In Silico Network Challenge

DREAM4, Challenge 2

Synopsis

The goal of the *in silico* network challenge is to reverse engineer gene regulation networks from simulated steady-state and time-series data. Participants are challenged to infer the network structure from the given *in silico* gene expression datasets. Optionally, participants may also predict the response of the networks to a set of novel perturbations that were not included in the provided datasets.

The three sub-challenges

There are three *in silico* sub-challenges called

- InSilico_Size10
- InSilico_Size100
- InSilico_Size100_Multifactorial

The sub-challenges differ in the size of the network and the type of data provided. Predictions are assessed independently for each sub-challenge. Thus, teams may choose to submit predictions to all three or only some of the challenges.

Each sub-challenge consists of five networks (the so-called gold standard networks). In order to participate in a challenge, predictions for all five networks of this sub-challenge must be submitted. The rational is that in this way it will be possible to assess how consistently a method predicts the topology in five independent networks of the same type and size.

InSilico_Size10 sub-challenge

In the first sub-challenge, we provide all of the datasets described in the next section (wild-type, knockouts, knockdowns, multifactorial perturbations, and time series) for five networks of size 10. Participants are challenged to predict the directed unsigned topology of these networks. In addition, participants can choose to predict the network response to previously unseen perturbations in the bonus round described below. Note that the best performer of the sub-challenge will be determined solely based on the prediction of the network topologies, and participation in the bonus round is optional.

Bonus round. Whereas some inference methods focus on predicting only network structures, others reverse engineer (potentially) predictive dynamical models, which could be used to predict the network response to novel perturbations that were not included in the original datasets. We invite participants that tackle inference of such models to predict, in addition to the network structure, also the steady-state levels of dual knockout experiments (knockout of two genes simultaneously, as described in the next section).

InSilico_Size100 sub-challenge

The second sub-challenge is similar to the first one, except that the five networks are of size 100. Furthermore, only the wild-type, knockout, knockdown, and time-series datasets are provided (the multifactorial perturbation datasets are not included as they are the subject of another sub-challenge). The primary goal is to predict the network structures, but there is an optional bonus round where participants can evaluate whether their inferred models correctly predict the effect of dual knockouts.

InSilico_Size100_Multifactorial sub-challenge

The third sub-challenge consists of five networks of size 100. In this challenge, we assume that extensive knockout / knockdown or time series experiments can't be performed. Instead, different variations of the network can be observed (e.g., samples from different patients). Thus, only the multifactorial perturbation dataset described below is provided. The goal is prediction of the network structure. There is no bonus round in this challenge.

The datasets

The data are given for each of the three sub-challenges in the following three files:

- DREAM4_InSilico_Size10.zip
- DREAM4_InSilico_Size100.zip
- DREAM4_InSilico_Size100_Multifactorial.zip

We will now describe the types of experiments that we simulated to produce gene expression datasets, and the name of the files where this data is included. In all cases, the data corresponds to noisy measurements of mRNA levels, which have been normalized such that the maximum normalized gene expression value in the datasets of a given network is one.

Wild-type

The files ***wildtype.tsv** contain the steady-state levels of the wild-type (the unperturbed network).

Knockouts

The files ***knockouts.tsv** contain the steady-state levels of single-gene knockouts (deletions). An independent knockout is provided for every gene of the network. A knockout is simulated by setting the transcription rate of this gene to zero. The k'th data line of the file *knockouts.tsv is the steady-state of the network after knockout of gene k.

Knockdowns

The files ***knockdowns.tsv** contain the 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. The k'th data line of the file *knockdowns.tsv is the steady state of the network after knockdown of gene k.

Multifactorial perturbations

The files ***multifactorial.tsv** contain steady-state levels of variations of the network, which are obtained by applying multifactorial perturbations to the original network. Each line gives the steady state of a different perturbation experiment, i.e., of a different variation of 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.

Time series

The files ***timeseries.tsv** contain time courses showing how the network responds to a perturbation and how it relaxes upon removal of the perturbation. For networks of size 10 we provide 5 different time series, for networks of size 100 we provide 10 time series. Each time series has 21 time points. The initial condition always corresponds to a steady-state measurement of the wild-type. At t=0, a perturbation is applied to the network as described below. The first half of the time series (until t=500) shows the response of the network to the perturbation. At t=500, the perturbation is removed (the wild-type network is restored). The second half of the time series (until t=1000) shows how the gene expression levels go back from the perturbed to the wild-type state.

In contrast to the multifactorial perturbations described in the previous section, which affect all the genes simultaneously, the perturbations applied here only affect about a third of all genes, but basal activation of these genes can be strongly increased or decreased. For example, these experiments could correspond to physical or chemical perturbations applied to the cells, which would cause (via regulatory mechanisms not explicitly modeled here) some genes to have an increased or decreased basal activation. The genes that are directly targeted by the perturbation may then cause a change in the expression level of their downstream target genes.

Dual knockouts

Dual knockouts consist of simulating each of the five networks in which two gene are knocked-out simultaneously. Gene expression data for dual knockouts is not provided to the participants. Instead, participants may predict steady-state levels for dual knockouts in the bonus round described in the previous section. The files ***dualknockouts_indexes.tsv** indicate the pairs of genes for which a dual knockout should be predicted. For example, the line "6 8" means that participants should predict the steady-state of the network after

knocking out genes 6 and 8. For networks of size 10 we ask for predictions for 5 dual knockout experiments, for networks of size 100 we ask for 20 predictions.

