BIOINFORMATICS

Vol. 00 no. 00 2011
Pages 1–3

Supplementary Material: GeneNetWeaver: *In silico* benchmark generation and performance profiling of network inference methods

Thomas Schaffter [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

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

In this supplement, we describe how to generate *in silico* gene regulatory networks and profile the performance of network inference methods using GeneNetWeaver (GNW) 3.1 Beta. For a more detailed description, please refer to the available user manual. Also, the section *Tutorial* and *Help* accessible in GNW aim to provide clear and quick insight into the different functionalities offered by GNW.

Furthermore, we briefly give a description of the network inference methods used and we provide the parameter values used for each inference methods we applied to reconstruct the *in silico* benchmark networks we generated using GNW.

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

1 GENENETWEAVER

1.1 Topology

We generated network structure by extracting modules from the biological interactions networks of *E. coli* (Gama-Castro *et al.*, 2011) and *S. cerevisiae* (Kim *et al.*, 2003) (the *source networks*). The benchmark suites A, B, and C contain 100-, 200-, and 500-gene networks. For each network, we set the minimum number of regulators (nodes with at least one outgoing link *in the source network*) to half of the size of the extracted network. The parameter *seed* has been set to *random vertex* and *neighbor selection* has been set to *random among top 50%*. Networks with more than one connected component were discarded.

1.2 Dynamical model

Network topologies are endowed with detailed dynamical models of gene regulation. Both transcription and translation are modeled using a standard thermodynamic approach (Ackers *et al.*, 1982) allowing for both independent ("additive") and synergistic ("multiplicative") regulatory interactions. A detailed description of the dynamical model used is given by Marbach *et al.* (2010). First, auto-regulatory interactions were removed from the extracted networks before setting the dynamical model. The most important parameters are the *mRNA* and *protein half-lives* in minutes sampled from a Gaussian distribution $\mathcal{N}(27.5, 56.25)$ bounded in the interval $[5, 50]$, the *dissociation constants* sampled from a uniform distribution $[0.01, 1]$, and the *Hill coefficients* sampled from a Gaussian distribution $\mathcal{N}(2, 4)$ bounded in the interval $[1, 10]$.

1.3 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. Systematic knockout and knockdown experiments were simulated to generate steady-state expression data. Also, 100 multifactorial perturbation experiments were simulated to generate steady-state expression data for each network from the benchmark suite C. The parameters were set to the same values used to generate the DREAM4 *In Silico* Challenge we provided. Those settings also correspond to the default parameter values provided by GNW. More specifically, we modeled molecular noise with the *coefficient of (molecular) noise term* set to 0.05 (Schaffter, 2010), in addition to a model of experimental noise observed in microarrays (Stolovitzky *et al.*, 2005).

1.4 Gold standards and network prediction format

Performance profiling of network inference methods using GNW requires *gold standards* and *network predictions* files to be provided. The gold standard files can be imported to GNW using either the format TSV, GML, DOT, or SBML[1]. The gold standard files in TSV format must be formatted as follows

```
G0    G1    1
G0    G2    1
...
G1    G0    0
...
```

*to whom correspondence should be addressed

[1] In the current version, only SBML files that have been generated by GNW can be opened.

Each line defines an interaction oriented from the first gene to the second gene. The third element is 1 if the interaction is present in the gold standard and 0 otherwise. Instead of listing the absent (0) interactions, they can also simply be omitted. The format for the predictions is the same as used for the DREAM challenges

```
G0    G1    0.98
G0    G2    0.8
...
G1    G0    0
...
```

As in the gold standard file, each line defines an interaction oriented from the first to the second gene. For each interaction, a *confidence level* between 0 and 1 is given that indicates the degree of belief that the interaction is included in the gold standard. The predictions must be listed in *descending* order relative to their confidence level (the first prediction in the list being the most confident). The confidence levels are only used to verify that the list of predictions is correctly ordered, they do not affect the PR and ROC curves and the motif analysis in any other way.

1.5 Evaluation of network inference methods

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 (network motifs). 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). The intuitive interface of GNW allows to easily evaluate several inference methods at a time to facilitate the comparison of their relative performance. Evaluation results are always saved in a text file (XML format). In addition, GNW can generate PDF reports with plots from these data (an internet connection is required). Without internet connection, the evaluation can still be run but no PDF report will be created.

2 NETWORK INFERENCE METHODS

2.1 Z-score

Z-score is one of the simplest inference methods (Prill *et al.*, 2010), yet it has relatively high accuracy in predicting directed network structures from knockout steady states (see Section 3.2 of the paper). For each gene of a network, Z-score computes the mean μ and standard deviation σ of the gene expression level from several experiments. Then for each single-gene knockout perturbation, a regulatory interaction is identified if the measured expression level of a given gene is below $\mu-\sigma$ (enhancing regulation) or above $\mu+\sigma$ (inhibitory regulation). The Matlab implementation of Z-score used is the one provided by Pinna *et al.* (2010). Z-score doesn't require any parameters to be set.

