Package 'CoGAPS'

October 15, 2018

Version 3.0.2

Date 2018-04-24

Title Coordinated Gene Activity in Pattern Sets

Author Thomas Sherman, Wai-shing Lee, Conor Kelton, Ondrej Maxian, Jacob Carey, Genevieve Stein-O'Brien, Michael Considine, Maggie Wodicka, John Stansfield, Shawn Sivy, Carlo Colantuoni, Alexander Favorov, Mike Ochs, Elana Fertig

Description Coordinated Gene Activity in Pattern Sets (CoGAPS) implements a Bayesian MCMC matrix factorization algorithm, GAPS, and links it to gene set statistic methods to infer biological process activity. It can be used to perform sparse matrix factorization on any data, and when this data represents biomolecules, to do gene set analysis.

Maintainer Elana J. Fertig <ejfertig@jhmi.edu>, Thomas D. Sherman <tomsherman159@gmail.com>

Depends R (>= 3.4.0), Rcpp (>= 0.11.0)

Imports RColorBrewer (>= 1.0.5), gplots (>= 2.8.0), graphics, grDevices, methods, cluster, shiny, stats, utils, doParallel, foreach, ggplot2, reshape2

Suggests testthat, lintr, knitr, rmarkdown, BiocStyle

LinkingTo Rcpp, BH

VignetteBuilder knitr

License GPL (==2)

biocViews GeneExpression, Transcription, GeneSetEnrichment, DifferentialExpression, Bayesian, Clustering, TimeCourse, RNASeq, Microarray, MultipleComparison, DimensionReduction

NeedsCompilation yes

RoxygenNote 6.0.1

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

git_branch RELEASE_3_7

git_last_commit 9d82c1f

git_last_commit_date 2018-07-26

Date/Publication 2018-10-15

R topics documented:

CoGAPS-package *CoGAPS: Coordinated Gene Activity in Pattern Sets*

Description

CoGAPS implements a Bayesian MCMC matrix factorization algorithm, GAPS, and links it to gene set statistic methods to infer biological process activity. It can be used to perform sparse matrix factorization on any data, and when this data represents biomolecules, to do gene set analysis.

Package:	CoGAPS
Type:	Package
Version:	2.99.0
Date:	2018-01-24
License:	LGPL

Author(s)

Maintainer: Elana J. Fertig <ejfertig@jhmi.edu>, Michael F. Ochs <ochsm@tcnj.edu>

References

Fertig EJ, Ding J, Favorov AV, Parmigiani G, Ochs MF. CoGAPS: an R/C++ package to identify patterns and biological process activity in transcriptomic data. Bioinformatics. 2010 Nov 1;26(21):2792-3

binaryA *Binary Heatmap for Standardized A Matrix*

Description

Binary Heatmap for Standardized A Matrix

Usage

```
binaryA(Amean, Asd, threshold = 3)
```

Arguments

Amean	the mean estimate for the A matrix
Asd	the standard deviations on Amean
threshold	the number of standard deviations above zero that an element of Amean must be to get a value of 1

Details

creates a binarized heatmap of the A matrix in which the value is 1 if the value in Amean is greater than threshold * Asd and 0 otherwise

Value

plots a heatmap of the A Matrix

Examples

```
data(SimpSim)
binaryA(SimpSim.result$Amean, SimpSim.result$Asd, threshold=3)
```

`calcCoGAPSStat` *Calculate Gene Set Statistics*

Description

Calculate Gene Set Statistics

Usage

```
calcCoGAPSStat(Amean, Asd, GStoGenes, numPerm = 500)
```

Arguments

`Amean`	A matrix mean values
`Asd`	A matrix standard deviations
`GStoGenes`	data.frame or list with gene sets
`numPerm`	number of permutations for null

Details

calculates the gene set statistics for each column of A using a Z-score from the elements of the A matrix, the input gene set, and permutation tests

