Package ‘cytofkit’

October 12, 2016

Type Package

Title cytofkit: an integrated mass cytometry data analysis pipeline

Version 1.4.10

Date 2016-09-27

Author Jinmiao Chen, Hao Chen

Description An integrated mass cytometry data analysis pipeline that enables simultaneous illustration of cellular diversity and progression.

License Artistic-2.0

Maintainer Jinmiao Chen <Chen_Jinmiao@immunol.a-star.edu.sg>, Hao Chen <chen_hao@immunol.a-star.edu.sg>

biocViews FlowCytometry, GUI, CellBiology, Clustering, DimensionReduction, BiomedicalInformatics

Depends R (>= 2.10.0), ggplot2, plyr

VignetteBuilder knitr

Suggests knitr, RUnit, testthat, BiocGenerics

Imports tcltk, stats, Rtsne, e1071, flowCore, gplots, VGAM, reshape2, ggrepel, shiny, vegan, Biobase, doParallel, parallel, pdist, methods, destiny, FlowSOM(>= 1.4.0), igraph(>= 1.0.1), RANN(>= 2.5), Rcpp (>= 0.12.0)

LinkingTo Rcpp

RoxygenNote 5.0.1

NeedsCompilation yes

R topics documented:

---

cytofkit-package *cytofkit: an integrated mass cytometry data analysis pipeline*

Description

This package is designed to facilitate the analysis workflow of mass cytometry data with automatic subset identification and mapping of cellular progression. Both command line and a GUI client are provided for executing the workflow easily.

Details

This package integrates merging methods of multiple FCS files, dimension reduction methods (PCA, t-SNE and ISOMAP) and clustering methods (DensVM, ClusterX, and Rphenograph) for rapid subset detection. Analysis reuslts can be visualized and explored interactively using a specially designed shiny web APP, see `cytofkitShinyAPP`. Moreover, the method isomap is provided to map the cellular progression. This workflow can be easily executed with the main function `cytofkit` or through the GUI client `cytofkit_GUI`.

Pre-processing

Using function `cytof_exprsMerge`, one or multiple FCS files will be loaded via the *read.FCS* function in the *flowCore* package. Then transformation was applied to the expression value of selected markers of each FCS file. Transformation methods include `autoLgcl`, `cytofAsinh`, `logicle` and `arcsinh`, where `cytofAsinh` is the default.Then mutilple FCS files are merged using one of the merging methods `all`, `min`, `fixed` or `ceil`.

Dimensionality reduction

Using function `cytof_dimReduction`, t-Distributed Stochastic Neighbor Embedding (`tsne`) is suggested for dimensionality reduction although we also provide methods like `isomap` and `pca`.

Cluster

Using function cytof_cluster, three cluster method are provided, DensVM, ClusterX Rphenograph and FlowSOM. DensVM, densityClustX are performend on the dimension reduced data, while Rphenograph works directed on the high dimensional expression data. Method FlowSOM is integrated from FlowSOM package (https://bioconductor.org/packages/release/bioc/html/FlowSOM.html).

Post-processing

- Using function cytof_clusterPlot to visualize the cluster results in a catter plot, in which dots represent cells, colours indicate their assigned clusters and point shapes represent their belonging samples.

- Using function cytof_heatmap to generate heat map to visualize the mean expression of every marker in every cluster. This heat maps is useful to interrogate marker expression to identify each cluster's defining markers.

- Using function cytof_progressionPlot to visualize the expression patter of selected markers against the estimated cellular progression order.

- Using function cytof_addToFCS to add any dimension reduced data, cluster results, progression data into the original FCS files, new FCS files will be saved for easy checking with other softwares like FlowJo.

All the above post processing can be automatically implemented and saved using one function cytof_writeResults.

Author(s)

Hao Chen, Jinmiao Chen

References

http://signbioinfo.github.io/cytofkit/

See Also

cytofkit, cytofkit_GUI

Examples

```
## Run on GUI
#cytofkit_GUI()  # remove the hash symbol to launch the GUI

## Run on command
dir <- system.file('extdata',package='cytofkit')
file <- list.files(dir, pattern='.fcs$', full=TRUE)
parameters <- list.files(dir, pattern='.txt$', full=TRUE)
## remove the hash symbol to run the following command
#cytofkit(fcsFile = file, markers = parameters)

## Checking the vignettes for more details
if(interactive()) browseVignettes(package = 'cytofkit')
```

ClusterX *Fast clustering by automatic search and find of density peaks*

Description

This package implement the clustering algorithm described by Alex Rodriguez and Alessandro Laio (2014) with improvements of automatic peak detection and parallel implementation

