The DREAM4 In-silico Network Challenge

Training data, gold standards, and supplementary information

Daniel Marbach[1,2,*], Thomas Schaffter[1], Dario Floreano[1], Robert J Prill[3], and Gustavo Stolovitzky[3]

[1]Laboratory of Intelligent Systems
Swiss Federal Institute of Technology in Lausanne, Lausanne, Switzerland

[2]Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge MA, USA

[3]IBM T.J. Watson Research Center, New York NY, USA

[*]Contact: dmarbach@mit.edu

Draft, version 0.3
November 28, 2009

Preface

In this manuscript we describe the DREAM4 *in silico* network challenge, a benchmark suite for performance evaluation of methods for gene network inference (reverse engineering). We released this challenge as a community-wide experiment within the context of the DREAM4 conference. This document is distributed together with the archive DREAM4 in-silico network challenge.zip, which is available from our website (gnw.sourceforge.net).

The archive *DREAM4 in-silico network challenge.zip* contains the complete challenge, including supplementary information. Here, we describe all provided files. In particular, we describe:

- the DREAM4 *in silico* network challenge,
- the training datasets of the challenge,
- the gold standards (solutions) of the challenge,
- supplementary information (additional datasets, the datasets without noise, details on the applied perturbations, network graphs in EPS format, signed gold standards, etc).

In this manuscript we do **not** discuss:

- the "science" behind the benchmarks, e.g., the models and simulation used to generate the data (refer to our published papers listed below, you may also contact Daniel Marbach for preprints),
- the evaluation of the predictions (refer to the information and evaluation scripts available on the DREAM results website `http://wiki.c2b2.columbia.edu/dream/results/DREAM4`),
- the results of the challenge (available on the DREAM results website).

How to cite us

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):229–239. [infoscience.epfl.ch/record/128148]
- Stolovitzky G, Prill RJ, Califano A (2009) Lessons from the DREAM2 Challenges. In Stolovitzky G, Kahlem P, Califano A, Eds, *Annals of the New York Academy of Sciences*, 1158:159–95.
- Stolovitzky G, Monroe D, Califano A (2007) 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.

GeneNetWeaver

All files have been generated automatically with our Java webstart tool GeneNetWeaver (GNW) version 2.0. You can launch GNW with a simple click right here or from our website (gnw.sourceforge.net), no installation is required. Using GNW you can, for example:

- open and visualize the gene network structures,
- open the dynamical models of the gold standards and simulate additional datasets,
- generate additional benchmarks similar to the DREAM4 challenges.

Important: GNW version 2.0 has not yet been released. With GNW version 1.x you can open the files, but if you generate additional datasets, they will be based on the model of the previous edition of the challenge. We plan to release GNW version 2.0 in December 2009.

Contact / feedback

For questions or feedback (highly appreciated) concerning the *in silico* challenge, please use the DREAM discussion forum or contact Daniel Marbach.

Structure of this document

This manuscript is split into 6 chapters. The content of the archive DREAM4 in-silico network challenge.zip is described in Chapters 2–4.

Chapter 1 is the description of the challenge as it was given to the participants.

Chapter 2 describes the files of the *Size10* and *Size100* subchallenges.

Chapter 3 describes the files of the *Size100_multifactorial* subchallenge.

Chapter 4 gives the *E. coli* and *yeast* networks from which the benchmark networks were extracted.

Chapter 5 lists some questions and answers from the DREAM discussion forum.

Abbreviations

ODE Ordinary Differential Equation

SDE Stochastic Differential Equation (Langevin equation)

Contents

Chapter 1

Challenge description

In this chapter we print the description of the *in silico* network challenge exactly as it was published on the DREAM website. Note that this was all the information that the participants had on the benchmarks of the challenge, the more detailed descriptions given in Chapters 2–5 were not available for the participants.

1.1 In Silico Network Challenge

1.1.1 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.

