JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 16, Number 2, 2009
© Mary Ann Liebert, Inc.
Pp. 229–239
DOI: 10.1089/cmb.2008.09TT

Generating Realistic *In Silico* Gene Networks for Performance Assessment of Reverse Engineering Methods

DANIEL MARBACH, THOMAS SCHAFFTER, CLAUDIO MATTIUSSI,
and DARIO FLOREANO

ABSTRACT

Reverse engineering methods are typically first tested on simulated data from *in silico* networks, for systematic and efficient performance assessment, before an application to real biological networks. In this paper, we present a method for generating biologically plausible *in silico* networks, which allow realistic performance assessment of network inference algorithms. Instead of using random graph models, which are known to only partly capture the structural properties of biological networks, we generate network structures by extracting modules from known biological interaction networks. Using the yeast transcriptional regulatory network as a test case, we show that extracted modules have a biologically plausible connectivity because they preserve functional and structural properties of the original network. Our method was selected to generate the "gold standard" networks for the gene network reverse engineering challenge of the third DREAM conference (Dialogue on Reverse Engineering Assessment and Methods 2008, Cambridge, MA).

Key words: DREAM challenge, gene regulatory networks, modularity, network motifs, reverse engineering.

1. INTRODUCTION

Reverse engineering algorithms hold the promise to unravel unknown gene regulatory networks from high-throughput data in an automated manner. Advances in experimental technologies have spurred the development of a wide range of methods for this purpose: for example, statistical methods (Anastassiou, 2007; Basso et al., 2005), Bayesian networks (Ferrazzi et al., 2007; Friedman, 2004; Mukherjee and Speed, 2008), and methods based on dynamical models (Brazhnik, 2005; De la Fuente et al., 2002; Gardner et al., 2003; Kholodenko et al., 2002; Marbach et al., 2009; Perkins et al., 2006). However, the strengths, weaknesses, and relative performance of different methods remain poorly understood (Stolovitzky et al., 2007).

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

This can be explained by an inherent difficulty in evaluating the performance of gene network inference methods, because it is seldom possible to systematically validate the predictions of unknown interactions *in vivo* (Fig. 1A). Consequently, simulated data from *in silico* gene networks is often the only possibility for systematic performance assessment (Fig. 1B). In simulation, all aspects of the networks and experiments are under full control. This allows characterization of reverse engineering methods for different types of data and levels of noise. In addition to performance assessment, *in silico* studies are of great relevance for optimal experimental design for subsequent real biological applications (Tegner et al., 2003).

However, results are only meaningful if the *in silico* benchmarks are biologically plausible. Creating such benchmarks involves (1) generating realistic gene network structures and (2) generating realistic data from these networks using adequate dynamical models. In this paper, we consider the first problem, that is, how to generate network topologies with the same structural properties as real gene networks.

Apart from manual design of small benchmark networks (Kremling et al., 2004; Tegner et al., 2003; Zak et al., 2003), Erdös–Rényi and scale-free (Albert–Barabási) random graph models are currently the predominant approaches for generating *in silico* gene network structures (Mendes et al., 2003; Wildenhain and Crampin, 2006). However, random graphs capture only few of the structural properties of real biological gene networks (Van den Bulcke et al., 2006). For example, scale-free random graphs approximate the power-law degree distribution of biological gene regulatory networks, but do not model other important properties such as modularity (Ravasz et al., 2002) or the occurence of network motifs, which are statistically overrepresented circuit elements (Shen-Orr et al., 2002). Instead of constructing more complex random graph models, which would be difficult to justify and might wrongly favor some reverse engineering algorithms over others, we believe that the fairest way to compare reverse engineering methods is based on real biological network structures. Nowadays, rough drafts of the complete gene regulatory network of model organisms are available in dedicated databases (we call such networks "global interaction networks"). Global interaction networks can be used as "templates" for generating realistic network structures. Rice et al. (2005) have generated a single *in silico* network from the structure of the global *E. coli* transcription network. In order to generate multiple networks, which is necessary for collecting statistics on the performance of reverse engineering methods, Van den Bulcke et al. (2006) extract random subnets from global interaction networks.

We argue that global interaction networks should be sampled in a biologically meaningful way for generating plausible benchmarks. The approach introduced in this article is based on the extraction of modules, that is, groups of genes that are more highly connected than expected in a random network. We show that topological modules extracted with the method described here correlate with functional modules of the global interaction network. Thus, the obtained network structures are realistic targets for reverse engineering, given that in a real application, one typically tries to reverse engineer the topology of a set of functionally related (and not randomly selected) genes.

