Package 'DREAM4'

October 12, 2020

Type Package

Title Synthetic Expression Data for Gene Regulatory Network Inference from the 2009 DREAM4 challenge

Version 1.24.0

Date 2013-03-15

Author Paul Shannon

Maintainer Paul Shannon <pshannon@systemsbiology.org>

Depends R (>= 2.15.1), SummarizedExperiment

Suggests RUnit, networkBMA

biocViews ExperimentData, Genome, Saccharomyces_cerevisiae_Data, SequencingData

Description
Simulated expression data for five 10-node, and five 100-node networks, with associated data (including solutions) from the 2009 DREAM4 challenge.

License GPL

LazyLoad yes

git_url https://git.bioconductor.org/packages/DREAM4

git_branch RELEASE_3_11

git_last_commit 2532ad7

git_last_commit_date 2020-04-27

Date/Publication 2020-10-12

R topics documented:

DREAM4 *DREAM4: Data from the DREAM4 In-silico Network Challenge*

Description

Please note that this documentation page draws liberally upon that provided by the DREAM project itself. Please see `http://wiki.c2b2.columbia.edu/dream/index.php/D4c2`.

Broadly speaking:

"DREAM is a Dialogue for Reverse Engineering Assessments and Methods. The main objective is to catalyze the interaction between experiment and theory in the area of cellular network inference and quantitative model building in systems biology."

More specifically:

"...the DREAM4 in silico network challenge [is] a benchmark suite for performance evaluation of methods for gene network inference (reverse engineering). We [the DREAM project] released this challenge as a community-wide experiment within the context of the DREAM4 conference."

DREAM4 includes ten synthetic expression sets, all of which are included here. There are ten genes in the first five sets, and one hundred genes in the second five.

Simulated expression is calculated by GeneNetWeaver, with these features:

- Extraction of modules from known transcriptional regulatory network patterns, drawn from E.coli and Yeast studies.
- Generation of realistic in-silico gene network benchmarks for network inference methods
- Simulation of realistic biological experiments (knowckout, knockdown, dual-knockout, multifactorial perturbations, time series, etc.)

Each dataset is a RangedSummarizedExperiment object, consisting of these slots:

- one assay (with one row per gene, and one column for each simulated condition, about which more below).
- a rowRanges object, a GRangesList, with one element for each row in the assay expression matrix. At present the only useful information is the 'gene' name; no metadata or futher description is offered.
- one colData object, a list of DataFrames, one for each column in the assay. At present, only the rownames of these one-row DataFrames are of interest. In time, we may adopt conventions so that inference programs can easily extract perturbation and timeseries metadata from these DataFrames. That information is currently only implicit in the names.
- Two matrices are found in the metadata slot: the goldStandardAdjacencyMatrix and doubleKnockoutGenePairs.

Expression Data Columns

The synthetic (that is, simulated) expression data in all of these data sets is organized into columns, each of which has a name and an implicit type. The DREAM4 challenge provides this explanation:

The [timeseries data] 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.

We use the following conventions to label the columns:

- `wt`: The wild-type 'genome', a steady-state measurement, with no perturbations applied
- `perturbation`: 105 or 210 columns,
- `ko`: each gene in turn is knocked out
- `kd`: each gene in turn is knocked down, with expression levels cut by half
- `mf`: multi-factorial: a variety of effects are simulated together

Examples

```
library(DREAM4)
library(SummarizedExperiment)
data(dream4_010_01)
names(assays(dream4_010_01))
expressionData <- assays(dream4_010_01)$simulated
names(metadata(dream4_010_01))
goldStandardMatrix <- metadata(dream4_010_01)$goldStandardAdjacencyMatrix
```

Index