1.1.2 The three subchallenges

There are three *in silico* subchallenges called

- InSilico_Size10
- InSilico_Size100
- InSilico_Size100_Multifactorial

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

Each subchallenge consists of five networks (the so-called gold standard networks). In order to participate in a subchallenge, predictions for all five networks of this subchallenge 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 subchallenge

In the first subchallenge, 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 subchallenge 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 subchallenge

The second subchallenge 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 subchallenge). 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 subchallenge

The third subchallenge 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.

1.2 The datasets

The data are given for each of the three subchallenges in the following three files [available on the DREAM website as well as in the archive accompanying this manuscript]:

- DREAM4_InSilico_Size10.zip [cf. Section 2.1]
- DREAM4_InSilico_Size100.zip [cf. Section 2.1]
- DREAM4_InSilico_Size100_Multifactorial.zip [cf. Section 3.1]

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.

1.2.1 Wild-type

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

1.2.2 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.

1.2.3 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.

1.2.4 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.

1.2.5 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 [e.g., a drug is added]. The first half of the time series (until t=500) shows the response of the network to the perturbation [the perturbation is constantly applied from t=0 until t=500]. At t=500, the perturbation is removed (the wild-type network is restored) [e.g., the drug is removed]. 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.

1.2.6 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.

1.3 Submission Information

1.3.1 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.

1.3.2 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

1.4 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).

1.5 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* [see Chapter 4]. 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 addi-

tion, 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 gnw.sourceforge.net.

Chapter 2

The Size10 and Size100 subchallenges

This chapter describes the content of the folders *Size 10* and *Size 100* of the archive distributed with this manuscript. These folders contain the complete set of files associated with the benchmarks of the *Size10* and *Size100* subchallenges (the *Size100_multifactorial* subchallenge will be described in the next chapter).

2.1 DREAM4 training data

This folder contains the noisy time-series and steady-state datasets that were provided to the participants of the challenge. These datasets are identical to the ones released on the DREAM website. 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. **Note that the time and concentrations are unitless** (we have non-dimensionalized the kinetic models as described by von Dassow et al. in the supplementary information of their paper [*Nature*, 406:188-192, 2000]).

Participants were free to use all steady-state and time-series datasets combined, or to use only a subset of the data to predict the network structures (and the steady-state levels for the double knockouts in the optional bonus round).

***wildtype.tsv** These files contain the steady-state levels of the wild-type (the unperturbed network), as described in Section 1.2.1.

***knockouts.tsv** Steady-state levels of single-gene knockouts (deletions), see Section 1.2.2.

***knockdowns.tsv** Steady-state levels of single-gene knockdowns, see Section 1.2.3.

***multifactorial.tsv (only for *Size10*)** Steady-state levels of variations of the network, which are obtained by applying multifactorial perturbations to the original network as described in Section 1.2.4.

***timeseries.tsv** Time courses showing how the network responds to a perturbation and how it relaxes upon removal of the perturbation, see Section 1.2.5.

2.1.1 Bonus round

There was no additional training data for the bonus round. Thus, the only file for the bonus round is the one specifying the dual knockouts that were asked to be predicted:

***dualknockouts_indexes.tsv** The pairs of genes for which a dual knockout was asked to be predicted in the bonus round, see Section 1.2.6.

2.2 DREAM4 gold standards

This folder contains the gold standards of the DREAM challenge. These files are identical to the ones released on the DREAM website.

***goldstandard.tsv** These files contain the true network structures. Edges marked with `1` are present in the network, edges marked with `0` are absent.

***nonoise_normalized_dualknockouts.tsv** These files, located in the *Bonus round* folder, contain the steady-states of the double knockouts that were asked to be predicted in the bonus round. The data is noise-free: it was simulated using Ordinary Differential Equations (ODEs) and no experimental noise (measurement error) was added. These files are identical to the ones available on the DREAM website, we just added the tag normalized to the filename to indicate that the data is normalized in the same way as the training data (this is not the case for some other datasets described in Section 2.3). Note that noisy dual-knockout datasets are available in the supplementary information described in the next section.