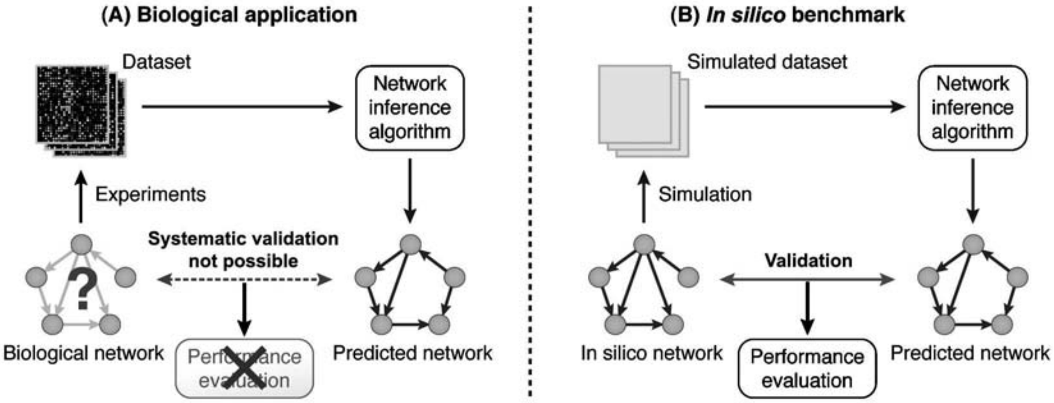

FIG. 1. Validation strategies for network inference methods. **(A)** The "true" network structure of biological gene networks is in general unknown or only partly known, which hinders systematic performance evaluation. **(B)** Since the structures of *in silico* networks are known, predictions can be validated.

2. RESULTS

2.1. Module extraction from global interaction networks

We have devised a method to generate *in silico* network structures by extracting modules from a given global interaction network (the so-called source network). Module extraction starts from a seed node that is selected randomly among the nodes of the source network. From this seed, a subnetwork is grown by iteratively adding nodes to it until a desired size is reached. At each step, from all neighbors of the subnetwork, we select the node that leads to the highest modularity Q, where

$$\begin{aligned} Q = & \text{(number of edges within the subnetwork)} \\ & -\text{(expected number of such edges in a randomized graph)}. \end{aligned} \qquad (1)$$

The method outlined above, and described in detail in Methods, can be applied repeatedly to extract different subnetworks of a desired size M from a source network of size $N \gg M$ by selecting different seed nodes.

Before applying the module extraction method to real biological networks, we demonstrate it on the hierarchical scale-free network model of Ravasz et al. (2002), which has a scale-free topology with embedded modularity similar to many biological networks. The network consists of a repeated four-node-motif, which is hierarchically grouped into clusters. As shown in Figure 2, the module extraction method tends to first add nodes from the four-node-motif of the seed, then it expands to other four-node-motifs of the same cluster, and only if the desired size has not yet been reached it will start to include nodes from another cluster.

The example in Figure 2 illustrates how subnetworks obtained with module extraction are more representative samples of the source network structure than randomly extracted subnetworks (random subnetwork extraction is described in Methods). Similar to the source network, the subnetworks resulting from

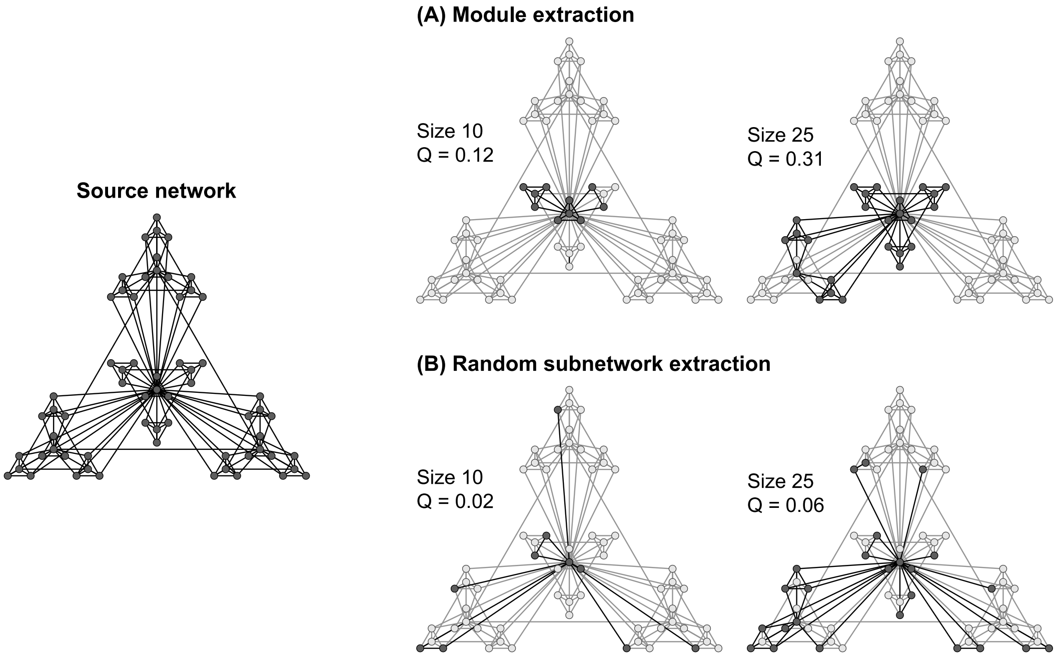

FIG. 2. Module extraction versus random sampling of a hierarchical scale-free network. Starting from the central hub (the seed node), subnetworks of size 10 and 25 are extracted. Module extraction leads to subnetworks of high modularity Q. Most importantly, these subnetworks have the same functional building blocks and structural properties as the source network. This is not the case for randomly sampled subnetworks.

module extraction are built from four-node-motifs that are organized in a hierarchical modular structure. In contrast, the randomly sampled subnetworks do not share these structural properties with the source network.

