

Package ‘UCell’

October 18, 2022

Type Package

Title Rank-based signature enrichment analysis for single-cell data

Version 2.0.1

Description UCell is a package for evaluating gene signatures in single-cell datasets. UCell signature scores, based on the Mann-Whitney U statistic, are robust to dataset size and heterogeneity, and their calculation demands less computing time and memory than other available methods, enabling the processing of large datasets in a few minutes even on machines with limited computing power. UCell can be applied to any single-cell data matrix, and includes functions to directly interact with SingleCellExperiment and Seurat objects.

Depends R(>= 4.1.0)

Imports methods, data.table(>= 1.13.6), Matrix, BiocParallel, SingleCellExperiment, SummarizedExperiment

Suggests Seurat, scater, scRNAseq, reshape2, patchwork, ggplot2, BiocStyle, knitr, rmarkdown

biocViews SingleCell, GeneSetEnrichment, Transcriptomics, GeneExpression, CellBasedAssays

VignetteBuilder knitr

BugReports <https://github.com/carmonalab/UCell/issues>

URL <https://github.com/carmonalab/UCell>

License GPL-3 + file LICENSE

Encoding UTF-8

LazyData FALSE

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

git_url <https://git.bioconductor.org/packages/UCell>

git_branch RELEASE_3_15

git_last_commit 2b5cd45

git_last_commit_date 2022-06-21

Date/Publication 2022-10-18

Author Massimo Andreatta [aut, cre] (<<https://orcid.org/0000-0002-8036-2647>>),
Santiago Carmona [aut] (<<https://orcid.org/0000-0002-2495-0671>>)

Maintainer Massimo Andreatta <massimo.andreatta@unil.ch>

R topics documented:

AddModuleScore_UCell	2
calculate_Uscore	4
check_genes	5
check_signature_names	6
data_to_ranks_data_table	6
rankings2Uscore	7
sample.matrix	8
ScoreSignatures_UCell	8
split_data.matrix	10
StoreRankings_UCell	10
UCell	12
u_stat	13
u_stat_signature_list	13

Index **15**

AddModuleScore_UCell *Calculate module enrichment scores from single-cell data (Seurat interface)*

Description

Given a Seurat object, calculates module/signature enrichment scores at single-cell level using the Mann-Whitney U statistic. UCell scores are normalized U statistics (between 0 and 1), and they are mathematically related to the Area under the ROC curve (see [Mason and Graham](#))

Usage

```
AddModuleScore_UCell(
  obj,
  features,
  maxRank = 1500,
  chunk.size = 1000,
  BPPARAM = NULL,
  ncores = 1,
  storeRanks = FALSE,
  w_neg = 1,
  assay = NULL,
  slot = "data",
```

```

    ties.method = "average",
    force.gc = FALSE,
    name = "_UCell"
  )

```

Arguments

<code>obj</code>	Seurat object
<code>features</code>	A list of signatures, for example: <code>list(Tcell_signature = c("CD2", "CD3E", "CD3D"), Myeloid_signature = c("SPI1", "FCER1G", "CSF1R"))</code> You can also specify positive and negative gene sets by adding a + or - sign to genes in the signature; see an example below
<code>maxRank</code>	Maximum number of genes to rank per cell; above this rank, a given gene is considered as not expressed.
<code>chunk.size</code>	Number of cells to be processed simultaneously (lower size requires slightly more computation but reduces memory demands)
<code>BPPARAM</code>	A <code>BiocParallel::bpparam()</code> object that tells UCell how to parallelize. If provided, it overrides the <code>ncores</code> parameter.
<code>ncores</code>	Number of processors to parallelize computation. If <code>BPPARAM = NULL</code> , the function uses <code>BiocParallel::bpparam(workers=ncores)</code>
<code>storeRanks</code>	Store ranks matrix in Seurat object ('UCellRanks' assay) for fast subsequent computations. This option may demand large amounts of RAM.
<code>w_neg</code>	Weight on negative genes in signature. e.g. <code>w_neg=1</code> weighs equally up- and down-regulated genes, <code>w_neg=0.5</code> gives 50% less importance to negative genes
<code>assay</code>	Pull out data from this assay of the Seurat object (if <code>NULL</code> , use <code>DefaultAssay(obj)</code>)
<code>slot</code>	Pull out data from this slot of the Seurat object
<code>ties.method</code>	How ranking ties should be resolved - passed on to <code>data.table::frank</code>
<code>force.gc</code>	Explicitly call garbage collector to reduce memory footprint
<code>name</code>	Name tag that will be appended at the end of each signature name, "_UCell" by default (e.g. signature score in meta data will be named: Myeloid_signature_UCell)