Value

gene set statistics for each column of A

Examples

```
data('SimpSim')
calcCoGAPSStat(SimpSim.result$Amean, SimpSim.result$Asd, GStoGenes=GSets,
numPerm=500)
```

calcGeneGSStat *Probability Gene Belongs in Gene Set*

Description

Probability Gene Belongs in Gene Set

Usage

```
calcGeneGSStat(Amean, Asd, GSGenes, numPerm, Pw = rep(1, ncol(Amean)),
  nullGenes = FALSE)
```

Arguments

Amean	A matrix mean values
Asd	A matrix standard deviations
GSGenes	data.frame or list with gene sets
numPerm	number of permutations for null
Pw	weight on genes
nullGenes	logical indicating gene adjustment

Details

calculates the probability that a gene listed in a gene set behaves like other genes in the set within the given data set

Value

gene similiarity statistic

Examples

```
data('SimpSim')
calcGeneGSStat(SimpSim.result$Amean, SimpSim.result$Asd, GSGenes=GSets[[1]],
numPerm=500)
```

calcZ *Compute Z-Score Matrix*

Description

Compute Z-Score Matrix

Usage

```
calcZ(meanMat, sdMat)
```

Arguments

meanMat	matrix of mean values
sdMat	matrix of standard deviation values

Details

calculates the Z-score for each element based on input mean and standard deviation matrices

Value

matrix of z-scores

Examples

```
data(SimpSim)
calcZ(SimpSim.result$Amean, SimpSim.result$Asd)
```

---

cellMatchR *cellMatchR*

Description

cellMatchR

Usage

```
cellMatchR(Atot, nSets, cnt, minNS = NA, maxNS = NA, ignore.NA = FALSE,
  bySet = FALSE, plotDen = FALSE, ...)
```

Arguments

Atot	a matrix containing the total by set estimates of Pmean output from reOrderBySet
nSets	number of parallel sets used to generate Atot
cnt	number of branches at which to cut dendrogram
minNS	minimum of individual set contributions a cluster must contain
maxNS	maximum of individual set contributions a cluster must contain
ignore.NA	logical indicating whether or not to ignore NAs from potential over dimensionalization. Default is FALSE.
bySet	logical indicating whether to return list of matched set solutions from Atot
plotDen	plot
...	additional parameters for agnes

Value

a matrix of concensus patterns by samples. If bySet=TRUE then a list of the set contributions to each concensus pattern is also returned.

CoGAPS *CoGAPS Matrix Factorization Algorithm*

Description

CoGAPS Matrix Factorization Algorithm

Usage

```
CoGAPS(D, S, nFactor = 7, nEquil = 1000, nSample = 1000,
  nOutputs = 1000, nSnapshots = 0, alphaA = 0.01, alphaP = 0.01,
  maxGibbmassA = 100, maxGibbmassP = 100, seed = -1, messages = TRUE,
  singleCellRNASeq = FALSE, whichMatrixFixed = "N",
  fixedPatterns = matrix(0), checkpointInterval = 0,
  checkpointFile = "gaps_checkpoint.out", ...)
```

Arguments

D	data matrix
S	uncertainty matrix (std devs for chi-squared of Log Likelihood)
nFactor	number of patterns (basis vectors, metagenes), which must be greater than or equal to the number of rows of FP
nEquil	number of iterations for burn-in
nSample	number of iterations for sampling
nOutputs	how often to print status into R by iterations
nSnapshots	the number of individual samples to capture
alphaA	sparsity parameter for A domain
alphaP	sparsity parameter for P domain
maxGibbmassA	limit truncated normal to max size
maxGibbmassP	limit truncated normal to max size
seed	a positive seed is used as-is, while any negative seed tells the algorithm to pick a seed based on the current time
messages	display progress messages
singleCellRNASeq	indicates if the data is single cell RNA-seq data
whichMatrixFixed	character to indicate whether A or P matric contains the fixed patterns
fixedPatterns	matrix of fixed values in either A or P matrix
checkpointInterval	time (in seconds) between creating a checkpoint
checkpointFile	name of the checkpoint file
...	keeps backwards compatibility with arguments from older versions

Details

calls the C++ MCMC code and performs Bayesian matrix factorization returning the two matrices that reconstruct the data matrix