It has been suggested that network motifs correspond to functional units of biological networks (Alon, 2007). Assuming that the four-node-motifs in the artificial source network considered here also correspond to functional units (organized hierarchically into higher-level functional modules), the method described above allows the extraction of different combinations of such functional units and modules. In the following sections, we show that module extraction indeed permits sampling of functional units and modules in real gene regulatory networks.

2.2. *Topological modules correlate with functional modules in the yeast gene network*

As a test source network, we used the transcriptional regulatory network of the yeast *Saccharomyces cerevisiae* as described by Balaji et al. (2006). With 4441 genes and 12,873 interactions, this network is one of the best global approximations of a eukaryotic transcriptional regulatory network so far.

We used module extraction and random subnetwork extraction to generate subnetworks of sizes 25, 100, and 400 from the yeast gene network. For each size, 20 subnetworks were generated starting from different randomly chosen seed nodes. We confirmed that the subnetworks obtained with the module extraction method described here indeed correspond to topological modules (densely interconnected subnetworks) of the gene network, as their modularity Q is positive and significantly higher than for randomly extracted subnetworks (Fig. 3). For small networks of size 25, the difference of the mean modularity Q compared to randomly extracted subnetworks is smaller than for large subnetworks, but it is still statistically significant (p-value $< 10^{-6}$ using the Wilcoxon-Mann-Whitney rank-sum test).

To check whether these topological modules also correspond to functional modules, that is, groups of genes that have a related function, we considered the Gene Ontology (GO) functional annotation of the genes in the Saccharomyces Genome Database (Hong et al., 2008). For a given subset of genes (a subnetwork), we identified all GO terms that are statistically overrepresented, i.e., that occur more frequently than expected compared to their background frequency in the complete set of all genes.

With module extraction, the total number of overrepresented GO terms is 111, 282, and 444 for the 20 subnetworks of sizes 25, 100, and 400, respectively. For random subnetwork extraction, the total number of terms is only 10, 9, and 112 for the same three network sizes. As can be seen in Figure 4 (left column) from the distribution of the p-values of these GO terms, module extraction leads not only to a higher number of overrepresented GO terms, but they also tend to be more statistically significant.

In the previous paragraph, we looked at the *total* number of overrepresented GO terms of the 20 subnetworks for the three sizes. Let's now consider the number of such terms *per subnetwork*. The median number of terms per subnetwork is significantly higher for module extraction than for random subnetwork extraction (p-value $< 10^{-3}$ using the Wilcoxon-Mann-Whitney rank-sum test), except for subnetworks of size 25, where both medians are zero (Fig. 4, right column). Remember, that for small subnetworks of size 25, the modularities Q obtained with module extraction are not very high and only slightly superior to those of random subnetworks (Fig. 3). Thus, it is not surprising that for this network size there are also fewer overrepresented GO terms than for larger subnetworks.

In summary, our results demonstrate that module extraction is a more biologically meaningful way of sampling gene networks than random subnetwork extraction (especially for medium and large subnetworks). Moreover, they confirm the hypothesis that topological modules correspond to functional entities of gene regulatory networks, as treated in more detail in the Discussion. As an example, Figure 5 shows the structure and function of two extracted modules.

2.3. *Module extraction preserves structural properties of the yeast gene network*

After having studied the functionality of extracted subnetworks, we turned our attention to their structural properties. First, we considered the degree distributions.[1] We found that both extracted modules and

[1] The degree distribution $P(k)$ is defined as the fraction of genes that have k connections (degree k).

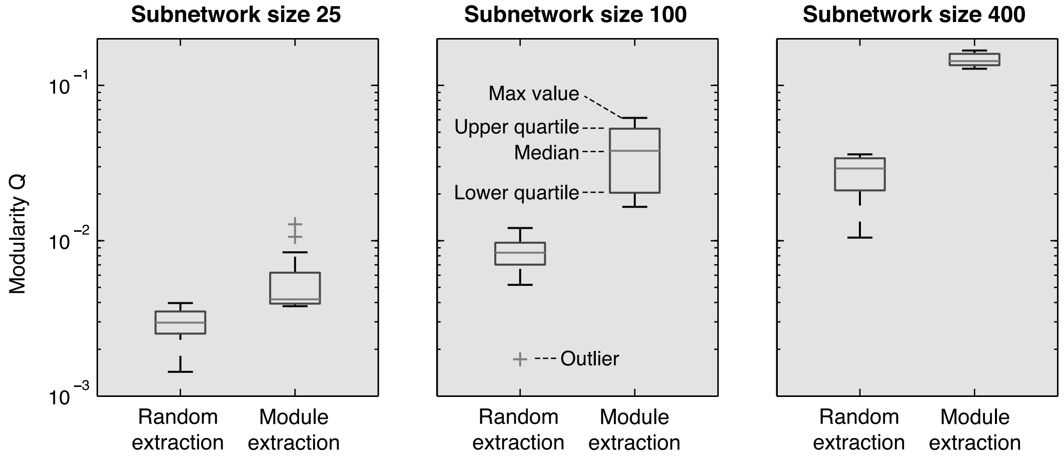

FIG. 3. Distribution of Q-values for subnetworks obtained with module extraction and random sampling from the yeast gene regulatory network. As expected, module extraction leads to subnetworks with higher modularity Q.

