Package 'ARTIVA'

May 20, 2015

Type Package

Title Time-Varying DBN Inference with the ARTIVA (Auto Regressive TIme VArying) Model

Version 1.2.3

Date 2015-05-19

Author S. Lebre and G. Lelandais.

Maintainer S. Lebre <sophie.lebre@icube.unistra.fr>

Depends MASS, igraph, gplots

Description Reversible Jump MCMC (RJ-MCMC)sampling for approximating the posterior distribution of a time varying regulatory network, under the Auto Regressive TIme VArying (ARTIVA) model (for a detailed description of the algorithm, see Lebre et al. BMC Systems Biology, 2010). Starting from time-course gene expression measurements for a gene of interest (referred to as ``target gene'') and a set of genes (referred to as ``parent genes'') which may explain the expression of the target gene, the ARTIVA procedure identifies temporal segments for which a set of interactions occur between the ``parent genes'' and the ``target gene''. The time points that delimit the different temporal segments are referred to as changepoints (CP).

License GPL (>= 2)

LazyLoad yes

LazyData yes

Repository CRAN

NeedsCompilation no

Date/Publication 2015-05-20 01:09:27

R topics documented:

ARTIVA-package 2
ARTIVA-internal 3
ARTIVAnet 3
ARTIVAsubnet 8
ARTIVAsubnetAnalysis 15

choosePriors 19
CP.postDist 20
drosophila 22
geneNetworkSummary 23
plotCP.postDist 25
priors 27
simulatedProfiles 28
traceGeneProfiles 29
traceNetworks 30
yeast 32

Index **35**

`ARTIVA-package` *Time-Varying DBN Inference with the ARTIVA (Auto Regressive TIme VArying) Model*

Description

This package generates Reversible Jump MCMC (RJ-MCMC) sampling for approximating the posterior distribution of a time varying regulatory network, under the Auto Regressive TIme VArying (ARTIVA) model (for a detailed description of the algorithm, see Lebre et al. BMC Systems Biology, 2010).

Starting from time-course gene expression measurements for a gene of interest (referred to as "target gene") and a set of genes (referred to as "parent genes") which may explain the expression of the target gene, the ARTIVA procedure identifies temporal segments for which a set of interactions occur between the "parent genes" and the "target gene". The time points that delimit the different temporal segments are referred to as changepoints (CP).

Details

Package:	ARTIVA
Type:	Package
Version:	1.2.3
Date:	2015-05-19
License:	GPL (>=2)
LazyLoad:	yes

Author(s)

S. Lebre and G. Lelandais.

Maintainer: S. Lebre <sophie.lebre@icube.unistra.fr>.

References

Statistical inference of the time-varying structure of gene-regulation networks S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais, BMC Systems Biology, 2010, 4:130.

`ARTIVA-internal` *Internal ARTIVA Functions*

Description

Internal ARTIVA functions

Details

These are not to be called by the user (or in some cases are just waiting for proper documentation to be written).

`ARTIVAnet` *Function to run the ARTIVA procedure for Auto Regressive TIme-VArying network inference on several target genes*

Description

This function runs function `ARTIVAsubnet` for all target genes in `targetData` successively. This function generates Reversible Jump MCMC (RJ-MCMC) sampling for approximating the posterior distribution of a time varying regulatory network, under the Auto Regressive TIme VArying (ARTIVA) model (for a detailed description of the algorithm, see `ARTIVAsubnet` and see Lebre et al. BMC Systems Biology, 2010). A network representing the interactions between the factor genes and the target genes is estimated and plotted.

Usage

```
ARTIVAnet(targetData, parentData, targetNames = NULL, parentNames =
NULL, dataDescription=NULL, saveEstimations=TRUE, saveIterations=FALSE,
savePictures = TRUE, outputPath=NULL, dyn=1, segMinLength=2, maxCP=NULL,
maxPred=NULL, nbCPinit=NULL, CPinit=NULL, niter=50000, burn_in=NULL,
PSRFactor=FALSE,PSRF_thres=1.05, segmentAnalysis=TRUE,
edgesThreshold=0.5, layout = "fruchterman.reingold", cCP= 0.5,
cEdges=0.5, alphaCP=1, betaCP=0.5, alphaEdges=1, betaEdges=0.5, v0=1,
gamma0=0.1, alphad2=2, betad2=0.2, silent=FALSE)
```

Arguments

targetData	A vector with the temporal gene expression measurements for the target gene (i.e. the gene whose regulation factors are looked for).
parentData	A matrix (or a vector if only 1 parent gene) with the temporal gene expression measurements for the proposed parent genes (i.e. potential regulation factors). Parent genes are shown in row and expression values in column. For computational reasons, we advise not to test simultaneously more than 10 parent genes.
targetNames	A vector with the names for target gene(s) (optional, default: targetNames=NULL
parentNames	A vector with the names for parent gene(s) (optional, default: parentNames=NULL).
dataDescription	(Required only when the gene expression measurements contain repeated values for the same time points). A vector indicating the ordering of the time measurements in the data. For example dataDescription=rep(1:n, each=2), if there are two measurements for each time point AND the repetitions for each time point are next to each other. Note that temporal gene expression measurements have to be organized identically in arguments targetData and parentData (optional, default: dataDescription=NULL).
saveEstimations	Boolean, if TRUE all estimated posterior distributions are saved as text files either in a new sub folder named "ARTIVA_Results" created by default in the current folder or in a folder specified with argument outputPath (see below) (optional, default: saveEstimations=TRUE).
saveIterations	Boolean, if TRUE the configuration for all iterations is saved as text files either in a new sub folder named "ARTIVA_Results" created by default in the current folder or in a folder specified with argument outputPath (see below) (optional, default: saveIterations=FALSE).
savePictures	Boolean, if TRUE all estimated posterior distributions and networks are plotted in a pdf file either in a new sub folder named "ARTIVA_Results" created by default in the current folder or in a folder specified with argument outputPath (see below) (optional, default: savePictures=TRUE).
outputPath	File path to a folder in which the output results have to be saved, either a complete path or the name of a folder to be created in the current directory (optional, default: outputPath=NULL).
dyn	Time delay to be considered in the auto-regressive process, between the temporal expression measurements of the analyzed target gene and the ones of the parent genes (optional, default: dyn=1).
segMinLength	Minimal length (number of time points) to define a temporal segment. Must be - strictly - greater than 1 if there is no repeated measurements for each time point in arguments targetData and parentData (optional, default: segMinLength=2).
maxCP	Maximal number of CPs to be considered by the ARTIVA inference procedure. Note that for a temporal course with n time points the maximal number of CPs is (n-1-dyn) (see before for a description of argument dyn). For long temporal courses (more than 20 time points), we advise - for computational reasons - to limit the maximal number of CP to 15 (optional, default: maxCP=NULL, if maxCP=NULL then the maximal number of CPs is set to maxCP=min(round((n-1-dyn)/segMinLength)-1 where n is the number of time points).

maxPred
: Maximal number of simultaneous incoming edges for each segment of the regulatory network estimated for the target gene (default: maxPred=NULL, if maxPred=NULL then the maximal number of incoming edges is set to

  maxPred=min(dim(parentData)[1],15))

nbCPinit
: Number of CPs to be considered at the algorithm initialization (optional, default: nbCPinit=NULL, if nbCPinit=NULL then the initial number of CPs is randomly set to a value between 0 and maxCP/2.).

CPinit
: A vector with the initial positions for potential CPs. (optional, default: CPinit = NULL, if CPinit = NULL then the initial vector is chosen randomly according to priors, with nbCPinit changepoints (see previous argument nbCPinit)).

niter
: Number of iterations to be performed in the RJ-MCMC sampling (optional, default: niter = 50000).

burn_in
: Number of initial iterations that are discarded for the estimation of the model distribution (posterior distribution). The ARTIVAnet function is a RJ-MCMC algorithm which, at each iteration, randomly samples a new configuration of the time-varying regulatory network from probability distributions based on constructing a Markov chain that has the network model distribution as its equilibrium distribution (The equilibrium distribution is obtained when the Markov Chain converges, which requires a large number of iterations). Typically, initial iterations are notconfident because the Markov Chain has not stabilized. The burn-in samples allow to not consider these initial iterations in the final analysis (optional, default: burn_in=NULL, if burn_in=NULL then the first 25% of the iterations is left for burn_in).

PSRFactor
: Boolean, if TRUE the RJ-MCMC procedure is stopped when the Potential Scale Ratio Factor or PSRF (which is a usual convergence criterion) is below a specified threshold (see documentation for argument PSRF_thres below) or when the maximal number of iterations is reached (see documentation for argument niter previously) (optional, default: PSRFactor=FALSE). For more details about the PSRF criterion, see Gelman and Rubin (1992).

