Package 'ClusterFoldSimilarity'

November 14, 2025

Type Package

Title Calculate similarity of clusters from different single cell samples using foldchanges

Version 1.7.0

Description This package calculates a similarity coefficient using the fold changes of shared features (e.g. genes) among clusters of different samples/batches/datasets. The similarity coefficient is calculated using the dot-product (Hadamard product) of every pairwise combination of Fold Changes between a source cluster i of sample/dataset n and all the target clusters j in sample/dataset m

License Artistic-2.0

Encoding UTF-8

Imports methods, igraph, ggplot2, scales, BiocParallel, graphics, stats, utils, Matrix, cowplot, dplyr, reshape2, Seurat, SeuratObject, SingleCellExperiment, ggdendro

Suggests knitr, rmarkdown, kableExtra, scRNAseq, BiocStyle

RoxygenNote 7.2.3

biocViews SingleCell, Clustering, FeatureExtraction, GraphAndNetwork, GeneTarget, RNASeq

VignetteBuilder knitr

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

git_branch devel

git_last_commit d5231f7

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-11-14

Author Oscar Gonzalez-Velasco [cre, aut] (ORCID: <https://orcid.org/0000-0002-5054-8635>)

Maintainer Oscar Gonzalez-Velasco <oscargvelasco@gmail.com>

Contents

`clusterFoldSimilarity` *Calculate cluster similarity between clusters from different single cell samples.*

Description

'clusterFoldSimilarity()' returns a dataframe containing the best top similarities between all possible pairs of single cell samples.

Usage

```
clusterFoldSimilarity(
  scList = NULL,
  sampleNames = NULL,
  topN = 1,
  topNFeatures = 1,
  nSubsampling = 15,
  parallel = FALSE
)
```

Arguments

`scList`	List. A list of Single Cell Experiments or Seurat objects. At least 2 are needed. The objects are expected to have cluster or label groups set as identity class.
`sampleNames`	Character Vector. Specify the sample names, if not a number corresponding with its position on (scList).
`topN`	Numeric. Specifies the number of target clusters with best similarity to report for each cluster comparison (default 1). If set to Inf, then all similarity values from all possible pairs of clusters are returned.
`topNFeatures`	Numeric. Number of top features that explains the clusters similarity to report for each cluster comparison (default 1). If topN = Inf then topNFeatures is automatically set to 1.
`nSubsampling`	Numeric. Number of random sampling of cells to achieve fold change stability (default 15).
`parallel`	Boolean. Whether to use parallel computing using BiocParallel or not (default FALSE).

Details

This function will calculate a similarity coeficient using the fold changes of shared features (e.g.:genes for a single-cell RNA-Seq, peaks for ATAC-Seq) among clusters, or user-defined-groups, from different samples/batches. The similarity coeficient is calculated using the dotproduct of every pairwise combination of Fold Changes between a source cluster/group i from sample n and all the target clusters/groups in sample j.

Value

The function returns a DataFrame containing the best top similarities between all possible pairs of single cell samples. Column values are:

`similarityValue`	The top similarity value calculated between datasetL:clusterL and datasetR.
`w`	Weight associated with the similarity score value.
`datasetL`	Dataset left, the dataset/sample which has been used to be compared.
`clusterL`	Cluster left, the cluster source from datasetL which has been compared.
`datasetR`	Dataset right, the dataset/sample used for comparison against datasetL.
`clusterR`	Cluster right, the cluster target from datasetR which is being compared with the clusterL from datas
`topFeatureConserved`	The features (e.g.: genes, peaks...) that most contributed to the similarity between clusterL & cluste
`featureScore`	The similarity score contribution for the specific topFeatureConserved (e.g.: genes, peaks...).