2.3 Supplementary information

This folder contains the complete set of files that we created when generating the benchmarks. In addition to the training data and the gold standards

used for the DREAM challenge, this includes much additional information such as further training datasets, noise-free datasets, the network graphs in EPS format, etc.

For a direct comparison of an inference method with the results of the challenge participants, only the DREAM4 training data described in Section 2.1 should be used. However, the additional datasets may be useful to test whether the same or better performance can be achieved using different types of data. It would also be interesting to know whether similar performance can be achieved using less data, and to study the effects of noise by comparing the results from the noise-free data, for instance.

insilico_size10_2_oscillating This folder contains a benchmark that was not part of the challenge, we included it here because it may be an interesting test case. In contrast to the other networks, it has a strong oscillatory behavior. Thus, the "steady-state" datasets are actually not true steady states in this network, they are just the state of the network after the maximum allowed simulation time. Note that this network has the same structure as the *insilico_size10_2* network of the challenge.

2.3.1 Gold standard

***goldstandard.tsv** This file is identical to the one in the DREAM4 gold standards folder (Section 2.2).

***goldstandard_signed.tsv** The directed signed network structures (`+` indicates enhancing, `-` repressing interactions).

***gene_names.tsv** In the DREAM4 challenge, genes were labeled G1, G2, etc. This table indicates for each gene the original label in the source network (see Chapter 4).

***.eps** The graphs of all gold standard networks in EPS format. You can also create and export suitable images yourself by opening the gold standards or kinetic models with GNW.

***.xml** The kinetic models of the gold standard networks. These files can be opened with GNW, for example to simulate additional datasets. (Note that even though we use SBML, these files can't be opened with other simulators than GNW. In the next version of GNW, we will implement a compatible SBML format).

2.3.2 SDEs with experimental noise

The datasets in this folder were generated using Stochastic Differential Equations (SDEs) to model internal noise in the dynamics of the networks (e.g.,

noise due to small numbers of molecules, noise in transcription and tranlsation, etc). In addition, experimental noise was added to the generated gene expression datasets using a model of noise in microarrays.

This folder includes the datasets that were provided as training data for the challenge plus additional datasets (for convenience, we put copies of the training-data files in the DREAM4 training data folder described in Section 2.1). The datasets always correspond to mRNA concentrations, except for those with the tag proteins in the filename.

***wildtype.tsv** Described in Section 2.1 (part of the challenge training data).

***knockouts.tsv**
***knockouts_timeseries.tsv** The first file gives the steady states for the knockouts, as described in Section 2.1 (part of the challenge training data). The second file gives the corresponding time series, which show how the network goes from the wild type to the perturbed steady state for every knockout (this was not included in the training data for participants of the challenge). The k'th time series corresponds to the knockout of gene k (time series are separated by an empty line). The first time point is the wild type. At that point, the gene is knocked out (it's transcription rate is set to zero). The following time points show the response of the network until t=1000. Note that the network has not necessarily reached its steady state at that point. The steady states given in the files **knockouts.tsv* do not correspond to the measurement at t=1000, but to the actual steady state, which may be reached after t=1000.

***knockdowns.tsv**
***knockdowns_timeseries.tsv** The first file gives the steady states for the knockdowns, as described in Section 2.1 (part of the challenge training data). The second file gives the corresponding time series, as described in the previous paragraph for the knockouts (not part of the challenge training data).

***multifactorial.tsv**
***multifactorial_timeseries.tsv** The steady states for the multifactorial perturbations described in Section 2.1 (part of the challenge training data), and the corresponding time series.

***timeseries_steadystates.tsv**
***timeseries.tsv** The second file is the time-series data provided in the challenge. As described in Section 2.1, time series were obtained by perturbing many genes of the networks at the same time from t=0 until t=500. The files **timeseries_steadystates.tsv* contain the steady states of the network for these perturbations (I admit, the filename *is*

confusing). In other words, these are the steady states corresponding to the perturbations that were applied in the time-series datasets of the challenge. In the time series, the perturbations are removed at t=500 (see Section 2.1). The steady states given here do not correspond to the time point at t=500, because at this time the networks have usually not yet reached a steady state. Instead, the given steady states are the true steady states that would be reached if the perturbations were not removed.