2.2 Pinnal *et al.*

The algorithm developed by Pinna *et al.* allows to choose between four possible different confidence matrices W to obtain the initial predictions (Pinna *et al.*, 2010). Here Z-score is applied on the raw gene expression data to generate the initial predictions (W^{ZR}). Then the method performs a refinement stage, which aims to suppress the errors made by Z-score on cascade motifs from knockout steady states. This improvement is achieved by reducing the confidence initially predicted to unnecessary feed-forward edges (Pinna *et al.*, 2010). The parameters used are the default ones. Especially, the threshold parameter t is set to 2 ($t = 0$ corresponds to not applying the refinement stage, i.e. Z-score alone), which is also the value Pinna *et al.* (2010) used to participate to the DREAM4 *In Silico* Challenge Size 100. Pinna *et al.* was best-performer in that challenge.

2.3 Yip *et al.*

The original Java tool developed by Yip *et al.* (2010) implements different techniques to infer gene regulations from both steady-state and time-series data. From steady-state expression data, a noise model is learnt to distinguish real signals from random fluctuations (*Batch 1*). Ordinary differential equations (ODE) are then used to model the change of expression levels of a gene along the time series due to the regulation of other genes (Yip *et al.*, 2010). Yet, Yip *et al.* (2010) applied their noise model alone to participate to the DREAM3 *In Silico* Challenge we provided, and their method was best-performer in all sub-challenge of size 10, 50, and 100 genes. Here the noise model was applied alone to predict directed networks from knockout expression data. This part of the method developed by Yip *et al.* (2010) doesn't require any parameters to be set.

2.4 CLR

The *context likelihood of relatedness (CLR)* algorithm developed by Faith *et al.* (2007) is an unsupervised network inference method using mutual information as a metric of similarity between the expression profiles of two genes. The method doesn't require systematic knockout gene expression data, which are not always available in practice, to infer undirected networks. We applied CLR 1.2.2 using the provided binary for Linux. All mutual information values were computed using *5 bins* and *third order B-splines* (Faith *et al.*, 2007).

2.5 ARACNE2

Similar to CLR, the inference method developed by Margolin *et al.* (2006) also uses mutual information as a metric of similarity between the expression profiles of two genes. The method allows the reconstruction of undirected networks from steady-state expression data, and doesn't require systematic knockout or knockdown experiments. The Java implementation of ARACNE2 was used with the provided default settings, that is, the *algorithm* set to *fixed_bandwidth*, the *p-value for MI threshold* to 1, the *DPI tolerance* to 1, the gene filter configured with *mean* and *cv* to 0, and the *MI threshold* to 0.

2.6 GENIE3

Huynh-Thu *et al.* (2010) decomposes the prediction of a regulatory network between p genes into p different regression problems

(Huynh-Thu *et al.*, 2010). GENIE3 has the potential ability to predict directed networks, while methods based on mutual information or correlation can only predict undirected networks unless additional information is used. The Matlab implementation of GENIE3 was used with the *Random Forests* procedure, the parameter K set to the square root of the number of input genes, and the number of trees grown in an ensemble set to 1000 (Huynh-Thu *et al.*, 2010).

REFERENCES

Ackers, G., Johnson, A., and Shea, M. (1982). Quantitative model for gene regulation by lambda phage repressor. *Proceedings of the National Academy of Sciences of the United States of America*, **79**(4), 1129.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM.

Faith, J., Hayete, B., Thaden, J., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J., and Gardner, T. (2007). Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol*, **5**(1), e8.

Gama-Castro, S., Salgado, H., Peralta-Gil, M., Santos-Zavaleta, A., Muñiz-Rascado, L., Solano-Lira, H., Jimenez-Jacinto, V., Weiss, V., García-Sotelo, J., López-Fuentes, A., *et al.* (2011). RegulonDB version 7.0: transcriptional regulation of Escherichia coli K-12 integrated within genetic sensory response units (Gensor Units). *Nucleic Acids Research*, **39**(suppl 1), D98.

Huynh-Thu, V., Irrthum, A., Wehenkel, L., Geurts, P., and Isalan, M. (2010). Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. *PLoS ONE*, **5**(9), e12776.

Kim, S., Imoto, S., and Miyano, S. (2003). Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in Bioinformatics*, **4**(3), 228.

Marbach, D., Prill, R. J., Schaffter, T., Mattiussi, C., Floreano, D., and Stolovitzky, G. (2010). Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences*, **107**(14), 6286–6291.

Margolin, A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R., and Califano, A. (2006). ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics*, **7**(Suppl 1), S7.

Pinna, A., Soranzo, N., and de la Fuente, A. (2010). From Knockouts to Networks: Establishing Direct Cause-Effect Relationships through Graph Analysis. *PloS one*, **5**(10), 218–223.

Prill, R., Marbach, D., Saez-Rodriguez, J., Sorger, P., Alexopoulos, L., Xue, X., Clarke, N., Altan-Bonnet, G., and Stolovitzky, G. (2010). Towards a rigorous assessment of systems biology models: the DREAM3 challenges. *PloS one*, **5**(2), e9202.

Schaffter, T. (2010). Numerical Integration of SDEs: A Short Tutorial. Technical report, Swiss Federal Institute of Technology in Lausanne (EPFL).

Stolovitzky, G., Kundaje, A., Held, G., Duggar, K., Haudenschild, C., Zhou, D., Vasicek, T., Smith, K., Aderem, A., and Roach, J. (2005). Statistical analysis of MPSS measurements: application to the study of LPS-activated macrophage gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(5), 1402.

Yip, K., Alexander, R., Yan, K., and Gerstein, M. (2010). Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data. *PloS one*, **5**(1).