Package 'M3C'

April 16, 2019

Title Monte Carlo Reference-based Consensus Clustering

Version 1.4.1

Description Genome-wide data is used to stratify large complex datasets into classes using class discovery algorithms. A widely applied technique is consensus clustering, however; the approach is prone to overfitting and false positives. These issues arise from not considering reference distributions while selecting the number of classes (K). As a solution, we developed Monte Carlo reference-based consensus clustering (M3C). M3C uses a multi-core enabled Monte Carlo simulation to generate null distributions along the range of K which are used to select its value. Using a reference, that maintains the correlation structure of the input features, eliminates the limitations of consensus clustering. M3C uses the Relative Cluster Stability Index (RCSI) and p values to decide on the value of K and reject the null hypothesis, K=1. M3C can also quantify structural relationships between clusters, and uses spectral clustering to deal with non-Gaussian and complex structures. M3C can automatically analyse biological or clinical data with respect to the discovered classes.

Depends R (>= 3.4.0)

License AGPL-3

Encoding UTF-8

LazyData true

Imports ggplot2, Matrix, doSNOW, NMF, RColorBrewer, cluster, parallel, foreach, doParallel, matrixcalc, dendextend, sigclust, Rtsne, survival

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.0.1

biocViews ImmunoOncology, Clustering, GeneExpression, Transcription, RNASeq, Sequencing

git_url https://git.bioconductor.org/packages/M3C

git_branch RELEASE_3_8

git_last_commit 4b3f65a

git_last_commit_date 2019-01-04

Date/Publication 2019-04-15

Author Christopher John [aut, cre]

Maintainer Christopher John <chris.r.john86@gmail.com>

R topics documented:

`clustersim` *clustersim: A cluster simulator for testing clustering algorithms*

Description

clustersim: A cluster simulator for testing clustering algorithms

Usage

```
clustersim(n, n2, r, K, alpha, wobble, redp = NULL, print = FALSE,
  seed = NULL)
```

Arguments

`n`	Numerical value: The number of samples, it must be square rootable
`n2`	Numerical value: The number of features
`r`	Numerical value: The radius to define the initial circle (use approx n/100)
`K`	Numerical value: How many clusters to simulate
`alpha`	Numerical value: How far to pull apart the clusters
`wobble`	Numerical value: The degree of noise to add to the sample co ordinates
`redp`	Numerical value: The fraction of samples to remove from one cluster
`print`	Logical flag: whether to print the PCA into current directory
`seed`	Numerical value: fixes the seed if you want to repeat results

Value

A list: containing 1) matrix with simulated data in it

Examples

```
res <- clustersim(225, 900, 8, 4, 0.75, 0.025, redp = NULL, print = TRUE, seed=123)
```

desx — *GBM clinical annotation data*

Description

This is the clinical annotation data from the GBM dataset, it contains the class of the tumour which is one of: classical, mesenchymal, neural, proneural. It is a data frame with 2 columns and 50 rows.

Author(s)

Chris John <chris.r.john86@gmail.com>

References

Verhaak, Roel GW, et al. "Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1." Cancer cell 17.1 (2010): 98-110.

featurefilter — *featurefilter: A function for filtering features based on the coefficient of variance*

Description

featurefilter: A function for filtering features based on the coefficient of variance

Usage

```
featurefilter(mydata, percentile = 10)
```

Arguments

mydata	Data frame: should have samples as columns and rows as features
percentile	Numerical value: the top X percent most variable features should be kept

Value

A filtered data frame

Examples

```
filtered <- featurefilter(mydata,percentile=10)
```

M3C *M3C: Monte Carlo Reference-based Consensus Clustering*

Description

This is the M3C core function, which is a reference-based consensus clustering algorithm. The basic idea is to use a multi-core enabled Monte Carlo simulation to drive the creation of a null distribution of stability scores. The Monte Carlo simulations maintains the feature correlation structure of the input data. Then the null distribution is used to compare the reference scores with the real scores and an empirical p value is calculated for every value of K to test the null hypothesis K=1. We derive the Relative Cluster Stability Index (RCSI) as a metric for selecting K, which is based on a comparison against the reference mean.

