Package: DAseq (via r-universe)

August 14, 2024

Type Package

Title Detecting regions of differential abundance between scRNA-seq datasets

Version 1.0.0

Author Jun Zhao <jun.zhao@yale.edu>

Maintainer Jun Zhao <jun.zhao@yale.edu>

Description DA-seq is a multiscale approach for detecting DA subpopulations. In contrast to clustering based approaches, our method can detect DA subpopulations that do not form well separated clusters. For each cell, we compute a multiscale differential abundance score measure. These scores are based on the k nearest neighbors in the transcriptome space for various values of k.

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports RANN, glmnet, caret, Seurat, e1071, reticulate, ggplot2, cowplot, scales, ggrepel

Repository https://blaserlab.r-universe.dev

RemoteUrl https://github.com/KlugerLab/DAseq

RemoteRef HEAD

RemoteSha b1f58e0028975dcbe3b38e76a01832d30401aaf6

Contents

`addDAslot` *Add DA slot*

Description

Add DA region information to the meta.data of a Seurat object

Usage

```
addDAslot(object, da.regions, da.slot = "da", set.ident = F)
```

Arguments

`object`	input Seurat object
`da.regions`	output from function getDAregion()
`da.slot`	character, variable name to put in Seurat meta.data, default "da"
`set.ident`	a logical value to indicate whether to set Idents of the Seurat object to DA information, default False

Value

updated Seurat object

DAseq *DAseq: Detecting regions of differential abundance between scRNA-seq datasets*

Description

DA-seq is a multiscale approach for detecting DA subpopulations. In contrast to clustering based approaches, our method can detect DA subpopulations that do not form well separated clusters. For each cell, we compute a multiscale differential abundance score measure. These scores are based on the k nearest neighbors in the transcriptome space for various values of k.

Author(s)

Jun Zhao, Ariel Jaffe, Henry Li, Ofir Lindenbaum, Xiuyuan Cheng, Yuval Kluger

getDAcells *DA-seq Step 1 & Step 2: select DA cells*

Description

Step 1: compute a multiscale score measure for each cell of its k-nearest-neighborhood for multiple values of k. Step 2: train a logistic regression classifier based on the multiscale score measure and retain cells that may reside in DA regions.

Usage

```
getDAcells(X, cell.labels, labels.1, labels.2, k.vector = NULL,
  save.knn = F, alpha = 0, k.folds = 10, n.runs = 5, n.rand = 2,
  pred.thres = NULL, do.plot = T, plot.embedding = NULL,
  size = 0.5)
```

Arguments

X	size N-by-p matrix, input merged dataset of interest after dimension reduction.
cell.labels	size N character vector, labels for each input cell
labels.1	character vector, label name(s) that represent condition 1
labels.2	character vector, label name(s) that represent condition 2
k.vector	vector, k values to create the score vector
save.knn	a logical value to indicate whether to save computed kNN result, default False
alpha	numeric, elasticnet mixing parameter passed to glmnet(), default 0 (Ridge)
k.folds	integer, number of data splits used in the neural network, default 10
n.runs	integer, number of times to run the neural network to get the predictions, default 5

n.rand	integer, number of random permutations to run, default 2
pred.thres	length-2 vector, top and bottom threshold on DA measure, default NULL, select significant DA cells based on permutation
do.plot	a logical value to indicate whether to return ggplot objects showing the results, default True
plot.embedding	size N-by-2 matrix, 2D embedding for the cells
size	cell size to use in the plot, default 0.5

Value

a list of results

cell.idx index of cells used in DA calculation

da.ratio score vector for each cell

da.pred (mean) prediction from the logistic regression

da.up index for DA cells more abundant in condition of labels.2

da.down index for DA cells more abundant in condition of labels.1

pred.plot ggplot object showing the predictions of logistic regression on plot.embedding

da.cells.plot ggplot object highlighting cells of da.cell.idx on plot.embedding

---

getDAregion *DA-seq Step 3: get DA regions*

Description

Cluster the DA cells retained from Step 1 and Step 2 of DA-seq to obtain spatially coherent DA regions.

