Package 'SPECK'

November 17, 2023

Type Package

Title Receptor Abundance Estimation using Reduced Rank Reconstruction and Clustered Thresholding

Version 1.0.0

Maintainer Azka Javaid <azka.javaid.gr@dartmouth.edu>

Description Surface Protein abundance Estimation using CKmeans-based clustered thresholding ('SPECK') is an unsupervised learning-based method that performs receptor abundance estimation for single cell RNA-sequencing data based on reduced rank reconstruction (RRR) and a clustered thresholding mechanism. Seurat's normalization method is described in: Hao et al., (2021) <doi:10.1016/j.cell.2021.04.048>, Stuart et al., (2019) <doi:10.1016/j.cell.2019.05.031>, Butler et al., (2018) <doi:10.1038/nbt.4096> and Satija et al., (2015) <doi:10.1038/nbt.3192>. Method for the RRR is further detailed in: Erichson et al., (2019) <doi:10.18637/jss.v089.i11> and Halko et al., (2009) <arXiv:0909.4061>. Clustering method is outlined in: Song et al., (2020) <doi:10.1093/bioinformatics/btaa613> and Wang et al., (2011) <doi:10.32614/RJ-2011-015>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Suggests ggplot2, gridExtra, knitr, rmarkdown, SeuratObject, usethis

VignetteBuilder knitr

Depends R (>= 2.10)

Imports Ckmeans.1d.dp, magrittr, Matrix (>= 1.6.1.1), rsvd, Seurat

NeedsCompilation no

Author H. Robert Frost [aut], Azka Javaid [aut, cre]

Repository CRAN

Date/Publication 2023-11-17 17:30:02 UTC

R topics documented:

`ckmeansThreshold` *Clustered thresholding of a vector.*

Description

Performs thresholding for a vector of length m from a corresponding mxn reduced rank reconstructed (RRR) matrix. Thresholding of a vector is only performed if more than one cluster is identified using the `Ckmeans.1d.dp::Ckmeans.1d.dp()` function based on a one-dimensional dynamic programming clustering algorithm, which functions by minimizing the sum of squares of within-cluster distances from an element to its associated cluster mean. If more than one cluster is present, then the RRR output corresponding to the nonzero elements of the least-valued cluster, as identified by the cluster mean, is set to zero. All other values in the least and higher-valued clusters are retained.

Usage

```
ckmeansThreshold(rrr.vector, max.num.clusters = 4, seed.ckmeans = 2)
```

Arguments

`rrr.vector`	Vector of length m from the corresponding mxn RRR matrix.
`max.num.clusters`	Maximum number of clusters for computation.
`seed.ckmeans`	Seed specified to ensure reproducibility of the clustered thresholding.

Value

- rrr.thresholded.vector - A thresholded vector of length m.
- num.centers - Number of identified clusters.
- max.clust.prop - Proportion of samples with the specified maximum number of clusters.

Examples

```
set.seed(10)
data.mat <- matrix(data = rbinom(n = 18400, size = 230, prob = 0.01), nrow = 80)
rrr.object <- randomizedRRR(counts.matrix = data.mat, rank.range.end = 60,
min.consec.diff = 0.01, rep.consec.diff = 2,
manual.rank = NULL, seed.rsvd = 1)
thresh.full.output <- ckmeansThreshold(rrr.vector = rrr.object$rrr.mat[,1],
```

```
max.num.clusters = 4, seed.ckmeans = 2)
head(thresh.full.output$rrr.thresholded.vector)
print(thresh.full.output$num.centers)
print(thresh.full.output$max.clust.prop)
```

`pbmc.rna.mat` *Single cell RNA-sequencing (scRNA-seq) peripheral blood (PBMC) data sample.*

Description

Single cell RNA-sequencing (scRNA-seq) subset of the Hao et al. 2021 human peripheral blood mononuclear cell (PBMC) data (GEO: GSE164378, DOI: 10.1016/j.cell.2021.04.048).

Usage

```
pbmc.rna.mat
```

Format

A scRNA-seq data of dgCMatrix class with 1000 rows and 33538 columns

pbmc.rna.mat RNA expression data for 1000 cells and 33538 genes

---

`randomizedRRR` *Reduced rank reconstruction (RRR) of a matrix.*

Description

Computes the rank and subsequent RRR of a mxn counts matrix. Log-normalization is first performed using the `Seurat::NormalizeData()` function. RRR is next performed on the normalized mxn matrix using randomized Singular Value Decomposition with the `rsvd::rsvd()` function. Estimated rank is selected via a construction of the standard deviations of non-centered sample principal components, which are used in a subsequent rate of change computation where each successive standard deviation value is compared to the previous to determine the rank at which the absolute value of the rate of change between consecutive values is at least 0.01 for at least two value pairs.

