Package 'CAMML'

November 13, 2023

Title Cell-Typing using Variance Adjusted Mahalanobis Distances with Multi-Labeling

Version 1.0.0

Maintainer Courtney Schiebout <courtney.t.schiebout.gr@dartmouth.edu>

Description Creates multi-label cell-types for single-cell RNA-sequencing data based on weighted VAM scoring of cell-type specific gene sets. Schiebout, Frost (2022) <https://psb.stanford.edu/psb-online/proceedings/psb22/schiebout.pdf>.

License GPL (>= 2)

Copyright Dartmouth College

Depends R (>= 3.6.0)

Imports VAM, Seurat (>= 4.0.0), MASS, Matrix (>= 1.3.3), utils, BiocManager, org.Hs.eg.db, org.Dr.eg.db, org.Mm.eg.db, AnnotationDbi, SeuratObject (>= 4.0.0), methods, edgeR

Suggests sctransform (>= 0.3.2)

Encoding UTF-8

NeedsCompilation no

Author H. Robert Frost [aut], Courtney Schiebout [aut, cre] (<https://orcid.org/0000-0002-4830-8237>)

Repository CRAN

Date/Publication 2023-11-13 15:03:24 UTC

R topics documented:

`BuildGeneSets` *Build Gene Sets for reference data to be applied to CAMML*

Description

BuildGeneSets takes in an expression matrix, either as a Seurat Object or as a simple matrix (where cells are the columns, and genes the rows), and labels for each of the cells. EdgeR differential expression analysis is then run within the function and gene sets are built based on a log fold change (FC) cut-off (the default is 2). Cut-offs can also be set by FC and -log10(p-value). Of note, when more than one cut-off is given, genes must meet ALL criteria. Gene weights are recorded for each gene in the gene set as the log2(FC), FC, or -log10(p-value) (the default is log2(FC)). Each gene in the gene set is converted into its corresponding Ensembl ID, necessitating that users also provide the species of interest. Currently humans "Hs", mice "Mm", and zebrafish "Dr" are accepted (the default is humans).

Usage

```
BuildGeneSets(exp.data, labels = as.character(Idents(exp.data)),
cutoff.type = "logfc", cutoff = 2, species = "Hs", weight.type = "logfc")
```

Arguments

`exp.data`	A Seurat Object or expression matrix for the reference data that has previously been normalized and scaled.
`labels`	A vector of the cell type labels for each cell in the expression matrix. This must have a label for each cell and be in the order the cells appear as columns in the expression matrix. The default will be the Idents of the Seurat Object.
`cutoff.type`	One or more of the following: "logfc", "fc", and "-logp". This value will determine what value(s) genes must have to be included in a given gene set. "logfc" and "fc" cut-off genes based on their log fold change and fold change values respectively. "-logp" will cut-off genes based on the significance of their differential expression according to the -log10(p-value). When more than one cut-off is given, genes must meet ALL criteria.
`cutoff`	A number or vector of numbers that correspond to the cutoff.types. The value should be greater than 0 if cutoff.type is "logfc", greater than 1 if cutoff.type is "fc", and greater than 1 if cutoff.type is "-logp". The number of values given should match the number of cutoff types listed and should be in the same order as the cutoff types vector.
`species`	Either "Hs", "Mm", or "Dr" for human, mouse, and zebrafish respectively. Used to convert gene symbols into Ensembl IDs.
`weight.type`	Can be one of the following: "logfc","fc", or "-logp". The former two options will assign gene weights by their log2FC or FC respectively in differential expression analysis. The final option will assign weight by the negative log10 of the p-values for each gene from differential expression analysis.

Value

A data.frame with the cell type, gene name, ensembl ID, and weight for each gene in each gene set.

