

Instead of using random graph models, which are known to only partly capture the structural properties of biological networks (Van den Bulcke et al., 2006), we generate network structures by extracting modules from known biological interaction networks such as those of E. coli (Gama-Castro et al., 2011) and S. cerevisiae (Kim et al., 2003) (the source networks). Our approach is based on the extraction of modules, that is, groups of genes that are more highly connected than expected in a random network (Marbach et al., 2009). We have shown that the topological modules extracted using our method correlate with functional modules of the source networks (Marbach et al., 2009). Hence, obtained network structures are meaningful targets for reverse engineering algorithms because in practice, one typically tries to infer the structure of a set of functionally related genes.

2.2 Dynamical model

Network topologies are endowed with detailed dynamical models of meaningful targets for reverse engineering algorithms because in practice, known biological interaction networks such as those of E. coli (van den Bulcke et al., 1982) allowing for both independent (“additive”) and synergistic (“multiplicative”) regulatory interactions. For each gene $i$ of a network, the rate of change of mRNA concentration $F_{RNA}^i$ and the rate of change of protein concentration $F_{Prot}^i$ are described by

$$\frac{d x_i}{dt} = m_i \cdot f_i(y) - \lambda_{RNA}^i \cdot x_i \quad (1)$$

$$\frac{d y_i}{dt} = r_i \cdot x_i - \lambda_{Prot}^i \cdot y_i \quad (2)$$

where $m_i$ is the maximum transcription rate, $r_i$ the translation rate, $\lambda_{RNA}^i$ and $\lambda_{Prot}^i$ are the mRNA and protein degradation rates, and $x$ and $y$ are vectors containing all mRNA and protein concentration levels, respectively. $f_i(\cdot)$ is the activation function of gene $i$, which 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 protein or transcription-factor (TF) concentrations $y$. A more detailed description of the activation function used is given by Marbach et al. (2010). Note that our approach conserves the nature of the gene interactions (enhancing or inhibitory) of the imported or extracted network structures.

The integration of the system of equations defined by (1) and (2) results in noiseless mRNA and protein concentration levels, respectively $x_i(t)$ and $y_i(t)$ for gene $i$. In living cells, molecular noise originates from thermal fluctuations and noisy processes such as transcription and translation (Becskei and Serrano, 2000). Hence, random fluctuations affect concentration levels of mRNA and protein, whose expression can be viewed as a stochastic process (Gardner and Collins, 2000). Both $F_{RNA}^i$ and $F_{Prot}^i$ are of the form

$$\frac{d X_i}{dt} = V(X_i) - D(X_i) \quad (3)$$

where $V(X_i)$ is the production and $D(X_i)$ the degradation term. The corresponding chemical Langevin equation (CLE) (Gillespie, 2000) we use to model molecular noise in transcription and translation processes is

$$\frac{d X_i}{dt} = V(X_i) - D(X_i) + c \left( \sqrt{V(X_i)} \eta_v + \sqrt{D(X_i)} \eta_d \right) \quad (4)$$

where $\eta_v$ and $\eta_d$ are independent Gaussian white-noise processes (Gillespie, 2000), $c$ is a multiplicative constant to control the amplitude of the molecular noise. For each gene $i$, we use the Stratonovich scheme and the Milstein method to integrate two equations of the form of 4, one describing the rate of change of mRNA concentration and one for the rate of change of protein concentration (Schaffter, 2010).

This model is derived from stochastic kinetics and the underlying assumptions are discussed by Gillespie (2000). Note that, according to this model, a gene that is not activated ($V(X_i)$ close to zero) has a very low level of noise (leakage) and it can not suddenly have a very high transcription rate due to noise. In contrast, a gene that is activated has a higher level of noise (which may be interpreted as transcriptional bursts, for instance).

The measurement noise depends on the technology used to monitor gene expression concentrations (Stolovitzky et al., 2005) and is modeled here independently of the molecular noise. GNW implements Gaussian and log-normal models of experimental noise as well as a model of noise observed in microarrays (Stolovitzky et al., 2005).

