Package ‘mineCitrus’

May 8, 2026

Type Package

Title Extract and Analyze Median Molecule Intensity from 'citrus' Output

Version 1.0.0

Date 2018-07-05

Author Allison Throm

Maintainer Allison Throm <`throm@wustl.edu`>

Description Citrus is a computational technique developed for the analysis of high dimensional cytometry data sets. This package extracts, statistically analyzes, and visualizes marker expression from 'citrus' data. This code was used to generate data for Figures 3 and 4 in the forthcoming manuscript: Throm et al. “Identification of Enhanced Interferon-Gamma Signaling in Polyarticular Juvenile Idiopathic Arthritis with Mass Cytometry”, JCI-Insight. For more information on Citrus, please see: Bruggner et al. (2014) <doi:10.1073/pnas.1408792111>. To download the 'citrus' package, please see <`https://github.com/nolanlab/citrus`>.

License GPL-2

Encoding UTF-8

LazyData true

Imports ggplot2

Depends R (>= 2.10)

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-05 13:40:03 UTC

Contents

`allmeds` *Gets matrices of medians for each individual sample for all measured parameters for all clusters*

Description

Gets matrices of medians for each individual sample for all measured parameters for all clusters

Usage

```
allmeds(citrus.combinedFCSSet, citrus.foldClustering, citrus.foldFeatureSet)
```

Arguments

`citrus.combinedFCSSet`	loaded from citrusClustering.RData file generated by Citrus run
`citrus.foldClustering`	loaded from citrusClustering.RData file generated by Citrus run
`citrus.foldFeatureSet`	computed from first two variables using citrus.calculateFoldFeatureSet function from citrus package

Value

Returns a list with each element corresponding to a matrix (rows as samples, columns as measured parameters) for a different cluster (for the minimum threshold specified)

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
              citrus.foldClustering=citrus.foldClustering,
              citrus.foldFeatureSet=citrus.foldFeatureSet)
```

`citrus.combinedFCSSet` *Cytometry data set for example of Citrus data set from nolanlab/citrus*

Description

A dataset containing the a simple example of cytometry data

Usage

```
citrus.combinedFCSSet
```

Format

A large citrus.combinedFCSSet object with 5 elements:

data Toy data set for cytometry

fileChannelNames Names of channels for measured parameters included in toy cytmetry data set

fileIds ID numbers for each file included in toy cytometry data set

fileNames Names of files included in toy cytmetry data set

fileReagentNames Names of measured channels in toy cytmetry data set ...

Source

`https://github.com/nolanlab/citrus`

`citrus.foldClustering` *Clustering data for example of Citrus data set from nolanlab/citrus*

Description

A dataset containing the clustering of different cell groups

Usage

```
citrus.foldClustering
```

Format

A large citrus.foldClustering object with 5 elements:

allClustering A list describing which events belong to which clusters

foldClustering A list describing which events belong to which clusters for each fold

foldMappingAssignments A list describing assignments with fold clustering

folds Descriptions of each data clustering

nFolds The number of times data is clustered ...

Source

`https://github.com/nolanlab/citrus`

`citrus.foldFeatureSet` *Correlation data for example of Citrus data set from nolanlab/citrus*

Description

A dataset containing the association of red and blue in clusters with different sample groups

Usage

```
citrus.foldFeatureSet
```

Format

A list with 8 elements:

allFeatures Data set for each sample for all markers and clusters

allLargeEnoughClusters Vector of clusters meeting size threshold

foldFeatures Data for each fold clustering

foldLargeEnoughClusters Clusters meeting size threshold for each fold clustering

folds Descriptions of each data clustering

leftoutFeatures Data omitted from analyses

minimumClusterSizePercent Minimum size threshold to retain clusters in analysis

nFolds The number of times data is clustered ...

Source

`https://github.com/nolanlab/citrus`

`classclustermeds` *Gets matrices of medians for each individual sample for all measured parameters for all clusters*

Description

Gets matrices of medians for each individual sample for all measured parameters for all clusters

Usage

```
classclustermeds(citrus.foldFeatureSet, citrus.foldClustering,
  citrus.combinedFCSSet, groupsizes, meds)
```

Arguments

`citrus.foldFeatureSet`	computed from first two variables using citrus.calculateFoldFeatureSet function from citrus package
`citrus.foldClustering`	loaded from citrusClustering.RData file generated by Citrus run
`citrus.combinedFCSSet`	loaded from citrusClustering.RData file generated by Citrus run
`groupsizes`	list of sizeso f the groups run in Citrus, in order of the selection for citrus run
`meds`	The names of the columns from citrus.combinedFCSSet$data of interest to extract medians for

Value

Returns a list of matrices with columns corresponding to selected features and rows corresponding to sample groups; each list element corresponds to data for a different cluster

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
              citrus.foldClustering=citrus.foldClustering,
              citrus.foldFeatureSet=citrus.foldFeatureSet)
medians<-classclustermeds(citrus.foldFeatureSet,citrus.foldClustering,
                          citrus.combinedFCSSet,groupsizes=c(10,10),meds=meds)
```