Details

In contrast to Seurat's `AddModuleScore`, which is normalized by binning genes of similar expression at the population level, UCell scores depend only on the gene expression ranks of individual cell, and therefore they are robust across datasets regardless of dataset composition.

Value

Returns a Seurat object with module/signature enrichment scores added to object meta data; each score is stored as the corresponding signature name provided in `features` followed by the tag given in `name` (or "_UCell" by default)

Examples

```

library(UCell)
gene.sets <- list(Tcell = c("CD2", "CD3E", "CD3D"),
                 Myeloid = c("SPI1", "FCER1G", "CSF1R"))
data(sample.matrix)
obj <- Seurat::CreateSeuratObject(sample.matrix)

obj <- AddModuleScore_UCell(obj, features = gene.sets)
head(obj[[[]])

## Using positive and negative gene sets
gene.sets <- list()
gene.sets$Tcell_gd <- c("TRDC+", "TRGC1+", "TRGC2+", "TRDV1+",
                      "TRAC-", "TRBC1-", "TRBC2-")
gene.sets$NKcell <- c("FGFBP2+", "SPON2+", "KLRF1+",
                    "FCGR3A+", "CD3E-", "CD3G-")
obj <- AddModuleScore_UCell(obj, features = gene.sets, name=NULL)
head(obj$NKcell)

```

calculate_Uscore

Calculate rankings and scores for query data and given signature set

Description

Calculate rankings and scores for query data and given signature set

Usage

```

calculate_Uscore(
  matrix,
  features,
  maxRank = 1500,
  chunk.size = 1000,
  BPPARAM = NULL,
  ncores = 1,
  w_neg = 1,
  ties.method = "average",
  storeRanks = FALSE,
  force.gc = FALSE,
  name = "_UCell"
)

```

Arguments

matrix	Input data matrix
features	List of signatures
maxRank	Rank cutoff (1500)

chunk.size	Cells per sub-matrix (1000)
BPPARAM	A BioParallel object to instruct UCell how to parallelize
ncores	Number of cores to use for parallelization
w_neg	Weight on negative signatures
ties.method	How to break ties, for data.table::frankv method ("average")
storeRanks	Store ranks? (FALSE)
force.gc	Force garbage collection? (FALSE)
name	Suffix for metadata columns ("_UCell")

Value

A list of signature scores

check_genes	<i>Check if all genes in signatures are found in data matrix - otherwise add zero counts in data-matrix to complete it</i>
-------------	--

Description

Check if all genes in signatures are found in data matrix - otherwise add zero counts in data-matrix to complete it

Usage

```
check_genes(matrix, features)
```

Arguments

matrix	Input data matrix
features	List of genes that must be present (otherwise they are added)

Value

Same input matrix, extended to comprise any missing genes

check_signature_names *Check signature names and add standard names is missing*

Description

Check signature names and add standard names is missing

Usage

```
check_signature_names(features)
```

Arguments

features List of signatures for scoring

Value

The input list of signatures, with standard names if provided un-named

data_to_ranks_data_table
Calculate per-cell feature rankings

Description

Calculate per-cell feature rankings

Usage

```
data_to_ranks_data_table(data, ties.method = "average")
```

Arguments

data Expression data matrix
ties.method How to break ties (passed on to data.table::frankv)

Value

A data.table of ranks

rankings2Uscore	<i>Get signature scores from pre-computed rank matrix</i>
-----------------	---

