Package 'MetaNeighbor'

April 15, 2020

Type Package

Title Single cell replicability analysis

Version 1.6.0

Description MetaNeighbor allows users to quantify cell type replicability across datasets using neighbor voting.

biocViews ImmunoOncology, GeneExpression, GO, MultipleComparison, SingleCell, Transcriptomics

License MIT + file LICENSE

Depends R(>= 3.5)

Imports beanplot (>= 1.2), GenomicRanges (>= 1.34.0), gplots (>= 3.0.1), Matrix (>= 1.2), matrixStats (>= 0.54), Rcpp, RColorBrewer (>= 1.1), stats (>= 3.4), SummarizedExperiment (>= 1.12), utils (>= 3.4)

LinkingTo Rcpp

Suggests knitr (>= 1.17), rmarkdown (>= 1.6), testthat (>= 1.0.2)

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

Maintainer Manthan Shah <shah@cshl.edu>

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

git_branch RELEASE_3_10

git_last_commit 6e44380

git_last_commit_date 2019-10-29

Date/Publication 2020-04-14

Author Megan Crow [aut, cre],
Sara Ballouz [ctb],
Manthan Shah [ctb],
Stephan Fischer [ctb],
Jesse Gillis [aut]

R topics documented:

GOhuman 2
GOmouse 2
MetaNeighbor 3
MetaNeighborUS 4
mn_data 5
neighborVoting 5
topHits 6
variableGenes 7

Index **8**

`GOhuman` *GOhuman*

Description

List containing gene symbols for 71 GO function

Usage

```
GOhuman
```

Format

genesets List containing gene symbols for 71 GO function (GO slim terms containing between 50 and 1,000 genes) downloaded from the Gene Ontology Consortium August 2015 http://www.geneontology.org/page/download-annotations

Source

Dataset: https://github.com/mm-shah/MetaNeighbor/tree/master/data | Paper: https://www.biorxiv.org/content/early/2017/06/16/150524

`GOmouse` *GOmouse*

Description

List containing gene symbols for 10 GO function

Usage

```
GOmouse
```

Format

genesets List containing gene symbols for 10 GO function (GO:0016853 , GO:0005615, GO:0005768, GO:0007067, GO:0065003, GO:0042592, GO:0005929, GO:0008565, GO:0016829, GO:0022857) downloaded from the Gene Ontology Consortium August 2015 http://www.geneontology.org/page/download-annotations

Source

Dataset: https://github.com/mm-shah/MetaNeighbor/tree/master/data | Paper: https://www.biorxiv.org/content/early/2017/06/16/150524

`MetaNeighbor` *Runs MetaNeighbor*

Description

For each gene set of interest, the function builds a network of rank correlations between all cells. Next,It builds a network of rank correlations between all cells for a gene set. Next, the neighbor voting predictor produces a weighted matrix of predicted labels by performing matrix multiplication between the network and the binary vector indicating cell type membership, then dividing each element by the null predictor (i.e., node degree). That is, each cell is given a score equal to the fraction of its neighbors (including itself), which are part of a given cell type. For cross-validation, we permute through all possible combinations of leave-one-dataset-out cross-validation, and we report how well we can recover cells of the same type as area under the receiver operator characteristic curve (AUROC). This is repeated for all folds of cross-validation, and the mean AUROC across folds is reported. Calls `neighborVoting`.

Usage

```
MetaNeighbor(dat, i = 1, experiment_labels, celltype_labels, genesets,
  bplot = TRUE, fast_version = FALSE)
```

Arguments

`dat`	A SummarizedExperiment object containing gene-by-sample expression matrix.
`i`	default value 1; non-zero index value of assay containing the matrix data
`experiment_labels`	A numerical vector that indicates the source of each sample.
`celltype_labels`	A matrix that indicates the cell type of each sample.
`genesets`	Gene sets of interest provided as a list of vectors.
`bplot`	default true, beanplot is generated
`fast_version`	default value FALSE; a boolean flag indicating whether to use the fast and low memory version of MetaNeighbor

Value

A matrix of AUROC scores representing the mean for each gene set tested for each celltype is returned directly (see `neighborVoting`).

See Also

`neighborVoting`

Examples

```
data("mn_data")
data("GOmouse")
library(SummarizedExperiment)
AUROC_scores = MetaNeighbor(dat = mn_data,
                            experiment_labels = as.numeric(factor(mn_data$study_id)),
                           celltype_labels = metadata(colData(mn_data))[["cell_labels"]],
                            genesets = GOmouse,
                            bplot = TRUE,
                            fast_version=FALSE)
```

`MetaNeighborUS` *Runs unsupervised version of MetaNeighbor*

Description

When it is difficult to know how cell type labels compare across datasets this function helps users to make an educated guess about the overlaps without requiring in-depth knowledge of marker genes

Usage

```
MetaNeighborUS(var_genes, dat, i = 1, study_id, cell_type,
  fast_version = FALSE)
```

Arguments

`var_genes`	vector of high variance genes.
`dat`	SummarizedExperiment object containing gene-by-sample expression matrix.
`i`	default value 1; non-zero index value of assay containing the matrix data
`study_id`	a vector that lists the Study (dataset) ID for each sample
`cell_type`	a vector that lists the cell type of each sample
`fast_version`	default value FALSE; a boolean flag indicating whether to use the fast and low memory version of MetaNeighbor

Value

The output is a cell type-by-cell type mean AUROC matrix, which is built by treating each pair of cell types as testing and training data for MetaNeighbor, then taking the average AUROC for each pair (NB scores will not be identical because each test cell type is scored out of its own dataset, and the differential heterogeneity of datasets will influence scores).