Usage

```
getDAregion(X, da.cells, cell.labels, labels.1, labels.2,
  prune.SNN = 1/15, resolution = 0.05, group.singletons = F,
  min.cell = NULL, do.plot = T, plot.embedding = NULL, size = 0.5,
  do.label = F, ...)
```

Arguments

X	size N-by-p matrix, input merged dataset of interest after dimension reduction
da.cells	output from getDAcells() or updateDAcells()
cell.labels	size N vector, labels for each input cell
labels.1	vector, label name(s) that represent condition 1
labels.2	vector, label name(s) that represent condition 2
prune.SNN	parameter for Seurat function FindNeighbors(), default 1/15

resolution	parameter for Seurat function FindClusters(), default 0.05
group.singletons	parameter for Seurat function FindClusters(), default True
min.cell	integer, number of cells below which a DA region will be removed as outliers, default NULL, use minimum k value in k-vector
do.plot	a logical value to indicate whether to return ggplot objects showing the results, default True
plot.embedding	size N-by-2 matrix, 2D embedding for the cells
size	cell size to use in the plot, default 0.5
do.label	a logical value to indicate whether to label each DA region with text, default False
...	other parameters to pass to Seurat FindClusters()

Value

a list of results

cell.idx index of cells used in DA calculation

da.region.label DA region label for each cell from the whole dataset, '0' represents non-DA cells.

DA.stat a table showing DA score and p-value for each DA region

da.region.plot ggplot object showing DA regions on plot.embedding

---

`plotCellLabel` *Plot cell labels*

Description

Produce a ggplot object with cells on 2D embedding, colored by given labels of each cell.

Usage

```
plotCellLabel(X, label, cell.col = NULL, size = 0.5, alpha = 1,
  shape = 16, do.label = T, label.size = 4, label.repel = F,
  label.plot = NULL)
```

Arguments

X	matrix, 2D embedding of each cell for the plot
label	vector, label for each cell
cell.col	string vector, color bar to use for cell labels, default ggplot default
size	numeric, dot size for each cell, default 0.5
alpha	numeric between 0 to 1, dot opacity for each cell, default 1
do.label	a logical value indicating whether to add text to mark each cell label

label.size	numeric, size of text labels, default 4
label.repel	a logical value indicating whether to repel the labels to avoid overlapping, default False
label.plot	cell labels to add text annotations, default NULL, add text for all labels

Value

a ggplot object

`plotCellScore` *Plot a score for each cell*

Description

Produce a ggplot object with cells on 2D embedding, colored by a given score for each cell.

Usage

```
plotCellScore(X, score, cell.col = c("blue", "white", "red"),
  size = 0.5, alpha = 1, shape = 16)
```

Arguments

X	matrix, 2D embedding of each cell for the plot
score	numeric vector, a single value to color each cell, continuous
cell.col	string vector, color bar to use for "score", defaul c("blue","white","red")
size	numeric, dot size for each cell, default 0.5
alpha	numeric between 0 to 1, dot opacity for each cell, default 1

Value

a ggplot object

plotDAsite *Plot da site*

Description

Plot da site

Usage

```
plotDAsite(X, site.list, size = 0.5, cols = NULL)
```

Arguments

X	matrix, 2D embedding of each cell for the plot
site.list	list, a list of cell indices for each site to plot
size	numeric, dot size for each cell, default 0.5
cols	string vector, color bar to use for each site, default ggplot default

runSTG *Run STG*

Description

Run STG to select a set of genes that separate cells with label.1 from label.2 (other labels)

Usage

```
runSTG(X, X.labels, label.1, label.2 = NULL, lambda = 1.5,
  n.runs = 5, return.model = T, python.use = "/usr/bin/python",
  GPU = "")
```