Author(s)

Oscar Gonzalez-Velasco

Examples

```
if (requireNamespace("Seurat") & requireNamespace("SeuratObject")){
library(ClusterFoldSimilarity)
library(Seurat)
library(SeuratObject)
# data dimensions
nfeatures <- 2000; ncells <- 400
# single-cell 1
counts <- matrix(rpois(n=nfeatures * ncells, lambda=10), nfeatures)
rownames(counts) <- paste0("gene",seq(nfeatures))
colnames(counts) <- paste0("cell",seq(ncells))
colData <- data.frame(cluster=sample(c("Cluster1","Cluster2","Cluster3"),size = ncells,replace = TRUE),
                      row.names=paste0("cell",seq(ncells)))
seu1 <- SeuratObject::CreateSeuratObject(counts = counts, meta.data = colData)
Idents(object = seu1) <- "cluster"
# single-cell 2
```

```
counts <- matrix(rpois(n=nfeatures * ncells, lambda=20), nfeatures)
rownames(counts) <- paste0("gene",seq(nfeatures))
colnames(counts) <- paste0("cell",seq(ncells))
colData <- data.frame(cluster=sample(c("Cluster1","Cluster2","Cluster3","Cluster4"),size = ncells,replace = TRUE
                      row.names=paste0("cell",seq(ncells)))
seu2 <- SeuratObject::CreateSeuratObject(counts = counts, meta.data = colData)
Idents(object = seu2) <- "cluster"
# Create a list with the unprocessed single-cell datasets
singlecellObjectList <- list(seu1, seu2)

similarityTable <- clusterFoldSimilarity(scList=singlecellObjectList, sampleNames = c("sc1","sc2"))
head(similarityTable)
}
```

`findCommunitiesSimmilarity` *Find cell-group communities by constructing and clustering a directed graph using the similarity values calculated by ClusterFoldSimilarity()*

Description

'findCommunitiesSimmilarity()' Find communities by constructing and clustering graph using the similarity values calculated by ClusterFoldSimilarity().

Usage

```
findCommunitiesSimmilarity(similarityTable = NULL)
```

Arguments

`similarityTable`
: Dataframe. A table obtained from ClusterFoldSimilarity that contains the similarity values as a column "similarityValue" that represents the similarity of a source cluster to a target cluster.

Details

This function will group together nodes of the network into communities using the InfoMap community detection algorithm.

Value

This function returns a data frame with the community that each node of the network (cell groups defined by the user) belongs to, and plots a graph in which the nodes are clusters from a specific dataset, the edges represent the similarity and the direction of that similarity between clusters.

sample	The sample name.
group	The group/cluster from sample defined by the user.
community	Community group to which the sample:group belongs to.

Author(s)

Oscar Gonzalez-Velasco

Examples

```
if (requireNamespace("Seurat") & requireNamespace("SeuratObject")){
library(ClusterFoldSimilarity)
library(Seurat)
library(SeuratObject)
# data dimensions
nfeatures <- 2000; ncells <- 400
# single-cell 1
counts <- matrix(rpois(n=nfeatures * ncells, lambda=10), nfeatures)
rownames(counts) <- paste0("gene",seq(nfeatures))
colnames(counts) <- paste0("cell",seq(ncells))
colData <- data.frame(cluster=sample(c("Cluster1","Cluster2","Cluster3"),size = ncells,replace = TRUE),
                      row.names=paste0("cell",seq(ncells)))
seu1 <- SeuratObject::CreateSeuratObject(counts = counts, meta.data = colData)
Idents(object = seu1) <- "cluster"
# single-cell 2
counts <- matrix(rpois(n=nfeatures * ncells, lambda=10), nfeatures)
rownames(counts) <- paste0("gene",seq(nfeatures))
colnames(counts) <- paste0("cell",seq(ncells))
colData <- data.frame(cluster=sample(c("Cluster1","Cluster2","Cluster3","Cluster4"),size = ncells,replace = TRUE
                       row.names=paste0("cell",seq(ncells)))
seu2 <- SeuratObject::CreateSeuratObject(counts = counts, meta.data = colData)
Idents(object = seu2) <- "cluster"
# Create a list with the unprocessed single-cell datasets
singlecellObjectList <- list(seu1, seu2)

similarityTable <- clusterFoldSimilarity(scList = singlecellObjectList, sampleNames = c("sc1","sc2"))
head(similarityTable)
findCommunitiesSimmilarity(similarityTable=similarityTable)
}
```

`foldchangeComposition` *Calculate the dot product between all the possible combinations of foldchanges from diferent clusters.*

Description

'foldchangeComposition()' returns a dataframe containing the best top similarities between all possible pairs of single cell samples.