Value

list with A and P matrix estimates

Examples

```
data(SimpSim)
result <- CoGAPS(SimpSim.D, SimpSim.S, nFactor=3, nOutputs=250)
```

`CoGapsFromCheckpoint` *Restart CoGAPS from Checkpoint File*

Description

Restart CoGAPS from Checkpoint File

Usage

```
CoGapsFromCheckpoint(D, S, path, checkpointFile = NA)
```

Arguments

`D`	data matrix
`S`	uncertainty matrix
`path`	path to checkpoint file
`checkpointFile`	name for future checkpooints made

Details

loads the state of a previous CoGAPS run from a file and continues the run from that point

Value

list with A and P matrix estimates

`computeGeneGSProb` *Compute Gene Probability*

Description

Compute Gene Probability

Usage

```
computeGeneGSProb(Amean, Asd, GSGenes, Pw = rep(1, ncol(Amean)),
  numPerm = 500, PwNull = FALSE)
```

Arguments

Amean	A matrix mean values
Asd	A matrix standard deviations
GSGenes	data.frame or list with gene sets
Pw	weight on genes
numPerm	number of permutations for null
PwNull	- logical indicating gene adjustment

Details

Computes the p-value for gene set membership using the CoGAPS-based statistics developed in Fertig et al. (2012). This statistic refines set membership for each candidate gene in a set specified in GSGenes by comparing the inferred activity of that gene to the average activity of the set.

Value

A vector of length GSGenes containing the p-values of set membership for each gene containined in the set specified in GSGenes.

Examples

```
data('SimpSim')
computeGeneGSProb(SimpSim.result$Amean, SimpSim.result$Asd, GSGenes=GSets[[1]],
numPerm=500)
```

---

createGWCoGAPSSets *Create Gene Sets for GWCoGAPS*

Description

Create Gene Sets for GWCoGAPS

Usage

```
createGWCoGAPSSets(D, S, nSets, simulationName)
```

Arguments

D	data matrix
S	uncertainty matrix
nSets	number of sets to partition the data into
simulationName	name used to identify files created by this simulation

Details

factors whole genome data into randomly generated sets for indexing

Value

simulationName used to identify saved files

Examples

```
data(SimpSim)
createGWCoGAPSSets(SimpSim.D, SimpSim.S, nSets=2, "example")
```

`createscCoGAPSSets` *Create Gene Sets for scCoGAPS*

Description

factors whole genome data into randomly generated sets for indexing

Usage

```
createscCoGAPSSets(D, nSets, simulationName, samplingRatio = NULL,
  path = "", anotionObj = NULL)
```

Arguments

`D`	data matrix
`nSets`	number of sets to partition the data into
`simulationName`	name used to identify files created by this simulation
`samplingRatio`	vector of relative quantities to use for sampling celltypes
`path`	character string indicating were to save resulting data objects. default is current working dir
`anotionObj`	vector of same length as number of columns of D

Value

simulationName used to identify saved files

Examples

```
data(SimpSim)
createscCoGAPSSets(SimpSim.D, nSets=2, simulationName="example")
```

`displayBuildReport` *Display Information About Package Compilation*

Description

Display Information About Package Compilation

Usage

```
displayBuildReport()
```

Details

displays information about how the package was compiled, i.e. which compiler/version was used, which compile time options were enabled, etc...

Value

display builds information

Examples

```
CoGAPS::displayBuildReport()
```

---

gapsMapRun *Backwards Compatibility with v2*

Description

Backwards Compatibility with v2

Usage

```
gapsMapRun(D, S, FP, ABins = data.frame(), PBins = data.frame(),
  nFactor = 5, simulation_id = "simulation", nEquil = 1000,
  nSample = 1000, nOutR = 1000, output_atomic = FALSE,
  fixedMatrix = "P", fixedBinProbs = FALSE, fixedDomain = "N",
  sampleSnapshots = TRUE, numSnapshots = 100, alphaA = 0.01,
  nMaxA = 1e+05, max_gibbmass_paraA = 100, alphaP = 0.01, nMaxP = 1e+05,
  max_gibbmass_paraP = 100, seed = -1, messages = TRUE)
```