Usage

```
ClusterX(data, dimReduction = NULL, outDim = 2, dc, gaussian = TRUE,
  alpha = 0.001, detectHalos = FALSE, SVMhalos = FALSE,
  parallel = FALSE, nCore = 4)
```

Arguments

`data`	A data matrix for clustering.
`dimReduction`	Dimenionality reduciton method.
`outDim`	Number of dimensions will be used for clustering.
`dc`	Distance cutoff value.
`gaussian`	If apply gaussian to esitmate the density.
`alpha`	Signance level for peak detection.
`detectHalos`	If detect the halos.
`SVMhalos`	Run SVM on cores to assign halos.
`parallel`	If run the algorithm in parallel.
`nCore`	Number of cores umployed for parallel compution.

Details

ClusterX works on low dimensional data analysis (Dimensionality less than 5). If input data is of high diemnsional, t-SNE is conducted to reduce the dimensionality.

Value

a list

Author(s)

Chen Hao

Examples

```
iris_unique <- unique(iris) # Remove duplicates
data <- as.matrix(iris_unique[,1:4])
ClusterXRes <- ClusterX(data)
```

cytofkit *cytofkit: an integrated mass cytometry data analysis pipeline*

Description

The main function to drive the cytofkit workflow.

Usage

```
cytofkit(fcsFiles = getwd(), markers = "parameter.txt",
  projectName = "cytofkit", ifCompensation = FALSE,
  transformMethod = c("autoLgcl", "cytofAsinh", "logicle", "arcsinh", "none"),
  mergeMethod = c("ceil", "all", "min", "fixed"), fixedNum = 10000,
  dimReductionMethod = c("tsne", "pca", "isomap"),
  clusterMethods = c("Rphenograph", "ClusterX", "DensVM", "FlowSOM", "NULL"),
  visualizationMethods = c("tsne", "pca", "isomap", "NULL"),
  progressionMethod = c("NULL", "diffusionmap", "isomap"), FlowSOM_k = 40,
  clusterSampleSize = 500, resultDir = getwd(), saveResults = TRUE,
  saveObject = TRUE, ...)
```

Arguments

`fcsFiles` It can be either the path where stores your FCS files or a vector of FCS file names.

`markers` It can be either a text file where contains the makers to be used for analysis or a vector of the marker names.

`projectName` A prefix that will be added to the names of all result files.

`ifCompensation` Either boolean value tells if do compensation (compensation matrix contained in FCS), or a compensation matrix to be applied.

`transformMethod` Data Transformation method, including `autoLgcl`, `cytofAsinh`, `logicle` and `arcsinh`, or `none` to avoid transformation.

`mergeMethod` When multiple fcs files are selected, cells can be combined using one of the four different methods including `ceil`, `all`, `min`, `fixed`. The default option is `ceil`, up to a fixed number (specified by `fixedNum`) of cells are sampled without replacement from each fcs file and combined for analysis. `all`: all cells from each fcs file are combined for analysis. `min`: The minimum number of cells among all the selected fcs files are sampled from each fcs file and combined for analysis. `fixed`: a fixed num (specified by fixedNum) of cells are sampled (with replacement when the total number of cell is less than fixedNum) from each fcs file and combined for analysis.

`fixedNum` The fixed number of cells to be extracted from each FCS file.

`dimReductionMethod` The method used for dimensionality reduction, including `tsne`, `pca` and `isomap`.

`clusterMethods` The clustering method(s) used for subpopulation detection, including `DensVM`, `ClusterX`, `Rphenograph` and `FlowSOM`. Multiple selection are accepted.

`visualizationMethods`
The method(s) used for visualize the cluster data, including `tsne`, `pca` and `isomap`. Multiple selection are accepted.

`progressionMethod`
Use the first ordination score of `isomap` to estimated the preogression order of cells, choose `NULL` to ignore.

`FlowSOM_k` Number of clusters for meta clustering in FlowSOM.

`clusterSampleSize`
The uniform size of each cluster.

`resultDir` The directory where result files will be generated.

`saveResults` If save the results, and the post-processing results including scatter plot, heatmap, and statistical results.