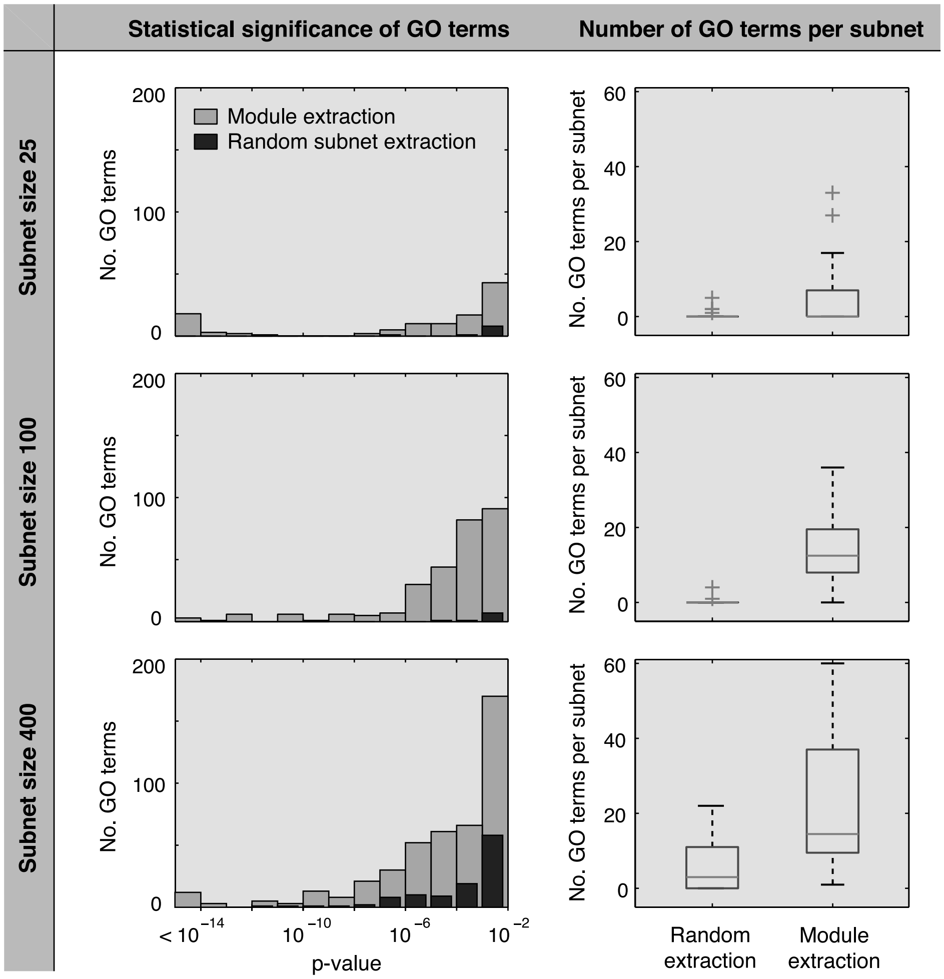

FIG. 4. Statistically overrepresented functional annotations (GO terms) in subnetworks obtained with module extraction and random subnetwork extraction from the yeast gene network. **(Left)** Histograms with the total number of GO terms of 20 subnets of sizes 25, 100, and 400. Module extraction leads to a higher number of overrepresented GO terms, and they tend to be more statistically significant (lower p-values). **(Right)** Boxplots (see Fig. 3 for legend) of the number of overrepresented GO terms per subnetwork. The median number of overrepresented terms is significantly higher with module extraction, except for small subnetworks of size 25.