Arguments

D	data matrix
S	uncertainty matrix
FP	data.frame with rows giving fixed patterns for P
ABins	unused
PBins	unused
nFactor	number of patterns (basis vectors, metagenes), which must be greater than or equal to the number of rows of FP
simulation_id	unused
nEquil	number of iterations for burn-in
nSample	number of iterations for sampling
nOutR	number of output messages
output_atomic	unused
fixedMatrix	unused
fixedBinProbs	unused
fixedDomain	unused

`sampleSnapshots`
: indicates if snapshots should be made

`numSnapshots`
: how many snapshots to take

`alphaA`
: sparsity parameter for A domain

`nMaxA`
: unused

`max_gibbmass_paraA`
: limit truncated normal to max size

`alphaP`
: sparsity parameter for P domain

`nMaxP`
: unused

`max_gibbmass_paraP`
: limit truncated normal to max size

`seed`
: a positive seed is used as-is, while any negative seed tells the algorithm to pick a seed based on the current time

`messages`
: display progress messages

`...`
: v2 style parameters

Value

list with A and P matrix estimates

Examples

```
data(SimpSim)
nC <- ncol(SimpSim.D)
patterns <- matrix(1:nC/nC, nrow=1, ncol=nC)
result <- gapsMapRun(SimpSim.D, SimpSim.S, FP=patterns, nFactor=3)
```

---

gapsRun *Backwards Compatibility with v2*

Description

Backwards Compatibility with v2

Usage

```
gapsRun(D, S, ABins = data.frame(), PBins = data.frame(), nFactor = 7,
  simulation_id = "simulation", nEquil = 1000, nSample = 1000,
  nOutR = 1000, output_atomic = FALSE, fixedBinProbs = FALSE,
  fixedDomain = "N", sampleSnapshots = TRUE, numSnapshots = 100,
  alphaA = 0.01, nMaxA = 1e+05, max_gibbmass_paraA = 100, alphaP = 0.01,
  nMaxP = 1e+05, max_gibbmass_paraP = 100, seed = -1, messages = TRUE)
```

Arguments

D	data matrix
S	uncertainty matrix
ABins	unused
PBins	unused
nFactor	number of patterns (basis vectors, metagenes), which must be greater than or equal to the number of rows of FP
simulation_id	unused
nEquil	number of iterations for burn-in
nSample	number of iterations for sampling
nOutR	number of output messages
output_atomic	unused
fixedBinProbs	unused
fixedDomain	unused
sampleSnapshots	indicates if snapshots should be made
numSnapshots	how many snapshots to take
alphaA	sparsity parameter for A domain
nMaxA	unused
max_gibbmass_paraA	limit truncated normal to max size
alphaP	sparsity parameter for P domain
nMaxP	unused
max_gibbmass_paraP	limit truncated normal to max size
seed	a positive seed is used as-is, while any negative seed tells the algorithm to pick a seed based on the current time
messages	display progress messages

Value

list with A and P matrix estimates

Examples

```
data(SimpSim)
result <- gapsRun(SimpSim.D, SimpSim.S, nFactor=3)
```

generateSeeds *Generate Seeds for Multiple Concurrent Runs*

Description

Generate Seeds for Multiple Concurrent Runs

Usage

```
generateSeeds(chains = 2, seed = -1)
```

Arguments

chains	number of seeds to generate (number of chains to run)
seed	positive values are kept, negative values will be overwritten by a seed generated from the current time

Value

vector of randomly generated seeds

GIST.D *Sample GIST gene expression data from Ochs et al. (2009).*

Description

Gene expression data from gastrointestinal stromal tumor cell lines treated with Gleevec.

Usage

```
GIST_TS_20084
```

Format

Matrix with 1363 genes by 9 samples of mean gene expression data.

References

Ochs, M., Rink, L., Tarn, C., Mburu, S., Taguchi, T., Eisenberg, B., and Godwin, A. (2009). Detection of treatment-induced changes in signaling pathways in gastrointestinal stromal tumors using transcriptomic data. Cancer Res, 69(23), 9125-9132.

