Package 'Seqtometry'

November 12, 2026

Title Signature scoring for single cell analysis

Version 1.0.1

URL https://github.com/HawigerLab/Seqtometry

BugReports https://github.com/HawigerLab/Seqtometry/issues

Description
This package provides functions used in Seqtometry (Kousnetsov et al. 2024), a method for analyzing single cell (scRNA-seq or scATAC-seq) data via signature (gene set) enrichment scores. The Seqtometry scores may be useful for annotating or characterizing cells, either in a flow cytometry like workflow (where scores are standalone features used for progressive partitoning as described in the Seqtometry publication) or in a cluster-based workflow (as features of clusters). The exported impute function (a port of Python's MAGIC-impute, van Dijk et al. 2018), may also be useful for single cell analysis on its own.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.5.0)

LinkingTo Rcpp, RcppArmadillo

Imports BiocSingular, checkmate, data.table, DelayedMatrixStats, future.apply, Matrix, MatrixGenerics, purrr, Rcpp, RcppHNSW, RSpectra, sparseMatrixStats, zeallot

Suggests BiocStyle, box, dplyr, future, ggplot2, harmony, knitr, MASS, patchwork, rmarkdown, scater, scuttle, SingleCellExperiment, stringr, TENxPBMCData, testthat (>= 3.0.0), tibble

biocViews SingleCell, GeneSetEnrichment, GeneExpression

VignetteBuilder knitr

Config/testthat/edition 3

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

git_branch RELEASE_3_23

git_last_commit 6ff53d7

git_last_commit_date 2026-07-27

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Robert Kousnetsov [aut, cre],
Daniel Hawiger [cph, fnd]

Maintainer Robert Kousnetsov <robert.kousnetsov@health.slu.edu>

Contents

Seqtometry-package 2
.apply_diff_op 3
.calc_diff_op 3
.calc_pca 4
.check_params 4
.gene_indices 5
.invert_pca 5
.minmax_scale 6
.normalize 6
.procrustes 7
impute 7
score 9
wks 11

Index **12**

`Seqtometry-package` *Signature scoring for single cell analysis*

Description

This package provides functions used in Seqtometry (Kousnetsov et al. 2024), a method for analyzing single cell (scRNA-seq or scATAC-seq) data via signature (gene set) enrichment scores. The Seqtometry scores may be useful for annotating or characterizing cells, either in a flow cytometry like workflow (where scores are standalone features used for progressive partitoning as demonstrated in the Seqtometry publication) or in a cluster-based workflow (as features of clusters). The exported impute function (a port of Python's MAGIC-impute, van Dijk et al. 2018), may also be useful for single cell analysis on its own.

Author(s)

Robert Kousnetsov <robert.kousnetsov@health.slu.edu>

Maintainer: Robert Kousnetsov <robert.kousnetsov@health.slu.edu>

.apply_diff_op *Perform data diffusion*

Description

Apply diffusion operator to data in order to perform imputation.

Usage

```
.apply_diff_op(gex, pcs, aff, dft, t_max, tol, exact_solver)
```

Arguments

gex	**matrix or Matrix** Gene expression matrix
pcs	**matrix** Principal components matrix
aff	**dgCMatrix** Markov affinity matrix
dft	**NULL or integer(1)** Diffusion time
t_max	**integer(1)** Maximum diffusion time
tol	**numeric(1)** Tolerance for Procrustes disparity
exact_solver	**logical(1)** Perform imputation in gene space

Value

list(matrix, integer(1)) Imputed matrix and diffusion time used

.calc_diff_op *Compute diffusion operator*

Description

Calculate graph diffusion operator (Markov affinity matrix).

Usage

```
.calc_diff_op(pcs, knn, ka, dist_metric)
```

Arguments

pcs	**matrix** Principal components matrix (used for kNN search)
knn	**integer(1)** Number of nearest neighbors to search for
ka	**integer(1)** Number of nearest neighbors to use for adaptive kernel
dist_metric	**character(1)** Type of metric to use for distance calculations during kNN search

Value

dgCMatrix Markov affinity matrix

.calc_pca *PCA wrapper*

Description

Calculate leading principal components (via truncated singular value decomposition).

Usage

```
.calc_pca(gex, npc, scale)
```

Arguments

gex	**matrix or Matrix** Gene expression matrix (without any zero variance genes)
npc	**numeric(1)** Number of leading principal components to compute
scale	**logical(1)** Whether to scale genes to unit variance

Value

list PC loading/rotation matrices as well as centering/scaling vectors

.check_params *Parameter validation*

Description

Checks that all parameters used in `impute` are valid.