2.3 Synthetic expression datasets

The next step in generating in silico benchmark networks consists in simulating the generated in silico regulatory networks to produce synthetic gene expression datasets. Available experiments in GNW are

- **Wild type.** The steady-state levels of the wild type (the unperturbed network).
- **Knockout (null-mutant).** Steady-state levels of single-gene knockouts (deletions). An independent knockout is provided for every gene of the network. A knockout experiment is simulated by setting the transcription rate of this gene to zero.
- **Knockdowns (heterozygous).** Steady-state levels of single-gene knockdowns. A knockdown of every gene of the network is simulated. Knockdowns are obtained by reducing the transcription rate of the corresponding gene by half.
- **Dual knockouts.** Dual knockouts consist of simulating a network with two genes knocked out simultaneously.
- **Multifactorial.** Steady-state levels of variations of the network, which are obtained by applying multifactorial perturbations to the network. One may think of each experiment as a gene expression profile from a different patient, for example. We simulate multifactorial perturbations by slightly increasing or decreasing the basal activation of all genes of the network simultaneously by different random amounts.
Custom perturbations can also be specified. Experiments can be simulated as steady states and/or time series with user-defined duration and number of measurement points.

2.4 Evaluation of network inference methods

We not only provide researchers with a method for generating in silico gene network models to be used as benchmarks for reverse engineering algorithms, but also tools to facilitate the evaluation of network predictions. From a set of predictions from one or several inference methods, GNW automatically generates a comprehensive report including the result of a network motif analysis, where the performance of inference methods is profiled on local connectivity patterns. The network motif analysis often reveals systematic prediction errors, thereby indicating potential ways of network reconstruction improvements (Marbach et al., 2010). Furthermore, precision-recall (PR) and receiver operating characteristic (ROC) curves are evaluated for each network prediction (Prill et al., 2010). The relation between ROC and PR curves is discussed by Davis and Goadrich (2006).

3 RESULTS

We assessed the performance of six inference methods to illustrate benchmarking and performance profiling of network inference methods using GNW (Table 1). We first describe how to generate suitable network benchmark suites for the testing of various hypotheses. Specifically, we designed benchmark suites to show how the performance of inference methods is affected by different sizes and structural properties of regulatory networks. In addition, we show how GNW can help to identify the most informative type of gene expression data that a given inference method could use to achieve the best possible reconstruction from in vivo experiments. Finally, we introduce the DREAM4 Network Inference Challenge we generated, which has been used to assess the performance of many inference methods (Klamt et al., 2010; Menéndez et al., 2010).

Table 1. Gene network inference methods evaluated using GNW. ARACNE2 and CLR are two of the most widely used inference methods. The following methods have been best-performer or co-best-performer in at least one DREAM challenge: Yip et al. (DREAM3 In Silico Challenge Size 10, 50, and 100), Pinna et al. (DREAM4 In Silico Challenge Size 100), and Huynh-Thu et al. (DREAM4 In Silico Challenge multifactorial).

<table>
<thead>
<tr>
<th>Inference method</th>
<th>Approach</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARACNE2</td>
<td>mutual information (MI)</td>
<td>Margolin et al., 2006</td>
</tr>
<tr>
<td>CLR</td>
<td>mutual information (MI)</td>
<td>Faith et al., 2007</td>
</tr>
<tr>
<td>GENIE3</td>
<td>regression</td>
<td>Huynh-Thu et al., 2010</td>
</tr>
<tr>
<td>Z-score</td>
<td>statistical</td>
<td>Prill et al., 2010</td>
</tr>
<tr>
<td>Pinna et al.</td>
<td>statistical</td>
<td>Pinna et al., 2010</td>
</tr>
<tr>
<td>Yip et al.</td>
<td>noise model</td>
<td>Yip et al., 2010</td>
</tr>
</tbody>
</table>

3.1 Generation of network benchmark suites

We generated several network benchmark suites using the approach described in Methods. Each benchmark suite is composed of several in silico regulatory networks (the so-called gold standards or target networks). Fig. 2A shows one gold standard extracted from a regulatory network of the yeast S. cerevisiae (Kim et al., 2003). The extracted structures have been endowed with stochastic dynamical models of gene regulation accounting for molecular noise in transcription and translation processes.