Arguments

X	matrix, normalized expression matrix of all cells in the dataset, genes are in rows, rownames must be gene names
X.labels	numeric vector, specify labels for each cell, must be 0 or 1
label.1	cell label to define markers for
label.2	second cell label to for comparison, if NULL, use all other labels
lambda	numeric, regularization parameter that weights the number of selected genes, a larger lambda leads to fewer genes, default 1.5
n.runs	integer, number of runs to run the model, default 5
return.model	a logical value to indicate whether to return the actual model of STG
python.use	character string, the Python to use, default "/usr/bin/python"
GPU	which GPU to use, default ", using CPU

Value

a list of results:

markers a list of data.frame with markers for each DA region

accuracy a numeric vector showing mean accuracy for each DA region

model a list of model for each DA region, each model contains:

- **model** the model of STG of the final run
- **cells** cell names/indices used to train the model
- **features** features used to train the model
- **selected.features** the selected features of the final run
- **pred** the linear prediction value for each cell from the model

---

`SeuratLocalMarkers` *Find local markers*

Description

Use Seurat FindMarkers() function to identify genes that distinguish a DA region from its local neighborhood

Usage

```
SeuratLocalMarkers(object, da.slot = "da", da.region.to.run,
  cell.label.slot, cell.label.to.run, ...)
```

Arguments

`object`	input Seurat object
`da.slot`	character, variable name that represents DA regions in Seurat meta.data, default "da"
`da.region.to.run`	numeric, which (single) DA region to find local markers for
`cell.label.slot`	character, variable name that represents cell labeling information in Seurat meta.data to combine with DA information
`cell.label.to.run`	cell label(s) that represent the local neiborhood for the input DA region
`...`	parameters passed to Seurat FindMarkers() function

Value

a data.frame of markers and statistics

SeuratMarkerFinder *DA-seq Step 4: Seurat marker finder to characterize DA regions*

Description

Use Seurat FindMarkers() function to identify characteristic genes for DA regions

Usage

```
SeuratMarkerFinder(object, da.slot = "da", da.regions.to.run = NULL,
  ...)
```

Arguments

object	input Seurat object
da.slot	character, variable name that represents DA regions in Seurat meta.data, default "da"
da.regions.to.run	numeric (vector), which DA regions to find markers for, default is to run all regions
...	parameters passed to Seurat FindMarkers() function

Value

a list of markers for DA regions with statistics

STGlocalMarkers *STG local markers Run STG to find a set of genes that separate a given DA region from a local subset of cells.*

Description

STG local markers Run STG to find a set of genes that separate a given DA region from a local subset of cells.

Usage

```
STGlocalMarkers(X, da.regions, da.region.to.run, cell.label.info,
  cell.label.to.run, ...)
```

Arguments

X	matrix, normalized expression matrix of all cells in the dataset, genes are in rows, rownames must be gene names
da.regions	output from the function getDAregion()
da.region.to.run	numeric, which (single) DA region to find local markers for
cell.label.info	vector, cell labeling information to select the local subset of cells to compare with input DA region
cell.label.to.run	cell label(s) to select from cell.label.info that represent the local neiborhood for the input DA region
lambda	numeric, regularization parameter that weights the number of selected genes, a larger lambda leads to fewer genes, default 1.5
n.runs	integer, number of runs to run the model, default 5
python.use	character string, the Python to use, default "/usr/bin/python"
return.model	a logical value to indicate whether to return the actual model of STG
GPU	which GPU to use, default ", using CPU

Value

a list of results:

markers a list of data.frame with markers for each DA region

accuracy a numeric vector showing mean accuracy for each DA region

model a list of model for each DA region, each model contains:

- **model** the model of STG of the final run
- **features** features used to train the model
- **selected.features** the selected features of the final run
- **pred** the linear prediction value for each cell from the model

---

STGmarkerFinder *DA-seq Step 4: STG feature selection*

Description

Use STG (stochastic gates) to select genes that separate each DA region from the rest of the cells. For a full description of the algorithm, see Y. Yamada, O. Lindenbaum, S. Negahban, and Y. Kluger. Feature selection using stochastic gates. arXiv preprint arXiv:1810.04247, 2018.