Submission Information

Network predictions

Network predictions must be directed and unsigned. There are no self-interactions (auto-regulatory loops) in the gold standard networks. Predictions of self-loops are ignored by the scoring.

Submit a ranked list of regulatory link predictions ordered according to the confidence you assign to the predictions, from the most reliable (first row) to the least reliable (last row) prediction. Use a 3 tab-separated column format as in the example below:

A \tab B \tab XYZ

where A and B are two different genes (no self-interactions). Links are directed: the gene in the first column regulates the gene in the second column. (If both A regulates B and B regulates A, then both lines should be included.) XYZ is a score between 0 and 1 that indicates the confidence level you assign to the prediction. (E.g., XYZ = 1 if gene A is deemed to regulate gene B with highest confidence and XYZ = 0 if A is deemed not to directly regulate B). All pairs omitted from the list will be considered to appear randomly ordered at the end of the list. Save the file as text, and name it:

- **DREAM4_TeamName_SubChallenge_Network.txt**

where TeamName is the name of the team with which you registered for the challenge, SubChallenge is either InSilico_Size10, InSilico_Size100, or InSilico_Size100_Multifactorial, and Network is one of the five networks of the indicated challenge (1,2,...5). As mentioned above, to participate in a challenge you need to submit predictions for all five networks of this challenge.

Bonus round predictions

Predictions for double knockouts in the bonus round should be submitted in the following format. The file should have M lines, where M is the number of double knockouts to be predicted (5 for networks of size 10 and 20 for networks of size 100). Line k should contain the steady-state levels of all genes in the k'th double knockout experiment

x_1 \tab x_2 \tab x_3 \tab ... x_N \newline

where x_i is the predicted expression level of gene i, and N is the size of the network. The two genes that should be knocked out in the k'th experiment are indicated in the file *doubleknockout_indexes.tsv, as described in the previous section. If the pair of genes (u, v) are knocked out in the k'th experiment, x_u and x_v must be equal zero in that line (we will verify this to check that the file format is correct).

Please submit a separate file for every network. Use the same naming convention as explained above for the network predictions and append _dualknockouts to the filename:

- **DREAM4_TeamName_SubChallenge_Network_dualknockouts.txt**

Scoring Metrics

We will score the results using the area under the precision versus recall curve for the whole set of link predictions for a network. For the first k predictions (ranked by score, and for predictions with the same score, taken in the order they were submitted in the prediction files), precision is defined as the fraction of correct predictions to k, and recall is the proportion of correct predictions out of all the possible true connections. Other metrics such as precision at 1%, 10%, 50%, and 80% recall, and the area under the ROC curve will also be evaluated. Teams will be ranked according to their overall performance over the five networks of a challenge.

Predictions for dual knockouts in the bonus round will be evaluated by comparing them to the true, noise-free gene expression values (e.g. using a sum of square error).

How were the *in silico* benchmarks generated?

Network structures. Network topologies were obtained by extracting subnetworks from transcriptional regulatory networks of *E. coli* and *S. cerevisiae*. We adapted the subnetwork extraction method to preferentially include parts of the network with cycles. Auto-regulatory interactions were removed, i.e., there are no self-interactions in the *in silico* networks.

Dynamical model. The dynamics of the networks were simulated using a detailed kinetic model of gene regulation. Both independent and synergistic gene regulation occur in the networks. Both transcription and translation are modeled. However, the protein concentrations are not included in the provided datasets. As mentioned above, the datasets correspond to the mRNA concentration levels.

Noise. The simulations are based on stochastic differential equations (Langevin equations) to model internal noise in the dynamics of the networks. In addition, we add measurement noise to the generated gene expression datasets. We use an existing model of noise observed in microarrays, which is very similar to a mix of normal and lognormal noise.

Software. All networks and data were generated with version 2.0 of GeneNetWeaver (GNW). The previous version of GNW, which was used to generate the DREAM3 challenges, is available at gnw.sourceforge.net.

Additional information. Additional information, including a short description of the dynamical model, will be posted on our website: gnw.sourceforge.net.

Authors

The challenge was provided by Daniel Marbach, Thomas Schaffter, and Dario Floreano from the Laboratory of Intelligent Systems (lis.epfl.ch) of the Swiss Federal Institute of Technology in Lausanne. The challenge has been designed in collaboration with Robert J. Prill and Gustavo Stolovitzky from the IBM T.J. Watson Research Center in New York.

All data can be freely used. If you use this data in your publication, please cite the following papers:

- Marbach, D., Schaffter, T., Mattiussi, C. and Floreano, D. (2009) Generating Realistic *in silico* Gene Networks for Performance Assessment of Reverse Engineering Methods. Journal of Computational Biology, 16(2) pp. 229-239.
- Stolovitzky G, Prill RJ, Califano A. "Lessons from the DREAM2 Challenges", in Stolovitzky G, Kahlem P, Califano A, Eds, Annals of the New York Academy of Sciences, 1158:159-95 (2009)
- Stolovitzky G, Monroe D, Califano A. "Dialogue on Reverse-Engineering Assessment and Methods: The DREAM of High-Throughput Pathway Inference", in Stolovitzky G and Califano A, Eds, Annals of the New York Academy of Sciences, 1115:11-22 (2007)

Download

- Download Data

Don't hesitate to post a question in the DREAM discussion board if you need any clarification on this challenge.