Package 'SacoGraci'

August 29, 2022

Type Package

Title SacoGraci, a data-driven optimization method for coarse-graining gene regulatory networks

Version 1.0

Author Cristian Caranica, Mingyang Lu

Maintainer Cristian Caranica <c.caranica@northeastern.edu>

Description SacoGraci R package for gene network coarse-graining

License MIT + file LICENSE

Encoding UTF-8

LazyData true

VignetteBuilder knitr

Imports sRACIPE, igraph, doParallel, visNetwork, SummarizedExperiment, foreach, parallel, utils, graphics, stats

Suggests knitr

RoxygenNote 7.1.2

R topics documented:

`adjMatCGrProb` *Builds the adjacency matrix with probabilities*

Description

Builds the adjacency matrix with probabilities

Usage

```
adjMatCGrProb(data_top, gene_list)
```

Arguments

`data_top`	topology of the full network
`gene_list`	gene clustering output

Value

adj_cgr: adjacency matrix with probabilities

`centMedVarCutDistPerc` *Identify the center and radius of each model cluster*

Description

Identify the center and radius of each model cluster

Usage

```
centMedVarCutDistPerc(data, clusterRef, percThr = 0.01)
```

Arguments

`data`	gene expression matrix
`clusterRef`	cluster indices of all models
`percThr`	Threshold of permutation test

Value

: a list of (center, variance, radius(centroid & medoid)) for each cluster

convAdjTop — *Converting the adjacency matrix of a gene circuit to the topology file (Source/Target/Interaction Type)*

Description

Converting the adjacency matrix of a gene circuit to the topology file (Source/Target/Interaction Type)

Usage

```
convAdjTop(adjMatL, numbG, colN)
```

Arguments

adjMatL	adjacency matrix
numbG	the number of genes
colN	a vector of gene names

Value

data_cgr_top: a data frame of the circuit topology

convTopAdj — *Converting the topology file (Source/Target/Interaction Type) of a gene circuit to the adjacency matrix d*

Description

Converting the topology file (Source/Target/Interaction Type) of a gene circuit to the adjacency matrix d

Usage

```
convTopAdj(data_t, numbG, colN)
```

Arguments

data_t	a data frame of the circuit topology
numbG	the number of genes
colN	a vector of gene names

Value

adjMatL: adjacency matrix

delEdges_topNew — *Deleting some edges from a circuit topology while trying to maintain a connected circuit*

Description

Deleting some edges from a circuit topology while trying to maintain a connected circuit

Usage

```
delEdges_topNew(samp_top, numb_genes, numb_del, pos_del, inNodes, list_top)
```

Arguments

samp_top	the current circuit topology
numb_genes	number of genes of the circuit
numb_del	number of edges to be deleted
pos_del	indices of edges that can be deleted
inNodes	a list of input nodes, output from the function find_inout
list_top	list of sampled circuit topologies, we make sure they are not sampled again

Value

adj_fin: the modified circuit topology

difLen — *Mismatches between the sampled and reference circuit topologies*

Description

Mismatches between the sampled and reference circuit topologies

Usage

```
difLen(samp_top, ref_top)
```

Arguments

samp_top	Adjacency matrix of the sampled circuit topology
ref_top	Adjacency matrix of the reference circuit topology

Value

: number of mismatching edges

find_inout *Identify input and output nodes from a circuit topology*

Description

Input nodes: nodes with only outward edges; Output nodes: nodes with only inward edges

Usage

```
find_inout(adjMatrPr, numb_cgnodes)
```

Arguments

adjMatrPr	Adjacency matrix with probabilities of CG-circuits derived from the full network
numb_cgnodes	number of CG nodes (an output of gene clustering)

Value

resNod: a list of input nodes (1) and output nodes (2)

find_in_list *Find an element in a list*

Description

Find an element in a list

Usage

```
find_in_list(xli, n, elem)
```

Arguments

xli	the list
n	the length of the list
elem	the element to search

Value

indli: the index of the found element in the list, or 0 if not found

`find_topology` *Check whether the circuit topology has been sampled or not*

Description

Check whether the circuit topology has been sampled or not

Usage

```
find_topology(top_list, samp_top, minN, maxN)
```

Arguments

`top_list`	a list of sampled topologies
`samp_top`	the current topology
`minN`	the range of circuit topologies in the list to be searched (minimum index value)
`maxN`	the range of circuit topologies in the list to be searched (maximum index value)