`saveObject` Save the resutls into RData objects for loading back to R for further analysis

`...` Other arguments passed to `cytof_exprsExtract`

Details

`cytofkit` works as the main funciton to perform the analysis of one or multiple FCS files. The workflow contains data merging from multiple FCS file, expression data transformation, dimensionality reduction with `PCA`, `isomap` or `tsne` (default), clustering analysis with methods includes `DensVM`, `ClusterX`, `Rphenograph)` and `FlowSOM` for subpopulation detection, and estimation of cellular progression using `isomap`. The analysis results can be visualized using scatter plot, heatmap plot or progression plot. Dimension reduced data and cluster labels will be saved back to new copies of FCS files. By default the analysis results will be automatically saved under `resultDir` for further annotation. Moreover An interactive web application is provided for interactive exploration of the analysis results, see `cytofkitShinyAPP`.

Value

a list containing `expressionData`, `dimReductionMethod`, `visualizationMethods`, `dimReducedRes`, `clusterRes`, `progressionRes`, `projectName`, `rawFCSdir` and `resultDir`. If choose 'saveResults = TRUE', results will be saved into files under `resultDir`.

Author(s)

Hao Chen, Jinmiao Chen

References

`http://signbioinfo.github.io/cytofkit/`

See Also

`cytofkit`, `cytofkit_GUI`, `cytofkitShinyAPP`

Examples

```
dir <- system.file('extdata',package='cytofkit')
file <- list.files(dir, pattern='.fcs$', full=TRUE)
parameters <- list.files(dir, pattern='.txt$', full=TRUE)
## remove the hash symbol to run the following command
#cytofkit(fcsFile = file, markers = parameters)
```

---

cytofkitNews *check the package update news*

Description

check the package update news

Usage

```
cytofkitNews()
```

---

cytofkitShinyAPP *A Shiny app to interactively visualize the analysis results*

Description

Load the RData object saved by cytofkit, explore the analysis results with interactive control

Usage

```
cytofkitShinyAPP()
```

Author(s)

Hao Chen

Examples

```
if (interactive()) cytofkit::cytofkitShinyAPP()
```

cytofkit_GUI *The user friendly GUI client for* cytofkit-package

Description

This GUI provides an easy way for CyToF data analysis using cytofkit package. Main parameters for running 'cytofkit' were integrated in this GUI, and each parameter has a help button to show the instruction. cytofkit analysis will be launched after submitting.

Usage

```
cytofkit_GUI()
```

Value

the GUI for cytofkit-package

Author(s)

Hao Chen

References

http://signbioinfo.github.io/cytofkit/

See Also

cytofkit-package, cytofkit

Examples

```
#cytofkit_GUI()  # remove the hash symbol to run
```

cytof_addToFCS *Add data to the original FCS files*

Description

Store the new dimension transformed data and cluster data into the exprs matrix in new fcs files under analyzedFCSdir

Usage

```
cytof_addToFCS(data, rawFCSdir, analyzedFCSdir, transformed_cols = c("tsne_1",
  "tsne_2"), cluster_cols = c("cluster"), inLgclTrans = TRUE)
```

Arguments

`data` The new data matrix to be added in.

`rawFCSdir` The directory containing the original fcs files.

`analyzedFCSdir` The directory to store the new fcs files.

`transformed_cols` the column name of the dimension transformend data in `data`.

`cluster_cols` the column name of the cluster data in `data`.

`inLgclTrans` Boolean value decides if apply the inverse lgcl transformation to the data before saving

Value

new fcs files stored under `analyzedFCSdir`

`cytof_cluster` *Subset detection by clustering*

Description

Apply clustering algorithms to detect cell subsets. `DensVM` and `ClusterX` clustering is based on the transformend ydata and use xdata to train the model. While `Rphenograph` directly works on the high dimemnional xdata. `FlowSOM` is integrated from FlowSOM pacakge (https://bioconductor.org/packages/release/bioc/html/Flo

Usage

```
cytof_cluster(ydata = NULL, xdata = NULL, method = c("Rphenograph",
  "ClusterX", "DensVM", "FlowSOM", "NULL"), FlowSOM_k = 40)
```

Arguments

`ydata` A matrix of the dimension reduced data.

`xdata` A matrix of the expression data.

`method` Cluster method including `DensVM`, `densityClustX`, `Rphenograph` and `FlowSOM`.

`FlowSOM_k` Number of clusters for meta clustering in FlowSOM.

Value

a vector of the clusters assigned for each row of the ydata

Examples

```
d<-system.file('extdata', package='cytofkit')
fcsFile <- list.files(d, pattern='.fcs$', full=TRUE)
parameters <- list.files(d, pattern='.txt$', full=TRUE)
markers <- as.character(read.table(parameters, sep = "\t", header = TRUE)[, 1])
xdata <- cytof_exprsMerge(fcsFile, markers = markers, mergeMethod = 'fixed', fixedNum = 100)
ydata <- cytof_dimReduction(xdata, method = "tsne")
clusters <- cytof_cluster(ydata, xdata, method = "ClusterX")
```

`cytof_clusterPlot` *Scatter plot of the cluster results*

Description

Dot plot visualization of the cluster results, with color indicating different clusters, and shape of different samples.