![Generation and simulation of in silico gene network models using GNW](image)

The dynamical models of gene regulation have then been simulated to reproduce wild-type, knockout, knockdown, and multifactorial perturbation experiments. Fig. 2B illustrates the evolution of mRNA concentration levels without noise, when only molecular noise is introduced, and with both molecular and experimental noise. We generated the following benchmark suites:

- **Benchmark suite A.** 40 500-gene networks (20 from E. coli / 20 from yeast). Systematic knockout experiments were simulated to generate steady-state expression data.
- **Benchmark suite B.** 20 100-gene networks (10 from E. coli / 10 from yeast), 20 200-gene networks (10 from E. coli / 10 from yeast), and 20 500-gene networks (10 from E. coli / 10 from yeast). Systematic knockout experiments were simulated to generate steady-state expression data.
- **Benchmark suite C.** 20 100-gene networks (10 from E. coli / 10 from yeast). Systematic knockout and knockdown, and 100 multifactorial perturbation experiments were simulated to generate steady-state expression data.

At least half of the genes included in each gold standard are regulators, i.e. genes which regulate the mRNA production of at least one other gene. This is to avoid structures where there are many genes that do not regulate any other genes (out-degree = 0). We used the default parameter values proposed by GNW to simulate the gene expression experiments (see Supplementary Material).
3.2 Effect of network structural properties on inference method performance

The performance of network inference methods may strongly vary depending on the structural properties of the target networks. Fig. 3 shows systematic errors made by each inference method on four three-node motifs over-represented in the in vivo regulatory network structures of E. coli and yeast (Marbach et al., 2009), and therefore in the gold standard structures we generated.

![Fig. 3. Systematic errors made by network inference methods in predicting network motifs. GNW analyses thirteen configurations of three-node motifs, including fan-out, fan-in, cascade, and feed-forward loop (FFL) motifs, which are over-represented motifs in E. coli and yeast regulatory network. The first column displays the network motifs to infer and additional columns show the systematic errors made by each inference method when trying to infer the corresponding network motif.](image)

*Z*-score, Pinna et al., and Yip et al. have different error profiles than CLR, ARACNE2 (both based on mutual information), and GENIE3, which make systematically false positive errors between gene 2 and 3 in predicting fan-out motifs. Note that ARACNE2 seems to make less errors on that particular motif because the gene interactions present in the gold standards are in general less reliably identified than with CLR or GENIE3, independently of any network motifs considered. On the other hand, *Z*-score, Pinna et al., and Yip et al. are strongly affected by cascade motifs, where these methods systematically predict false positive interactions between gene 1 and gene 3.

We show that inference methods have changing performance when used to make predictions about the structure of regulatory networks having specific structural properties. Thus we evaluated the selected inference methods (Table 1) against the benchmark suite A described in Section 3.1. Fig. 4 shows the AUROC and AUPR values obtained by those methods when applied to infer E. coli and yeast network structures from knockout expression data.

![Fig. 4. Effect of structural properties of target networks on performance of inference methods. 20 benchmark networks containing 500 genes each have been generated for each condition using GNW (benchmark suite A, see Section 3.1). The inference methods have been applied to predict the directed structure of each benchmark network from knockout expression data and the corresponding AUROC and AUPR values have been evaluated. Methods strongly impeded by the cascade motif (*Z*-score, Pinna et al., and Yip et al.) as shown in Fig. 3 exhibit a performance degradation on yeast because yeast structure is composed of more cascade motifs than E. coli network structure.](image)

and because structures extracted from yeast contain more cascade motifs than in E. coli structures (data not shown). We observe a linear correlation between the number of cascade motifs to predict in a regulatory network and the AUROC and AUPR values obtained for *Z*-score, Pinna et al., and Yip et al. (Pearson’s correlation, \( r = -0.703, p < 0.05 \)). ARACNE2, CLR, and GENIE3 are less affected by the cascade motif (see Fig. 3).

Interestingly, Fig. 3 also shows that *Z*-score and Pinna et al. exhibit very similar error profiles. *Z*-score is one of the simplest inference methods (Prill et al., 2010), yet it has relatively high accuracy in predicting network structures from knockout steady states. Pinna et al. first performs a *Z*-score analysis followed by a refinement stage, which aims to suppress the errors made by *Z*-score on cascade motifs (Pinna et al., 2010). Fig. 3 doesn’t show any noticeable difference between *Z*-score and Pinna et al. This is confirmed by the fact that AUROC and AUPR values for *Z*-score and Pinna et al. are not significantly different (Mann-Whitney U-test, \( p > 0.05 \)).