Usage

```
.check_params(args)
```

Arguments

args	**list** The arguments to the magic_impute function

Value

NULL (but stops execution for invalid parameters)

.gene_indices *Finds row indices of signature genes*

Description

For converting from character to integer based indexing

Usage

```
.gene_indices(mat, gss)
```

Arguments

mat	**matrix-like** Gene expression data (genes x cells)
gss	**named list of character** Signature genes (with same nomenclature system as mat)

Value

integer 0-based indices (for passing to Rcpp function) of signature genes

.invert_pca *PCA inversion*

Description

Reverses operations done for PCA: back-rotation, unscaling, and uncentering.

Usage

```
.invert_pca(pcs, rot, ctr, sdv, low_mem)
```

Arguments

pcs	**matrix** The principal components (scaled left singular vectors)
rot	**matrix** The rotation matrix (right singular vectors)
ctr	**integer** The centering vector
sdv	**integer or NULL** The scaling vector (or NULL if no scaling was applied)
low_mem	**logical(1)** Whether to use delayed operations to reduce memory usage

Value

matrix or DelayedMatrix `rot %*% t(pcs) * sdv + ctr`

.minmax_scale *Minmax transform*

Description

Scales input vector to unit range

Usage

```
.minmax_scale(x)
```

Arguments

x	**numeric** Values to be scaled

Value

numeric Minmax transformed values

.normalize *LogCP10K transform*

Description

Simple normalization method for scRNA-seq data.

Usage

```
.normalize(gex)
```

Arguments

gex	**matrix or Matrix** Gene expression matrix (cells x genes)

Value

matrix or Matrix Transformed (normalized) matrix

.procrustes *Procrustes disparity*

Description

Calculates symmetric Procrustes distance (adapted from MATLAB procrustes).

Usage

```
.procrustes(x, y)
```

Arguments

x	**matrix**
y	**matrix**

Value

numeric(1) Procrustes disparity between input matrices

impute *MAGIC imputation (van Dijk et al. 2018)*

Description

Calculates a graph diffusion operator for the given input matrix and applies it to produce an imputed matrix.

Usage

```
impute(
  gex,
  transpose = TRUE,
  do_norm = FALSE,
  pca = NULL,
  npc = 100L,
  scale = TRUE,
  knn = 16L,
  ka = 6L,
  dist_metric = "euclidean",
  dft = NULL,
  t_max = 16L,
  tol = 0.001,
  exact_solver = TRUE,
  conserve_memory = FALSE,
```

```
  env_ret = FALSE,
  verbose = FALSE
)
```

Arguments

gex	**matrix or Matrix** Gene expression values (that has passed quality control).
transpose	**logical(1)** Whether to transpose gex (make it cells x genes) prior to downstream operations.
do_norm	**logical(1)** Whether to perform LogCP10K normalization on gex.
pca	**matrix (cells x PCs) or NULL** Precomputed principal component matrix (or NULL to derive it from gex).
npc	**integer(1)** Number of principal components (min = 1) to calculate.
scale	**logical(1)** Whether to scale columns of input matrix to unit variance prior to PCA.
knn	**integer(1)** Number of nearest neighbors (min = 2) to consider during distance calculation.
ka	**integer(1)** Number of nearest neighbors (min = 2, max <= knn) to use for the adaptive kernel.
dist_metric	**character(1)** Type of metric to use for distance calculations during kNN search.
dft	**NULL or integer(1)** Automatic (NULL) or user-defined (integer) diffusion time (min = 1, max = 16).
t_max	**integer(1)** Maximum diffusion time to test when using automatic diffusion time (min = 1, max = 16).
tol	**numeric(1)** Threshold for Procrustes disparity (min = 0, max = 1) between successive diffusion times.
exact_solver	**logical(1)** Whether to perform imputation in gene space (TRUE) or PCA space (FALSE).
conserve_memory	**logical(1)** Whether to avoid allocating a large dense matrix when `exact_solver = FALSE`.
env_ret	**logical(1)** Return all variables in the environment (TRUE) or just the imputed matrix (FALSE).
verbose	**logical(1)** Whether to print messages at different major parts of the algorithm.