PSRF_thres
: (Only when PSRFactor=TRUE) RJ-MCMC stopping threshold: the RJ-MCMC procedure is stopped when the Potential Scale Ratio Factor (PSRF) is below this threshold (see documentation for argument PSRFactor previously). Values of the PSRF below 1.1 are usually taken as indication of sufficient convergence (optional, default: PSRF_threshold=1.05).

segmentAnalysis
: Boolean, if TRUE the posterior distribution for the edges in each temporal segment delimited by the estimated changepoints (CPs) is computed. (The estimated CPs are the k CPs with maximal posterior probability, where k is the number of CPs having the maximal porsterior probability (such that each temporal segment is larger than segMinLength) (see documentation below for output values CPpostDist, nbSegs, CPposition. If segmentAnalysis=FALSE only the posterior distributions of the CPs number and CPs positions are computed (optional, default: segmentAnalysis=TRUE).

edgesThreshold
: Probability threshold for the selection of the edges of the time-varying regulatory network when segmentAnalysis=TRUE (optional, default: edgesThreshold=0.5).

layout — Name of the function determining the placement of the vertices for drawing a graph, possible values among others: "random", "circle", "sphere", "fruchterman.reingold", "kamada.kawai", "spring", "reingold.tilford", "fruchterman.reingold.grid", see package igraph0 for more details (default: layout="fruchterman.reingold").

cCP — Maximal probability to propose the birth (resp. death) of a changepoint (CP) during the RJ-MCMC iterations (optional, default: cCP=0.5).

cEdges — Maximal probability - when a move update of the network topology is chosen - to propose ecah edge move (birth or death of an edge) within a temporal segment (optional, default: cEdges=0.5).

alphaCP — Hyperparameter for sampling the number k of CPs. k follows a Gamma distribution: Gamma(alphaCP, betaCP (see below)) (optional, default: alphaCP=1). Note that function choosePriors can be used to choose alphaCP (or betaCP) according to the desired dimension penalisation.

betaCP — Hyperparameter for sampling the number k of CPs. k follows a Gamma distribution: Gamma(alphaCP (see before), betaCP) (optional, default: betaCP=0.5). Note that function choosePriors can be used to choose betaCP (or alphaCP) according to the desired dimension penalisation.

alphaEdges — Hyperparameter for sampling the number l of regulatory interactions between the target gene and the parent genes. l follows a Gamma distribution: rgamma(1, shape=alphaEdges, rate=betaEdges) (see betaEdges below), (default: alphaEdges=1). Function choosePriors can be used to choose alphaEdges (or betaEdges) according to the desired dimension penalisation.

betaEdges — Hyperparameter for sampling the number l of regulatory interactions between the target gene and the parent genes. l follows a Gamma distribution: rgamma(1, shape=alphaEdges, rate=betaEdges) (see alphaEdges upper) (optional, default: betaEdges=0.5). Function choosePriors can be used to choose betaEdges (or alphaEdges) according to the desired dimension penalisation.

v0 — Hyperparameter for sampling the noise variance (denoted by Sig2) in the auto-regressive model defining the regulatory time vrarying network. The prior distribution of the noise variance is an Inverse Gamma distribution with shape parameter v0/2 and scale parameter gamma0/2: rinvgamma(1,shape=v0/2, scale = gamma0), (optional, default: v0=1).

gamma0 — Hyperparameter for sampling the noise variance (denoted by Sig2) in the ARTIVA package) in the auto-regressive model defining the regulatory time vrarying network. The prior distribution of the noise variance is an Inverse Gamma distribution with shape parameter v0/2 and scale parameter gamma0/2: rinvgamma(1, shape=v0/2, scale = gamma0), (optional, default: gamma0=0.1).

alphad2 — Hyperparameter for sampling a parameter that represents the expected signal-to-noise ratio (denoted by delta2 in the ARTIVA package). It is sampled according to an Inverse Gamma distribution: rinvgamma(1, shape=alphad2, scale=betad2), (optional, default: alphad2=2).

betad2 — Hyperparameter for sampling a parameter that represents the expected signal-to-noise ratio (denoted by delta2 in the ARTIVA package). It is sampled according to an Inverse Gamma distribution: rinvgamma(1, shape=alphad2, scale=betad2), (optional, default: betad2=0.2).

`silent` Boolean, if `TRUE` messages are printed along the ARTIVA procedure (optional, default: `silent=FALSE`).

Value

A table containing the information to plot (see function `traceNetworks`) the global network estimated by the `ARTIVAnet` procedure. All results are plotted in a pdf file (when choosing `savePictures = TRUE`) in folder `outputPath`. All numerical results (see `ARTIVAsubnet` output values documentation)) are saved in text files (when choosing `saveEstimations=TRUE` and/or `saveIterations=TRUE`) in folder `outputPath`.

Author(s)

S. Lebre and G. Lelandais

References

Statistical inference of the time-varying structure of gene-regulation networks S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais, BMC Systems Biology 2010, 4:130.

Inference from iterative simulation using multiple sequences. Gelman, A. and D. Rubin, Statistical science 7 (4), 457-472, 1992.

See Also

`ARTIVAsubnet`, `choosePriors`, `ARTIVAsubnetAnalysis`,`CP.postDist`,

`segmentModel.postDist`

Examples

```
library(ARTIVA)

# Load the dataset with simulated gene expression profiles
data(simulatedProfiles)

# List of target genes to be analyzed independantly with ARTIVA
targetGenes = c(1, 10, 20, "TF3", 45, 50)

# Names of the parent genes (typically transcription factors)
parentGenes = c("TF1", "TF2", "TF3", "TF4", "TF5")

###
# ARTIVA analysis searching for potential interactions between each target
# genes and a predefined list of parent genes.
###

# Note that the number of iterations in the RJ-MCMC sampling is reduced
# to 'niter=20000' in this example, but it should be increased (e.g. up to
# 50000) for a better estimation.

## Not run:
ARTIVAtest1=ARTIVAnet(targetData = simulatedProfiles[targetGenes,],
```

```
  parentData = simulatedProfiles[parentGenes,],
  targetNames= targetGenes,
  parentNames = parentGenes,
  niter = 20000,
  savePictures=FALSE)

# Note that function ARTIVAnet calls fonction ARTIVAsubnet for each
# target gene successively and provides a global estimated regulatory
# network entitled "ARTIVA_FinalNetwork.pdf" in addition to the output
# results given by function ARTIVAsubnet.

## Gene names for the target and the parent genes, minimum segment length,
## threshold for the edges selection can be specified as follow:
ARTIVAtest2=ARTIVAnet(targetData = simulatedProfiles[targetGenes,],
  parentData = simulatedProfiles[parentGenes,],
  targetNames= targetGenes,
  parentNames = parentGenes,
  segMinLength = 2,
  edgesThreshold = 0.5,
  niter = 20000,
  outputPath = "ARTIVA-test")

## End(Not run)

# By default, the output results (pictures and estimation values) are
# saved in a folder named "ARTIVAnet" created in the current directory
# In order to save the results in a specific folder, for example a
# folder entitled "ARTIVA-test" in the current directory:
```

`ARTIVAsubnet` *Function to recover Auto Regressive TIme-VArying interactions between a gene of interest (referred to as "target gene") and a set of genes which may explain the expression of the target gene.*

Description

This function generates Reversible Jump MCMC (RJ-MCMC) sampling for approximating the posterior distribution of a time varying regulatory network, under the Auto Regressive TIme VArying (ARTIVA) model (for a detailed description of the algorithm, see Lebre et al. BMC Systems Biology, 2010).

Starting from time-course gene expression measurements for a gene of interest (referred to as "target gene") and a set of genes (referred to as "parent genes") which may explain the expression of the target gene, the ARTIVA procedure identifies temporal segments for which a set of interactions occur between the "parent genes" and the "target gene". The time points that delimit the different temporal segments are referred to as changepoints (CP).

If the measurements time delay is short enough so that the expression of the target gene depends more likely on the expression of the parent genes at some previous time points, then the time delay

for the interactions can be chosen with argument dyn. In that case the set of parent genes may contain the target gene (auto-regulation). Otherwise, contemporaneous measurements of the parent genes are used to explain the expression of the target gene and argument dyn is set to 0. In the latter case, the target gene must be kept out of the set of possible parent genes.

The ARTIVA algorithm uses a combination of efficient and robust methods: (1) dynamical Bayesian networks (DBN) to model directed regulatory interactions between the parent genes and the analyzed target gene (2) RJ-MCMC sampling for inferring - simultaneously - (a) the time position (changepoints) at which the regulatory interactions between the parent genes and the target gene changes and (b) the regulatory network topologies (interactions between parent and target genes) associated with the temporal segments delimited by the changepoints.

If available, repeated measurements can be used, the design of experiments must be specified with argument dataDescription.