GIST.S *Sample GIST gene expression data from Ochs et al. (2009).*

Description

Standard deviation of gene expression data from gastrointestinal stromal tumor cell lines treated with Gleevec.

Usage

```
GIST_TS_20084
```

Format

Matrix with 1363 genes by 9 samples containing standard deviation (GIST.S) of the gene expression data.

References

Ochs, M., Rink, L., Tarn, C., Mburu, S., Taguchi, T., Eisenberg, B., and Godwin, A. (2009). Detection of treatment-induced changes in signaling pathways in gastrointestinal stromal tumors using transcriptomic data. Cancer Res, 69(23), 9125-9132.

GSets *Simulated dataset to quantify gene set membership.*

Description

Simulated gene sets used to generate amplitude matrix in SimpSim.A and corresponding data SimpSim.D.

Usage

```
GSets
```

Format

A list containing names of genes in two simulated gene sets used to generate the data in SimpSim.D.

GWCoGAPS *GWCoGAPS*

Description

GWCoGAPS

Usage

```
GWCoGAPS(simulationName, nFactor, nCores = NA, cut = NA, minNS = NA,
  manualMatch = FALSE, consensusPatterns = NULL,
  saveUnmatchedPatterns = FALSE, ...)
```

Arguments

simulationName	name of this simulation
nFactor	number of patterns (basis vectors, metagenes), which must be greater than or equal to the number of rows of FP
nCores	number of cores for parallelization. If left to the default NA, nCores = nSets.
cut	number of branches at which to cut dendrogram used in patternMatch4Parallel
minNS	minimum of individual set contributions a cluster must contain
manualMatch	logical indicating whether or not to stop after initial phase for manual pattern matching
consensusPatterns	fixed pattern matrix to be used to ensure reciprocity of A weights accross sets
saveUnmatchedPatterns	option to save intermediate results for each set, before the pattern matching happens
...	additional parameters to be fed into gapsRun and gapsMapRun

Details

calls the C++ MCMC code and performs Bayesian matrix factorization returning the two matrices that reconstruct the data matrix for whole genome data;

Value

list of A and P estimates

See Also

gapsRun, patternMatch4Parallel, and gapsMapRun

Examples

```
data(SimpSim)
sim_name <- "example"
createGWCoGAPSSets(SimpSim.D, SimpSim.S, nSets=2, sim_name)
result <- GWCoGAPS(sim_name, nFactor=3, nEquil=200, nSample=200)
```

GWCoGapsFromCheckpoint *Restart a GWCoGaps Run from Checkpoint*

Description

Restart a GWCoGaps Run from Checkpoint

Usage

```
GWCoGapsFromCheckpoint(simulationName, nCores, cut = NA, minNS = NA, ...)
```

Arguments

simulationName	name of this simulation
nCores	number of cores for parallelization. If left to the default NA, nCores = nSets.
cut	number of branches at which to cut dendrogram used in patternMatch4Parallel
minNS	minimum of individual set contributions a cluster must contain
...	additional parameters to be fed into gapsRun and gapsMapRun

Value

list of A and P estimates

Examples

```
data(SimpSim)
sim_name <- "example"
createGWCoGAPSSets(SimpSim.D, SimpSim.S, nSets=2, sim_name)
trash <- GWCoGAPS(sim_name, nFactor=3, nEquil=200, nSample=200)
result <- GWCoGapsFromCheckpoint(sim_name, 2)
```

patternMarkers *patternMarkers*

Description

patternMarkers

Usage

```
patternMarkers(Amatrix = NA, scaledPmatrix = FALSE, Pmatrix = NA,
  threshold = "all", lp = NA, full = FALSE)
```

Arguments

Amatrix	A matrix of genes by weights resulting from CoGAPS or other NMF decomposition
scaledPmatrix	logical indicating whether the corresponding pattern matrix was fixed to have max 1 during decomposition
Pmatrix	the corresponding Pmatrix (patterns X samples) for the provided Amatrix (genes x patterns). This must be supplied if scaledPmatrix is FALSE.
threshold	# the type of threshold to be used. The default "all" will distribute genes into pattern with the lowest ranking. The "cut" thresholding by the first gene to have a lower ranking, i.e. better fit to, a pattern.
lp	a vector of weights for each pattern to be used for finding markers. If NA markers for each pattern of the A matrix will be used.
full	logical indicating whether to return the ranks of each gene for each pattern

Value

By default a non-overlapping list of genes associated with each lp. If full=TRUE a data.frame of genes rankings with a column for each lp will also be returned.