Usage

```
cytof_clusterPlot(data, xlab, ylab, cluster, sample, title = "cluster",
  type = 1, point_size = NULL, addLabel = TRUE, labelSize = 10,
  sampleLabel = TRUE, labelRepel = FALSE, fixCoord = TRUE)
```

Arguments

`data` The data frame of cluster results, which should contains at least xlab, ylab and cluster.

`xlab` The column name of the x axis in input `data`.

`ylab` The column name of the y axis in input `data`.

`cluster` The column name of cluster in input `data`.

`sample` the column name of the sample in input `data`.

`title` the title of the plot.

`type` plot type, 1 indicates combined plot, 2 indicated grid facet plot seperated by samples.

`point_size` the size of the dot.

`addLabel` Boolean, if add cluster labels.

`labelSize` the size of cluster labels.

`sampleLabel` If use point shapes to represent different samples.

`labelRepel` If repel the cluste labels to avoid label overlapping.

`fixCoord` If fix the Cartesian coordinates.

Value

the ggplot object of the scatter cluster plot.

Examples

```
x <- c(rnorm(100, mean = 1), rnorm(100, mean = 3), rnorm(100, mean = 9))
y <- c(rnorm(100, mean = 2), rnorm(100, mean = 8), rnorm(100, mean = 5))
c <- c(rep(1,100), rep(2,100), rep(3,100))
rnames <- paste(paste('sample_', c('A','B','C'), sep = ''), rep(1:100,each = 3), sep='_')
data <- data.frame(dim1 = x, dim2 = y, cluster = c)
rownames(data) <- rnames
data$sample <- "data"
cytof_clusterPlot(data, xlab="dim1", ylab="dim2", cluster="cluster", sample = "sample")
```

cytof_clusterStat *Statistics of the cluster relusts*

Description

Calculate the mean or median expression level of each marker for each cluster, or percentage of cell numbers of each cluster for each sample.

Usage

```
cytof_clusterStat(data, markers, cluster = "cluster", sample,
  statMethod = c("mean", "median", "percentage", "NULL"))
```

Arguments

data	Input data frame.
markers	The names of markers used for calcualtion.
cluster	The column name contatining cluster labels.
sample	The samples used for calculation.
statMethod	Statistics contatining mean, median or percentage.

Value

A matrix of the satatistics results

Examples

```
m1 <- c(rnorm(300, 10, 2), rnorm(400, 4, 2), rnorm(300, 7))
m2 <- c(rnorm(300, 4), rnorm(400, 16), rnorm(300, 10, 3))
m3 <- c(rnorm(300, 16), rnorm(400, 40, 3), rnorm(300, 10))
m4 <- c(rnorm(300, 7, 3), rnorm(400, 30, 2), rnorm(300, 10))
m5 <- c(rnorm(300, 27), rnorm(400, 40, 1),rnorm(300, 10))
c <- c(rep(1,300), rep(2,400), rep(3,300))
rnames <- paste(paste('sample_', c('A','B','C','D'), sep = ''),
rep(1:250,each = 4), sep='_')
exprs_cluster <- data.frame(cluster = c, m1 = m1, m2 = m2, m3 = m3, m4 = m4, m5 = m5)
row.names(exprs_cluster) <- rnames
cytof_clusterStat(data = exprs_cluster, cluster = "cluster", statMethod = "mean")
```

`cytof_colorPlot` *Plot the data with color-coded marker values*

Description

Plot the data with color-coded marker values

Usage

```
cytof_colorPlot(data, xlab, ylab, zlab, colorPalette = c("bluered", "topo",
  "heat", "terrain", "cm"), pointSize = 1, removeOutlier = TRUE)
```

Arguments

`data`	A dataframe containing the xlab, ylab and zlab.
`xlab`	The column name of data for x lab.
`ylab`	The column name of data for y lab.
`zlab`	The column name of data for z lab.
`colorPalette`	Color Palette.
`pointSize`	The size of the point.
`removeOutlier`	If remove the outliers.

Value

A ggplot object.