See Also

edgeR,org.Hs.eg.db,org.Mm.eg.db

Examples

```
#Only run code if Seurat package is available
if (requireNamespace("Seurat", quietly=TRUE) & requireNamespace("SeuratObject", quietly=TRUE)) {
  #See vignettes for more examples
  BuildGeneSets(exp.data=SeuratObject::pbmc_small,
  labels = c(rep(1,40),rep(2,40)), cutoff.type = "logfc",
  cutoff = 2, species = "Hs", weight.type = "logfc")
}
```

CAMML *Cell-typing using variance Adjusted Mahalanobis distances with Multi-Labeling (CAMML)*

Description

Multi-label cell-typing method for single-cell RNA-sequencing data. CAMML takes in cell type-specific gene sets, with weights for each gene, and builds weighted Variance-Adjusted Mahalanobis (VAM) scores for each of them. CAMML then outputs a Seurat Object with an assay for CAMML that has the weighted VAM score for each cell type in each cell. CAMML takes in several arguments: seurat: a Seurat Object of the scRNA-seq data, gene.set.df: a data frame with a row for each gene and the following required columns: "ensemb.id" and "cell.type" and optional columns of "gene.weight" and "gene.symbol".

Usage

```
CAMML(seurat, gene.set.df)
```

Arguments

seurat	A Seurat Object that has previously been normalized and scaled.
gene.set.df	A list of lists of genes in each gene set, with each gene set list named for the cell type it represents.

Value

A SeuratObject with a CAMML assay with the scores for each cell type in each cell. This will be in the form of a matrix with columns for each cell and rows for each cell type that was scored.

See Also

vamForSeurat

Examples

```
# Only run example code if Seurat package is available
if (requireNamespace("Seurat", quietly=TRUE) & requireNamespace("SeuratObject", quietly=TRUE)) {
  # See vignettes for more examples
  seurat <- CAMML(seurat=SeuratObject::pbmc_small,
  gene.set.df = data.frame(cbind(ensembl.id = c("ENSG00000172005",
  "ENSG00000173114","ENSG00000139187"),
  cell.type = c("T cell","T cell","T cell"))))
  seurat@assays$CAMML@data
}
```

ChIMP *CAMML with the Integration of Marker Proteins (ChIMP)*

Description

ChIMP takes in the output of CAMML and a list of the CITE-seq markers designated for each cell type. For each marker, a k=2 means clustering will be applied to discretize their presence, resulting in a 0 in cells where the marker expression is in the lower value cluster and a 1 in cells where the marker expression is in the higher value cluster. Additionally, if a quantile cutoff is desired instead, this method can be designated and a cutoff can be given (the default is .5). These discretized scores are then multiplied by the CAMML score for each cell type in each cell. The function also takes in a vector of booleans the length of the number of cell types being evaluated that designates whether each cell type is required to have all markers score 1 or any marker score a 1 in order for the CAMML score to be maintained. If the boolean is true, ChIMP will weight CAMML by the maximum marker score for each cell type. For example, if both CD4 and CD8 are listed markers for T cells and either marker scoring a 1 is sufficient, the boolean will be true. If it is false, all markers designated for a cell type need to be in the higher value cluster for a given cell. ChIMP can also use the absence of a CITE-seq marker as support for a cell type by designating it "FALSE" with the greater argument. For example, if one is looking to identify non-immune cell types, CD45 can be used with greater = FALSE to support cell-type scores for a non-immune cell type.

Usage

```
ChIMP(seurat, citelist, method = "k", cutoff = .5,
anyMP = rep(T, length(rownames(seurat))),
greater = rep(T, length(unlist(citelist))))
```

Arguments

seurat	A Seurat Object that has previously been run on CAMML.
citelist	A list of all the surface markers for each cell type, named by their cell type.

method	Either a "k" or a "q" to designate the desired method. "k" will use a k=2 k-means clustering method for discretization. "q" will use a quantile cutoff method.
cutoff	A value between 0 and 1 designating the cutoff to be used if the quantile method is selected.
anyMP	A vector of booleans regarding whether the CITE-seq weighting will take any positive marker protein score (TRUE) or requires all positive marker scores (FALSE)
greater	A vector of booleans for every CITE-seq marker designating whether to evaluate it as present (TRUE) in a cell type or absent (FALSE) in a cell type.

Value

A `SeuratObject` with a ChIMP assay with the scores for each cell type in each cell, weighted by their CITE-seq score. This will be in the form of a matrix with columns for each cell and rows for each cell type that was scored.