Usage

```
foldchangeComposition(root = NULL, comparative = NULL)
```

Arguments

root	Dataframe. Foldchanges between a source cluster and all the other clusters found on a sample.
comparative	Dataframe. Foldchanges between a cluster and all the other clusters found on a second sample to be compared with the (root) cluster foldchanges.

Details

This function will perform the dot product of each possible combination of foldchanges, by constructing two dataframes: one with the source cluster's foldchanges and the other with the foldchange values of a target sample's cluster. The computation of all the possible combinations is the hadamard product of the matrix.

Value

A dataframe containing the hadamard product of all the possible combinations of foldchanges.

---

pairwiseClusterFoldChange

Calculate the gene mean expression Fold Change between all possible combinations of clusters.

Description

'pairwiseClusterFoldChange()' returns a list of dataframes containing the pairwise fold changes between all combinations of cluster.

Usage

```
pairwiseClusterFoldChange(countData, clusters, nSubsampling, functToApply)
```

Arguments

countData	Matrix. Normalized counts containing gene expression.
clusters	Factor. A vector of corresponding cluster for each sample of (x).
nSubsampling	Numeric. Number of random sampling of cells to achieve fold change stability.

Details

This function will perform fold change estimation from the mean feature´s expression between all possible combination of clusters specified on colLabels inside the sc object. Bayesian Estimation of FoldChanges and Pseudocounts adapted from: Florian Erhard, Estimating pseudocounts and fold changes for digital expression measurements, Bioinformatics, Volume 34, Issue 23, December 2018, Pages 4054–4063, https://doi.org/10.1093/bioinformatics/bty471 Please consider citing also Erhard et. al. paper when using ClusterFoldSimilarity.

Value

A list of dataframes containing the pairwise fold changes between all combinations of cluster.

Author(s)

Oscar Gonzalez-Velasco

`plotClustersGraph` *Creates a graph plot using the similarity values calculated with ClusterFoldSimilarity().*

Description

'plotClustersGraph()' Creates a graph plot using the similarity values calculated with ClusterFoldSimilarity().

Usage

```
plotClustersGraph(similarityTable = NULL)
```

Arguments

`similarityTable`
: Dataframe. A table obtained from ClusterFoldSimilarity that contains the similarity values as a column "similarityValue" that represents the similarity of a source cluster to a target cluster.

Details

This function will calculate a similarity coeficient using the fold changes of shared genes among clusters of different samples/batches. The similarity coeficient is calculated using the dotproduct of every pairwise combination of Fold Changes between a source cluster i of sample n and all the target clusters in sample j.

Value

This function plots a graph in which the nodes are clusters from a specific dataset, the edges represent the similarity and the direction of that similarity between clusters.

Author(s)

Oscar Gonzalez-Velasco

Examples

```
if (requireNamespace("Seurat") & requireNamespace("SeuratObject")){
library(ClusterFoldSimilarity)
library(Seurat)
library(SeuratObject)
# data dimensions
nfeatures <- 2000; ncells <- 400
# single-cell 1
counts <- matrix(rpois(n=nfeatures * ncells, lambda=10), nfeatures)
rownames(counts) <- paste0("gene",seq(nfeatures))
colnames(counts) <- paste0("cell",seq(ncells))
colData <- data.frame(cluster=sample(c("Cluster1","Cluster2","Cluster3"),size = ncells,replace = TRUE),
                      row.names=paste0("cell",seq(ncells)))
seu1 <- SeuratObject::CreateSeuratObject(counts = counts, meta.data = colData)
Idents(object = seu1) <- "cluster"
# single-cell 2
counts <- matrix(rpois(n=nfeatures * ncells, lambda=10), nfeatures)
rownames(counts) <- paste0("gene",seq(nfeatures))
colnames(counts) <- paste0("cell",seq(ncells))
colData <- data.frame(cluster=sample(c("Cluster1","Cluster2","Cluster3","Cluster4"),size = ncells,replace = TRUE
                       row.names=paste0("cell",seq(ncells)))
seu2 <- SeuratObject::CreateSeuratObject(counts = counts, meta.data = colData)
Idents(object = seu2) <- "cluster"
# Create a list with the unprocessed single-cell datasets
singlecellObjectList <- list(seu1, seu2)

similarityTable <- clusterFoldSimilarity(scList = singlecellObjectList, sampleNames = c("sc1","sc2"))
head(similarityTable)
plotClustersGraph(similarityTable=similarityTable)
}
```