Description

Get signature scores from pre-computed rank matrix

Usage

```
rankings2Uscore(  
  ranks_matrix,  
  features,  
  chunk.size = 1000,  
  w_neg = 1,  
  BPPARAM = NULL,  
  ncores = 1,  
  force.gc = FALSE,  
  name = "_UCell"  
)
```

Arguments

ranks_matrix	A rank matrix
features	List of signatures
chunk.size	How many cells per matrix chunk
w_neg	Weight on negative signatures
BPPARAM	A BioParallel object to instruct UCell how to parallelize
ncores	How many cores to use for parallelization?
force.gc	Force garbage collection to recover RAM? (FALSE)
name	Name suffix for metadata columns ("_UCell")

Value

A list of signature scores

sample.matrix *Sample dataset to test UCell installation*

Description

A sparse matrix (class "dgCMatrix") of single-cell transcriptomes (scRNA-seq) for 600 cells and 20729 genes. Single-cell UMI counts were normalized using a standard log-normalization: counts for each cell were divided by the total counts for that cell and multiplied by 10,000, then natural-log transformed using $\log_1 p$.

This is a subsample of T cells from the large scRNA-seq PBMC dataset published by [Hao et al.](https://atlas.fredhutch.org/data/nygc/multimodal/pbmc_multimodal.h5seurat) and available as UMI counts at https://atlas.fredhutch.org/data/nygc/multimodal/pbmc_multimodal.h5seurat

Usage

```
sample.matrix
```

Format

A sparse matrix of 600 cells and 20729 genes.

Source

<https://doi.org/10.1016/j.cell.2021.04.048>

ScoreSignatures_UCell *Calculate module enrichment scores from single-cell data*

Description

Given a gene vs. cell matrix, calculates module/signature enrichment scores on single-cell level using Mann-Whitney U statistic. UCell scores are normalized U statistics (between 0 and 1), and they are mathematically related to the Area under the ROC curve (see [Mason and Graham](#)) These scores only depend on the gene expression ranks of individual cell, and therefore they are robust across datasets regardless of dataset composition.

Usage

```
ScoreSignatures_UCell(  
  matrix = NULL,  
  features,  
  precalc.ranks = NULL,  
  maxRank = 1500,  
  w_neg = 1,  
  name = "_UCell",
```



```

    assay = "counts",
    chunk.size = 1000,
    BPPARAM = NULL,
    ncores = 1,
    ties.method = "average",
    force.gc = FALSE
  )

```

Arguments

matrix	Input matrix, either stored in a SingleCellExperiment object or as a raw matrix. dgCMatrx format supported.
features	A list of signatures, for example: <code>list(Tcell_signature = c("CD2", "CD3E", "CD3D"), Myeloid_signature = c("SPI1", "FCER1G", "CSF1R"))</code> You can also specify positive and negative gene sets by adding a + or - sign to genes in the signature; see an example below
precalc.ranks	If you have pre-calculated ranks using StoreRankings_UCell , you can specify the pre-calculated ranks instead of the gene vs. cell matrix.
maxRank	Maximum number of genes to rank per cell; above this rank, a given gene is considered as not expressed. Note: this parameter is ignored if <code>precalc.ranks</code> are specified
w_neg	Weight on negative genes in signature. e.g. <code>w_neg=1</code> weighs equally up- and down-regulated genes, <code>w_neg=0.5</code> gives 50% less importance to negative genes
name	Name suffix appended to signature names
assay	The sce object assay where the data is to be found
chunk.size	Number of cells to be processed simultaneously (lower size requires slightly more computation but reduces memory demands)
BPPARAM	A BiocParallel::bpparam() object that tells UCell how to parallelize. If provided, it overrides the <code>ncores</code> parameter.
ncores	Number of processors to parallelize computation. If <code>BPPARAM = NULL</code> , the function uses <code>BiocParallel::bpparam(workers=ncores)</code>
ties.method	How ranking ties should be resolved - passed on to data.table::frank
force.gc	Explicitly call garbage collector to reduce memory footprint