`clustermeds` *Gets matrix of medians for desired measured features for all clusters meeting threshold requirements specified in Citrus*

Description

Gets matrix of medians for desired measured features for all clusters meeting threshold requirements specified in Citrus

Usage

```
clustermeds(citrus.foldFeatureSet, citrus.foldClustering, medsofinterest,
  citrus.combinedFCSSet)
```

Arguments

`citrus.foldFeatureSet`	computed from first two variables using citrus.calculateFoldFeatureSet function from citrus package
`citrus.foldClustering`	loaded from citrusClustering.RData file generated by Citrus run
`medsofinterest`	The names of the columns from citrus.combinedFCSSet$data of interest to extract medians for
`citrus.combinedFCSSet`	loaded from citrusClustering.RData file generated by Citrus run

Value

Returns a matrix with columns corresponding to selected features and rows corresponding to samples

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
medians<-clustermeds(citrus.foldFeatureSet=citrus.foldFeatureSet,
                     citrus.foldClustering=citrus.foldClustering,
                     medsofinterest=c("Red","Blue"),
                     citrus.combinedFCSSet=citrus.combinedFCSSet)
```

`difMarkerPlots` *Plot dot plots of features where both clusters are significantly different from the reference cluster without processing data before hand*

Description

Plot dot plots of features where both clusters are significantly different from the reference cluster without processing data before hand

Usage

```
difMarkerPlots(data, clusters, markers, diffclust, strat)
```

Arguments

`data`	output from call to allmeds function
`clusters`	clusterIDs of the desired clusters to compare and plot
`markers`	indices of the columns of the data matrix for features to be analyse
`diffclust`	clusterID of for cluster to statisticaly compare others to
`strat`	clusterIDs for stratifying clusters as indicated by Citrus results

Value

Dot plots for all features where both clusters are significantly different from the reference cluster

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
              citrus.foldClustering=citrus.foldClustering,
              citrus.foldFeatureSet=citrus.foldFeatureSet)
graphs<-difMarkerPlots(data=meds,clusters=c(19999,19972,19988),
                       markers=c(2,3),diffclust=19999,strat=19999)
```

`difMarkerPlots2` *Plot dot plots of features where one cluster is significantly different from the reference cluster without processing data before hand*

Description

Plot dot plots of features where one cluster is significantly different from the reference cluster without processing data before hand

Usage

```
difMarkerPlots2(data, clusters, markers, diffclust, strat)
```

Arguments

`data`	output from call to allmeds function
`clusters`	clusterIDs of the desired clusters to compare and plot
`markers`	indices of the columns of the data matrix for features to be analyse
`diffclust`	clusterID of for cluster to statisticaly compare others to
`strat`	clusterIDs for stratifying clusters as indicated by Citrus results

Value

Dot plots for all features where one cluster is significantly different from the reference cluster

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
             citrus.foldClustering=citrus.foldClustering,
             citrus.foldFeatureSet=citrus.foldFeatureSet)
graphs<-difMarkerPlots2(data=meds,clusters=c(19999,19972,19988),markers=c(2,3),
                        diffclust=19999,strat=19999)
```

`filterMarker` *Filters list of data matrices with columns corresponding to the measured parameters of interest*

Description

Filters list of data matrices with columns corresponding to the measured parameters of interest

Usage

```
filterMarker(clustdat, markers)
```

Arguments

`clustdat`	a list of data matrices with list elements corresponding to clusters and matrices of intensities of measured parameters
`markers`	Indices of the columns of parmeters to keep

Value

A list of data matrices with columns of data matrices only corresponding to measured parameters of interest

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
              citrus.foldClustering=citrus.foldClustering,
              citrus.foldFeatureSet=citrus.foldFeatureSet)
meds2<-filterMarker(clustdat=meds,markers=c(2,3))
```

`findclust` *Filters list to contain only desired clusters*

Description

Filters list to contain only desired clusters

Usage

```
findclust(data, clusters)
```

Arguments

`data`	a list of data matrices with list elements corresponding to clusters and matrices of intensities of measured parameters
`clusters`	indices of the clusters to retain

Value

A list of data matrices for the desired clusters

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
              citrus.foldClustering=citrus.foldClustering,
              citrus.foldFeatureSet=citrus.foldFeatureSet)
filteredmeds<-findclust(data=meds,clusters=c(19999,19972,19988))
```

`findSig` *Assesses significance of ANOVA and t-test results*

Description

Assesses significance of ANOVA and t-test results

Usage