---

patternMatch4Parallel *patternMatch4Parallel*

Description

patternMatch4Parallel

Usage

```
patternMatch4Parallel(Ptot, nSets, cnt, minNS = NA, maxNS = NULL,
  cluster.method = "complete", ignore.NA = FALSE, bySet = FALSE, ...)
```

Arguments

Ptot	a matrix containing the total by set estimates of Pmean output from reOrderBySet
nSets	number of parallel sets used to generate Ptot
cnt	number of branches at which to cut dendrogram
minNS	minimum of individual set contributions a cluster must contain
maxNS	max of individual set contributions a cluster must contain. default is nSets+minNS
cluster.method	the agglomeration method to be used for clustering
ignore.NA	logical indicating whether or not to ignore NAs from potential over dimensionalization. Default is FALSE.
bySet	logical indicating whether to return list of matched set solutions from Ptot
...	additional parameters for agnes

Value

a matrix of concensus patterns by samples. If bySet=TRUE then a list of the set contributions to each concensus pattern is also returned.

See Also

agnes

`patternMatcher` *PatternMatcher Shiny Ap*

Description

PatternMatcher Shiny Ap

Usage

```
patternMatcher(PBySet = NULL, out = NULL, order = NULL,
  sample.color = NULL)
```

Arguments

`PBySet`	list of matched set solutions for the Pmatrix from an NMF algorithm
`out`	optional name for saving output
`order`	optional vector indicating order of samples for plotting. Default is NULL.
`sample.color`	optional vector of colors of same length as colnames. Default is NULL.

Value

either an index of selected sets' contributions or the editted `PBySet` object

---

`plotAtoms` *Plot Number of Atoms*

Description

Plot Number of Atoms

Usage

```
plotAtoms(gapsRes, type = "sampA")
```

Arguments

`gapsRes`	the list resulting from applying GAPS
`type`	the atoms to plot, values are "sampA", "sampP" , "equilA", or "equilP" to plot sampling or equilibration teop atom numbers

Details

a simple plot of the number of atoms from one of the vectors returned with atom numbers

Value

plot

Examples

```
data(SimpSim)
plotAtoms(SimpSim.result, type="sampA")
```

`plotDiag` *Diagnostic Plots*

Description

Diagnostic Plots

Usage

```
plotDiag(gapsRes)
```

Arguments

`gapsRes`	list returned by CoGAPS

Details

plots a series of diagnostic plots

Value

plot

Examples

```
data(SimpSim)
plotDiag(SimpSim.result)
```

`plotGAPS` *Plot Decomposed A and P Matrices*

Description

Plot Decomposed A and P Matrices

Usage

```
plotGAPS(A, P, outputPDF = "")
```

Arguments

A	the mean A matrix
P	the mean P matrix
outputPDF	optional root name for PDF output, if not specified, output goes to screen

Details

plots the output A and P matrices as a heatmap and line plot respectively

Value

plot

Examples

```
data(SimpSim)
plotGAPS(SimpSim.result$Amean, SimpSim.result$Pmean)
```

---

plotP *Plot the P Matrix*

Description

Plot the P Matrix

Usage

```
plotP(Pmean, Psd = NULL)
```

Arguments

Pmean	matrix of mean values of P
Psd	matrix of standard deviation values of P

Details

plots the P matrix in a line plot with error bars

Value

plot

Examples

```
data(SimpSim)
plotP(SimpSim.result$Pmean, SimpSim.result$Psd)
```

plotPatternMarkers *plotPatternMarkers*

Description

plotPatternMarkers

Usage

```
plotPatternMarkers(data = NA, patternMarkers = NA, patternPalette = NA,
  sampleNames = NA, samplePalette = NULL, colDenogram = TRUE, heatmapCol,
  scale = "row", ...)
```