3.3 Effect of network size on inference method performance

We are interested in showing how the performances of inference methods scale with the size of the regulatory networks to reconstruct. Using GNW, it is very simple to generate in silico benchmark network of size \( N < M \), where \( M \) is the size of the source network used (e.g. E. coli or yeast). Here we used the benchmark suite B described in Section 3.1, where each benchmark network has been simulated using the above methodology to produce knockout gene expression data. Fig. 5 shows the performance of the inference methods listed in Table 1 when applied to infer regulatory networks containing 100, 200, and 500 genes.

CLR has both AUROC and AUPR values significantly higher than those obtained by ARACNE2 for gold standards of size...
Fig. 5. Performance assessment of inference methods on GNW-generated in silico benchmark networks of size 100, 200, and 500 genes. 20 benchmark networks have been generated for each condition (benchmark suite B, see Section 3.1). The inference methods have been applied to predict the directed structures of benchmark networks from knockout expression data and the corresponding AUROC and AUPR values have been evaluated. We observed that the performance of inference methods decreases with the size of the regulatory networks to reconstruct.

Fig. 6. Identification of the most informative type of gene expression data required by a given inference method using in silico benchmark networks. Knockout (ko), knockdown (kd), and multifactorial (mf) perturbations were applied on 20 gold standards to generate three datasets containing each 100 measured steady states (benchmark suite C, see Section 3.1). Note. Z-score, Pinna et al., and Yip et al. are not applicable to the multifactorial data.

3.4 Design of in vivo gene expression experiments

A given inference method may require a very specific type of expression data in order to enable accurate network reconstruction. We show that in silico benchmark networks have also the ability to support the design of suitable in vivo gene expression experiments, which are typically time-consuming and expensive (Haynes and Brent, 2009). The benchmark suite C described in Section 3.1 is formed of 20 in silico networks consisting of 100 genes each, which we simulated using GNW to produce steady-state data for systematic knockout and knockdown, as well as 100 multifactorial perturbation experiments. Fig. 6 shows the AUROC and AUPR values obtained by the inference methods reviewed here (Table 1).

The most accurate network reconstructions are obtained using GENIE3, Z-score, and the methods developed by Pinna et al. and Yip et al. on knockout data. Knockout experiments are very informative because they provide network responses to individual and large perturbations (genes are “deleted”). Knockdown expression data, where the maximum transcription rate of genes is halved, are less informative than knockout data and thus lead to less accurate network reconstructions. Fig. 6 shows that ARACNE2 obtained AUROC and AUPR values comparable to CLR and GENIE3 when using multifactorial perturbation data. In addition, we considered providing knockout, knockdown, and multifactorial perturbation data together to ARACNE2, CLR, and GENIE3. We observed that AUROC and AUPR values obtained were slightly higher than when providing individually the three expression datasets (data not shown). We also added successively 100, 200, 300, and 400 additional multifactorial perturbations, however, the AUROC and AUPR values didn’t improve significantly for all methods (Mann-Whitney U-test, p < 0.05). Furthermore, it has been shown using GNW and time-series data that the inference accuracy of inference methods reaches a saturation point after a specific data size (Vijender et al., 2010). This reveals that simply adding more expression data does not necessarily imply performance improvement.

3.5 DREAM Network Inference Challenges

We have used GNW to generate the target networks for three international competitions on gene network reverse engineering: DREAM3 (2008), DREAM4 (2009), and DREAM5 (2010). Participants of the DREAM4 In Silico Challenge were asked to provide network predictions for two sub-challenges made of networks of size 10 and 100, respectively. Each sub-challenge was composed of five in silico gene networks (two extracted from E. coli and three from yeast), which have been simulated to produce steady-state wild-type, knockout, knockdown, and multifactorial perturbation experiments. In addition, time-series data have been made available.

For each sub-challenge, network predictions made by participating teams have been evaluated by computing P-values, which indicate the probability that random lists of genetic interaction predictions would be of the same or better quality (Prill et al., 2010). The
The most accurate reconstruction of the five gene networks of size 100 genes was achieved by Pinna et al. (2010). They participated to the DREAM4 In Silico Challenge 100 Challenge, in which their method was best-performer (OS = 71.589). Hence, both first bars in Fig. 7 correspond to the score of Pinna et al. We have shown in Fig. 3 that AUROC and AUPR values obtained by Pinna et al. are not significantly higher than those obtained using the original Z-score method. This can be explained by the fact that transitive causal effects are almost always weaker than the direct effects. We expect that if many amplifying cascades occur, the refinement stage introduced by Pinna et al. (2010) will enable more reliable network predictions as compared to Z-score alone.

