Package ‘PAC’

Type Package

Title Partition-Assisted Clustering and Multiple Alignments of Networks

Version 1.1.4

Date 2021-02-17

Author Ye Henry Li, Dangna Li

Maintainer Ye Henry Li <h1owl2@gmail.com>

Description Implements partition-assisted clustering and multiple alignments of networks. It 1) utilizes partition-assisted clustering to find robust and accurate clusters and 2) discovers coherent relationships of clusters across multiple samples. It is particularly useful for analyzing single-cell data set. Please see Li et al. (2017) <doi:10.1371/journal.pcbi.1005875> for detail method description.

URL https://doi.org/10.1371/journal.pcbi.1005875

License GPL-3

Imports Rcpp (>= 0.12.2),igraph,parmigene,infotheo,dplyr, Rtsne, ggplot2, ggrepel

Suggests knitr, rmarkdown

VignetteBuilder knitr

LinkingTo Rcpp

RoxygenNote 5.0.1

NeedsCompilation yes

SystemRequirements C++11

Repository CRAN

Date/Publication 2021-02-18 07:00:02 UTC

R topics documented:

aggregateData .......................................................... 2
annotateClades ......................................................... 3
annotationMatrix_withSubpopProp ................................. 4
aggregateData

**Description**

Aggregates results from the clustering and merging step.

**Usage**

`aggregateData(dataInput, labelsInput)`

**Arguments**

- `dataInput` Data matrix, with first column being SampleID.
- `labelsInput` cluster labels from PAC.

**Value**

The aggregated data of `dataInput`, with average signal levels for all clusters and sample combinations.
annotateClades

Examples

\begin{verbatim}
n = 5e3  # number of observations
p = 1  # number of dimensions
K = 3  # number of clusters
w = rep(1,K)/K  # component weights
mu <- c(0,2,4)  # component means
sd <- rep(1,K)/K  # component standard deviations
g <- sample(1:K,prob=w,size=n,replace=TRUE)  # ground truth for clustering
X <- as.matrix(rnorm(n=n,mean=mu[g],sd=sd[g]))
y <- PAC(X, K)
X2<-as.matrix(rnorm(n=n,mean=mu[g],sd=sd[g]))
y2<-PAC(X2,K)
X<-cbind("Sample1", as.data.frame(X)); colnames(X)<-c("SampleID", "Value")
X2<-cbind("Sample2", as.data.frame(X2)); colnames(X2)<-c("SampleID", "Value")
aggregateData(rbind(X,X2),c(y,y2))
\end{verbatim}

---

**annotateClades**

Creates annotation matrix for the clades in aggregated format. The matrix contains average signals of each dimension for each clade in each sample

**Description**

Creates annotation matrix for the clades in aggregated format. The matrix contains average signals of each dimension for each clade in each sample

**Usage**

\begin{verbatim}
annotateClades(sampleIDs, topHubs)
\end{verbatim}

**Arguments**

- **sampleIDs**: sampleID vector
- **topHubs**: number of top ranked genes to output for annotation; annotation is a concatenated list of top ranked genes.

**Value**

Annotated clade matrix
**annotationMatrix_withSubpopProp**

*Adds subpopulation proportion for the annotation matrix for the clades*

**Description**

Adds subpopulation proportion for the annotation matrix for the clades

**Usage**

`annotationMatrix_withSubpopProp(aggregateMatrix_withAnnotation)`

**Arguments**

- `aggregateMatrix_withAnnotation` the annotated clade matrix

**Value**

Annotated clade matrix with subpopulation proportions

---

**BSPLLeafCenter**

*Finds N Leaf centers in the data*

**Description**

Finds N Leaf centers in the data

**Usage**

`BSPLLeafCenter(data, N = 40, method = "dsp")`

**Arguments**

- `data` a \( n \times p \) data matrix
- `N` number of leaves centers
- `method` partition method, either "dsp (discrepancy based partition)", or "ll (bayesian sequential partition limited-look ahead)"

**Value**

leafctr N leaves centers
constellationPlot

**Description**

Makes constellation plot, in which the centroids are clusters embedded in the t-SNE 2D plane and the cross-sample relationships are plotted as lines connecting related sample clusters (clades).