See Also

`vamForSeurat`

Examples

```
# Only run example code if Seurat package is available
if (requireNamespace("Seurat", quietly=TRUE) &&
requireNamespace("SeuratObject", quietly=TRUE)) {
  # See vignettes for more examples
  seurat <- CAMML(seurat=SeuratObject::pbmc_small,
  gene.set.df = data.frame(cbind(ensembl.id = c("ENSG00000172005",
  "ENSG00000173114","ENSG00000139187"),
  cell.type = c("T cell","T cell","T cell"))))
  cite <- matrix(c(rnorm(40), rnorm(40,2,1)),
  nrow = length(rownames(seurat@assays$CAMML)),
  ncol = length(colnames(seurat@assays$CAMML)))
  rownames(cite) <- "marker"
  colnames(cite) <- colnames(seurat)
  assay <- SeuratObject::CreateAssayObject(counts = cite)
  seurat[["ADT"]] <- assay
  citelist <- list()
  citelist[[1]] = "marker"
  names(citelist) = "T cell"
  seurat <- ChIMP(seurat, citelist)
  seurat@assays$ChIMP@data
}
```

GetCAMMLLabels *Single- and Multi-Labelling for CAMML*

Description

This function takes in the Seurat Object output from the CAMML function and returns one of four labelling options. "top1" will return the top cell type for each cell. "top2" will return the top two highest scoring cell types for each cell. "top10p" will return the top scoring cell type and all other cell types with 10% of that score for each cell. "2xmean" will return all cell types with scores greater than twice the mean of all scores for a given cell.

Usage

```
GetCAMMLLabels(seurat, labels = "top1")
```

Arguments

seurat	A Seurat Object with a CAMML assay with weighted VAM scores for each cell type in each query cell. This is the output from the CAMML function.
labels	One of the following: "top1", "top2", "top10p", or "top2xmean". "top1" will return the single-label for the top-scoring cell type for each cell. "top2" will return the labels for the two top-scoring cell types for each cell. "top10p" will return the top scoring cell type and any other cell types with scores within 10% of the top score for each cell. "top2xmean" will return any cell types with scores two times the average of all cell type scores for each cell.

Value

A list with the labels designated by the "labels" argument.

See Also

CAMML

Examples

```
# Only run example code if Seurat and CAMML packages are available
if (requireNamespace("Seurat", quietly=TRUE) &
requireNamespace("SeuratObject", quietly=TRUE) &
requireNamespace("CAMML", quietly=TRUE)) {
  # See vignettes for more examples
  seurat <- CAMML(seurat=SeuratObject::pbmc_small,
  gene.set.df=data.frame(cbind(ensembl.id = c("ENSG00000172005",
  "ENSG00000173114","ENSG00000139187"),
  cell.type = c("T cell","T cell","T cell"))))
  GetCAMMLLabels(seurat, labels = "top1")
}
```

GetGeneSets *Get Pre-Built Gene Sets to be applied to CAMML*

Description

GetGeneSets takes in a one of the following: "immune.cells","skin.immune.cells","T.subset.cells", or "mouse.cells" and a Seurat Object that will be cell-typed using CAMML. The function will then build a gene.set.collection and a list of gene.weights based on one of the pre-built gene sets.

Usage

```
GetGeneSets(data = "immune.cells")
```

Arguments

data	One of the following: "immune.cells","skin.immune.cells","T.subset.cells", or "mouse.cells". • "immune.cells" provides gene sets for 5 human immune cells: B, T, NK, Monocytes, and HSCs. • "skin.immune.cells" provides gene sets for 6 human cell types: B, T, NK, Endothelial, Fibroblast, and Monocytes. • "T.subset.cells" provides 6 gene sets for 5 human T cell subtypes: naive CD4, CD4, naive CD8, CD8, memory, and regulatory T cells. • "mouse.cells" provides gene sets for 7 mouse cell types: B, T, NK, DC, Endothelial, Fibroblasts, and Macrophages. All datasets were built using differential expression of data in the package celldex using the package EdgeR.

Value

A `data.frame` with the cell type, gene name, ensembl ID, and weight for each gene in each gene set.

See Also

`org.Hs.eg.db`,`org.Mm.eg.db`

Examples

```
GetGeneSets("immune.cells")
```

Index