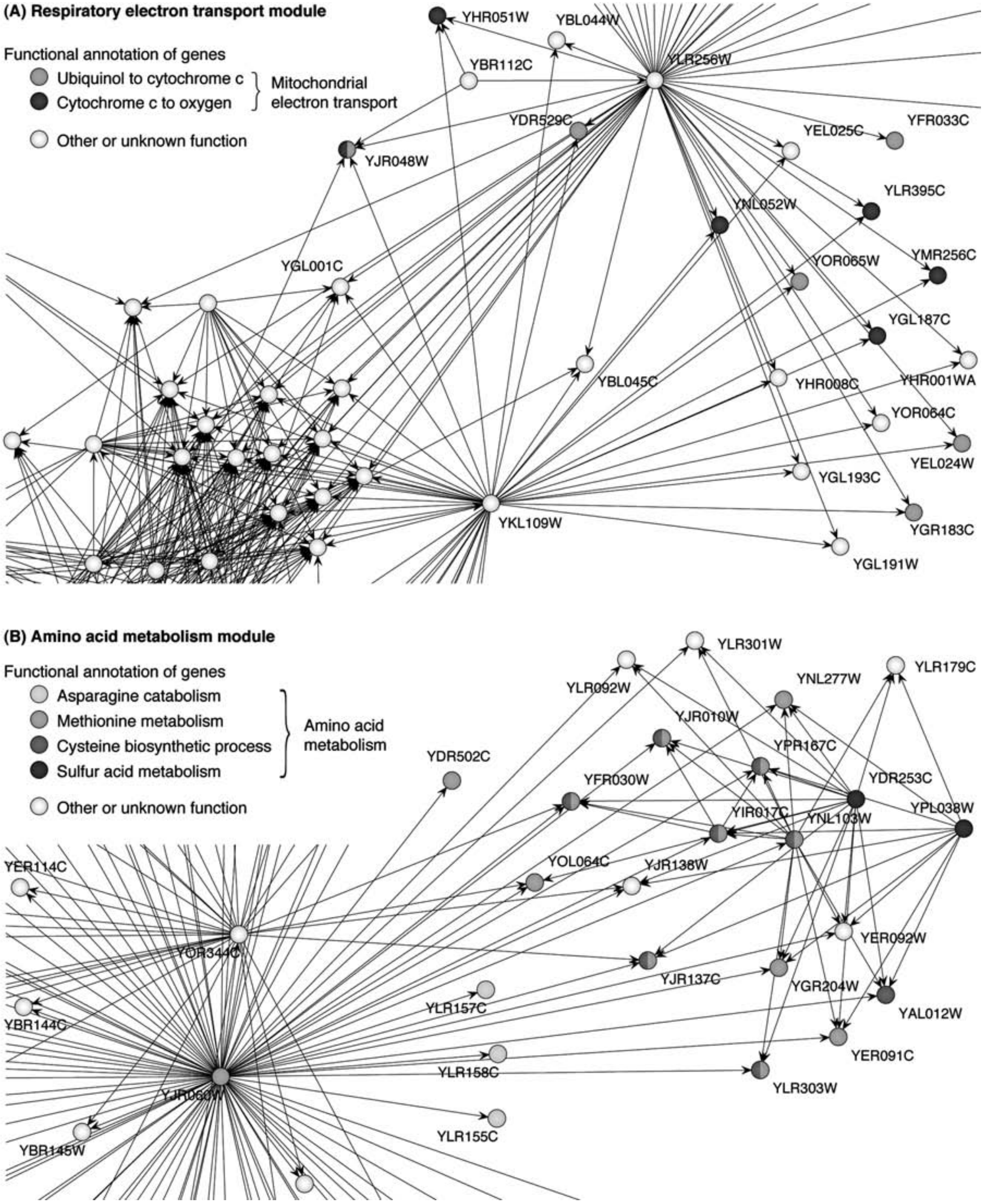

FIG. 5. Two example subnetworks of size 100 obtained with module extraction from the yeast gene network. We have zoomed in on two groups of functionally related genes (shaded circles), illustrating two different types of modules. The functional annotation of genes is taken from the Saccharomyces Genome Database (Hong et al., 2008), and p-values are calculated as described in Methods. **(A)** In the first subnetwork, genes related to mitochondrial electron transport are statistically overrepresented with a p-value $< 10^{-10}$. The structure of this module can be described as a set of co-regulated genes. **(B)** The module of the second subnetwork is a more complex regulatory circuit related to amino acid metabolism (p-value $< 10^{-9}$). Note that in both subnetworks there are additional functional modules, which are not shown here.

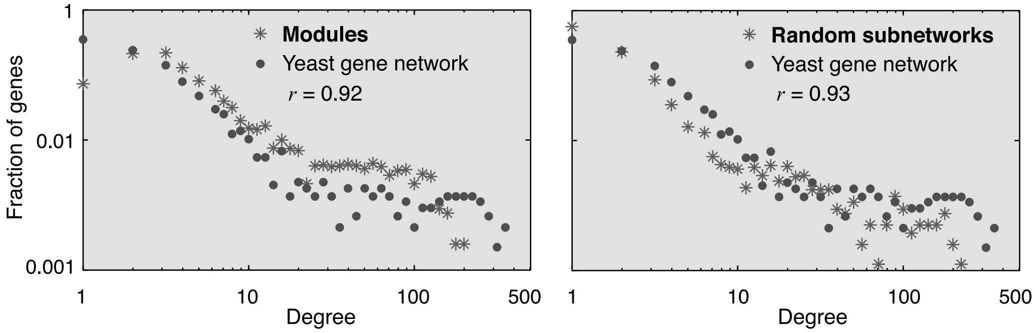

FIG. 6. Degree distribution of the complete yeast gene network compared to the mean degree distribution of 20 subnetworks of size 400 obtained with module extraction and random subnetwork extraction. Both strategies lead to subnetworks with similar degree distributions as the complete network (r is the Pearson correlation).

randomly extracted subnetworks have a very similar degree distribution as the complete yeast gene network: the Pearson correlation between the degree distribution of the complete network and the mean degree distribution of the 20 subnetworks of size 400 is 0.92 for module extraction, and it is 0.93 for random subnetwork extraction (Fig. 6). Thus, both strategies yield network structures with a biologically plausible degree distribution.

In the previous section, we have shown that subnetworks obtained with module extraction correlate with functional modules, whereas randomly extracted subnetworks do not. It has been hypothesized that network motifs (statistically overrepresented sub-circuits) are functional building blocks of gene networks (Alon, 2007). Thus, we would expect to find these motifs in the subnetworks obtained with module extraction, and not in the randomly sampled subnetworks.