**Usage**

```r
collect_element_set(pacman_results, perplexity, max_iter, seed,
    plotTitle = "Constellations of Clades", nudge_x = 0.3, nudge_y = 0.3)
```

**Arguments**

- `pacman_results`: PAC-MAN analysis result matrix that contains network annotation, clade IDs and mean (centroid) clade expression levels.
- `perplexity`: perplexity setting for running t-SNE.
- `max_iter`: max_iter setting for running t-SNE.
- `seed`: set seed to make t-SNE and constellation plot to be reproducible.
- `plotTitle`: max_iter setting for running t-SNE.
- `nudge_x`: nudge on x coordinate of centroid labels.
- `nudge_y`: nudge on y coordinate of centroid labels.

---

fmeasure

**F-measure Calculation**

**Description**

Compute the F measure between the ground truth and the estimated label.

**Usage**

```r
fmeasure(g, t)
```

**Arguments**

- `g`: the ground truth.
- `t`: estimated labels.

**Value**

- `f`: the F measure.
getAverageSpreadOf2SubpopClades

Calculate the (global) average spread of subpopulations in clades with 2 subpopulations on the constellation plot.

Description

Calculate the (global) average spread of subpopulations in clades with 2 subpopulations on the constellation plot.

Usage

getAddressAverageSpreadOf2SubpopClades(tsneResults, pacman_results)

Arguments

tsneResults  t-SNE output of clade centroids’ embedding.
pacman_results  PAC-MAN analysis result matrix that contains network annotation, clade IDs and mean (centroid) clade expression levels.

Value

Returns global average of 2-subpopulation clade spread on the constellation plot.

getExtraneousCladeSubpopulations

Calculates subpopulations in clades (with two or more subpopulations) that are too far away from other subpopulations (within the same clade) on the constellation plot; these far away subpopulations should be pruned away from the original clades.

Description

Calculates subpopulations in clades (with two or more subpopulations) that are too far away from other subpopulations (within the same clade) on the constellation plot; these far away subpopulations should be pruned away from the original clades.

Usage

getAddressExtraneousCladeSubpopulations(tsneResults, pacman_results, threshold_multiplier, max_threshold)
**Arguments**

- `tsneResults`: t-SNE output of clade centroids’ embedding.
- `pacman_results`: PAC-MAN analysis result matrix that contains network annotation, clade IDs and mean (centroid) clade expression levels.
- `threshold_multiplier`: how many times the threshold ((a) spread from center of clade for clades with three or more sample subpopulations and (b) distance from each subpopulation centroid for clades with exactly two subpopulations).
- `max_threshold`: the maximum distance (on t-SNE plane) allowed for sample subpopulations to be categorized into the same clade.

**Value**

Returns clade subpopulations to be pruned.

**Description**

Outputs representative networks for clades/subpopulations larger than a size filter (very small subpopulations are not considered in downstream analyses).

**Usage**

```r
getRepresentativeNetworks(sampleIDs, dim Subset, SubpopSizeFilter, num_networkEdge)
```

**Arguments**

- `sampleIDs`: sampleID vector
- `dim_subset`: a string vector of string names to subset the data columns for PAC; set to NULL to use all columns
- `SubpopSizeFilter`: the cutoff for small subpopulations. Smaller subpopulations have unstable covariance structure, so no network structure is calculated
- `num networkEdge`: the number of edges to draw for each subpopulation mutual information network
JaccardSM

Calculate the Jaccard similarity matrix.

Description

Calculates the Jaccard similarity matrix.

Usage

JaccardSM(network1, network2)

Arguments

network1 first network matrix input
network2 second network matrix input

Value

the alignment/co-occurrence score

-----------------------------------

heatmapInput

Creates the matrix that can be easily plotted with a heatmap function available in an R package

Description

Creates the matrix that can be easily plotted with a heatmap function available in an R package

Usage

heatmapInput(aggregateMatrix_withAnnotation)

Arguments

aggregateMatrix_withAnnotation
the annotated clade matrix

Value

the heatmap input matrix

-----------------------------------
MAN

*Creates network alignments using network constructed from subpopulations after PAC*

**Description**

Creates network alignments using network constructed from subpopulations after PAC