Usage

```
ARTIVAsubnet(targetData, parentData, targetName="Target",parentNames=NULL,
dataDescription=NULL, saveEstimations=TRUE, saveIterations=FALSE,
savePictures = TRUE, outputPath=NULL, dyn=1,  segMinLength=2, maxCP=NULL,
maxPred=NULL, nbCPinit=NULL, CPinit=NULL, niter=50000, burn_in=NULL,
PSRFactor=FALSE, PSRF_thres=1.05, segmentAnalysis=TRUE,  edgesThreshold=0.5,
layout = "fruchterman.reingold", cCP= 0.5, cEdges=0.5, alphaCP=1,
betaCP=0.5, alphaEdges=1, betaEdges=0.5, v0=1, gamma0=0.1, alphad2=2,
betad2=0.2, silent=FALSE)
```

Arguments

targetData	A vector with the temporal gene expression measurements for the target gene (i.e. the gene whose regulation factors are looked for).
parentData	A matrix (or a vector if only 1 parent gene) with the temporal gene expression measurements for the proposed parent genes (i.e. potential regulation factors). Parent genes are shown in row and expression values in column. For computational reasons, we advise not to test simultaneously more than 10 parent genes.
targetName	Name of the target gene (optional, default: targetName="Target").
parentNames	A vector with the names for parent gene(s) (optional, default: parentNames=NULL).
dataDescription	(Required only when the gene expression measurements contain repeated values for the same time points). A vector indicating the ordering of the time measurements in the data. For example dataDescription=rep(1:n, each=2), if there are two measurements for each time point AND the repetitions for each time point are next to each other. Note that temporal gene expression measurements have to be organized identically in arguments targetData and parentData (optional, default: dataDescription=NULL).
saveEstimations	Boolean, if TRUE all estimated posterior distributions are saved as text files either in a new sub folder named "ARTIVA_Results" created by default in the current folder or in a folder specified with argument outputPath (see below) (optional, default: saveEstimations=TRUE).

saveIterations Boolean, if TRUE the configuration for all iterations is saved as text files either in a new sub folder named "ARTIVA_Results" created by default in the current folder or in a folder specified with argument outputPath (see below) (optional, default: saveIterations=FALSE).

savePictures Boolean, if TRUE all estimated posterior distributions and networks are plotted in a pdf file either in a new sub folder named "ARTIVA_Results" created by default in the current folder or in a folder specified with argument outputPath (see below) (optional, default: savePictures=TRUE).

outputPath File path to a folder in which the output results have to be saved, either a complete path or the name of a folder to be created in the current directory (optional, default: outputPath=NULL).

dyn Time delay to be considered in the auto-regressive process, between the temporal expression measurements of the analyzed target gene and the ones of the parent genes (optional, default: dyn=1).

segMinLength Minimal length (number of time points) to define a temporal segment. Must be - strictly - greater than 1 if there is no repeated measurements for each time point in arguments targetData and parentData (optional, default: segMinLength=2).

maxCP Maximal number of CPs to be considered by the ARTIVA inference procedure. Note that for a temporal course with n time points the maximal number of CPs is (n-1-dyn) (see before for a description of argument dyn). For long temporal courses (more than 20 time points), we advise - for computational reasons - to limit the maximal number of CP to 15 (optional, default: maxCP=NULL, if maxCP=NULL then the maximal number of CPs is set to maxCP=min(round((n-1-dyn)/segMinLength)-1 where n is the number of time points).

maxPred Maximal number of simultaneous incoming edges for each segment of the regulatory network estimated for the target gene (default: maxPred=NULL, if maxPred=NULL then the maximal number of incoming edges is set to

maxPred=min(dim(parentData)[1],15))

nbCPinit Number of CPs to be considered at the algorithm initialization (optional, default: nbCPinit=NULL, if nbCPinit=NULL then the initial number of CPs is randomly set to a value between 0 and maxCP/2.).

CPinit A vector with the initial positions for potential CPs. (optional, default: CPinit = NULL, if CPinit = NULL then the initial vector is chosen randomly according to priors, with nbCPinit changepoints (see previous argument nbCPinit)).

niter Number of iterations to be performed in the RJ-MCMC sampling (optional, default: niter = 50000).

burn_in Number of initial iterations that are discarded for the estimation of the model distribution (posterior distribution). The ARTIVAsubnet function is a RJ-MCMC algorithm which, at each iteration, randomly samples a new configuration of the time-varying regulatory network from probability distributions based on constructing a Markov chain that has the network model distribution as its equilibrium distribution (The equilibrium distribution is obtained when the Markov Chain converges, which requires a large number of iterations). Typically, initial iterations are notconfident because the Markov Chain has not stabilized. The burn-in samples allow to not consider these initial iterations in the final analysis

(optional, default: burn_in=NULL, if burn_in=NULL then the first 25% of the iterations is left for burn_in).

PSRFactor — Boolean, if TRUE the RJ-MCMC procedure is stopped when the Potential Scale Ratio Factor or PSRF (which is a usual convergence criterion) is below a specified threshold (see documentation for argument PSRF_thres below) or when the maximal number of iterations is reached (see documentation for argument niter previously) (optional, default: PSRFactor=FALSE). For more details about the PSRF criterion, see Gelman and Rubin (1992).

PSRF_thres — (Only when PSRFactor=TRUE) RJ-MCMC stopping threshold: the RJ-MCMC procedure is stopped when the Potential Scale Ratio Factor (PSRF) is below this threshold (see documentation for argument PSRFactor previously). Values of the PSRF below 1.1 are usually taken as indication of sufficient convergence (optional, default: PSRF_threshold=1.05).

segmentAnalysis — Boolean, if TRUE the posterior distribution for the edges in each temporal segment delimited by the estimated changepoints (CPs) is computed. (The estimated CPs are the k CPs with maximal posterior probability, where k is the number of CPs having the maximal porsterior probability (such that each temporal segment is larger than segMinLength) (see documentation below for output values CPpostDist, nbSegs, CPposition. If segmentAnalysis=FALSE only the posterior distributions of the CPs number and CPs positions are computed (optional, default: segmentAnalysis=TRUE).

edgesThreshold — Probability threshold for the selection of the edges of the time-varying regulatory network when segmentAnalysis=TRUE (optional, default: edgesThreshold=0.5).

layout — Name of the function determining the placement of the vertices for drawing a graph, possible values among others: "random", "circle", "sphere", "fruchterman.reingold", "kamada.kawai", "spring", "reingold.tilford", "fruchterman.reingold.grid", see package igraph0 for more details (default: layout="fruchterman.reingold").

cCP — Maximal probability to propose the birth (resp. death) of a changepoint (CP) during the RJ-MCMC iterations (optional, default: cCP=0.5).

cEdges — Maximal probability - when a move update of the network topology is chosen - to propose ecah edge move (birth or death of an edge) within a temporal segment (optional, default: cEdges=0.5).

alphaCP — Hyperparameter for sampling the number k of CPs. k follows a Gamma distribution: Gamma(alphaCP, betaCP (see below)) (optional, default: alphaCP=1). Note that function choosePriors can be used to choose alphaCP (or betaCP) according to the desired dimension penalisation.

betaCP — Hyperparameter for sampling the number k of CPs. k follows a Gamma distribution: Gamma(alphaCP (see before), betaCP) (optional, default: betaCP=0.5). Note that function choosePriors can be used to choose betaCP (or alphaCP) according to the desired dimension penalisation.

alphaEdges — Hyperparameter for sampling the number l of regulatory interactions between the target gene and the parent genes. l follows a Gamma distribution: rgamma(1, shape=alphaEdges, rate=betaEdges) (see betaEdges below), (default: alphaEdges=1).

Function choosePriors can be used to choose alphaEdges (or betaEdges) according to the desired dimension penalisation.

betaEdges — Hyperparameter for sampling the number l of regulatory interactions between the target gene and the parent genes. l follows a Gamma distribution: rgamma(1, shape=alphaEdges, rate=betaEdges) (see alphaEdges upper) (optional, default: betaEdges=0.5). Function choosePriors can be used to choose betaEdges (or alphaEdges) according to the desired dimension penalisation.

v0 — Hyperparameter for sampling the noise variance (denoted by Sig2) in the auto-regressive model defining the regulatory time vrarying network. The prior distribution of the noise variance is an Inverse Gamma distribution with shape parameter v0/2 and scale parameter gamma0/2: rinvgamma(1,shape=v0/2, scale = gamma0), (optional, default: v0=1).

gamma0 — Hyperparameter for sampling the noise variance (denoted by Sig2) in the ARTIVA package) in the auto-regressive model defining the regulatory time vrarying network. The prior distribution of the noise variance is an Inverse Gamma distribution with shape parameter v0/2 and scale parameter gamma0/2: rinvgamma(1, shape=v0/2, scale = gamma0), (optional, default: gamma0=0.1).

alphad2 — Hyperparameter for sampling a parameter that represents the expected signal-to-noise ratio (denoted by delta2 in the ARTIVA package). It is sampled according to an Inverse Gamma distribution: rinvgamma(1, shape=alphad2, scale=betad2), (optional, default: alphad2=2).

betad2 — Hyperparameter for sampling a parameter that represents the expected signal-to-noise ratio (denoted by delta2 in the ARTIVA package). It is sampled according to an Inverse Gamma distribution: rinvgamma(1, shape=alphad2, scale=betad2), (optional, default: betad2=0.2).

silent — Boolean, if TRUE messages are printed along the ARTIVA procedure (optional, default: silent=FALSE).

Value

Samples — Results obtained at each iteration of the RJ-MCMC procedure. Samples is list composed of the following elements:

1) Samples$CP: a matrix with in row the different iterations performed with the ARTIVAsubnet function and in column the identified positions for CPs (if the parameter dyn=1, first CP=2 and final CP=n+1, with n the number of time points);

2) Samples$Edges: a matrix with in row the different ARTIVA iterations and in column the number of regulatory interactions identified in each temporal phases (-1 values are default values, when no temporal phase exist);

3) Samples$coeff: a matrix with in row the different ARTIVA iterations and in column the coefficient values corresponding to the identified regulatory interactions.