To verify this, we compared the subnetworks based on their triad significance profile (Milo et al., 2004), which indicates for each three-node-motif the degree to which it is statistically over or underrepresented (see Methods). As shown in Figure 7, the significance profile of the complete gene network is indeed very similar to the mean significance profile of extracted modules (it falls within the range of one standard deviation). In contrast, randomly extracted subnetworks have a completely different profile.

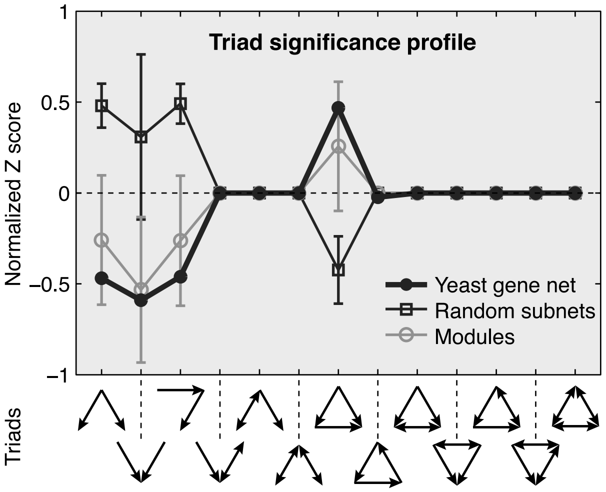

FIG. 7. The significance profile (SP) of the complete yeast gene network compared to the mean SP of 20 subnetworks of size 400 obtained with module extraction and random subnetwork extraction. The error bars indicate the standard deviation. The SP of the complete network falls within the range of one standard deviation from the mean SP of modules. In contrast, the SP of randomly extracted subnetworks is opposite to the SP of the complete network.

In summary, we have confirmed that subnetworks obtained with module extraction have a biologically plausible connectivity because they preserve functional and structural properties of gene networks such as degree distribution and network motifs. Incidentally, our finding that functional modules preserve network motifs, whereas non-functional subnetworks (random subnetworks) do not, supports the hypothesis that network motifs are functional building blocks of modules in gene regulatory networks.

3. DISCUSSION

We have presented a method that permits the generation of biologically meaningful network structures for performance assessment of reverse engineering methods. The method is based on the extraction of topological modules from global interaction networks. In this paper, we have focused on gene regulatory networks, but the same approach could be used for other types of cellular networks.

Using the yeast transcriptional regulatory network as a test case, we have shown that topological modules extracted with the method described here have a high number of statistically overrepresented functional annotations, indicating that they correlate with functional modules of this network. Furthermore, extracted modules preserve structural properties of the original network such as degree distribution and network motifs. We conclude that subnetwork extraction is a biologically meaningful way of sampling gene networks both from a functional and structural point of view.

The approach described in this article was originally motivated by the hypothesis that topological modules in gene networks coincide with functional modules. Such a separation of functions into more or less structurally isolated modules is thought to favor network evolvability and robustness (Hartwell et al., 1999). Indeed, it has been shown that topological modules in the *Escherichia coli* transcriptional regulatory network can be assigned specific functions (Ma et al., 2004; Resendis-Antonio et al., 2005). Our results from the yeast network indicate that this may also be true for eukaryotic transcriptional regulatory networks, which have an increased number of interconnections and cannot be as clearly decomposed into distinct topological modules.

An example application of module extraction is the generation of realistic gold standard networks for the DREAM challenges (the international cellular network reverse engineering competition) (Stolovitzky et al., 2007). For the DREAM2 In-Silico-Network Challenges (New York, 2007), Mendes et al. (2003) provided one gold standard network with a random Erdös–Rényi structure and one with a random scale-free structure. The goal of the challenge was to infer the structure of these networks (which was not disclosed to the participants) from simulated gene expression data. Interestingly, the winning method of this challenge (Gardner et al., 2003) had a much better performance on the Erdös–Rényi network than on the scale-free network. Indeed, the performance of reverse engineering methods can be very sensitive to the type of network structure to which they are applied (Mukherjee and Speed, 2008; Rice et al., 2005; Wildenhain and Crampin, 2006). Thus, for a fair comparison of methods in the DREAM challenge, it is crucial that the gold standard networks have a realistic structure.

Our method has been selected for the In-Silico-Network Challenges of DREAM3 (Cambridge, MA, 2008). We provide three challenges corresponding to networks of size 10, 50, and 100. For each size, we generated two gold standard networks using an *E. coli* transcriptional regulatory network (Shen-Orr et al., 2002) as source, and three gold standard networks from a yeast genetic interaction network (Reguly et al., 2006). The three challenges had 29, 27, and 22 participants each, making them by far the most widely used gene network reverse engineering benchmarks so far. A detailed description of the challenges will be reported elsewhere. Additional information is available on the websites of DREAM3 and our institution.[2]

In this paper, we have focused on the generation of realistic gene network structures. The design of dynamical models and initialization of numerical parameters to obtain biologically plausible network dynamics are equally important challenges (Nykter et al., 2006; Roy et al., 2008). We have developed a framework for this purpose (manuscript in preparation), but module extraction can be employed with any dynamical modeling approach of choice. Note that the structure of biological networks often confers

[2] DREAM website: *http://wiki.c2b2.columbia.edu/dream*. Our website: *http://lis.epfl.ch/grn*

robustness towards perturbations and variations in kinetic parameters (von Dassow et al., 2000). Thus, network structures obtained with module extraction may actually facilitate the initialization of models with biologically plausible network dynamics and enable the design of truly realistic *in silico* reverse engineering benchmarks.