Value

matrix-like or list If `env_ret = FALSE`, then just the imputed matrix. Otherwise the function environment as a list containing all parameters (possibly modified) as well as

- imp **matrix or DelayedMatrix** Imputed matrix.
- aff **dgCMatrix** Markov affinity matrix (graph diffusion operator).
- pca **list** Possibly computed (if pca was NULL), yielding a four element list, where:
  - x **matrix (cells x PCs)** The principal components matrix (scaled left singular vectors).

- `v` **matrix (genes x PCs)** The rotation matrix (right singular vectors).
- `center` **integer (cells)** The centering vector.
- `scale` **integer (cells) or NULL** The scaling vector (or NULL if no scaling was applied).

Examples

```
box::use(
  TENxPBMCData[TENxPBMCData],
  SingleCellExperiment[rowData, logcounts],
  scuttle[quickPerCellQC, logNormCounts],
  scater[runUMAP, plotReducedDim],
  patchwork[wrap_plots])

# PBMC data, basic processing pipeline
dat <- TENxPBMCData(dataset = "pbmc3k")
dimnames(dat) <- list(
  rowData(dat)[["Symbol_TENx"]],
  dat[["Barcode"]])
dat <- dat |>
  quickPerCellQC() |>
  logNormCounts() |>
  runUMAP()

# MAGIC imputation
imp <- logcounts(dat) |>
  as("dgCMatrix") |>
  impute()

# Visualize unimputed versus imputed expression
# on UMAP plots for a gene of interest (GOI)
goi <- "CD19"
dat[["Imputed_GOI"]] <- imp[goi, ]
p1 <- plotReducedDim(dat, "UMAP", color_by = goi)
p2 <- plotReducedDim(dat, "UMAP", color_by = "Imputed_GOI")
wrap_plots(p1, p2, ncol = 2)
```

---

score *Seqtometry scoring (Kousnetsov et al. 2024)*

Description

Computes signature scores (a weighted KS-like statistic) for single cell expression data

Usage

```
score(mat, signatures, minmax = TRUE)
```

Arguments

mat	**matrix, Matrix, or DelayedMatrix** Gene expression data (genes x cells)
signatures	**named list of character** Signature genes (with same nomenclature system used in mat)
minmax	**logical(1)** Whether to perform minmax transform on scoring results (default: TRUE)

Value

data.table Single cell scores (cells x signatures) for each signature, where cell barcodes are stored in the "id" column

Examples

```
box::use(
  TENxPBMCData[TENxPBMCData],
  SingleCellExperiment[rowData, logcounts],
  scuttle[quickPerCellQC, logNormCounts],
  scater[runUMAP, plotReducedDim],
  patchwork[wrap_plots])

# PBMC data, basic processing pipeline
dat <- TENxPBMCData(dataset = "pbmc3k")
dimnames(dat) <- list(
  rowData(dat)[["Symbol_TENx"]],
  dat[["Barcode"]])
dat <- dat |>
  quickPerCellQC() |>
  logNormCounts() |>
  runUMAP()

# MAGIC imputation
imp <- logcounts(dat) |>
  as("dgCMatrix") |>
  impute()

# Score with a B cell signature (gene set)
options(future.globals.maxSize = 1024^3)
b_cell_sig <- list("B_cell" = c("CD19", "MS4A1", "CD79A", "CD79B"))
dat[["B cell signature"]] <- Seqtometry::score(imp, b_cell_sig)[["B_cell"]]

# Visualize a hallmark B cell gene versus a B cell signature score
p1 <- plotReducedDim(dat, "UMAP", color_by = "CD19")
p2 <- plotReducedDim(dat, "UMAP", color_by = "B cell signature")
wrap_plots(p1, p2, ncol = 2)
```

wks *Helper function for performing a weighted Kolmogorov-Smirnov-like procedure.*

Description

Helper function for performing a weighted Kolmogorov-Smirnov-like procedure.

Usage

```
wks(gex, gss, mus, sds)
```

Arguments

gex	numeric: normalized gene expression values
gss	list: indices of genes in each gene set
mus	numeric: means of all genes
sds	numeric: standard deviations of all genes

Value

Modified Kuiper statistic (sum of minimal and maximal deviations during running sum)

Index

.apply_diff_op, 3
.calc_diff_op, 3
.calc_pca, 4
.check_params, 4
.gene_indices, 5
.invert_pca, 5
.minmax_scale, 6
.normalize, 6
.procrustes, 7

impute, 7

score, 9
Seqtometry *(*Seqtometry-package*)*, 2
Seqtometry-package, 2

wks, 11