Examples

```
data(mn_data)
var_genes = variableGenes(dat = mn_data, exp_labels = mn_data$study_id)
celltype_NV = MetaNeighborUS(var_genes = var_genes,
                             dat = mn_data,
                             study_id = mn_data$study_id,
                             cell_type = mn_data$cell_type,
                             fast_version=FALSE)
celltype_NV
```

mn_data *mn_data*

Description

A SummarizedExperiment object containing: a gene matrix, cell type labels, experiment labels, sets of genes, sample ID, study id and cell types.

Usage

```
mn_data
```

Format

Gene matrix A gene-by-sample expression matrix consisting of 3157 rows (genes) and 1051 columns (cell types)

cell_labels 1051x1 binary matrix that indicates whether a cell belongs to the SstNos cell type (1=yes, 0 = no)

sample_id A character vector of length 1051 that indicates the sample_id of each sample

study_id A character vector of length 1051 that indicates the study_id of each sample ("GSE60361" = Zeisel et al, "GSE71585" = Tasic et al)

cell_type A character vector of length 1051 that indicates the cell-type of each sample

Source

Dataset:https://github.com/mm-shah/MetaNeighbor/tree/master/data 1. Zeisal et al. http://science.sciencemag.org/content/347/6226/1138 2. Tasic et al. http://www.nature.com/neuro/journal/v19/n2/full/nn.4216.html

---

neighborVoting *Runs the neighbor voting algorithm.*

Description

The function performs cell type identity prediction based on 'guilt by association' using cross validation. Performance is evaluated by calculating the AUROC for each cell type.

Usage

```
neighborVoting(exp_labels, cell_labels, network, means = TRUE)
```

Arguments

exp_labels	numerical vector that indicates the dataset source of each sample
cell_labels	sample by cell type matrix that indicates the cell type of each sample (0-absent; 1-present)
network	sample by sample adjacency matrix, ranked and standardized between 0-1
means	default TRUE, determines output formatting

Value

If means = TRUE (default) a vector containing the mean of AUROC values across cross-validation folds will be returned. If FALSE a list is returned containing a cell type by dataset matrix of AUROC scores, for each fold of cross-validation. Default is over-ridden when more than one cell type is assessed.

See Also

MetaNeighbor

Examples

```
data("mn_data")
data("GOmouse")
library(SummarizedExperiment)
AUROC_scores = MetaNeighbor(dat = mn_data,
                            experiment_labels = as.numeric(factor(mn_data$study_id)),
                           celltype_labels = metadata(colData(mn_data))[["cell_labels"]],
                            genesets = GOmouse,
                            bplot = TRUE)
AUROC_scores
```

---

topHits *Find reciprocal top hits*

Description

Identifies reciprocal top hits and high scoring cell type pairs

Usage

```
topHits(cell_NV, dat, i = 1, study_id, cell_type, threshold = 0.95)
```

Arguments

cell_NV	matrix of celltype-to-celltype AUROC scores (output from MetaNeighborUS)
dat	a SummarizedExperiment object containing gene-by-sample expression matrix.
i	default value 1; non-zero index value of assay containing the matrix data
study_id	a vector that lists the Study (dataset) ID for each sample
cell_type	a vector that lists the cell type of each sample
threshold	default value 0.95. Must be between [0,1]

Value

Function returns a dataframe with cell types that are either reciprocal best matches, and/or those with AUROC values greater than or equal to threshold value

Examples

```
data(mn_data)
var_genes = variableGenes(dat = mn_data, exp_labels = mn_data$study_id)
celltype_NV = MetaNeighborUS(var_genes = var_genes,
                             dat = mn_data,
                             study_id = mn_data$study_id,
                             cell_type = mn_data$cell_type)
top_hits = topHits(cell_NV = celltype_NV,
                   dat = mn_data,
                   study_id = mn_data$study_id,
                   cell_type = mn_data$cell_type,
                   threshold = 0.9)
top_hits
```

`variableGenes` *Identify a highly variable gene set*

Description

Identifies genes with high variance compared to their median expression (top quartile) within each experimentCertain function

Usage

```
variableGenes(dat, i = 1, exp_labels)
```

Arguments

`dat`	SummarizedExperiment object containing gene-by-sample expression matrix.
`i`	default value 1; non-zero index value of assay containing the matrix data
`exp_labels`	character vector that denotes the source (Study ID) of each sample.

Value

The output is a vector of gene names that are highly variable in every experiment (intersect)

Examples

```
data(mn_data)
var_genes = variableGenes(dat = mn_data, exp_labels = mn_data$study_id)
var_genes
```

Index

∗Topic **datasets**
GOhuman, 2
GOmouse, 2
mn_data, 5

GOhuman, 2
GOmouse, 2

MetaNeighbor, 3, *6*
MetaNeighborUS, 4, *6*
mn_data, 5

neighborVoting, *3*, 5

topHits, 6

variableGenes, 7