4) Samples$variance: a matrix with in row the different ARTIVA iterations and in column the variance values modelling data noise for each identified temporal phase.

`Counters` Results obtained at each iteration of the RJ-MCMC procedure. `Counters` is list composed of the following elements:

1) `Counters$CPMovesCount`: Number of modifications PROPOSED during ARTIVA iterations in term of CPs (i.e. birth of a new CP, death of an existing CP, move of an existing CP or upate of regulatory models in each temporal phases).

2) `Counters$CPMovesAcceptationPrct`: Percentage of modifications ACCEPTED during ARTIVA iterations in term of CPs (i.e. birth of a new CP, death of an existing CP, move of an existing CP or upate of regulatory models in each temporal phases).

3) `Counters$EdgesMoveCount`: Number of modifications PROPOSED during ARTIVA iterations in term of regulatory models (i.e. birth of a new edge between parent and target genes, death of an existing edge or update of the regression coefficient for existing edges).

4) `Counters$EdgesMovesAcceptationPrct`: Percentage of modifications ACCEPTED during ARTIVA iterations in term of regulatory models (i.e. birth of a new edge between parent and target genes, death of an existing edge or update of the regression coefficient for existing edges).

5) `Counters$iterations`: Total number of iterations generated by the ARTIVA procedure.

6) `Counters$rcvgce`: (only if `PSRFactor=TRUE`) Number of iterations before the stopping criterion is reached, i.e. before the PSRF factor is below the threshold specified with argument `PSRF_thres`. The returned value is NULL if the stopping criterion is not reached.

`CPpostDist` A list of 2 tables :

1) `CPpostDist$CPnumberPostDist`: A table containing the approximated distribution for the number of CPs.

2)`CPpostDist$CPpositionPostDist`: A table containing the approximated distribution for the position of the CPs.

`nbSegs` An integer equal to the number of temporal segments with the largest value observed in the posterior distribution (see previously `CPnumberPostDist`).

`SegmentPostDist`

(only when parameter `segmentAnalysis=TRUE`) A list of tables:

1) `SegmentPostDist$CPpos`: A table containing the most significant CP positions that delimit `nbSegs` temporal segments, according to `CPpostDist$CPnumber`, `CPpostDist$CPposition` and `segMinLength` (if parameter `dyn=1`, first CP is `2` and final CP is `n+1`, where `n` is the number of time points).

2) `SegmentPostDist$edgesPostDist`: A table containing the approximate posterior distribution for the incoming edges (regulatory interaction between parent and target genes) for each temporal segment delimited by the CP given in `SegmentPostDist$CPpos` (see previously). Each raw corresponds to a segment, ordered by time.

3) `SegmentPostDist$edgesCoeff` A table containing the estimated coefficients for the incoming edges (regulatory interaction between parent and target genes) for each temporal segment delimited by the CP given in `SegmentPostDist$CPpos` (see previously). Each raw corresponds to a segment, ordered by time.

`network` A table containing the information to plot (see function `traceNetworks`) the network estimated with the `ARTIVAsubnet` procedure.

GLOBvar	A list of parameters used in the ARTIVAsubnet procedure
HYPERvar	A list of hyperparameters used in the ARTIVAsubnet procedure
OUTvar	A list of output parameters used in the ARTIVAsubnet procedure.
targetData	targetData vector given as input of the ARTIVAsubnet procedure.
parentData	parentData matrix given as input of the ARTIVAsubnet procedure.

Author(s)

S. Lebre and G. Lelandais

References

S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais (2010) Statistical inference of the time-varying structure of gene-regulation networks BMC Systems Biology, 4:130.

Gelman, A. and D. Rubin (1992) Inference from iterative simulation using multiple sequences. Statistical science 7 (4), 457-472.

See Also

ARTIVAnet,ARTIVAsubnetAnalysis, choosePriors, CP.postDist, plotCP.postDist, segmentModel.postDist, traceNetworks

Examples

```
# Load the ARTIVA R package
library(ARTIVA)

# Load the dataset with simulated gene expression profiles
data(simulatedProfiles)

# Name of the target gene to be analyzed with ARTIVA
targetGene = 1

# Names of the parent genes (typically transcription factors)
parentGenes = c("TF1", "TF2", "TF3", "TF4", "TF5")

###
# ARTIVA analysis searching for potential interactions between the target
# genes and a predefined list of parent genes.
###

# Note that the number of iterations in the RJ-MCMC sampling is reduced
# to in this example to 'niter=20000' (in order obtain a quick overview
# of the ARTIVAnet fonction, but it should be increased (e.g. up to
# 50000) for a better parameter estimation.

# Without saving the output pictures "savePictures=FALSE"
## Not run:
ARTIVAtest = ARTIVAsubnet(targetData = simulatedProfiles[targetGene,],
  parentData = simulatedProfiles[parentGenes,],
```

```
  targetName = targetGene,
  parentNames = parentGenes,
  segMinLength = 2,
  edgesThreshold = 0.5,
  niter = 20000,
  savePictures=FALSE)

# By default, the output results (pictures and estimation values) are
# saved in a folder named "ARTIVAsubnet" created in the current directory
ARTIVAtest = ARTIVAsubnet(targetData = simulatedProfiles[targetGene,],
  parentData = simulatedProfiles[parentGenes,],
  targetName = targetGene,
  parentNames = parentGenes,
  segMinLength = 2,
  edgesThreshold = 0.5,
  niter = 20000)

# In order to save the results in a specific folder, for example a
# folder entitled "ARTIVA-test" in the current directory:

ARTIVAtest2 = ARTIVAsubnet(targetData = simulatedProfiles[targetGene,],
  parentData = simulatedProfiles[parentGenes,],
  targetName = targetGene,
  parentNames = parentGenes,
  segMinLength = 2,
  edgesThreshold = 0.5,
  niter = 20000,
  outputPath = "ARTIVA-test")

## End(Not run)
```

`ARTIVAsubnetAnalysis` *Function to estimate a regulatory time-varying network from the output of function ARTIVAsubnet.*

Description

This function estimates a regulatory time-varying network from the output of function `ARTIVAsubnet`. A graphical representation in a pdf file and estimated values are provided in text files. This function is used in function `ARTIVAsubnet` when parameter `segmentAnalysis=TRUE`. This function can be used separately for re-computing a time-varying network from the output of function `ARTIVAsubnet` with new analysis parameters `segMinLength`, `edgesThreshold`, `CPpos`, `layout`, ... see detail below.

Usage

```
ARTIVAsubnetAnalysis(ARTIVAsubnet=NULL, CPpostDist=NULL, CPsamples=NULL,
coefSamples=NULL, TFnumber=NULL, segMinLength=2, edgesThreshold=0.5,
burn_in=NULL, CPpos=NULL,targetData=NULL, parentData=NULL,
```

```
targetName=NULL,parentNames=NULL, savePictures=TRUE,saveEstimations=TRUE,
outputPath=NULL,layout="fruchterman.reingold", silent=FALSE,
inARTIVAsubnet=FALSE , onepage= FALSE)
```

Arguments

ARTIVAsubnet	Ouput of function ARTIVAsubnet, a list containing Samples, Counters, CPpostDist, nbSegs, SegmentPostDist, network, ... (optional, default: ARTIVAsubnet=NULL, if ARTIVAsubnet=NULL then parameters CPpostDist, CPsamples, coefSamples, TFnumber must be not null.
CPpostDist	A list of 2 tables : 1)CPpostDist$CPnumberPostDist: A table containing the distribution for the number of CPs approximated with ARTIVAsubnet. 2)CPpostDist$CPpositionPostDi A table containing the distribution for the position of the CPs approximated with ARTIVAsubnet. (optional, default: CPpostDist=NULL, but CPpostDist must be given when parameter ARTIVAsubnet=NULL)
CPsamples	A matrix with the different iterations (in row) performed with the ARTIVAsubnet function and in column the identified positions for CPs. (optional, default: CPsamples=NULL, but CPsamples must be given when parameter ARTIVAsubnet=NULL)
coefSamples	A matrix with the different (in row)performed with the ARTIVAsubnet function and in column the coefficient values corresponding to the identified regulatory interactions. (optional, default: coefSamples=NULL, but coefSamples must be given when parameter ARTIVAsubnet=NULL)
TFnumber	Number of parent genes in the data parentData used in the ARTIVAsubnet function. (optional, default: TFnumber=NULL, but TFnumber must be given when parameter ARTIVAsubnet=NULL)
segMinLength	Minimal length (number of time points) to define a temporal segment. Must be - strictly - greater than 1 if there is no repeated measurements for each time point in arguments targetData and parentData (optional, default: segMinLength=2).
edgesThreshold	Probability threshold for the selection of the edges of the time-varying regulatory network when segmentAnalysis=TRUE (optional, default: edgesThreshold=0.5).
burn_in	Number of initial iterations that are discarded for the estimation of the model distribution (posterior distribution). The ARTIVAsubnet function is a RJ-MCMC algorithm which, at each iteration, randomly samples a new configuration of the time-varying regulatory network from probability distributions based on constructing a Markov chain that has the network model distribution as its equilibrium distribution (The equilibrium distribution is obtained when the Markov Chain converges, which requires a large number of iterations). Typically, initial iterations are notconfident because the Markov Chain has not stabilized. The burn-in samples allow to not consider these initial iterations in the final analysis (optional, default: burn_in=NULL, if burn_in=NULL then the first 25% of the iterations is left for burn_in).
CPpos	A table containing the desired most significant CP positions (optional, default: CPpos=NULL, if CPpos=NULL then CPpos is evaluated as in CP.postDist function.)
targetData	A vector with the temporal gene expression measurements for the target gene (i.e. the gene whose regulation factors are looked for). (optional, default: targetData=NULL, if not null then the target data is plotted).