Examples

```
x <- c(rnorm(100, mean = 1), rnorm(100, mean = 3), rnorm(100, mean = 9))
y <- c(rnorm(100, mean = 2), rnorm(100, mean = 8), rnorm(100, mean = 5))
c <- rnorm(300, 10, 5)
data <- data.frame(dim1 = x, dim2 = y, marker = c)
cytof_colorPlot(data = data, xlab = "dim1", ylab = "dim2", zlab = "marker")
```

`cytof_dimReduction` *Dimension reduction for high dimensional data*

Description

Apply dimension reduction on the cytof expression data, with method `pca`, `tsne`, `diffusionmap` or `isomap`.

Usage

```
cytof_dimReduction(data, method = c("tsne", "pca", "isomap", "diffusionmap",
  "NULL"), distMethod = "euclidean", out_dim = 2, tsneSeed = 42,
  isomap_k = 5, isomap_ndim = NULL, isomapFragmentOK = TRUE, ...)
```

Arguments

data	Input expression data matrix.
method	Method chosed for dimension reduction, must be one of isomap, pca , diffusionmap or tsne.
distMethod	Method for distance calcualtion, default is "euclidean", other choices like "manhattan", "cosine", "rankcor"....
out_dim	The dimensionality of the output.
tsneSeed	Set a seed if you want reproducible t-SNE results.
isomap_k	Number of shortest dissimilarities retained for a point, parameter for isomap method.
isomap_ndim	Number of axes in metric scaling, parameter for isomap method.
isomapFragmentOK	What to do if dissimilarity matrix is fragmented, parameter for isomap method.
...	Other parameters passed to the method, check Rtsne, DiffusionMap, isomap.

Value

a matrix of the dimension reducted data, with colnames method_ID, and rownames same as the input data.

Examples

```
data(iris)
in_data <- iris[, 1:4]
out_data <- cytof_dimReduction(in_data, method = "tsne")
```

cytof_exprsExtract *Extract the expression data from a FCS file with preprocessing*

Description

Extract the FCS expresssion data with preprocessing of compensation (for FCM data only) and transformation. Transformtion methods includes autoLgcl, cytofAsinh, logicle (customizable) and arcsinh (customizable).

Usage

```
cytof_exprsExtract(fcsFile, verbose = FALSE, comp = FALSE, markers = NULL,
  transformMethod = c("autoLgcl", "cytofAsinh", "logicle", "arcsinh", "none"),
  scaleTo = NULL, q = 0.05, l_w = 0.1, l_t = 4000, l_m = 4.5,
  l_a = 0, a_a = 1, a_b = 1, a_c = 0)
```

Arguments

`fcsFile` The name of the FCS file.

`verbose` Boolean value detecides if print the massage details of FCS loading.

`comp` Either boolean value tells if do compensation (compensation matrix contained in FCS), or a compensation matrix to be applied.

`markers` Selected markers for analysis, either marker names/descriptions or marker IDs.

`transformMethod`
Data Transformation method, including `autoLgcl`, `cytofAsinh`, `logicle` and `arcsinh`, or `none` to avoid transformation.

`scaleTo` Scale the expression to a specified range c(a, b), default is NULL.

`q` quantile of negative values removed for auto w estimation, default is 0.05, parameter for autoLgcl transformation.

`l_w` Linearization width in asymptotic decades, parameter for logicle transformation.

`l_t` Top of the scale data value, parameter for logicle transformation.

`l_m` Full width of the transformed display in asymptotic decades, parameter for logicle transformation.

`l_a` Additional negative range to be included in the display in asymptotic decades, parameter for logicle transformation.

`a_a` Positive double that corresponds to the base of the arcsinh transformation, `arcsinh` = asinh(a + b * x) + c).

`a_b` Positive double that corresponds to a scale factor of the arcsinh transformation, `arcsinh` = asinh(a + b * x) + c).

`a_c` Positive double that corresponds to another scale factor of the arcsinh transformation, `arcsinh` = asinh(a + b * x) + c).

Value

A transformend expression data matrix with selected markers, row names added as `filename_cellID`, column mamed added as `name<desc>`.

Examples

```
d <- system.file('extdata',package='cytofkit')
fcsFile <- list.files(d,pattern='.fcs$',full=TRUE)
parameters <- list.files(d, pattern='.txt$', full=TRUE)
markers <- as.character(read.table(parameters, sep = "\t", header = TRUE)[, 1])
transformed <- cytof_exprsExtract(fcsFile, markers = markers)
```

cytof_exprsMerge *Merge the expression matrix from multiple FCS files with preprocessing*

Description

Apply preprocessing on each FCS file including compensation (for FCM data only) and transformation with selected markers, then expression matrix are extracted and merged using one of the methods, `all`, `min`, `fixed` or `ceil`

Usage

```
cytof_exprsMerge(fcsFiles, comp = FALSE, markers = NULL,
  transformMethod = c("autoLgcl", "cytofAsinh", "logicle", "arcsinh", "none"),
  scaleTo = NULL, mergeMethod = c("ceil", "all", "fixed", "min"),
  fixedNum = 10000, sampleSeed = 123, ...)
```

Arguments

`fcsFiles` A vector of FCS file names.