```
findSig(posHocRes)
```

Arguments

`posHocRes`	results from a call to the posthoc function

Value

A dataframe indicating the significances of results

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
              citrus.foldClustering=citrus.foldClustering,
              citrus.foldFeatureSet=citrus.foldFeatureSet)
```

```
filteredmeds<-findclust(data=meds,clusters=c(19999,19972,19988))
meds2<-filterMarker(clustdat=filteredmeds,markers=c(2,3))
foranova<-processforanova(filtereddata=meds2)
ttests<-posthoc(processedDat=foranova,clustIDdif=19999)
sig<-findSig(posHocRes=ttests)
```

`plotdif` *Plot dot plots of features where both clusters are significantly different from the reference cluster*

Description

Plot dot plots of features where both clusters are significantly different from the reference cluster

Usage

```
plotdif(BJHdf, anovadata, strat)
```

Arguments

`BJHdf`	results of a call to findsig
`anovadata`	results of call to processforanova
`strat`	clusterIDs for clusters that are stratifying

Value

Dot plots for all features where both clusters are significantly different from the reference cluster

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
              citrus.foldClustering=citrus.foldClustering,
              citrus.foldFeatureSet=citrus.foldFeatureSet)
filteredmeds<-findclust(data=meds,clusters=c(19999,19972,19988))
meds2<-filterMarker(clustdat=filteredmeds,markers=c(2,3))
foranova<-processforanova(filtereddata=meds2)
ttests<-posthoc(processedDat=foranova,clustIDdif=19999)
sig<-findSig(posHocRes=ttests)
graphs<-plotdif(BJHdf=sig,anovadata=foranova,strat=19999)
```

plotdif2 *Plot dot plots of features where one cluster is significantly different from the reference cluster*

Description

Plot dot plots of features where one cluster is significantly different from the reference cluster

Usage

```
plotdif2(BJHdf, anovadata, strat)
```

Arguments

`BJHdf`	results of a call to findsig
`anovadata`	results of call to processforanova
`strat`	clusterIDs for clusters that are stratifying

Value

Dot plots for all features where one cluster is significantly different from the reference cluster

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
              citrus.foldClustering=citrus.foldClustering,
              citrus.foldFeatureSet=citrus.foldFeatureSet)
filteredmeds<-findclust(data=meds,clusters=c(19999,19972,19988))
meds2<-filterMarker(clustdat=filteredmeds,markers=c(2,3))
foranova<-processforanova(filtereddata=meds2)
ttests<-posthoc(processedDat=foranova,clustIDdif=19999)
sig<-findSig(posHocRes=ttests)
graphs<-plotdif2(BJHdf=sig,anovadata=foranova,strat=19999)
```

posthoc *Runs ANOVA and t-tests comparing clusters and markers in clusters*

Description

Runs ANOVA and t-tests comparing clusters and markers in clusters

Usage

```
posthoc(processedDat, clustIDdif)
```

Arguments

`processedDat`	data that has been processed using the processforanova function
`clustIDdif`	ID number of the cluster to compare the others to

Value

A list of t-test results for each of the comparisons

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
             citrus.foldClustering=citrus.foldClustering,
             citrus.foldFeatureSet=citrus.foldFeatureSet)
filteredmeds<-findclust(data=meds,clusters=c(19999,19972,19988))
meds2<-filterMarker(clustdat=filteredmeds,markers=c(2,3))
foranova<-processforanova(filtereddata=meds2)
ttests<-posthoc(processedDat=foranova,clustIDdif=19999)
```

`processforanova` *Processes cluster signaling data in form for statistical analysis*

Description

Processes cluster signaling data in form for statistical analysis

Usage

```
processforanova(filtereddata)
```

Arguments

`filtereddata`	a list with each element corresonding to a cluster of interest and matrices containing individual sample data for desired markers

Value

A dataframe sufficient for using the posthoc function to compute statistics

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
meds<-allmeds(citrus.combinedFCSSet=citrus.combinedFCSSet,
              citrus.foldClustering=citrus.foldClustering,
              citrus.foldFeatureSet=citrus.foldFeatureSet)
filteredmeds<-findclust(data=meds,clusters=c(19999,19972,19988))
foranova<-processforanova(filtereddata=filteredmeds)
```

`sortmat` *Reorders to rows (corresponding to different clusters) of a matrix of medians to a desired order*

Description

Reorders to rows (corresponding to different clusters) of a matrix of medians to a desired order

Usage

```
sortmat(mat, desiredorder)
```

Arguments

`mat`	matrix of median data
`desiredorder`	row labels from matrix in desired order

Value

Returns a matrix with rows rearranged in desired order

Examples

```
library(mineCitrus)
data("citrus.combinedFCSSet")
data("citrus.foldClustering")
data("citrus.foldFeatureSet")
medians<-clustermeds(citrus.foldFeatureSet=citrus.foldFeatureSet,
                     citrus.foldClustering=citrus.foldClustering,
                     medsofinterest=c("Red","Blue"),
                     citrus.combinedFCSSet=citrus.combinedFCSSet)
names<-rownames(medians)
names<-names[c(31,1:30)]
sortedmedians<-sortmat(mat=medians,desiredorder=names)
```

Index