parentData — A matrix (or a vector if only 1 parent gene) with the temporal gene expression measurements for the proposed parent genes (i.e. potential regulation factors). Parent genes are shown in row and expression values in column. (optional, default: parentData=NULL, if not null then the parent data is plotted).

targetName — Name of the target gene (optional, default: targetName="Target").

parentNames — A vector with the names for parent gene(s) (optional, default: parentNames=NULL).

savePictures — Boolean, if TRUE all estimated posterior distributions and networks are plotted in a pdf file either in a new sub folder named "ARTIVA_Results" created by default in the current folder or in a folder specified with argument outputPath (see below) (optional, default: savePictures=TRUE).

saveEstimations — Boolean, if TRUE all estimated posterior distributions are saved as text files either in a new sub folder named "ARTIVA_Results" created by default in the current folder or in a folder specified with argument outputPath (see below) (optional, default: saveEstimations=TRUE).

outputPath — File path to a folder in which the output results have to be saved, either a complete path or the name of a folder to be created in the current directory (optional, default: outputPath=NULL).

layout — Name of the function determining the placement of the vertices for drawing a graph, possible values among others: "random", "circle", "sphere", "fruchterman.reingold", "kamada.kawai", "spring", "reingold.tilford", "fruchterman.reingold.grid", see package igraph0 for more details (default: layout="fruchterman.reingold").

silent — Boolean, if TRUE messages are printed along the ARTIVA procedure (optional, default: silent=FALSE).

inARTIVAsubnet — Boolean, if TRUE, general information already printed in function ARTIVAsubnet are not printed a second time (optional, default: inARTIVAsubnet=FALSE).

onepage — Boolean, if TRUE, all output pictures are plotted on one page only (optional, default: onepage=FALSE.

Value

nbSegs — An integer equal to the number of temporal segments with the largest value observed in the CP number posterior distribution (from CPpostDist$CPnumberPostDist).

CPposition — A table containing the most significant CP positions that delimit nbSegs temporal segments, according to CPnumber, CPposition and segMinLength (if parameter dyn=1, first CP is 2 and final CP is n+1, where n is the number of time points).

SegmentPostDist — Output of function segmentModel.postDist. A list of tables:

1) SegmentPostDist$CPpos: A table containing the most significant CP positions that delimit nbSegs temporal segments, according to CPpostDist$CPnumber, CPpostDist$CPposition and segMinLength (if parameter dyn=1, first CP is 2 and final CP is n+1, where n is the number of time points).

2) SegmentPostDist$edgesPostDist: A table containing the approximate posterior distribution for the incoming edges (regulatory interaction between parent and target genes) for each temporal segment delimited by the CP given in SegmentPostDist$CPpos (see previously). Each raw corresponds to a segment, ordered by time.

3) SegmentPostDist$edgesCoeff A table containing the estimated coefficients for the incoming edges (regulatory interaction between parent and target genes) for each temporal segment delimited by the CP given in SegmentPostDist$CPpos (see previously). Each raw corresponds to a segment, ordered by time.

network — A table containing the information to plot (see function traceNetworks) the network estimated with the ARTIVAsubnet procedure.

Author(s)

S. Lebre and G. Lelandais

References

S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais (2010) Statistical inference of the time-varying structure of gene-regulation networks BMC Systems Biology, 4:130.

See Also

ARTIVAsubnet, ARTIVAnet, traceNetworks, traceGeneProfiles, CP.postDist, plotCP.postDist.

Examples

```
# Load the ARTIVA R package
library(ARTIVA)

# Load the dataset with simulated gene expression profiles
data(simulatedProfiles)

# Name of the target gene to be analyzed with ARTIVA
targetGene = 1

# Names of the parent genes (typically transcription factors)
parentGenes = c("TF1", "TF2", "TF3", "TF4", "TF5")


# Note that the number of iterations in the RJ-MCMC sampling is reduced
# to 'niter=20000' in this example, but it should be increased (e.g. up to
# 50000) for a better estimation.

# Run the ARTIVAsubnet function
## Not run:
ARTIVAtest = ARTIVAsubnet(targetData = simulatedProfiles[targetGene,],
  parentData = simulatedProfiles[parentGenes,],
  targetName = targetGene,
  parentNames = parentGenes,
  segMinLength = 2,
```

```
  edgesThreshold = 0.6,
  niter= 20000,
  savePictures=FALSE)

# Re-compute a time-varying network from the output of function
# ARTIVAsubnet with new analysis parameters
analysis2 = ARTIVAsubnetAnalysis(ARTIVAsubnet=ARTIVAtest,
  segMinLength = 3,
  edgesThreshold = 0.5,
  outputPath="ARTIVAsubnet2",
  savePictures=FALSE)

# Trace the obtained network.
traceNetworks(analysis2$network, edgesThreshold = 0.3)

## End(Not run)
```

`choosePriors` *Function to plot an overview of possible priors for the number of changepoints and/or edges.*

Description

Plots an overview of some possible priors used in the `ARTIVAsubnet` function for the number of changepoints (resp. incoming edges) according to a given number of maximum changepoints `maxCP` (resp. incoming Edges `maxPred`) when parameters (`alphaCP`, `betaCP` for the CPs or `alphaEdges`, `betaEdges` for the edges) in function `ARTIVAsubnet` are set to default (alpha=1, beta=0.5). In the `ARTIVAsubnet` procedure, the number of CPs (respectively the number of incoming edges) is sampled from a truncated Poisson with mean lambda, where lambda is drawn from an Inverse Gamma distribution (alpha, beta), see Lebre et al. (2010) for more details.

Usage

```
choosePriors(kmax,priors)
```

Arguments

`kmax`	Maximum number of changepoints or incomins Edges (parents)
`priors`	Table describing the priors which can be loaded with data(priors)

Value

NULL, a graph is plotted.

Author(s)

S. Lebre and G. Lelandais.

References

S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais (2010) Statistical inference of the time-varying structure of gene-regulation networks, BMC Systems Biology, 4:130.

See Also

ARTIVAsubnet, ARTIVAnet

Examples

```
# See some prior probabbility density when the maximal number of
# CPs/parents is equal to 5
data(priors)
choosePriors(kmax=5,priors)
```

`CP.postDist` *Function to compute the CPs posterior distribution for the ARTIVA network model from the the ouput samples of function ARTIVAsubnet.*

Description

Using the ouput RJ-MCMC samples of functions ARTIVAsubnet, this function estimates posterior distributions for the number of CPs and their position.

Usage

```
CP.postDist(CPsamples, burn_in=NULL, segMinLength=2)
```

Arguments

`CPsamples`	A matrix with the different iterations (in row) performed with the ARTIVAsubnet function and in column the identified positions for CPs.
`burn_in`	Number of initial iterations that are discarded for the estimation of the model distribution (posterior distribution). The ARTIVAsubnet function is a RJ-MCMC algorithm which, at each iteration, randomly samples a new configuration of the time-varying regulatory network from probability distributions based on constructing a Markov chain that has the network model distribution as its equilibrium distribution (The equilibrium distribution is obtained when the Markov Chain converges, which requires a large number of iterations). Typically, initial iterations are notconfident because the Markov Chain has not stabilized. The burn-in samples allow to not consider these initial iterations in the final analysis (optional, default: `burn_in=NULL`, if `burn_in=NULL` then the first 25% of the iterations is left for `burn_in`).
`segMinLength`	Minimal length (number of time points) to define a temporal segment. Must be - strictly - greater than 1 if there is no repeated measurements for each time point in arguments `targetData` and `parentData` (optional, default: `segMinLength=2`).

Value

A list of 4 elements:

1) `CPnumber`: a table containing the approximate posterior distribution for the number of CPs.

2) `CPposition`: a table containing the approximate posterior distribution for the CPs position.

3) `estimatedCPnumber`: number of CP position with the greatest posterior probability according to the approximate posterior distribution for the number of CPs `CPnumber`.

4) `estimatedCPpos`: a table containing the `estimatedCPnumber` most significant CP positions according to `CPnumber`, `CPposition` and `segMinLength` (if parameter `dyn=1`, first CP is `2` and final CP is `n+1`, where `n` is the number of time points).

Author(s)

S. Lebre and G. Lelandais.

References

Statistical inference of the time-varying structure of gene-regulation networks S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais, BMC Systems Biology, 4:130, 2010.