4. METHODS

4.1. Subnetwork extraction

The module extraction method grows a subnetwork of desired size M starting from a seed node, which can be selected randomly or manually from the source network. The procedure starts with a subnetwork containing only the seed node. Additional nodes are added iteratively until the subnetwork reaches the desired size.

Nodes are selected for addition as follows. First, the set of all neighbors of the subnetwork is constructed (a neighbor is a node of the source network that is connected by a direct link to at least one node of the subnetwork). Second, we compute for each neighbor the modularity Q of the subnetwork after adding only this neighbor to the subnetwork. Finally, we select the neighbor that obtained the highest modularity Q for addition to the subnetwork (if several neighbors obtained the same modularity, we randomly choose one of them).

The modularity Q is defined as the number of edges within the subnetwork minus the expected number of such edges in a randomized network with the same degree sequence (Newman, 2006)

$$Q = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s}, \tag{2}$$

where m is the total number of edges in the network, $\mathbf{s}$ is the index vector defining the subnetwork ($s_i = 1$ if node i is part of the subnetwork, $s_i = -1$ if not), and $\mathbf{B}$ is the so-called *modularity matrix* with elements $B_{ij} = A_{ij} - P_{ij}$. A_{ij} is the actual number of edges between node i and j and $P_{ij} = k_i k_j / 2m$ is the expected number of edges in a randomized network (k_i being the degree of node i).

There exist methods to find a globally optimal decomposition of a complex network into a set of modules (Newman, 2006). Here, our goal is not the identification of optimal modules, but the extraction of *diverse* subnetworks of *prespecified size* and reasonably high (not necessarily globally optimal) modularity Q. Classical modularity detection algorithms are not well suited for this purpose.

Note that neighboring seeds may converge to identical or very similar (overlapping) subnetworks. The diversity of subnetworks can be increased by adding some randomness to the module extraction: instead of always selecting the neighbor that leads to the highest modularity Q, one can randomly select among the top k percent of all neighbors. For $k = 100\%$, this amounts to the random neighbor addition strategy used by Van den Bulcke et al. (2006). Varying k between 0% and 100% allows for tuning of the sampling strategy from pure module extraction to random subnetwork extraction.

Apart from the case of neighboring seeds mentioned in the previous paragraph, module extraction from different random seed nodes typically leads to very diverse subnetworks. In principle, every extracted subnetwork may be considered a realistic network structure for a reverse engineering benchmark because they all correspond to modules of a real biological network. In practice, one may have additional criteria and discard certain types of modules. For example, for the gold standard networks of DREAM3 we did not include subnetworks that only contained a global regulator and its direct targets, because this is not a very challenging network structure for a reverse engineering benchmark.

4.2. Identification of statistically overrepresented functional annotations

We identify statistically overrepresented functional annotations in a subset of genes (a subnetwork) using the GO::TermFinder tool (Boyle et al., 2004).

GO::TermFinder calculates p-values to determine whether any GO term occurs more frequently in the subset of genes than expected by chance. The p-value of a term corresponds to the probability of obtaining an equal or greater frequency of this term when randomly selecting genes from the background set of genes (in our case, the background set is the set of all genes of the network). Bonferroni correction is used for multiple hypothesis testing.

4.3. Network motif significance profiles

The triad significance profile (SP) indicates for each type of three-node-subgraph whether it is over or underrepresented in a given network. The statistical significance of triad i is measured by its Z score

$$Z_i = \frac{N\text{real}_i - \langle N\text{rand}_i \rangle}{\text{std}(N\text{rand}_i)}, \tag{3}$$

where $N\text{real}_i$ is the number of times the triad occurs in the network. $\langle N\text{rand}_i \rangle$ and $\text{std}(N\text{rand}_i)$ are the mean and standard deviation of its occurrences in an ensemble of randomized networks with the same degree sequence. The SP corresponds to the normalized Z vector. For details, refer to Milo et al. (2004).

ACKNOWLEDGMENTS

We thank Gustavo Stolovitzky, Sabine Hauert, Sara Mitri, Peter Dürr, and Fred Marbach for helpful discussions. This work was supported by the Swiss National Science Foundation (grant 200021–112060), and the Swiss SystemsX.ch initiative (WingX project, evaluated by the Swiss National Science Foundation).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Alon, U. 2007. Network motifs: theory and experimental approaches. *Nat. Rev. Genet.* 8, 450–461.

Anastassiou, D. 2007. Computational analysis of the synergy among multiple interacting genes. *Mol. Syst. Biol.* 3, 83.

Balaji, S., Babu, M.M., Iyer, L.M., et al. 2006. Comprehensive analysis of combinatorial regulation using the transcriptional regulatory network of yeast. *J. Mol. Biol.* 360, 213–227.

Basso, K., Margolin, A.A., Stolovitzky, G., et al. 2005. Reverse engineering of regulatory networks in human B cells. *Nat. Genet.* 37, 382–390.

Boyle, E.I., Weng, S., Gollub, J., et al. 2004. GO::termfinder—open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics* 20, 3710–3715.