Note that the **multifactorial.tsv* and the **timeseries_steadystates.tsv* datasets are similar, they both correspond to steady states after multifactorial perturbations of the network. However, as described in Section 2.1, the type of multifactorial perturbation is different in the two cases. In **multifactorial.tsv*, all genes of the network are slightly perturbed. In **timeseries_steadystates.tsv*, only about a third of the genes are perturbed, but these genes are perturbed more strongly.

***dualknockouts.tsv**

***dualknockouts_timeseries.tsv** The steady states of the dual knockouts that were asked to be predicted in the bonus round of the challenge (see Section 2.2) and the corresponding time series. Note that the indexes of the genes that were knocked out in each double knockout experiment are given in the files **dualknockouts_indexes.tsv* described in Section 2.1.

***proteins.tsv** The protein concentrations for all the experiments described above.

***normalization.tsv** After adding noise, we have normalized the mRNA concentrations by dividing them by the maximum mRNA concentration value across all datasets. This maximum value is saved in the file **normalization.tsv*. All of the above datasets (including the protein concentration data) have been divided by this same value. Note that even though the protein concentrations have also been divided by this constant for consistency, they are not necessarily normalized because the constant is the maximum mRNA concentration value.

2.3.3 SDEs without experimental noise (*noexpnoise*)

This folder contains the exact same datasets as described in Section 2.3.2, but before adding experimental noise. In other words, these are the datasets simulated using SDEs to model noise in the dynamics of the networks, but without modeling additional experimental noise (measurement error). The tag *noexpnoise* (no experimental noise) was added to the filenames of the datasets in this folder.

Note that these datasets have not been normalized, for this reason there is no file **noexpnoise_normalization.tsv* (normalization was only done for the datasets described in the previous section, because this was the type of data provided in the challenge).

2.3.4 ODEs without experimental noise (*nonoise*)

This folder contains the exact same datasets as described in Section 2.3.2, but generated using ODEs instead of SDEs. No experimental noise was added. In other words, the data is completely noise free. The tag *nonoise* (no noise) was added to the filenames of the datasets in this folder.

Note that these datasets have not been normalized, for this reason there is no file **nonoise_normalization.tsv* (normalization was only done for the datasets described in Section 2.3.2, because this was the type of data provided in the challenge).

2.3.5 Perturbations

This folder contains information on the dual knockouts and on the multifactorial perturbations that were applied to the networks. These files can be loaded with GNW version 2.0 to generate additional datasets with these same perturbations.

***dualknockouts_indexes.tsv** The indexes of the genes that were knockouted out in each dual knockout experiment of the bonus rounds, as described in Section 2.1 (given to the participants along with the challenge training data).

***timeseries_perturbations.tsv** The perturbations that were applied in each time series (see Section 1.2.5 for a description of the time series datasets and the type of perturbations). This information was not given to the participants of the challenge. The k'th data line gives the perturbation of the k'th time series. The given values are the perturbations of the basal transcrption rate for every gene. For example, a value of 0.5 means that the basal transcription rate of that gene was increased by 0.5. A value of zero means that this gene was not directly perturbed (it may have been indirectly perturbed due to perturbations of its regulators).

***multifactorial_perturbations.tsv** The perturbations that were applied in the multifactorial perturbation datasets (see Section 1.2.4 for a description of these perturbations). The format is the same as desribed in the pervious paragraph for time-series perturbations. Note that the perturbations tend to be weaker than those in **timeseries_perturbations.tsv*, but all genes are being perturbed.