Arguments

data	the dataset from which the patterns where generated
patternMarkers	the list of genes generated from the patternMarkers function
patternPalette	a vector indicating what color should be used for each pattern
sampleNames	names of the samples to use for labeling
samplePalette	a vector indicating what color should be used for each sample
colDenogram	logical indicating whether to display sample denogram
heatmapCol	pallelet giving color scheme for heatmap
scale	character indicating if the values should be centered and scaled in either the row direction or the column direction, or none. The default is "row".
...	additional graphical parameters to be passed to heatmap.2

Value

heatmap of the data values for the patternMarkers

See Also

heatmap.2

plotSmoothPatterns *Plot Smooth Patterns*

Description

Plot Smooth Patterns

Usage

```
plotSmoothPatterns(P, x = NULL, breaks = NULL, breakStyle = TRUE,
  orderP = !all(is.null(x)), plotPTS = FALSE, pointCol = "black",
  lineCol = "grey", add = FALSE, ...)
```

Arguments

P	the mean A matrix
x	optional variables
breaks	breaks in plots
breakStyle	style of breaks
orderP	whether to order patterns
plotPTS	whether to plot points on lines
pointCol	color of points
lineCol	color of line
add	logical specifying if bars should be added to an already existing plot; defaults to 'FALSE'.
...	arguments to be passed to/from other methods. For the default method these can include further arguments (such as 'axes', 'asp' and 'main') and graphical parameters (see 'par') which are passed to 'plot.window()', 'title()' and 'axis'.

Details

plots the output A and P matrices as a heatmap and a line plot respectively

Value

plot

---

postFixed4Parallel *Post Processing of Parallel Output*

Description

Post Processing of Parallel Output

Usage

```
postFixed4Parallel(AP.fixed, setValues, setMatrix = "P")
```

Arguments

AP.fixed	output of parallel gapsMapRun calls with same FP
setValues	data.frame with rows giving fixed patterns for P used as input for gapsMapRun
setMatrix	which matrix, A or P

Value

list of two data.frames containing the A matrix estimates or their corresponding standard deviations from output of parallel CoGAPS

postFixed4SC *Post Processing of Parallel Output*

Description

Post Processing of Parallel Output

Usage

```
postFixed4SC(AP.fixed, setAs)
```

Arguments

AP.fixed	output of parallel gapsMapRun calls with same FP
setAs	data.frame with rows giving fixed patterns for A used as input for gapsMapRun

Value

list of two data.frames containing the A matrix estimates or their corresponding standard deviations from output of parallel CoGAPS

reconstructGene *Reconstruct Gene*

Description

Reconstruct Gene

Usage

```
reconstructGene(A, P, genes = NA)
```

Arguments

A	A matrix estimates
P	corresponding P matrix estimates
genes	an index of the gene or genes of interest

Value

the D' estimate of a gene or set of genes

Examples

```
data(SimpSim)
reconstructGene(SimpSim.result$Amean, SimpSim.result$Pmean)
```

reorderByPatternMatch *Reorder By Pattern Match*

Description

Reorder By Pattern Match

Usage

```
reorderByPatternMatch(P, matchTo)
```

Arguments

P	matrix to be matched
matchTo	matrix to match P to

Value

matched patterns

reOrderBySet *reOrderBySet*

Description

<restructures output of gapsRun into a list containing each sets solution for Amean, Pmean, and Asd>

Usage

```
reOrderBySet(AP, nFactor, nSets, match = "P")
```

Arguments

AP	output of gapsRun in parallel
nFactor	number of patterns
nSets	number of sets
match	which matrix to use for downstream matching. default is P