See Also

`ARTIVAsubnet`, `ARTIVAnet`, `plotCP.postDist`, `ARTIVAsubnetAnalysis`

Examples

```
# Load the ARTIVA R package
library(ARTIVA)

# Load the dataset with simulated gene expression profiles
data(simulatedProfiles)

# Name of the target gene to be analyzed with ARTIVA
targetGene = 1

# Names of the parent genes (typically transcription factors)
parentGenes = c("TF1", "TF2", "TF3", "TF4", "TF5")

# run ARTIVAsubnet

# Note that the number of iterations in the RJ-MCMC sampling is reduced
# to 'niter=20000' in this example, but it should be increased (e.g. up to
# 50000) for a better estimation.
## Not run:
ARTIVAtest = ARTIVAsubnet(targetData = simulatedProfiles[targetGene,],
  parentData = simulatedProfiles[parentGenes,],
  targetName = targetGene,
  parentNames = parentGenes,
  segMinLength = 2,
  edgesThreshold = 0.6,
  niter= 20000,
```

```
  savePictures=FALSE)

# compute the PC posterior distribution with other parameters
outCPpostDist = CP.postDist(ARTIVAtest$Samples$CP, burn_in=10000,
    segMinLength=3)

# plot the CP posterior distribution
plotCP.postDist(outCPpostDist, targetName=paste("Target", targetGene),
  estimatedCPpos=outCPpostDist$estimatedCPpos)

## End(Not run)
```

`drosophila` *Drosphila life cycyle time series by Arbeitman et al 2002.*

Description

Gene expression time series for 4028 genes involved in the life cycle of Drosophila melanogaster. The microarray data measured gene expression levels during all four major stages of morphogenesis: embryo, larva, pupa and adult (67 time points). Data published in Arbeitman et al. (2002) and used in Lebre et al. (2010)

Usage

```
data(drosophila)
```

Format

A matrix of 4028 row (genes) by 67 columns (timepoints) containing the gene expression time series.

Source

The data has been published in Arbeitman et al (2002).

References

Gene expression during the life cycle of Drosophila melanogaster. M.N Arbeitman, E.E.M. Furlong, F. Imam, E. Johson, B.H. Null, B.S. Baker, M.A. Krasnow, M.P. Scott, R.W. Davis and K.P. White. Science, 297(5590):2270-2275, 2002.

Statistical inference of the time-varying structure of gene-regulation networks S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais, BMC Systems Biology, 2010, 4:130.

Examples

```
#load the data
data(drosophila)

#plot time serie for a chosen gene
gene=1
plot(1:67,drosophila[gene,],type="l",xlab="Timepoints", ylab="LogRatio",
main=row.names(drosophila)[gene])
```

`geneNetworkSummary` *Function to*

Description

This function is used for printing a summary of the gene network estimated with the ARTIVA procedure (`ARTIVAnet`, `ARTIVAsubnet`) for Auto Regressive TIme-VArying network inference.

Usage

```
geneNetworkSummary(ARTIVAnet, edgesThreshold)
```

Arguments

`ARTIVAnet`	Table containing the information to plot a time-varying regulatory network. In particular, this table can be obtained with function `ARTIVAsubnet`, `ARTIVAsubnetAnalysis` (output value `network`) or `ARTIVAnet` (unique output value). Each row of the table describes one edge. The columns, entitled Target, CPini, CPfinal, Parent, PostProb, describe the name of the target gene, the changepoints defining the start and the end of the regulation, the parent name and the estimated posterior probability of the edge.
`edgesThreshold`	Probability threshold for the selection of the edges to be plotted.

Value

NULL

Author(s)

Original version by S. Lebre and G. Lelandais, contribution of D. Servillo to the final version.

References

Statistical inference of the time-varying structure of gene-regulation networks S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais, BMC Systems Biology, 4:130, 2010.

See Also

ARTIVAnet,ARTIVAsubnet, ARTIVAsubnetAnalysis, CP.postDist, segmentModel.postDist, plotCP.postDist

Examples

```
# Load the ARTIVA R package
library(ARTIVA)

# Load the dataset with simulated gene expression profiles
data(simulatedProfiles)

# Name of the target gene to be analyzed with ARTIVA
targetGene = 1

# Names of the parent genes (typically transcription factors)
parentGenes = c("TF1", "TF2", "TF3", "TF4", "TF5")


# Run the ARTIVAsubnet function
# Note that the number of iterations in the RJ-MCMC sampling is reduced
# to 'niter=20000' in this example, but it should be increased (e.g. up to
# 50000) for a better estimation.

## Not run:
ARTIVAtest = ARTIVAsubnet(targetData = simulatedProfiles[targetGene,],
  parentData = simulatedProfiles[parentGenes,],
  targetName = targetGene,
  parentNames = parentGenes,
  segMinLength = 2,
  edgesThreshold = 0.6,
  niter= 2000,
  savePictures=FALSE)

# Print a summary of the obtained network
geneNetworkSummary(ARTIVAtest$network, edgesThreshold = 0.3)

# List of target genes to be analyzed independantly with ARTIVA
targetGenes = c("TF3", 45, 50)
ARTIVAtest2 = ARTIVAnet(targetData = simulatedProfiles[targetGenes,],
  parentData = simulatedProfiles[parentGenes,],
  targetName = targetGenes,
  parentNames = parentGenes,
  segMinLength = 2,
  edgesThreshold = 0.6,
  niter= 2000,
  savePictures=FALSE)

# Print a summary of the obtained network
geneNetworkSummary(ARTIVAtest2, edgesThreshold = 0.3)

# Re-compute a time-varying network from the output of function
```

```
# ARTIVAsubnet with new analysis parameters
analysis2 = ARTIVAsubnetAnalysis(ARTIVAsubnet=ARTIVAtest,
  segMinLength = 3,
  edgesThreshold = 0.5,
  outputPath="ARTIVAsubnet2",
  savePictures=FALSE)

# Print a summary of the network obtained with the 2nd analysis.
geneNetworkSummary(analysis2$network, edgesThreshold = 0.3)

## End(Not run)
```

`plotCP.postDist` *Function to plot the estimated posterior distribution for the changepoints (CPs) number and position*

Description

This function is used for plotting the estimated changepoint number and position posterior distribution after running the ARTIVA procedure (function `ARTIVAsubnet`) for Auto Regressive TIme-VArying network inference.

Usage

```
plotCP.postDist(CPpostDist, targetName = NULL, onepage = TRUE,
color1 = "green", color2 = "black", estimatedCPpos=NULL)
```

Arguments

`CPpostDist`	A list of 2 tables : 1)`CPpostDist$CPnumberPostDist`: A table containing the distribution for the number of CPs approximated with `ARTIVAsubnet`. 2)`CPpostDist$CPpositionPostDi` A table containing the distribution for the position of the CPs approximated with function `ARTIVAsubnet` or `CP.postDist`
`targetName`	Name of the target gene (optional, default: `targetName=NULL`).
`onepage`	Boolean, if TRUE the two estimated posterior distributions are plotted in one window next to each other (optional, default: `mfrow=TRUE`).
`color1`	Color for plotting the estimated posterior distribution for the changepoints (CPs) number (default: `color1="green"`).
`color2`	Color for plotting the estimated posterior distribution for the changepoints (CPs) position (default: `color2="black"`).
`estimatedCPpos`	CP positions to be highlighted as most significant, e.g. CP positions estimated with function `CP.postDist` (optional, default: `estimatedCPpos=NULL`, if `estimatedCPpos=NULL` then the number of highlighted CPs is the maximum of `CPpostDist$CPnumberPostDist` and the positions are the top best of `CPpostDist$CPpositionPostDist`).

Value

NULL

Author(s)

S. Lebre and G. Lelandais.

References

Statistical inference of the time-varying structure of gene-regulation networks S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais, BMC Systems Biology, 4:130, 2010.

See Also

ARTIVAnet, ARTIVAsubnet, CP.postDist, segmentModel.postDist, ARTIVAsubnetAnalysis

Examples

```
# Load the ARTIVA R package
library(ARTIVA)

# Load the dataset with simulated gene expression profiles
data(simulatedProfiles)

# Name of the target gene to be analyzed with ARTIVA
targetGene = 1

# Names of the parent genes (typically transcription factors)
parentGenes = c("TF1", "TF2", "TF3", "TF4", "TF5")

# run ARTIVAsubnet
# Note that the number of iterations in the RJ-MCMC sampling is reduced
# to 'niter=20000' in this example, but it should be increased (e.g. up to
# 50000) for a better estimation.

## Not run:
ARTIVAtest = ARTIVAsubnet(targetData = simulatedProfiles[targetGene,],
  parentData = simulatedProfiles[parentGenes,],
  targetName = targetGene,
  parentNames = parentGenes,
  segMinLength = 2,
  edgesThreshold = 0.6,
  niter= 20000,
  savePictures=FALSE)

# compute the PC posterior distribution with other parameters
outCPpostDist = CP.postDist(ARTIVAtest$Samples$CP, burn_in=500,
    segMinLength=3)

# plot the CP posterior distribution
plotCP.postDist(outCPpostDist, targetName=paste("Target", targetGene),
  estimatedCPpos=outCPpostDist$estimatedCPpos)


## End(Not run)
```

priors — *Set of possible priors for the number of changepoints (CPs) or incoming edges.*

Description

Set of possible priors used in the ARTIVAsubnet function for the number of changepoints (resp. incoming edges) according to a given number of maximum changepoints maxCP (resp. incoming Edges maxPred) when parameters (alphaCP, betaCP for the CPs or alphaEdges, betaEdges for the edges) in function ARTIVAsubnet are set to default (alpha=1, beta=0.5). In the ARTIVAsubnet procedure, the number of CPs (respectively the number of incoming edges) is sampled from a truncated Poisson with mean lambda, where lambda is drawn from an Inverse Gamma distribution (alpha, beta), see Lebre et al. (2010) for more details.