`comp` Either boolean value tells if do compensation (compensation matrix contained in FCS), or a compensation matrix to be applied.

`markers` Selected markers for analysis, either marker names/descriptions or marker IDs.

`transformMethod` Data Transformation method, including `autoLgcl`, `cytofAsinh`, `logicle` and `arcsinh`, or `none` to avoid transformation.

`scaleTo` Scale the expression to a specified range c(a, b), default is NULL.

`mergeMethod` Merge method for mutiple FCS expression data. cells can be combined using one of the four different methods including `ceil`, `all`, `min`, `fixed`. The default option is `ceil`, up to a fixed number (specified by `fixedNum`) of cells are sampled without replacement from each fcs file and combined for analysis. `all`: all cells from each fcs file are combined for analysis. `min`: The minimum number of cells among all the selected fcs files are sampled from each fcs file and combined for analysis. `fixed`: a fixed num (specified by fixedNum) of cells are sampled (with replacement when the total number of cell is less than fixedNum) from each fcs file and combined for analysis.

`fixedNum` The fixed number of cells to be extracted from each FCS file.

`sampleSeed` A sampling seed for reproducible expression matrix merging.

`...` Other arguments passed to `cytof_exprsExtract`

Value

A matrix containing the merged expression data, with selected markers, row names added as `filename_cellID`, column mamed added as `name<desc>`.

See Also

cytof_exprsExtract

Examples

```
d<-system.file('extdata',package='cytofkit')
fcsFiles <- list.files(d,pattern='.fcs$',full=TRUE)
parameters <- list.files(d, pattern='.txt$', full=TRUE)
markers <- as.character(read.table(parameters, sep = "\t", header = TRUE)[, 1])
merged <- cytof_exprsMerge(fcsFiles, markers = markers)
```

cytof_heatmap *Heatmap plot of cluster mean value results*

Description

Heatmap plot of cluster mean value results

Usage

```
cytof_heatmap(data, baseName = "Cluster", scaleMethod = "none",
  cex_row_label = NULL, cex_col_label = NULL, key.par = list(mgp = c(1.5,
  0.5, 0), mar = c(2.5, 2.5, 3, 1.5)), keysize = 1.4, margins = c(5, 5))
```

Arguments

data	a matrix with rownames and colnames
baseName	The name as a prefix in the title of the heatmap.
scaleMethod	Method indicating if the values should be centered and scaled in either the row direction or the column direction, or none. The default is 'none'.
cex_row_label	Text size for row labels.
cex_col_label	Text size for column labels.
key.par	graphical parameters for the color key.
keysize	numeric value indicating the size of the key.
margins	numeric vector of length 2 containing the margins (see par(mar= *)) for column and row names, respectively.

Value

a heatmap object from gplots

Examples

```
m1 <- c(rnorm(300, 10, 2), rnorm(400, 4, 2), rnorm(300, 7))
m2 <- c(rnorm(300, 4), rnorm(400, 16), rnorm(300, 10, 3))
m3 <- c(rnorm(300, 16), rnorm(400, 40, 3), rnorm(300, 10))
m4 <- c(rnorm(300, 7, 3), rnorm(400, 30, 2), rnorm(300, 10))
m5 <- c(rnorm(300, 27), rnorm(400, 40, 1),rnorm(300, 10))
c <- c(rep(1,300), rep(2,400), rep(3,300))
rnames <- paste(paste('sample_', c('A','B','C','D'), sep = ''),
rep(1:250,each = 4), sep='_')
exprs_cluster <- data.frame(cluster = c, m1 = m1, m2 = m2, m3 = m3, m4 = m4, m5 = m5)
row.names(exprs_cluster) <- sample(rnames, 1000)
cluster_mean <- aggregate(. ~ cluster, data = exprs_cluster, mean)
rownames(cluster_mean) <- paste("cluster_", cluster_mean$cluster, sep = "")
cytof_heatmap(cluster_mean[, -which(colnames(cluster_mean) == "cluster")])
```

cytof_progression *Progression estimation of cytof expression data*

Description

Infer the progression based on the relationship of cell subsets estimated using ISOMAP or Diffusion map.