Value

i: the index of the found topology in the list, or 0 if not found

`gaInitial_gen` *Generating a series of randomly generated initial CG circuit topologies*

Description

Generating a series of randomly generated initial CG circuit topologies

Usage

```
gaInitial_gen(circuit_top, gene_list, numbNewTop)
```

Arguments

`circuit_top`	the original circuit topology
`gene_list`	gene clustering output
`numbNewTop`	number of new circuit topologies to be generated

Value

resM: a list of randomly generated initial CG circuit topologies

geneClustInd *GENE CLUSTERING BY INDIVIDUAL MODELS (HCA)*

Description

GENE CLUSTERING BY INDIVIDUAL MODELS (HCA)

Usage

```
geneClustInd(logscData, numbGeneClust)
```

Arguments

logscData	simulated data after log scaling and standardization
numbGeneClust	number of gene clusters

Value

gene_list: gene clustering output

geneClustMedian *GENE CLUSTERING BY MEDIAN VALUES*

Description

GENE CLUSTERING BY MEDIAN VALUES

Usage

```
geneClustMedian(clustData, clSize, numbGeneClust)
```

Arguments

clustData	model clustering data
clSize	size of each model clusters (an output of the model clustering results)
numbGeneClust	number of gene clusters

Value

gene_list: gene clustering output

gen_heatmap_hca *Generating HCA Heatmap*

Description

Generating HCA Heatmap

Usage

```
gen_heatmap_hca(logscData)
```

Arguments

logscData	simulated data after log scaling and standardization

gen_pca_plot *Generating PCA Scatterplot*

Description

Generating PCA Scatterplot

Usage

```
gen_pca_plot(logscData)
```

Arguments

logscData	simulated data after log scaling and standardization

gen_RACIPE *RACIPE simulations*

Description

RACIPE simulations

Usage

```
gen_RACIPE(dftop, nModels, integrateStepSize = 0.02, simulationTime = 200)
```

Arguments

dftop	circuit topology
nModels	number of RACIPE models generated
integrateStepSize	step size for the ODE integration
simulationTime	simulation time of ODE for each RACIPE model

Value

logscData: simulated data after log scaling and standardization

---

initialize_topology *Build the most dense circuit topology (adj. matrix in a vector“)*

Description

Build the most dense circuit topology (adj. matrix in a vector“)

Usage

```
initialize_topology(adj_matrProb, numb_cgnodes)
```

Arguments

adj_matrProb	adjacency matrix with probabilities
numb_cgnodes	the number of CG nodes

Value

adj_Matr: the adj. matrix of the most dense topology as a vector

---

inNrepsample_topology *Circuit sampling*

Description

Circuit sampling

Usage

```
inNrepsample_topology(adj_matrPr, numb_cgnodes, inNodes, outNodes, old_top)
```

Arguments

adj_matrPr	adjacency matrix of the edge probability
numb_cgnodes	the nuymber of CG nodes
inNodes	a list of input nodes
outNodes	a list of output nodes
old_top	the last circuit topology

Value

new_top: the new circuit topology

inNWsimilarityRefCGr *Circuit scoring*

Description

Circuit scoring

Usage

```
inNWsimilarityRefCGr(
  dataRow,
  clusterRef,
  cenMedRef,
  cutOffM,
  gene_list,
  inNodes,
  topol_cgr,
  modelsCGr = 10000
)
```

Arguments

dataRow	gene expression matrix with gene in rows
clusterRef	the cluster indices of all models
cenMedRef	cluster centers
cutOffM	cluster radii
gene_list	gene clustering output
inNodes	a list of input nodes
topol_cgr	the current CG circuit topology
modelsCGr	the number of RACIPE models to be simulated (10000)

Value

scoresOuts: 1: the total score, 2: the number of noisy models

modClustHCA *Model Clustering By hierarchical clustering analysis (HCA)*

Description

Model Clustering By hierarchical clustering analysis (HCA)

Usage

```
modClustHCA(logscData, numbClust)
```

Arguments

logscData	simulated data after log scaling and standardization
numbClust	number of model clusters

Value

res: model clustering output: (1: cluster sizes; 2: clustering rearranged data; 3: cluster indices)

---

modClustKmeans *MODEL CLUSTERING BY K-MEANS*

Description

MODEL CLUSTERING BY K-MEANS

Usage

```
modClustKmeans(data, numbClust, clustCenters)
```

Arguments

data	data matrix for k-means clustering (either logscData or projected data)
numbClust	number of model clusters
clustCenters	coordinates of the cluster center