Usage

```
randomizedRRR(
  counts.matrix,
  rank.range.end = 100,
  min.consec.diff = 0.01,
  rep.consec.diff = 2,
  manual.rank = NULL,
  seed.rsvd = 1
)
```

Arguments

counts.matrix	A mxn counts matrix.
rank.range.end	Upper value of the rank for RRR.
min.consec.diff	Minimum difference in the rate of change between a pair of successive standard deviation estimate.
rep.consec.diff	Frequency of the minimum difference in the rate of change between a pair of successive standard deviation estimate.
manual.rank	Optional, user-specified upper value of the rank used for RRR as an alternative to automatically computed rank.
seed.rsvd	Seed specified to ensure reproducibility of the RRR.

Value

- rrr.mat - A mxn RRR matrix.
- rrr.rank - Automatically computed rank.
- component.stdev - A vector corresponding to standard deviations of non-centered sample principal components.

Examples

```
set.seed(10)
data.mat <- matrix(data = rbinom(n = 18400, size = 230, prob = 0.2), nrow = 80)
rrr.object <- randomizedRRR(counts.matrix = data.mat, rank.range.end = 60,
min.consec.diff = 0.01, rep.consec.diff = 2,
manual.rank = NULL, seed.rsvd = 1)
print(rrr.object$component.stdev)
print(rrr.object$rrr.rank)
dim(rrr.object$rrr.mat); str(rrr.object$rrr.mat)
```

---

speck *Abundance estimation for single cell RNA-sequencing (scRNA-seq) data.*

Description

Performs normalization, reduced rank reconstruction (RRR) and thresholding for a mxn scRNA-seq matrix with m samples and n genes. The `speck()` function calls the `randomizedRRR()` function on the scRNA-seq matrix. Thresholding is next applied to each gene from the mxn RRR matrix using the `ckmeansThreshold()` function, resulting in a mxn thresholded matrix. See documentation for the `randomizedRRR()` and `ckmeansThreshold()` functions for individual implementation details.

Usage

```
speck(
  counts.matrix,
  rank.range.end = 100,
  min.consec.diff = 0.01,
  rep.consec.diff = 2,
  manual.rank = NULL,
  max.num.clusters = 4,
  seed.rsvd = 1,
  seed.ckmeans = 2
)
```

Arguments

`counts.matrix`	mxn scRNA-seq counts matrix with m samples and n genes.
`rank.range.end`	Upper value of the rank for RRR.
`min.consec.diff`	Minimum difference in the rate of change between a pair of successive standard deviation estimate.
`rep.consec.diff`	Frequency of the minimum difference in the rate of change between a pair of successive standard deviation estimate.
`manual.rank`	Optional, user-specified upper value of the rank used for RRR as an alternative to automatically computed rank.
`max.num.clusters`	Maximum number of clusters for computation.
`seed.rsvd`	Seed specified to ensure reproducibility of the RRR.
`seed.ckmeans`	Seed specified to ensure reproducibility of the clustered thresholding.

Value

- thresholded.mat - A mxn thresholded RRR matrix with m samples and n genes.
- rrr.mat - A mxn RRR matrix with m samples and n genes.
- rrr.rank - Automatically computed rank.
- component.stdev - A vector corresponding to standard deviations of non-centered sample principal components.
- clust.num - A vector of length n indicating the number of clusters identified by the `Ckmeans.1d.dp()` algorithm for each gene.
- clust.max.prop - A vector of length n indicating the proportion of samples with the specified maximum number of clusters for each gene.

Examples

```
set.seed(10)
data.mat <- matrix(data = rbinom(n = 18400, size = 230, prob = 0.01), nrow = 80)
speck.full <- speck(counts.matrix = data.mat, rank.range.end = 60,
```

```
min.consec.diff = 0.01, rep.consec.diff = 2,
manual.rank = NULL, max.num.clusters = 4,
seed.rsvd = 1, seed.ckmeans = 2)
print(speck.full$component.stdev)
print(speck.full$rrr.rank)
head(speck.full$clust.num); table(speck.full$clust.num)
head(speck.full$clust.max.prop); table(speck.full$clust.max.prop)
speck.output <- speck.full$thresholded.mat
dim(speck.output); str(speck.output)
```

Index