Value

Returns input `SingleCellExperiment` object with UCell scores added to `altExp`

Examples

```

library(UCell)
# Using sparse matrix
data(sample.matrix)
gene.sets <- list( Tcell_signature = c("CD2", "CD3E", "CD3D"),
                  Myeloid_signature = c("SPI1", "FCER1G", "CSF1R"))
scores <- ScoreSignatures_UCell(sample.matrix, features=gene.sets)

```

```

head(scores)

# Using sce object
library(SingleCellExperiment)
data(sample.matrix)
my.sce <- SingleCellExperiment(list(counts=sample.matrix))
gene.sets <- list( Tcell_signature = c("CD2", "CD3E", "CD3D"),
                  Myeloid_signature = c("SPI1", "FCER1G", "CSF1R"))
my.sce <- ScoreSignatures_UCell(my.sce, features=gene.sets)
altExp(my.sce, 'UCell')

```

split_data.matrix	<i>Split data matrix into smaller sub-matrices ('chunks')</i>
-------------------	---

Description

Split data matrix into smaller sub-matrices ('chunks')

Usage

```
split_data.matrix(matrix, chunk.size = 1000)
```

Arguments

matrix	Input data matrix
chunk.size	How many cells to include in each sub-matrix

Value

A list of sub-matrices, each with size `n_features` x `chunk_size`

StoreRankings_UCell	<i>Calculate and store gene rankings for a single-cell dataset</i>
---------------------	--

Description

Given a gene vs. cell matrix, calculates the rankings of expression for all genes in each cell.

Usage

```
StoreRankings_UCell(
  matrix,
  maxRank = 1500,
  chunk.size = 1000,
  BPPARAM = NULL,
  ncores = 1,
  assay = "counts",
  ties.method = "average",
  force.gc = FALSE
)
```

Arguments

matrix	Input matrix, either stored in a SingleCellExperiment object or as a raw matrix. dgCMatrx format supported.
maxRank	Maximum number of genes to rank per cell; above this rank, a given gene is considered as not expressed
chunk.size	Number of cells to be processed simultaneously (lower size requires slightly more computation but reduces memory demands)
BPPARAM	A BiocParallel::bpparam() object that tells UCell how to parallelize. If provided, it overrides the ncores parameter.
ncores	Number of processors to parallelize computation. If BPPARAM = NULL, the function uses BiocParallel::bpparam(workers=ncores)
assay	Assay where the data is to be found (for input in 'sce' format)
ties.method	How ranking ties should be resolved - passed on to data.table::frank
force.gc	Explicitly call garbage collector to reduce memory footprint

Details

While [ScoreSignatures_UCell](#) can be used 'on the fly' to evaluate signatures in a query dataset, it requires recalculating gene ranks at every execution. If you have a large dataset and plan to experiment with multiple signatures, evaluating the same dataset multiple times, this function allows you to store pre-calculated ranks so they do not have to be recomputed every time. Pre-calculated ranks can then be applied to the function [ScoreSignatures_UCell](#) to evaluate gene signatures in a significantly faster way on successive iterations.

Value

Returns a sparse matrix of pre-calculated ranks that can be used multiple times to evaluate different signatures

Examples

```
library(UCell)
data(sample.matrix)
ranks <- StoreRankings_UCell(sample.matrix)
```

```

ranks[1:5,1:5]
gene.sets <- list( Tcell_signature = c("CD2", "CD3E", "CD3D"),
                  Myeloid_signature = c("SPI1", "FCER1G", "CSF1R"))
scores <- ScoreSignatures_UCell(features=gene.sets, precalc.ranks=ranks)
head(scores)

```

UCell

UCell: Robust and scalable single-cell gene signature scoring

Description

UCell is an R package for scoring gene signatures in single-cell datasets. UCell scores, based on the Mann-Whitney U statistic, are robust to dataset size and heterogeneity, and their calculation demands relatively less computing time and memory than most other methods, enabling the processing of large datasets ($> 10^5$ cells). UCell can be applied to any cell vs. gene data matrix, and includes functions to directly interact with Seurat and SingleCellExperiment objects.