It is also interesting to note that the method of Yip et al. has been best-performer on all DREAM3 In Silico Challenges of size 10, 50, and 100 genes we also provided. Yet it would have been ranked 7th on the DREAM4 size 100 challenge (OS = 57.079). While the original algorithm is composed of several batches using both steady-state and time-series data, Yip et al. only used the first batch to build a noise model from knockout steady-state data (Yip et al., 2010). The achievement of the 7th rank in DREAM4 can be partially explained by the fact that Yip et al. made a strong and correct assumption on the Gaussian measurement noise we used in DREAM3, which is no longer valid in DREAM4. Indeed, we modeled molecular noise in addition to a model of experimental noise observed in microarrays (Stolovitzky et al., 2005).

4 DISCUSSION

We propose a comprehensive and powerful framework for in silico benchmark generation and performance profiling of network inference methods. We implemented this framework as an open-source tool called GeneNetWeaver (GNW). Biologically plausible network structures are generated by extracting modules from known biological interaction networks such as those of E. coli and the yeast S. cerevisiae. Network structures are then endowed with detailed dynamical models of gene regulation describing both transcription and translation processes. Transcriptional regulation is modeled using a thermodynamic approach accounting for both independent ("additive") and synergistic ("multiplicative") interactions. In addition, our models account for stochastic molecular noise as well as experimental noise observed in microarrays. The generated in silico benchmark networks can be simulated in GNW to reproduce wild-type, knockout (null-mutant), knockdown (heterozygous), and multifactorial perturbation gene expression experiments. As an example of the application, we have used GNW to generate the target networks for three international competitions on gene network reverse engineering: DREAM3 (2008), DREAM4 (2009), and DREAM5 (2010). In total, 91 teams have submitted over 900 network predictions on GNW-generated networks, making GNW one of the most widely used benchmark generators by the community.

In contrast to previously proposed benchmark generators, GNW also integrates tools for systematic evaluation of the predictions from inference methods on benchmark networks. A unique feature of GNW is the ability to perform a network motif analysis from a set of network predictions and their corresponding benchmark networks. The network motif analysis reveals systematic prediction errors made by inference method on specific network motifs, thereby indicating potential ways of network reconstruction improvements. The accuracy of network inference is assessed using standard metrics such as precision-recall and receiver operating characteristic (ROC) curves.

We have used GNW to generate in silico benchmark suites to assess the performance and identify the strengths and weaknesses of six network inference methods. We show that Z-score, and the inference methods developed by Pinna et al. and Yip et al. make more accurate network predictions than two widely used methods, ARACNE2 and CLR. This good performance is achieved apparently because those methods target the inference of causal relationships between genes. However, ARACNE2 and CLR do not require systematic knockout gene expression data, which are not always available in practice, to infer undirected networks. Yet ARACNE2, CLR, and GENIE3 methods can be applied to infer regulatory networks even if no systematic knockout or knockdown experiments are provided. Furthermore, our results show that at some point simply giving more expression data to inference methods does not necessarily imply performance improvement. Therefore, the integration of additional information about the target regulatory networks should be considered, for instance using prior knowledge about the network structures.
The novelty of GNW is that it additionally provides a unique network motif analysis, which we used to show that the structural properties of the target regulatory networks affect the performance of inference methods. We observed that the performances of Z-score, and the methods developed by Pinna et al. and Yip et al. are impeded by the presence of cascade motifs in the target networks. Thus, we show that those methods make significantly less accurate network predictions on the yeast S. cerevisiae, whose structure includes more cascade motifs than E. coli transcriptional network structure. Finally, we also provide evidence that in silico benchmark networks can be used to identify the most informative type of gene expression data that a given inference method could use to achieve the best possible reconstruction from in vivo experiments.

ACKNOWLEDGEMENT
The authors would like to express their thanks to Gilles Roulet for his collaboration in software development, and Steffen Wischmann, Peter Dürr, and Pradeep Fernando for their careful reading and suggestions on the manuscript.

Funding: The work was supported by the SystemsX.ch initiative (WingX project), the Swiss National Science Foundation (grant numbers 200021-112060 and 200021-127143), and a postdoctoral fellowship for D.M.

Conflict of Interest: none declared.

REFERENCES