Brazhnik, P. 2005. Inferring gene networks from steady-state response to single-gene perturbations. *J. Theor. Biol.* 237, 427–440.

De la Fuente, A., Brazhnik, P., and Mendes, P. 2002. Linking the genes: inferring quantitative gene networks from microarray data. *Trends Genet.* 18, 395–398.

Ferrazzi, F., Sebastiani, P., Ramoni, M., et al. 2007. Bayesian approaches to reverse engineer cellular systems: a simulation study on nonlinear Gaussian networks. *BMC Bioinform.* 8, S2.

Friedman, N. 2004. Inferring cellular networks using probabilistic graphical models. *Science* 303, 799–805.

Gardner, T.S., di Bernardo, D., Lorenz, D., et al. 2003. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 301, 102–105.

Hartwell, L.H., Hopfield, J.J., Leibler, S., et al. 1999. From molecular to modular cell biology. *Nature* 402, C47–C52.

Hong, E.L., Balakrishnan, R., Dong, Q., et al. 2008. Gene ontology annotations at SGD: new data sources and annotation methods. *Nucleic Acids Res.* 36, D577–D581.

Kholodenko, B.N., Kiyatkin, A., Bruggeman, F.J., et al. 2002. Untangling the wires: a strategy to trace functional interactions in signaling and gene networks. *Proc. Natl. Acad. Sci. USA* 99, 12841–12846.

Kremling, A., Fischer, S., Gadkar, K., et al. 2004. A benchmark for methods in reverse engineering and model discrimination: problem formulation and solutions. *Genome Res.* 14, 1773–1785.

Ma, H., Buer, J., and Zeng, A. 2004. Hierarchical structure and modules in the *Escherichia coli* transcriptional regulatory network revealed by a new top-down approach. *BMC Bioinform.* 5, 199.

Marbach, D., Mattiussi, C., and Floreano, D. 2009. Replaying the evolutionary tape: biomimetic reverse engineering of gene networks. *Ann. NY Acad. Sci.* (to appear).

Mendes, P., Sha, W., and Ye, K. 2003. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics* 19, Suppl 2, ii122–ii129.

Milo, R., Itzkovitz, S., Kashtan, N., et al. 2004. Superfamilies of evolved and designed networks. *Science* 303, 1538–1542.

Mukherjee, S., and Speed, T.P. 2008. Network inference using informative priors. *Proc. Natl. Acad. Sci. USA* 105, 14313–14318.

Newman, M.E.J. 2006. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74 (3 Pt 2): 036104.

Nykter, M., Aho, T., Andesmäk, M., et al. 2006. Simulation of microarray data with realistic characteristics. *BMC Bioinform.* 7, 349.

Perkins, T.J., Jaeger, J., Reinitz, J., et al. 2006. Reverse engineering the gap gene network of *Drosophila melanogaster*. *PLoS Comput. Biol.* 2, e51.

Ravasz, E., Somera, A.L., Mongru, D.A., et al. 2002. Hierarchical organization of modularity in metabolic networks. *Science* 297, 1551–1555.

Reguly, T., Breitkreutz, A., Boucher, L., et al. 2006. Comprehensive curation and analysis of global interaction networks in *Saccharomyces cerevisiae*. *J. Biol.* 5, 11.

Resendis-Antonio, O., Freyre-González, J.A., Menchaca-Méndez, R., et al. 2005. Modular analysis of the transcriptional regulatory network of *E. coli*. *Trends Genet.* 21, 16–20.

Rice, J.J., Tu, Y., and Stolovitzky, G. 2005. Reconstructing biological networks using conditional correlation analysis. *Bioinformatics* 21, 765–773.

Roy, S., Werner-Washburne, M., and Lane, T. 2008. A system for generating transcription regulatory networks with combinatorial control of transcription. *Bioinformatics* 24, 1318–1320.

Shen-Orr, S.S., Milo, R., Mangan, S., et al. 2002. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.* 31, 64–68.

Stolovitzky, G., Monroe, D., and Califano, A. 2007. Dialogue on reverse-engineering assessment and methods: the dream of high-throughput pathway inference. *Ann. NY Acad. Sci.* 1115, 1–22.

Tegner, J., Yeung, M.K.S., Hasty, J., et al. 2003. Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling. *Proc. Natl. Acad. Sci. USA* 100, 5944–5949.

Van den Bulcke, T., van Leemput, K., Naudts, B., et al. 2006. SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinform.* 7, 43.

Von Dassow, G., Meir, E., Munro, E.M., et al. 2000. The segment polarity network is a robust developmental module. *Nature* 406, 188–192.

Wildenhain, J., and Crampin, E.J. 2006. Reconstructing gene regulatory networks: from random to scale-free connectivity. *Syst. Biol.* 153, 247–256.

Zak, D.E., Gonye, G.E., Schwaber, J.S., et al. 2003. Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: insights from an identifiability analysis of an in silico network. *Genome Res.* 13, 2396–2405.

Address reprint requests to:
Prof. Dario Floreano
Laboratory of Intelligent Systems
EPFL–I2S–LIS
Station 11
CH–1015 Lausanne, Switzerland

E-mail: Dario.Floreano@epfl.ch