Usage

```
data(priors)
```

Format

A matrix of 96 rows by 44 columns (kmax, alpha, beta and the probability for k=0 to 40 according to the chosen values of kmax, alpha and beta).

References

Statistical inference of the time-varying structure of gene-regulation networks S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais, BMC Systems Biology, 2010, 4:130.

See Also

choosePriors,ARTIVAsubnet, ARTIVAnet

Examples

```
# See some prior probability density when the maximal number of
# CPs/parents is equal to 5
data(priors)
choosePriors(kmax=5,priors)
```

simulatedProfiles *Simulated gene expression profiles dataset.*

Description

Simulated gene expression time series for 55 genes and 30 timepoints generated as in Lebre et al (2010). In the simulation model, every gene may be target gene but only genes 51 to 55 are parent genes.

Usage

```
data(simulatedProfiles)
```

Format

A matrix of 55 row (genes) by 30 columns (timepoints).

References

Statistical inference of the time-varying structure of gene-regulation networks S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais, BMC Systems Biology, 2010, 4:130.

Examples

```
# Load the ARTIVA R package
library(ARTIVA)

# Load the simulated profiles dataset
data(simulatedProfiles)

# Name of the target gene to be analyzed with ARTIVA
targetGene = 1

# Names of the parent genes (typically transcription factors)
parentGenes = c("TF1", "TF2", "TF3", "TF4", "TF5")

# Plot of the gene expression profiles for target gene and parent genes
traceGeneProfiles(targetData= simulatedProfiles[targetGene,],
  parentData= simulatedProfiles[parentGenes,])

###
# ARTIVA analysis searching for potential interactions between the target
# genes and a predefined list of parent genes.
###

# Note that the number of iterations in the RJ-MCMC sampling is reduced
# to in this example to 'niter=20000' in order obtain a quick overview of
# the ARTIVAnet fonction, but it should be increased (e.g. up to 50000)
```

```
# for a better parameter estimation.
## Not run:
ARTIVAtest = ARTIVAsubnet(targetData = simulatedProfiles[targetGene,],
  parentData = simulatedProfiles[parentGenes,],
  targetName = targetGene,
  parentNames = parentGenes,
  niter = 5000,
  savePictures = FALSE)

## End(Not run)
```

---

traceGeneProfiles *Function to plot the gene expression profiles*

Description

This function is used for plotting the target and parent gene expression profiles.

Usage

```
traceGeneProfiles(targetData, parentData,  dataDescription=NULL,
targetColor = "grey", parentColor = "blue", onepage=TRUE)
```

Arguments

targetData	A matrix (or a vector if only one target gene) with the temporal gene expression measurements for the target genes (i.e. the genes whose regulation factors are looked for). Target genes are shown in row and expression values in column.
parentData	A matrix (or a vector if only 1 parent gene) with the temporal gene expression measurements for the proposed parent genes (i.e. potential regulation factors). Parent genes are shown in row and expression values in column.
dataDescription	(Required only when the gene expression measurements contain repeated values for the same time points). A vector indicating the ordering of the time measurements in the data. For example dataDescription=rep(1:n, each=2), if there are two measurements for each time point AND the repetitions for each time point are next to each other. Note that temporal gene expression measurements have to be organized identically in arguments targetData and parentData (optional, default: dataDescription=NULL).
targetColor	Color for plotting the target genes expression profiles (optional, default targetColor= "grey")
parentColor	Color for plotting the parent genes expression profiles (optional, default parentColor= "blue")
onepage	Boolean, if TRUE, all output pictures are plotted on one page only (optional, default: onepage=TRUE.

Value

NULL

Author(s)

S. Lebre and G. Lelandais.

References

Statistical inference of the time-varying structure of gene-regulation networks S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais, BMC Systems Biology, 4:130, 2010.

See Also

ARTIVAsubnet, ARTIVAnet

Examples

```
# Load the R package ARTIVA
library(ARTIVA)

# Load the dataset with simulated gene expression profiles
data(simulatedProfiles)

# Name of the target gene to be analyzed with ARTIVA
targetGene = 1

# Names of the parent genes (typically transcription factors)
parentGenes = c("TF1", "TF2", "TF3", "TF4", "TF5")

# Plot of the gene expression profiles for target gene and parent genes
traceGeneProfiles(targetData= simulatedProfiles[targetGene,],
  parentData= simulatedProfiles[parentGenes,])
```

---

traceNetworks *Function to plot the network estimated with functions ARTIVAnet or ARTIVAsubnet*

Description

This function is used for plotting the network estimated with the ARTIVA procedure (ARTIVAnet, ARTIVAsubnet) and ARTIVAsubnetAnalysis for Auto Regressive TIme-VArying network inference.

Usage

```
traceNetworks(ARTIVAnet, edgesThreshold, parentColor = "blue",
targetColor = "grey", parentgeneNames = TRUE, targetgeneNames = TRUE,
layout = "fruchterman.reingold", onepage=TRUE)
```

Arguments

ARTIVAnet	Table containing the information to plot a time-varying regulatory network. In particular, this table can be obtained with function ARTIVAsubnet, ARTIVAsubnetAnalysis (output value network) or ARTIVAnet (unique output value). Each row of the table describes one edge. The columns, entitled Target, CPini, CPfinal, Parent, PostProb, describe the name of the target gene, the changepoints defining the start and the end of the regulation, the parent name and the estimated posterior probability of the edge.
edgesThreshold	Probability threshold for the selection of the edges to be plotted.
parentColor	Color for plotting the node representing parent genes (optional, default: parentColor= "blue").
targetColor	Color for plotting the node representing target genes (optional, default: targetColor= "grey").
parentgeneNames	Boolean, if TRUE the name of the parent gene is plotted (optional, default: geneNames = TRUE).
targetgeneNames	Boolean, if TRUE the name of the target gene is plotted (optional, default: geneNames = TRUE).
layout	Name of the function determining the placement of the vertices for drawing a graph, possible values among others: "fruchterman.reingold", "geneLines", "random", "circle", "sphere", "kamada.kawai","spring", "reingold.tilford", "fruchterman.reingold.grid", see package igraph0 for more details (default: layout="fruchterman.reingold").
onepage	Boolean, if TRUE, all output pictures are plotted on one page only (optional, default: onepage=TRUE.

Value

NULL

Author(s)

Original version by S. Lebre and G. Lelandais, contribution of D. Servillo to the final version.

References

Statistical inference of the time-varying structure of gene-regulation networks S. Lebre, J. Becq, F. Devaux, M. P. H. Stumpf, G. Lelandais, BMC Systems Biology, 4:130, 2010.

See Also

ARTIVAnet,ARTIVAsubnet, ARTIVAsubnetAnalysis, CP.postDist, segmentModel.postDist, plotCP.postDist

Examples

```
# Load the ARTIVA R package
library(ARTIVA)

# Load the dataset with simulated gene expression profiles
```

```
data(simulatedProfiles)

# Name of the target gene to be analyzed with ARTIVA
targetGene = 1

# Names of the parent genes (typically transcription factors)
parentGenes = c("TF1", "TF2", "TF3", "TF4", "TF5")


# Run the ARTIVAsubnet function
# Note that the number of iterations in the RJ-MCMC sampling is reduced
# to 'niter=20000' in this example, but it should be increased (e.g. up to
# 50000) for a better estimation.

## Not run:
ARTIVAtest = ARTIVAsubnet(targetData = simulatedProfiles[targetGene,],
  parentData = simulatedProfiles[parentGenes,],
  targetName = targetGene,
  parentNames = parentGenes,
  segMinLength = 2,
  edgesThreshold = 0.6,
  niter= 2000,
  savePictures=FALSE)

# Re-compute a time-varying network from the output of function
# ARTIVAsubnet with new analysis parameters
analysis2 = ARTIVAsubnetAnalysis(ARTIVAsubnet=ARTIVAtest,
  segMinLength = 3,
  edgesThreshold = 0.5,
  outputPath="ARTIVAsubnet2",
  savePictures=FALSE)

# Trace the obtained network.
traceNetworks(analysis2$network, edgesThreshold = 0.3)

## End(Not run)
```

`yeast` *Yeast stress response data*

Description

This dataset was produced by Lucau-Danila et al. (2005). In this study the authors measured the changes in the mRNA concentrations for each gene at successive times after addition of an antimitotic drug (benomyl), in the growth media of Saccharomyces cerevisiae cells. Parallel experiments were conducted in different genetic contexts: the wild type strain and knock out (KO) strains in which the genes coding for different transcription factors connected to drug response (YAP1, PDR1, PDR3 and YRR1) were deleted. For 78 genes, the measured expression values for 5 different time points (at 30s, 2min, 4min, 10min and 20min) are available, in each yeast strains (WT, DeltaYAP1, DeltaPDR1, DeltaPDR3 and DeltaYRR1). These genes are divided into 18 clusters of genes with

condordant transcription profiles. In this context, regulatory associations between parent and target genes are proposed if the deletion of a parent gene significantly alters the expression measurements of the target genes.