Value

a list containing the nSets sets solution for Amean under "A", Pmean under "P", and Asd under "Asd"

residuals *Plot of Residuals*

Description

Plot of Residuals

Usage

```
residuals(AMean_Mat, PMean_Mat, D, S)
```

Arguments

AMean_Mat	matrix of mean values for A from GAPS
PMean_Mat	matrix of mean values for P from GAPS
D	original data matrix run through GAPS
S	original standard deviation matrix run through GAPS

Details

calculate residuals and produce heatmap

Value

creates a residual plot

Examples

```
data(SimpSim)
residuals(SimpSim.result$Amean, SimpSim.result$Pmean, SimpSim.D, SimpSim.S)
```

scCoGAPS *scCoGAPS*

Description

scCoGAPS

Usage

```
scCoGAPS(simulationName, nFactor, nCores = NA, cut = NA, minNS = NA,
  manualMatch = FALSE, consensusAs = NULL, ...)
```

Arguments

simulationName	name of this simulation
nFactor	number of patterns (basis vectors, metagenes), which must be greater than or equal to the number of rows of FP
nCores	number of cores for parallelization. If left to the default NA, nCores = nSets.
cut	number of branches at which to cut dendrogram used in patternMatch4singleCell
minNS	minimum of individual set contributions a cluster must contain
manualMatch	logical indicating whether or not to stop after initial phase for manual pattern matching
consensusAs	fixed pattern matrix to be used to ensure reciprocity of A weights accross sets
...	additional parameters to be fed into gapsRun and gapsMapRun

Details

calls the C++ MCMC code and performs Bayesian matrix factorization returning the two matrices that reconstruct the data matrix for whole genome data;

Value

list of A and P estimates

---

scCoGapsFromCheckpoint

Restart a scCoGAPS run from a Checkpoint

Description

Restart a scCoGAPS run from a Checkpoint

Usage

```
scCoGapsFromCheckpoint(simulationName, nCores, cut = NA, minNS = NA, ...)
```

Arguments

simulationName	name of this simulation
nCores	number of cores for parallelization. If left to the default NA, nCores = nSets.
cut	number of branches at which to cut dendrogram used in patternMatch4Parallel
minNS	minimum of individual set contributions a cluster must contain
...	additional parameters to be fed into gapsRun and gapsMapRun

Value

list of A and P estimates

SimpSim.A *Simulated data*

Description

True amplitude matrix generated from gene sets in GSets used to generate simulated data in SimpSim.D.

Usage

```
SimpSim.A
```

Format

Matrix with 30 genes by 3 patterns of true amplitude used to generate simulated data.

SimpSim.D *Simulated data*

Description

Simulated gene expression data from true patterns in SimpSim.P and amplitude in SimpSim.A.

Usage

```
SimpSim.D
```

Format

Matrix with 30 genes by 20 samples of simulated gene expression data.

SimpSim.P *Simulated data*

Description

True pattern matrix used to generate simulated data in SimpSim.D.

Usage

```
SimpSim.P
```

Format

Matrix with 3 patterns by 20 samples of true patterns used to generate simulated data.

SimpSim.result — *Simulated Data Results*

Description

Resulting list created by calling CoGAPS on simulated data

Usage

```
SimpSim.result
```

Format

list

SimpSim.S — *Simulated data*

Description

Standard deviation of simulated gene expression data from true patterns in SimpSim.P and amplitude in SimpSim.A.

Usage

```
SimpSim.S
```

Format

Matrix with 30 genes by 20 samples of containing standard deviation of simulated gene expression data.

tf2ugFC — *Gene sets defined by transcription factors defined from TRANSFAC.*

Description

List of genes contained in gastrointestinal stromal tumor cell line measurements that are regulated by transcription factors in the TRANSFAC database. Used for the gene set analysis in Ochs et al. (2009).

Usage

```
TFGSList
```

Format

Data.frame containing genes (rows) regulated by each transcription factor (columns).

References

Ochs, M., Rink, L., Tarn, C., Mburu, S., Taguchi, T., Eisenberg, B., and Godwin, A. (2009). Detection of treatment-induced changes in signaling pathways in gastrointestinal stromal tumors using transcriptomic data. Cancer Res, 69(23), 9125-9132.

Index