**Usage**

`MAN(sampleIDs, num_PACSupop, smallSubpopCutoff, k_clades)`

**Arguments**

- `sampleIDs` sampleID vector
- `num_PACSupop` number of subpopulations learned in PAC step for each sample
- `smallSubpopCutoff` Population size cutoff for subpopulations in clade calculation. The small sub-populations will be considered in the refinement step.
- `k_clades` number of clades to output before refinement

**Value**

- `clades_network` only the clades constructed without small subpopulations (by cutoff) using mutual information network alignments

---

MINetworkPlot_topEdges

*Plots mutual information network (mrnet algorithm) connection using the parmigene package. Mutual information calculated with infotheo package.*

**Description**

Plots mutual information network (mrnet algorithm) connection using the parmigene package. Mutual information calculated with infotheo package.

**Usage**

`MINetworkPlot_topEdges(dataMatrix, threshold)`

**Arguments**

- `dataMatrix` data matrix
- `threshold` the maximum number of edges to draw for each subpopulation mutual information network
MINetwork_matrix_topEdges

*Mutual information network connection matrix generation (mrnet algorithm) using the parmigene package. Mutual information calculated with infotheo package.*

---

**Description**

Mutual information network connection matrix generation (mrnet algorithm) using the parmigene package. Mutual information calculated with infotheo package.

**Usage**

```r
MINetwork_matrix_topEdges(dataMatrix, threshold)
```

**Arguments**

- `dataMatrix`: data matrix
- `threshold`: the number of edges to draw for each subpopulation mutual information network

**Value**

the mutual information network connection matrix with top edges

---

MINetwork_simplified_topEdges

*Outputs the vectorized summary of a network based on the number of edges connected to a node*

---

**Description**

Outputs the vectorized summary of a network based on the number of edges connected to a node

**Usage**

```r
MINetwork_simplified_topEdges(dataMatrix, threshold)
```

**Arguments**

- `dataMatrix`: data matrix
- `threshold`: the number of edges to draw for each subpopulation mutual information network
outputNetworks_topEdges_matrix

Wrapper to output the mutual information networks for subpopulations with size larger than a desired threshold.

Description

Wrapper to output the mutual information networks for subpopulations with size larger than a desired threshold.

Usage

outputNetworks_topEdges_matrix(dataMatrix, subpopulationLabels, threshold)

Arguments

dataMatrix: data matrix with first column being the sample ID
subpopulationLabels: the subpopulation labels
threshold: the number of edges to draw for each subpopulation mutual information network

outputRepresentativeNetworks_topEdges

Outputs the representative/clade networks (plots and summary vectors) for subpopulations with size larger than a desired threshold. Saves the networks and the data matrices without the smaller subpopulations.

Description

Outputs the representative/clade networks (plots and summary vectors) for subpopulations with size larger than a desired threshold. Saves the networks and the data matrices without the smaller subpopulations.

Usage

outputRepresentativeNetworks_topEdges(dataMatrix, subpopulationLabels, threshold)

Arguments

dataMatrix: data matrix with first column being the sample ID
subpopulationLabels: the subpopulation labels
threshold: the number of edges to draw for each subpopulation mutual information network
PAC

**Partition Assisted Clustering (PAC)**

1) utilizes DSP or BSP-ll to recursively partition the data space and 2) applies a short round of kmeans style postprocessing to efficiently output clustered labels of data points.

**Usage**

PAC(data, K, maxlevel = 40, method = "dsp", max.iter = 50)

**Arguments**

- **data**  
  a n x p data matrix
- **K**  
  number of final clusters in the output
- **maxlevel**  
  the maximum level of the partition
- **method**  
  partition method, either "dsp(discrepancy based partition)", or "bsp(bayesian sequential partition)"
- **max.iter**  
  maximum iteration for the kmeans step

**Value**

- y cluster labels for the input

**Examples**

```r
n = 5e3
p = 1
K = 3
w = rep(1, K)/K
mu <- c(0, 2, 4)
sd <- rep(1, K)/K
g <- sample(1:K, prob=w, size=n, replace=TRUE)  
X <- as.matrix(rnorm(n=n, mean=mu[g], sd=sd[g]))
y <- PAC(X, K)
print(fmeasure(g, y))
```
recordWithinClusterSpread

Calculates the within cluster spread

Description

Calculates the within cluster spread

Usage

recordWithinClusterSpread(sampleIDs, dim_subset = NULL, SubpopSizeFilter)

Arguments

sampleIDs A vector of sample names.
dim_subset a string vector of string names to subset the data columns for PAC; set to NULL to use all columns.
SubpopSizeFilter threshold to filter out very small clusters with too few points; these very small subpopulations may not be outliers and not biologically relevant.