Value

res: model clustering output

---

opt_MH *Circuit optimization with Metropolis-Hastings (MH) algorithm*

Description

Circuit optimization with Metropolis-Hastings (MH) algorithm

Usage

```
opt_MH(
  network_top,
  data,
  clusterRef,
  cenMedRef,
  cutOffM,
  gene_list,
  init_top,
  output = "Results",
  nRepeat = 5,
  nIter = 1400,
  modelsCGr = 10000,
  tempM = 60
)
```

Arguments

network_top	topology of the full network
data	processed gene expression matrix
clusterRef	cluster indices of all models
cenMedRef	cluster centers
cutOffM	cluster radii
gene_list	gene clustering output
init_top	initial circuit topology
output	a string of file prefix for saving results ("Results")
nRepeat	number of repeats of RACIPE simulations for each new circuit topology (5) A new circuit is simulated by RACIPE nRepeat times for robust score evaluation; The scores will then be saved and used in future iterations, when the circuits are sampled again.
nIter	number of iterations for each simulation (1400)
modelsCGr	number of RACIPE models to be simulated (10000)
tempM	temperature for MH (60)

Value

df: topology of the optimized CG circuit

---

opt_MH_multi *Circuit optimization with Metropolis-Hastings (MH) algorithm (multiple threads)*

Description

Circuit optimization with Metropolis-Hastings (MH) algorithm (multiple threads)

Usage

```
opt_MH_multi(
  network_top,
  data,
  clusterRef,
  cenMedRef,
  cutOffM,
  gene_list,
  inTopsM,
  output = "Results",
  nRepeat = 5,
  nIter = 1400,
  modelsCGr = 10000,
  tempM = 60,
  numbThr = 40,
  nSim = 20
)
```

Arguments

network_top	topology of the full network
data	processed gene expression matrix
clusterRef	cluster indices of all models
cenMedRef	cluster centers
cutOffM	cluster radii
gene_list	gene clustering output
inTopsM	a list of all initial circuit topologies
output	a string of file prefix for saving results ("Results")
nRepeat	number of repeats of RACIPE simulations for each new circuit topology (5) A new circuit is simulated by RACIPE nRepeat times for robust score evaluation; The scores will then be saved and used in future iterations, when the circuits are sampled again.
nIter	number of iterations for each simulation (1400)
modelsCGr	number of RACIPE models to be simulated (10000)
tempM	temperature for MH (60)
numbThr	number of requested threads for HPC (40)
nSim	number of parallel simulations (20)

Value

df: topology of the optimized CG circuit

---

opt_SA *Circuit optimization with Simulated Annealing (SA) algorithm*

Description

Circuit optimization with Simulated Annealing (SA) algorithm

Usage

```
opt_SA(
  network_top,
  data,
  clusterRef,
  cenMedRef,
  cutOffM,
  gene_list,
  init_top,
  output = "Results",
  nRepeat = 5,
  modelsCGr = 10000,
  maxT = 150,
  decayRate1 = 0.8,
  decayRate2 = 0.6,
  threshT = 40,
  iter_per_temp = 100
)
```

Arguments

network_top	topology of the full network
data	processed gene expression matrix
clusterRef	cluster indices of all models
cenMedRef	cluster centers
cutOffM	cluster radii
gene_list	gene clustering output
init_top	initial circuit topology
output	a string of file prefix for saving results ("Results")
nRepeat	number of repeats of RACIPE simulations for each new circuit topology (5) A new circuit is simulated by RACIPE nRepeat times for robust score evaluation; The scores will then be saved and used in future iterations, when the circuits are sampled again.
modelsCGr	number of RACIPE models to be simulated (10000)
maxT	maximum/initial temperature in SA (150)
decayRate1	1st temperature decaying rate (geometrically decaying) (0.8)
decayRate2	2nd temperature decaying rate (0.6), until temperature = 1 (current implementation)
threshT	a second temperature in SA, below which SA has a slower temperature decaying rate (40)
iter_per_temp	number of iterations for each fixed temperature (100)

Value

df: topology of the optimized CG circuit

---

opt_SA_multi *Circuit optimization with Simulated Annealing (SA) algorithm (multiple threads)*

Description

Circuit optimization with Simulated Annealing (SA) algorithm (multiple threads)

Usage

```
opt_SA_multi(
  network_top,
  data,
  clusterRef,
  cenMedRef,
  cutOffM,
  gene_list,
  inTopsM,
  output = "Results",
  nRepeat = 5,
```

```
  modelsCGr = 10000,
  maxT = 150,
  decayRate1 = 0.8,
  decayRate2 = 0.6,
  threshT = 40,
  iter_per_temp = 100,
  numbThr = 40,
  nSim = 20
)
```