Usage

```
cytof_progression(data, cluster, method = c("diffusionmap", "isomap", "NULL"),
  distMethod = "euclidean", out_dim = 2, clusterSampleMethod = c("ceil",
  "all", "fixed", "min"), clusterSampleSize = 500, sampleSeed = 123)
```

Arguments

`data` Expression data matrix.

`cluster` A vector of cluster results for the data.

`method` Method for estimation of cell progression, isomap or diffusionmap.

`distMethod` Method for distance calcualtion, default is "euclidean", other choices like "manhattan", "cosine", "rankcor".

`out_dim` Number of transformed dimenions choosed for output.

`clusterSampleMethod`
Cluster sampling method including `ceil`, `all`, `min`, `fixed`. The default option is `ceil`, up to a fixed number (specified by `fixedNum`) of cells are sampled without replacement from each cluster and combined for analysis. `all`: all cells from each cluster are combined for analysis. `min`: The minimum number of cells among all clusters are sampled from cluster and combined for analysis. `fixed`: a fixed num (specified by fixedNum) of cells are sampled (with replacement when the total number of cell is less than fixedNum) from each cluster and combined for analysis.

clusterSampleSize
: The number of cells to be sampled from each cluster.

sampleSeed
: The seed for random down sample of the clusters.

Value

a list includes sampleData, sampleCluster and progressionData.

Examples

```
d<-system.file('extdata', package='cytofkit')
fcsFile <- list.files(d, pattern='.fcs$', full=TRUE)
parameters <- list.files(d, pattern='.txt$', full=TRUE)
markers <- as.character(read.table(parameters, sep = "\t", header = TRUE)[, 1])
xdata <- cytof_exprsMerge(fcsFile, markers = markers, mergeMethod = 'fixed', fixedNum = 2000)
clusters <- cytof_cluster(xdata = xdata, method = "Rphenograph")
prog <- cytof_progression(data = xdata, cluster = clusters, clusterSampleSize = 100)
d <- as.data.frame(cbind(prog$progressionData, cluster = factor(prog$sampleCluster)))
cytof_clusterPlot(data =d, xlab = "diffusionmap_1", ylab="diffusionmap_2", cluster = "cluster", sampleLabel = FAl
```

cytof_progressionPlot *Progression plot*

Description

Plot the expression trend along the estimated cell progressing order

Usage

```
cytof_progressionPlot(data, markers, clusters, orderCol = "isomap_1",
  clusterCol = "cluster", reverseOrder = FALSE, addClusterLabel = TRUE,
  clusterLabelSize = 5, segmentSize = 0.5, min_expr = NULL,
  trend_formula = "expression ~ sm.ns(Pseudotime, df=3)")
```

Arguments

data	The data frame for progression plot.
markers	The column names of the selected markers for visualization.
clusters	Selecte clusters for plotting, defauls select all.
orderCol	The column name of the estimated cell progression order.
clusterCol	The column name of the cluster results.
reverseOrder	If reverse the value of orderCol.
addClusterLabel	If add the cluster label on the plot.
clusterLabelSize	The size of the cluster label.

`segmentSize` The size of the cluster label arrow.

`min_expr` the threshold of the minimal expression value for markers.

`trend_formula` a symbolic description of the model to be fit.

Value

a ggplot2 object

Examples

```
m1 <- c(rnorm(300, 10, 2), rnorm(400, 4, 2), rnorm(300, 7))
m2 <- c(rnorm(300, 4), rnorm(400, 16), rnorm(300, 10, 3))
m3 <- c(rnorm(300, 16), rnorm(400, 40, 3), rnorm(300, 10))
m4 <- c(rnorm(300, 7, 3), rnorm(400, 30, 2), rnorm(300, 10))
m5 <- c(rnorm(300, 27), rnorm(400, 40, 1),rnorm(300, 10))
c <- c(rep(1,300), rep(2,400), rep(3,300))
rnames <- paste(paste('sample_', c('A','B','C','D'), sep = ''),
rep(1:250,each = 4), sep='_')
exprs_cluster <- data.frame(cluster = c, m1 = m1, m2 = m2, m3 = m3, m4 = m4, isomap_1 = m5)
row.names(exprs_cluster) <- sample(rnames, 1000)
cytof_progressionPlot(exprs_cluster, markers = c("m1","m2","m3","m4"))
```

---

cytof_writeResults *Save the cytofkit analysis results*

Description

Save analysis results from cytofkit main function to RData, csv files and PDF files and add them to a new copy of FCS files.