Usage

```
STGmarkerFinder(X, da.regions, da.regions.to.run = NULL, lambda = 1.5,
  n.runs = 5, return.model = T, python.use = "/usr/bin/python",
  GPU = "")
```

Arguments

X	matrix, normalized expression matrix of all cells in the dataset, genes are in rows, rownames must be gene names
da.regions	output from the function getDAregion()
da.regions.to.run	numeric (vector), which DA regions to run the marker finder, default is to run all regions
lambda	numeric, regularization parameter that weights the number of selected genes, a larger lambda leads to fewer genes, default 1.5
n.runs	integer, number of runs to run the model, default 5
return.model	a logical value to indicate whether to return the actual model of STG
python.use	character string, the Python to use, default "/usr/bin/python"
GPU	which GPU to use, default ", using CPU

Value

a list of results:

da.markers a list of data.frame with markers for each DA region

accuracy a numeric vector showing mean accuracy for each DA region

model a list of model for each DA region, each model contains:

model the model of STG of the final run

features features used to train the model

selected.features the selected features of the final run

pred the linear prediction value for each cell from the model

---

updateDAcells *Update DA cells*

Description

Use different threshold to select DA cells based on an output from getDAcells().

Usage

```
updateDAcells(X, pred.thres = NULL, force.thres = F, alpha = NULL,
  k.folds = 10, n.runs = 10, cell.labels = NULL, labels.1 = NULL,
  labels.2 = NULL, do.plot = T, plot.embedding = NULL, size = 0.5)
```

Arguments

X	output from getDAcells()
pred.thres	length-2 vector, top and bottom threshold on DA measure, default NULL, select significant DA cells based on permutation
force.thres	a logical value to indicate whether to forcefully use pred.thres without considering significance, default False
alpha	set this parameter to not NULL to rerun Logistic regression
do.plot	a logical value to indicate whether to return ggplot objects showing the results, default True
plot.embedding	size N-by-2 matrix, 2D embedding for the cells
size	cell size to use in the plot, default 0.5

Value

a list of results with updated DA cells

X.2d.melanoma *t-SNE embedding of the melanoma dataset*

Description

A matrix containing 2D t-SNE embedding of the melanoma dataset

Usage

```
X.2d.melanoma
```

Format

An object of class matrix with 16291 rows and 2 columns.

Source

https://www.sciencedirect.com/science/article/pii/S0092867418313941 Sade-Feldman, Moshe, et al. (Cell. 2018)

X.label.info *Sample label information*

Description

A dataset containing information of each sample, indicating whether this sample is a "responder" (R) or a "non-responder" (NR)

Usage

```
X.label.info
```

Format

a data.frame with 48 rows and 2 columns

label sample label, matching with labels in X.label.melanoma

condition condition of the sample label, either R or NR

Source

https://www.sciencedirect.com/science/article/pii/S0092867418313941 Sade-Feldman, Moshe, et al. (Cell. 2018)

X.label.melanoma *Cell sample labels of the melanoma dataset*

Description

A string vector with the length equal to number of cells, indicating sample labels of each cell: which sample each cell comes from

Usage

```
X.label.melanoma
```

Format

An object of class `character` of length 16291.

Source

https://www.sciencedirect.com/science/article/pii/S0092867418313941 Sade-Feldman, Moshe, et al. (Cell. 2018)

`X.melanoma` *Top 10 PCs of the melanoma dataset*

Description

A dataset containing the top 10 PCs (principal components) of the melanoma dataset

Usage

```
X.melanoma
```

Format

An object of class `matrix` with 16291 rows and 10 columns.

Source

https://www.sciencedirect.com/science/article/pii/S0092867418313941 Sade-Feldman, Moshe, et al. (Cell. 2018)

Index