UCell functions

- `ScoreSignatures_UCell` Calculate module enrichment scores from single-cell data. Given a gene vs. cell matrix (either as sparse matrix or stored in a `SingleCellExperiment` object), it calculates module/signature enrichment scores. This score depends only on the gene activity ranks of individual cell, and therefore is robust across datasets.
- `AddModuleScore_UCell` A wrapper for UCell to interact directly with Seurat objects. Given a Seurat object and a set of signatures, it calculates enrichment scores on single-cell level and returns them into the meta.data of the input Seurat object.
- `StoreRankings_UCell` Calculates and stores gene rankings for a single-cell dataset. Given a gene vs. cell matrix and a set of signatures, it calculates the rankings of expression for all genes in each cell. It can then be applied to the function `ScoreSignatures_UCell` to evaluate gene signatures on the gene expression ranks of individual cells.

Gene signatures

UCell evaluates the strength of gene signatures (or gene sets) in individual cells of your dataset. You may specify positive and negative (up- or down-regulated) genes in signatures. See the examples below:

```

markers <- list()
markers$Tcell_CD4 <- c("CD4", "CD40LG")
markers$Tcell_CD8 <- c("CD8A", "CD8B")
markers$Tcell_Treg <- c("FOXP3", "IL2RA")
markers$Tcell_gd <- c("TRDC+", "TRGC1+", "TRGC2+",
                    "TRDV1+", "TRAC-", "TRBC1-", "TRBC2-")
markers$Tcell_NK <- c("FGFBP2+", "SPON2+", "KLRF1+",
                    "FCGR3A+", "CD3E-", "CD3G-")

```

If you don't specify +/- for genes, they are assumed to be all as a positive set. The UCell score is calculated as:

$$U = \max(0, U^+ - w_{neg} * U^-)$$

where U^+ and U^- are respectively the UCell scores for the positive and negative set, and w_{neg} is a weight on the negative set. When no negative set of genes is present, $U = U^+$

References

UCell: robust and scalable single-cell gene signature scoring. Massimo Andreatta & Santiago J Carmona (2021) CSBJ <https://doi.org/10.1016/j.csbj.2021.06.043>

u_stat	<i>Calculate Mann Whitney U from a vector of ranks</i>
--------	--

Description

Calculate Mann Whitney U from a vector of ranks

Usage

```
u_stat(rank_value, maxRank = 1000, sparse = FALSE)
```

Arguments

rank_value	A vector of ranks
maxRank	Max number of features to include in ranking
sparse	Whether the vector of ranks is in sparse format

Value

Normalized AUC (as U statistic) for the vector

u_stat_signature_list	<i>Calculate U scores for a list of signatures, given a rank matrix</i>
-----------------------	---

Description

Calculate U scores for a list of signatures, given a rank matrix

Usage

```
u_stat_signature_list(
  sig_list,
  ranks_matrix,
  maxRank = 1000,
  sparse = FALSE,
  w_neg = 1
)
```

Arguments

<code>sig_list</code>	A list of signatures
<code>ranks_matrix</code>	Matrix of pre-computed ranks
<code>maxRank</code>	Max number of features to include in ranking, for <code>u_stat</code> function
<code>sparse</code>	Whether the vector of ranks is in sparse format
<code>w_neg</code>	Weight on negative signatures

Value

A matrix of U scores

Index

* datasets

sample.matrix, 8

AddModuleScore_UCell, 2

BiocParallel::bparam(), 3, 9, 11

calculate_Uscore, 4

check_genes, 5

check_signature_names, 6

data.table::frank, 3, 9, 11

data_to_ranks_data_table, 6

rankings2Uscore, 7

sample.matrix, 8

ScoreSignatures_UCell, 8, 11

SingleCellExperiment, 9, 11

split_data.matrix, 10

StoreRankings_UCell, 9, 10

u_stat, 13

u_stat_signature_list, 13

UCell, 12