Chapter 3

The Size100_multifactorial subchallenge

This chapter describes the content of the folder named *Size 100 multifactorial* of the archive distributed with this manuscript. This folder contains the complete set of files associated with the benchmarks of the *Size100_multifactorial* subchallenge. Note that there is no supplementary information associated with this subchallenge because the benchmark networks are identical to those of the *Size100* subchallenge, as discussed below.

3.1 DREAM4 training data

***multifactorial.tsv** The only training data available in this subchallenge were the steady states of the network after application of multifactorial perturbations. The multifactorial perturbation datasets have been described in Section 1.2.4.

3.2 DREAM4 gold standards

***goldstandard.tsv** The true network structures (i.e., a list of edges that are present in the network).

***goldstandard_ref.tsv** The gene networks of the *Size100_multifactorial* subchallenge are identical to the ones of the *Size100* subchallenge, we just randomly renamed all the nodes. The files **goldstandard_ref.tsv* contain the mapping of gene names. The original gene names in the *Size100* subchallenge are given in the second column, and the corresponding gene names in the *Size100_multifactorial* subchallenge are given in the first column.

Chapter 4

Source Networks

This chapter describes the content of the folder *Source networks*. This folder contains the two source networks from which we extracted modules to generate the gold-standard network structures.

As mentioned in Section 1.5, network topologies were obtained by extracting subnetworks from the transcriptional regulatory networks of model organisms. The following two networks were used:

ecoli_transcriptional_network_regulonDB_6_2.tsv *Escherichia coli* transcriptional regulatory network: 1502 nodes, 3587 edges. Corresponds to the TF-gene interactions of RegulonDB release 6.2. (Gama-Castro et al. 2008. *Nucleic Acids Res*, 36:D120-4).

yeast_transcriptional_network_Balaji2006.tsv *Saccharomyces cerevisiae* (yeast) transcriptional regulatory network: 4441 nodes, 12873 edges (Balaji et al. 2006. *J Mol Biol*, 360:213-27).

Chapter 5

FAQ

In this chapter we list some questions and answers from the DREAM discussion forum.

Can we use the knockouts, knockdowns, time series, and multifactorial datasets all together, or can we use only one type of dataset at a time to predict a network?

In the DREAM4 challenge, participants were provided the data in the folders *DEAM4 training data* (Sections 2.1 and 3.1). They could use this data to predict the networks and optionally also to predict the steady-state levels for dual knockouts. They were free to use all of this data combined, or only a subset of the data.

Are the perturbations time varying or time invariant?

The perturbations are time invariant. For multifactorial perturbations (as well as time-series), each gene may be perturbed by a different amount, but the perturbation is constant in time for a given experiment.

Are the initial points for all of the time series the same steady state, up to noise?

In principle yes. The perturbation is applied at t=0, but did not yet have an effect. Thus, the initial points of each time series are independent samples of the wild-type steady state.

Are the last time points (t=1000) the same as the initial state up to noise?

Not necessarily, because the time from t=500 when the perturbation is removed until t=1000 may not be sufficient for the networks to completely

go back to the initial steady state. Also, in exceptional cases the network could go to a different attractor after the perturbation. We did not analyze whether this actually occurs, so this possible issue should be considered part of the challenge.

The description of the data leads one to assume that both the wildtype and the single deletion networks have a unique steady state. Is this assumption correct?

We generated the data using numerical simulations, without a detailed analysis of the different attractors that the *in silico* gene networks have. Some networks probably have several steady states. The wild-type steady state was defined arbitrarily as one of possibly several steady states of the network. The steady states for the genetic perturbations (knockouts and knockdowns) are those that the network converged to from this wild-type steady state after applying the genetic perturbation. Note that the networks typically converge to a steady state, but as in a biological experiment, there is no absolute guarantee. In exceptional cases, there may be oscillations in the *in silico* networks. Again, this possible issue is part of the challenge.

In the time series experiments, are there 5 distinct perturbation effects (chemicals), or a single perturbation at 5 different concentration levels?

Each time series corresponds to a distinct perturbation.