Arguments

network_top	topology of the full network
data	processed gene expression matrix
clusterRef	cluster indices of all models
cenMedRef	cluster centers
cutOffM	cluster radii
gene_list	gene clustering output
inTopsM	a list of all initial circuit topologies
output	a string of file prefix for saving results ("Results")
nRepeat	number of repeats of RACIPE simulations for each new circuit topology (5) A new circuit is simulated by RACIPE nRepeat times for robust score evaluation; The scores will then be saved and used in future iterations, when the circuits are sampled again.
modelsCGr	number of RACIPE models to be simulated (10000)
maxT	maximum/initial temperature in SA (150)
decayRate1	1st temperature decaying rate (geometrically decaying) (0.8)
decayRate2	2nd temperature decaying rate (0.6), until temperature = 1 (current implementation)
threshT	a second temperature in SA, below which SA has a slower temperature decaying rate (40)
iter_per_temp	number of iterations for each fixed temperature (100)
numbThr	number of requested threads for HPC (40)
nSim	number of parallel simulations (20)

Value

df: topology of the optimized CG circuit

opt_TE *Circuit optimization with temperature tempting (TE) algorithm*

Description

Circuit optimization with temperature tempting (TE) algorithm

Usage

```
opt_TE(
  network_top,
  data,
  clusterRef,
  cenMedRef,
  cutOffM,
  gene_list,
  inTopsM,
  output = "Results",
  nRepeat = 5,
  modelsCGr = 10000,
  numbThr = 2,
  temp_Array = c(1, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.5, 2, 2.5, 3, 3.5, 4, 6, 9, 11,
    13, 20, 28, 40, 55, 70, 90, 120),
  iter_temp_add = c(50, 100, 150),
  numb_iter_extra = 1100,
  logAlpha = log(0.4)
)
```

Arguments

network_top	topology of the full network
data	processed gene expression matrix
clusterRef	cluster indices of all models
cenMedRef	cluster centers
cutOffM	cluster radii
gene_list	gene clustering output
inTopsM	a list of all initial circuit topologies
output	a string of file prefix for saving results ("Results")
nRepeat	number of repeats of RACIPE simulations for each new circuit topology (5) A new circuit is simulated by RACIPE nRepeat times for robust score evaluation; The scores will then be saved and used in future iterations, when the circuits are sampled again.
modelsCGr	number of RACIPE models to be simulated (10000)
numbThr	number of requested threads for HPC (40)
temp_Array	temperatures for all replicas Default: a total of 24 replicas with temperatures: c(1,1.05,1.1,1.15,1.2,1.25,1.3,1.5,2.0,2.5,3.0,3.5,4.0,6,9,11,13,20,28,40,55,70,90,120)
iter_temp_add	the number of iterations (proposed swaps per replica) during the temperature addition process Default: three runs of the procedure; (c(50,100,150))

numb_iter_extra
: number of extra iterations after the temperature addition process (1100)

logAlpha
: log of the target swap rate (log(0.4))

Value

df: topology of the optimized CG circuit

---

plot_network *Visualize network topology*

Description

Visualize network topology

Usage

```
plot_network(tf_links, height = "300px")
```

Arguments

tf_links
: circuit topology

height
: plot height ("300px")

Value

network plot

---

reordering *Reordering the gene clusters in the gene expression matrix*

Description

Reordering the gene clusters in the gene expression matrix

Usage

```
reordering(logscData, gene_list, geneGroupOrder = NULL)
```

Arguments

logscData
: current gene expression data matrix

gene_list
: gene clustering output

geneGroupOrder
: desired order of gene groups

Value

a list of reordered data (logscData) and an updated gene list (gene_list)

signChanges_top *Making sign changes to some edges of a circuit topology*

Description

this process will be tried 3 times if no new circuit topology is sampled

Usage

```
signChanges_top(
  samp_top,
  numb_genes,
  numb_changes,
  pos_changes,
  inNodes,
  list_top
)
```

Arguments

samp_top	the current circuit topology
numb_genes	number of genes of the circuit
numb_changes	number of edges with sign changes (>0)
pos_changes	indices of edges that can have sign changes
inNodes	a list of input nodes, output from the function find_inout
list_top	list of sampled circuit topologies, we make sure they are not sampled again

Value

cTop: the modified circuit topology

Index