`similarityHeatmap` *Plot a heatmap of the similarity values obtained using cluster fold similarity*

Description

'similarityHeatmap()' returns a ggplot heatmap representing the similarity values between pairs of clusters as obtained from clusterFoldSimilarity.

Usage

```
similarityHeatmap(
  similarityTable = NULL,
```

```
  mainDataset = NULL,
  otherDatasets = NULL,
  highlightTop = TRUE
)
```

Arguments

similarityTable	A DataFrame containing the similarities between all possible pairs of single cell samples obtained with clusterFoldSimilarity using the option n_top=Inf.
mainDataset	Numeric. Specify the main dataset (y axis). It corresponds with the datasetL column from the similarityTable
otherDatasets	Numeric. Specify some specific dataset to be ploted along the mainDataset (x axis, default: all other datasets found on datasetR column from similarity_table).
highlightTop	Boolean. If the top 2 similarity values should be highlighted on the heatmap (default: TRUE)

Details

This function plots a heatmap using ggplot. It is intended to be used with the output table from clusterFoldSimilarity, which includes the columns: datasetL (the dataset used for comparison) datasetR (the dataset against datasetL has been contrasted), clusterL (clusters from datasetL), clusterR (clusters from datasetR) and the similarityValue.

Value

The function returns a heatmap ggplot object.

Author(s)

Oscar Gonzalez-Velasco

Examples

```
if (requireNamespace("Seurat") & requireNamespace("SeuratObject")){
library(ClusterFoldSimilarity)
library(Seurat)
library(SeuratObject)
# data dimensions
nfeatures <- 2000; ncells <- 400
# single-cell 1
counts <- matrix(rpois(n=nfeatures * ncells, lambda=10), nfeatures)
rownames(counts) <- paste0("gene",seq(nfeatures))
colnames(counts) <- paste0("cell",seq(ncells))
colData <- data.frame(cluster=sample(c("Cluster1","Cluster2","Cluster3"),size = ncells,replace = TRUE),
                      row.names=paste0("cell",seq(ncells)))
seu1 <- SeuratObject::CreateSeuratObject(counts = counts, meta.data = colData)
Idents(object = seu1) <- "cluster"
# single-cell 2
counts <- matrix(rpois(n=nfeatures * ncells, lambda=10), nfeatures)
```

```
rownames(counts) <- paste0("gene",seq(nfeatures))
colnames(counts) <- paste0("cell",seq(ncells))
colData <- data.frame(cluster=sample(c("Cluster1","Cluster2","Cluster3","Cluster4"),size = ncells,replace = TRUE
                      row.names=paste0("cell",seq(ncells)))
seu2 <- SeuratObject::CreateSeuratObject(counts = counts, meta.data = colData)
Idents(object = seu2) <- "cluster"
# Create a list with the unprocessed single-cell datasets
singlecellObjectList <- list(seu1, seu2)
# Using topN = Inf by default plots a heatmap using the similarity values:
similarityTableAll <- clusterFoldSimilarity(scList=singlecellObjectList, topN=Inf)
# Using the dataset 2 as a reference on the Y-axis of the heatmap:
similarityHeatmap(similarityTable=similarityTableAll, mainDataset=2, highlightTop=FALSE)
}
```

Index