Usage

```
data(yeast)
```

Format

A list that comprises information for the 18 clusters of genes whose expression is identically modified in strains deleted for YAP1, PDR1, PDR3 and YRR1 transcription factors, compared to the wild type strain. Detailed description and more information concerning these clusters can be found in Lebre et al. (2010).

`Cluster 1` Experimental data for genes that belong to the Cluster 1. `yeast$Cluster1` is a list composed of the following elements: 1) `yeast$Cluster1$InitialCluster`: A matrix with in row the genes that belong to Cluster 1 and in column the different experimental measurements related to the wild type strain (WT), and strains deleted for the transcription factor YAP1 (DeltaYAP1), PDR1 (DeltaPDR1), PDR3 (DeltaPDR3) and YRR1 (DeltaYRR1). For each genetic context, 5 time points are available T1 = 30s, T2 = 2min, T3 = 4min, T4 = 10min and T5 = 20min. 2) `yeast$Cluster1$targetData`: A vector with all gene expression measurements correctly formatted to be analyzed with the `ARTIVAsubnet` function. 3) `yeast$Cluster1$targetName`: The name of the analyzed cluster. 4) `yeast$Cluster1$GeneList`: A vector with the names of the genes that belong to the analyzed cluster. 5) `yeast$Cluster1$parentData`: A matrix with in row the four transcription factors for which the corresponding genes were deleted (independantly) in knock out strains. 1 = the gene coding for the transcription factor is present and 0 = the gene coding for the transcription factor is deleted. This matrix is correctly formatted to be analyzed with the `ARTIVAsubnet` function, therefore searching for regulatory interactions between YAP1, PDR1, PDR3 and YRR1 transcription factors and the genes that belong to the analyzed cluster. 6) `yeast$Cluster1$dataDescription` : A vector indicating the ordering of the time measurements in the `targetData` and `parentData` variables.

`Cluster2 to 18` Experimental data for genes that belong to the Cluster 2 to 18 (see the documentation for Cluster 1).

Source

Expression data were obtained from the website:

`http://www.biologie.ens.fr/lgmgml/publication/benomyl/`

References

The Early Expression of Yeast Genes Affected by Chemical Stress A. Lucau-Danila, G. Lelandais, Z. Kozovska, V. Tanty, T. Delaveau, F. Devaux and C. Jacq., Mol Cell Biol., 25(5):1860-8, 2005.

Statistical inference of the time-varying structure of gene regulation networks S. Lebre, J. Becq, F. Devaux, MP Stumpf, G. Lelandais., BMC Systems Biology 4:130, 2010.

Examples

```
####
# Datasets related to the analysis of the genomic response of the yeast
# Saccharomyces cerevisiae to an environmental stress induced by
# benomyl (a toxic compound).
# Analysis of the yeast data is presented in the original article of
# ARTIVA (Lebre et al. BMC Syst. Biol, 2010)
####

# Load the yeast dataset
data(yeast)
# This is a a list that comprises information for the 18 clusters of genes
# whose expression is identically modified in strains deleted for
# YAP1, PDR1, PDR3 and YRR1 transcription factors,
# compared to the wild type strain.

# As an illustration : analysis of one cluster
cluster=4

# Different genes in a cluster is considered as repeated measurements.
# Organisation of the different time point measurements is described in
# variable : yeast[[cluster]]$dataDescription
# Beacause of repeated measurements, the minimum segment length is set to
# segMinLength = 1.
# The parentdata is the experiment design (YAP1, PDR1, PDR3 and YRR1
# deletion) described in variable: yeast[[cluster]]$parentData
# Time delay between parent and target genes is fixed to dyn=0.
## Not run:
ARTIVAtest = ARTIVAsubnet(targetData = yeast[[cluster]]$targetData,
  targetName = yeast[[cluster]]$targetName,
  parentData = yeast[[cluster]]$parentData,
  parentNames = row.names(yeast[[cluster]]$parentData),
  dataDescription = yeast[[cluster]]$dataDescription,
  outputPath = paste("ARTIVA_Results_Cluster", cluster, sep = ""),
  dyn = 0,
  segMinLength = 1,
  edgesThreshold = 0.7,
  niter = 20000)

## End(Not run)
# Detailed results can be found in the folder named
# "ARTIVA_Results_Cluster4" (with the subfolders "Estimations" for
# detailled results of the estimated parameters and "Pictures" for
# graphical representations).
```

Index

∗Topic **DBN**
ARTIVA-package, 2
ARTIVAnet, 3
ARTIVAsubnet, 8
ARTIVAsubnetAnalysis, 15
∗Topic **datasets**
drosophila, 22
priors, 27
simulatedProfiles, 28
yeast, 32
∗Topic **graphical model**
ARTIVA-package, 2
ARTIVAnet, 3
ARTIVAsubnet, 8
ARTIVAsubnetAnalysis, 15
∗Topic **graphics**
geneNetworkSummary, 23
plotCP.postDist, 25
traceGeneProfiles, 29
traceNetworks, 30
∗Topic **inference**
ARTIVA-package, 2
ARTIVAnet, 3
ARTIVAsubnet, 8
ARTIVAsubnetAnalysis, 15
∗Topic **network inference**
ARTIVA-package, 2
ARTIVAnet, 3
ARTIVAsubnet, 8
ARTIVAsubnetAnalysis, 15
∗Topic **time series**
ARTIVA-package, 2
ARTIVAnet, 3
ARTIVAsubnet, 8
ARTIVAsubnetAnalysis, 15
∗Topic **util**
choosePriors, 19
CP.postDist, 20
geneNetworkSummary, 23
plotCP.postDist, 25
traceGeneProfiles, 29
traceNetworks, 30

ARTIVA *(*ARTIVA-package*)*, 2
ARTIVA-internal, 3
ARTIVA-package, 2
ARTIVAnet, 3, *5*, *14*, *18*, *20, 21*, *23, 24*, *26, 27*, *30, 31*
ARTIVAsubnet, *3*, *7*, 8, *10*, *12–21*, *23–27*, *30, 31*, *33*
ARTIVAsubnetAnalysis, *7*, *14*, 15, *21*, *23, 24*, *26*, *30, 31*

bdu *(*ARTIVA-internal*)*, 3
buildXY *(*ARTIVA-internal*)*, 3

choosePriors, *6, 7*, *11, 12*, *14*, 19, *27*
computePx *(*ARTIVA-internal*)*, 3
computeRho3 *(*ARTIVA-internal*)*, 3
computeRho4 *(*ARTIVA-internal*)*, 3
cp.birth *(*ARTIVA-internal*)*, 3
cp.computeAlpha *(*ARTIVA-internal*)*, 3
cp.death *(*ARTIVA-internal*)*, 3
CP.postDist, *7*, *14*, *16*, *18*, 20, *24–26*, *31*
cp.shift *(*ARTIVA-internal*)*, 3
CPposition *(*ARTIVA-internal*)*, 3

dinvgamma *(*ARTIVA-internal*)*, 3
drosophila, 22

fast.svd *(*ARTIVA-internal*)*, 3

geneNetworkSummary, 23

hasSeg *(*ARTIVA-internal*)*, 3

init *(*ARTIVA-internal*)*, 3

main *(*ARTIVA-internal*)*, 3

nsmall.svd *(*ARTIVA-internal*)*, 3

plotCP.postDist, *14*, *18*, *21*, *24*, 25, *31*
PlotFunction *(*ARTIVA-internal*)*, 3
plotSegmentModel.postDist
*(*ARTIVA-internal*)*, 3
positive.svd *(*ARTIVA-internal*)*, 3
priors, 27
pseudoinverse *(*ARTIVA-internal*)*, 3
psmall.svd *(*ARTIVA-internal*)*, 3
PSRF *(*ARTIVA-internal*)*, 3

readDataTS *(*ARTIVA-internal*)*, 3
readInput *(*ARTIVA-internal*)*, 3
rinvgamma *(*ARTIVA-internal*)*, 3

sampleBinit *(*ARTIVA-internal*)*, 3
sampleBxy *(*ARTIVA-internal*)*, 3
sampleDelta2 *(*ARTIVA-internal*)*, 3
sampleK *(*ARTIVA-internal*)*, 3
sampleParms *(*ARTIVA-internal*)*, 3
sampleSig2 *(*ARTIVA-internal*)*, 3
segment.update *(*ARTIVA-internal*)*, 3
segmentModel.postDist, *7*, *14*, *17*, *24*, *26*, *31*
segmentModel.postDist
*(*ARTIVA-internal*)*, 3
segPos *(*ARTIVA-internal*)*, 3
simulatedProfiles, 28

traceGeneProfiles, *18*, 29
traceNetworks, *13*, *14*, *18*, 30

updateSigMulti *(*ARTIVA-internal*)*, 3

yeast, 32