Usage

```
cytof_writeResults(analysis_results, projectName, saveToRData = TRUE,
  saveToFCS = TRUE, saveToFiles = TRUE, resultDir, rawFCSdir,
  inverseLgclTrans = TRUE)
```

Arguments

`analysis_results` result data from output of `cytofkit`

`projectName` a prefix that will be added to the names of result files.

`saveToRData` boolean value determines if save the results object into RData file, for loading back to R and to shiny APP.

`saveToFCS` boolean value determines if save the results back to the FCS files, new FCS files will be generated under folder XXX_analyzedFCS.

`saveToFiles` boolean value determines if parse the results and automatically save to csv files and pdf figures.

`resultDir` the directory where result files will be generated.

`rawFCSdir` the directory that contains fcs files to be analysed.

`inverseLgclTrans` boolean if inverse logicle transform the cluster cor1 and cor2 channels.

Value

save all results in the `resultDir`

See Also

`cytofkit`

Examples

```
d <- system.file('extdata',package='cytofkit')
f <- list.files(d, pattern='.fcs$', full=TRUE)
p <- list.files(d, pattern='.txt$', full=TRUE)
#tr <- cytofkit(fcsFile=f,markers=p,projectName='t',saveResults=FALSE)
#cytof_write_results(tr,projectName = 'test',resultDir=d,rawFCSdir =d)
```

`DensVM` *Density-based local maxima cluster with SVM*

Description

Density-based local maxima peak finding, subpopulation assigning with the power of SVM

Usage

```
DensVM(ydata, xdata)
```

Arguments

`ydata` a matrix of the dimension reduced(transformed) data

`xdata` a matrix of the expression data

Value

a list contains a matrix `peakdata` of the peak numbers with different kernel bandwidth, and a matrix `clusters` of the cluster results

Author(s)

Chen Jinmiao

Examples

```
d<-system.file('extdata',package='cytofkit')
fcsFile <- list.files(d,pattern='.fcs$',full=TRUE)
xdata <- cytof_exprsMerge(fcsFile, mergeMethod = 'fixed', fixedNum = 100)
ydata <- cytof_dimReduction(xdata)
#clusters <- DensVM(ydata, xdata)
```

getParameters_GUI *GUI for marker selection*

Description

Extract the markers from the fcsfiles

Usage

```
getParameters_GUI(fcsFile, rawFCSdir)
```

Arguments

fcsFile	The name of the FCS file
rawFCSdir	The path of the FCS file

Examples

```
#getParameters_GUI()
```

launchShinyAPP_GUI *GUI for launching shiny APP*

Description

A shiny APP for interactive exploration of the analysis results

Usage

```
launchShinyAPP_GUI(message = "cytofkit", dir = getwd())
```

Arguments

message	A message to determine if open the shiny APP
dir	Result direcroty.

Examples

```
# launchShinyAPP_GUI()
```

Rphenograph *RphenoGraph clustering*

Description

R implementation of the phenograph algorithm

Usage

```
Rphenograph(data, k = 30)
```

Arguments

`data`	Input data matrix.
`k`	Number of nearest neighbours, default is 30.

Details

A simple R implementation of the phenograph [PhenoGraph](http://www.cell.com/cell/abstract/S0092-8674(15)00637-6) algorithm, which is a clustering method designed for high-dimensional single-cell data analysis. It works by creating a graph ("network") representing phenotypic similarities between cells by calclating the Jaccard coefficient between nearest-neighbor sets, and then identifying communities using the well known [Louvain method](https://sites.google.com/site/findcommunities/) in this graph.

Value

a communities object, the operations of this class contains:

`print`	returns the communities object itself, invisibly.
`length`	returns an integer scalar.
`sizes`	returns a numeric vector.
`membership`	returns a numeric vector, one number for each vertex in the graph that was the input of the community detection.
`modularity`	returns a numeric scalar.
`algorithm`	returns a character scalar.
`crossing`	returns a logical vector.
`is_hierarchical`	returns a logical scalar.
`merges`	returns a two-column numeric matrix.
`cut_at`	returns a numeric vector, the membership vector of the vertices.
`as.dendrogram`	returns a dendrogram object.
`show_trace`	returns a character vector.
`code_len`	returns a numeric scalar for communities found with the InfoMAP method and NULL for other methods.
`plot`	for communities objects returns NULL, invisibly.

Author(s)

Chen Hao

References

Jacob H. Levine and et.al. Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. Cell, 2015.

Examples

```
iris_unique <- unique(iris) # Remove duplicates
data <- as.matrix(iris_unique[,1:4])
Rphenograph_out <- Rphenograph(data, k = 45)
```

Index