Value

Returns the sample within cluster spread

refineSubpopulationLabels

Refines the subpopulation labels from PAC using network alignment and small subpopulation information. Outputs a new set of files containing the representative labels.

Description

Refines the subpopulation labels from PAC using network alignment and small subpopulation information. Outputs a new set of files containing the representative labels.

Usage

refineSubpopulationLabels(sampleIDs, dim_subset, clades_network_only, expressionGroupClamp)
renamePrunedSubpopulations

Prune away specified subpopulations in clades that are far away.

Description

Prune away specified subpopulations in clades that are far away.

Usage

renamePrunedSubpopulations(pacman_results, subpopulationsToPrune)

Arguments

pacman_results  PAC-MAN analysis result matrix that contains network annotation, clade IDs
                and mean (centroid) clade expression levels.
subpopulationsToPrune  A vector of clade IDs; these clades will be pruned.

Value

Returns PAC-MAN analysis result matrix with pruned clades. The pruning process creates new
clades to replace the original clade ID of the specified subpopulations.
runElbowPointAnalysis

**Description**

Runs elbow point analysis to find the practical optimal number of clades to output. Outputs the average within sample cluster spread for all samples and the elbow point analysis plot with loess line fitted through the results.

**Usage**

```r
runElbowPointAnalysis(ks, sampleIDs, dim_subset, num_PACSupop, smallSubpopCutoff, expressionGroupClamp, SubpopSizeFilter)
```

**Arguments**

- `ks` Vector that is a sequence of clade sizes.
- `sampleIDs` A vector of sample names.
- `dim_subset` a string vector of string names to subset the data columns for PAC; set to NULL to use all columns.
- `num_PACSupop` Number of PAC subpopulation explored in each sample.
- `smallSubpopCutoff` Cutoff of minor subpopulation not used in multiple alignments of networks
- `expressionGroupClamp` clamps the subpopulations into desired number of expression groups for assigning small subpopulations into larger groups or their own groups.
- `SubpopSizeFilter` threshold to filter out very small clusters with too few points in the calculation of cluster spreads; these very small subpopulations may be outliers and not biologically relevant.

**samplePass**

*Run PAC for Specified Samples*

**Description**

A wrapper to run PAC and output subpopulation mutual information networks. Please use the PAC function itself for individual samples or if the MAN step is not needed.
Usage

```r
samplePass(sampleIDs, dim_subset, hyperrectangles, num_PACSupop, max.iter,
            num_networkEdge)
```

Arguments

- **sampleIDs**: sampleID vector
- **dim_subset**: a string vector of string names to subset the data columns for PAC; set to NULL to use all columns
- **hyperrectangles**: number of hyperrectangles to learn for each sample
- **num_PACSupop**: number of subpopulations to output for each sample using PAC
- **max.iter**: postprocessing kmeans iterations
- **num_networkEdge**: a threshold on the number of edges to output for each subpopulation mutual information network
Index

aggregateData, 2
annotateClades, 3
annotationMatrix_withSubpopProp, 4
BSPLeaveCenter, 4
constellationPlot, 5
fmeasure, 5
getAverageSpreadOf2SubpopClades, 6
getExtraneousCladeSubpopulations, 6
getRepresentativeNetworks, 7
heatmapInput, 8
JaccardSM, 8
MAN, 9
MINetwork_matrix_topEdges, 10
MINetwork_simplified_topEdges, 10
MINetworkPlot_topEdges, 9
outputNetworks_topEdges_matrix, 11
outputRepresentativeNetworks_topEdges, 11
PAC, 12
recordWithinClusterSpread, 13
refineSubpopulationLabels, 13
renamePrunedSubpopulations, 14
runElbowPointAnalysis, 15
samplePass, 15