Package ‘GENIE3’

March 31, 2025

Type Package

Title GEne Network Inference with Ensemble of trees

Version 1.28.0

Date 2022-04-05

Author Van Anh Huynh-Thu, Sara Aibar, Pierre Geurts

Maintainer Van Anh Huynh-Thu <`vahuynh@uliege.be`>

Description This package implements the GENIE3 algorithm for inferring gene regulatory networks from expression data.

License GPL (>= 2)

LazyData TRUE

Imports stats, reshape2, dplyr

Suggests knitr, rmarkdown, foreach, doRNG, doParallel, Biobase, SummarizedExperiment, testthat, methods, BiocStyle

NeedsCompilation yes

RoxygenNote 7.1.2

VignetteBuilder knitr

biocViews NetworkInference, SystemsBiology, DecisionTree, Regression, Network, GraphAndNetwork, GeneExpression

git_url https://git.bioconductor.org/packages/GENIE3

git_branch RELEASE_3_20

git_last_commit ccc2873

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2025-03-31

Contents

`GENIE3` *GENIE3*

Description

`GENIE3` Infers a gene regulatory network (in the form of a weighted adjacency matrix) from expression data, using ensembles of regression trees.

Usage

```
GENIE3(
  exprMatrix,
  regulators = NULL,
  targets = NULL,
  treeMethod = "RF",
  K = "sqrt",
  nTrees = 1000,
  nCores = 1,
  returnMatrix = TRUE,
  verbose = FALSE
)

## S4 method for signature 'matrix'
GENIE3(
  exprMatrix,
  regulators = NULL,
  targets = NULL,
  treeMethod = "RF",
  K = "sqrt",
  nTrees = 1000,
  nCores = 1,
  returnMatrix = TRUE,
  verbose = FALSE
)

## S4 method for signature 'SummarizedExperiment'
GENIE3(
  exprMatrix,
  regulators = NULL,
  targets = NULL,
  treeMethod = "RF",
  K = "sqrt",
  nTrees = 1000,
  nCores = 1,
  returnMatrix = TRUE,
  verbose = FALSE
)

## S4 method for signature 'ExpressionSet'
GENIE3(
  exprMatrix,
```

```
  regulators = NULL,
  targets = NULL,
  treeMethod = "RF",
  K = "sqrt",
  nTrees = 1000,
  nCores = 1,
  returnMatrix = TRUE,
  verbose = FALSE
)
```

Arguments

`exprMatrix`	Expression matrix (genes x samples). Every row is a gene, every column is a sample. The expression matrix can also be provided as one of the Bioconductor classes: • `ExpressionSet`: The matrix will be obtained through exprs(exprMatrix) • `RangedSummarizedExperiment`: The matrix will be obtained through assay(exprMatrix), wich will extract the first assay (usually the counts)
`regulators`	Subset of genes used as candidate regulators. Must be either a vector of gene names, e.g. `c("at_12377", "at_10912")` or a vector of indices, e.g. `c(1,5,6,7)`. The default value NULL means that all the genes are used as candidate regulators (which is NOT recommended). To provide different regulators for each gene, providethem as named list.
`targets`	Subset of genes to which potential regulators will be calculated. Must be either a vector of indices, e.g. `c(1,5,6,7)`, or a vector of gene names, e.g. `c("at_12377", "at_10912")`. If NULL (default), regulators will be calculated for all genes in the input matrix.
`treeMethod`	Tree-based method used. Must be either "RF" for Random Forests (default) or "ET" for Extra-Trees.
`K`	Number of candidate regulators randomly selected at each tree node (for the determination of the best split). Must be either "sqrt" for the square root of the total number of candidate regulators (default), "all" for the total number of candidate regulators, or a stricly positive integer.
`nTrees`	Number of trees in an ensemble for each target gene. Default: 1000.
`nCores`	Number of cores to use for parallel computing. Default: 1.
`returnMatrix`	Returns output as weight matrix (TRUE). Otherwise (FALSE) it is returned as a list.
`verbose`	If set to TRUE, a feedback on the progress of the calculations is given. Default: FALSE.

Value

Weighted adjacency matrix of inferred network. Element w_ij (row i, column j) gives the importance of the link from regulatory gene i to target gene j.

Examples

```
## Generate fake expression matrix
exprMatrix <- matrix(sample(1:10, 100, replace=TRUE), nrow=20)
rownames(exprMatrix) <- paste("Gene", 1:20, sep="")
```

```
colnames(exprMatrix) <- paste("Sample", 1:5, sep="")

## Run GENIE3
set.seed(123) # For reproducibility of results
weightMatrix <- GENIE3(exprMatrix, regulators=paste("Gene", 1:5, sep=""))

## Get ranking of edges
linkList <- getLinkList(weightMatrix)
head(linkList)

## Different regulators for each gene & return as list
regulatorsList <- list("Gene1"=rownames(exprMatrix)[1:10],
                       "Gene2"=rownames(exprMatrix)[10:20],
                       "Gene20"=rownames(exprMatrix)[15:20])

set.seed(123)
weightList <- GENIE3(exprMatrix, nCores=1, targets=names(regulatorsList), regulators=regulatorsList, returnMa
```

`getLinkList` *getLinkList*

Description

`getLinkList` Converts the weight matrix, as returned by `GENIE3`, to a sorted list of regulatory links (most likely links first).

Usage

```
getLinkList(weightMatrix, reportMax = NULL, threshold = 0)
```

Arguments

`weightMatrix`	Weighted adjacency matrix as returned by `GENIE3`.
`reportMax`	Maximum number of links to report. The default value NULL means that all the links are reported.
`threshold`	Only links with a weight equal or above the threshold are reported. Default: threshold = 0, i.e. all the links are reported.

Value

List of regulatory links in a data frame. Each line of the data frame corresponds to a link. The first column is the regulatory gene, the second column is the target gene, and the third column is the weight of the link.

See Also

`GENIE3`

Examples

```
## Generate fake expression matrix
exprMat <- matrix(sample(1:10, 100, replace=TRUE), nrow=20)
rownames(exprMat) <- paste("Gene", 1:20, sep="")
colnames(exprMat) <- paste("Sample", 1:5, sep="")

## Run GENIE3
weightMat <- GENIE3(exprMat, regulators=paste("Gene", 1:5, sep=""))

## Get ranking of edges
linkList <- getLinkList(weightMat)
head(linkList)
```

Index