Usage

```
M3C(mydata, montecarlo = TRUE, cores = 1, iters = 100, maxK = 10,
  des = NULL, ref_method = c("reverse-pca", "chol"), repsref = 100,
  repsreal = 100, clusteralg = c("pam", "km", "spectral", "hc"),
  distance = "euclidean", pacx1 = 0.1, pacx2 = 0.9,
  printres = FALSE, printheatmaps = FALSE, showheatmaps = FALSE,
  seed = NULL, removeplots = FALSE, dend = FALSE, silent = FALSE,
  doanalysis = FALSE, analysistype = c("survival", "kw", "chi"),
  variable = NULL)
```

Arguments

mydata	Data frame or matrix: Contains the data, with samples as columns and rows as features
montecarlo	Logical flag: whether to run the Monte Carlo simulation or not (recommended: TRUE)
cores	Numerical value: how many cores to split the monte carlo simulation over
iters	Numerical value: how many Monte Carlo iterations to perform (default: 100, recommended: 5-200)
maxK	Numerical value: the maximum number of clusters to test for, K (default: 10)
des	Data frame: contains annotation data for the input data for automatic reordering
ref_method	Character string: refers to which reference method to use (recommended: leaving as default)
repsref	Numerical value: how many resampling reps to use for reference (default: 100, recommended: 100-250)
repsreal	Numerical value: how many resampling reps to use for real data (default: 100, recommended: 100-250)
clusteralg	String: dictates which inner clustering algorithm to use for M3C
distance	String: dictates which distance metric to use for M3C (recommended: leaving as default)
pacx1	Numerical value: The 1st x co-ordinate for calculating the pac score from the CDF (default: 0.1)
pacx2	Numerical value: The 2nd x co-ordinate for calculating the pac score from the CDF (default: 0.9)

printres	Logical flag: whether to print all results into current directory
printheatmaps	Logical flag: whether to print all the heatmaps into current directory
showheatmaps	Logical flag: whether to show the heatmaps on screen
seed	Numerical value: fixes the seed if you want to repeat results, set the seed to 123 for example here
removeplots	Logical flag: whether to remove all plots
dend	Logical flag: whether to compute the dendrogram and p values for the optimal K or not
silent	Logical flag: whether to remove messages or not
doanalysis	Logical flag: whether to analyse the clinical variable supplied (univariate only)
analysistype	Character string: refers to which kind of statistical analysis to do on the data, survival, Kruskal-Wallis (kw), or chi-squared (chi)
variable	Character string: if not doing survival what is the dependant variable (column name) called in the data frame

Value

A list, containing: 1) the stability results and 2) all the output data (another list) 3) reference stability scores (see vignette for more details on how to easily access)

Examples

```
res <- M3C(mydata, cores=1, iters=100, ref_method = 'reverse-pca', montecarlo = TRUE,printres = FALSE,
maxK = 10, showheatmaps = FALSE, repsreal = 100, repsref = 100,printheatmaps = FALSE, seed = 123, des = desx)
```

---

mydata *GBM expression data*

Description

This is the expression data from the GBM dataset. It is a data frame with 50 columns and 1740 rows.

Author(s)

Chris John <chris.r.john86@gmail.com>

References

Verhaak, Roel GW, et al. "Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1." Cancer cell 17.1 (2010): 98-110.

pca *pca: A principal component analysis function*

Description

pca: A principal component analysis function

Usage

```
pca(mydata, K = FALSE, printres = FALSE, labels = FALSE,
  text = FALSE, axistextsize = 30, legendtextsize = 30,
  dotsize = 6, textlabelsize = 4)
```

Arguments

mydata	Data frame or matrix or M3C results object: if dataframe/matrix should have samples as columns and rows as features
K	Numerical value: if running on the M3C results object, which value was the optimal K?
printres	Logical flag: whether to print the PCA into current directory
labels	Character vector: if we want to just label with gender for example
text	Character vector: if we wanted to label the samples with text IDs to look for outliers
axistextsize	Numerical value: axis text size
legendtextsize	Numerical value: legend text size
dotsize	Numerical value: dot size
textlabelsize	Numerical value: text inside plot label size

Value

A PCA plot object

Examples

```
PCA <- pca(mydata)
```

tsne *tsne: A tsne function*

Description

tsne: A tsne function

Usage

```
tsne(mydata, K = FALSE, labels = FALSE, perplex = 15,
  printres = FALSE, seed = FALSE, axistextsize = 30,
  legendtextsize = 30, dotsize = 6, textlabelsize = 4)
```

Arguments

mydata	Data frame or matrix or M3C results (list) object: if dataframe/matrix should have samples as columns and rows as features
K	Numerical value: if running on the M3C results object, which value was the optimal K? Needs manual input from user.
labels	Factor: if we want to just display gender for example, only for when running without K parameter and with a matrix or data frame
perplex	Numerical value: this is the perplexity parameter for tsne, it usually requires adjusting for each dataset
printres	Logical flag: whether to print the plot into current directory
seed	Numerical value: to repeat the results exactly, setting seed is required
axistextsize	Numerical value: axis text size
legendtextsize	Numerical value: legend text size
dotsize	Numerical value: dot size
textlabelsize	Numerical value: text inside plot label size

Value

A tsne plot object

Examples

```
TSNE <- tsne(mydata,perplex=15)
```

Index