Package ‘bbknnR’

May 27, 2022

Title Perform Batch Balanced KNN in R

Version 1.0.0

Date 2022-05-10

Description A fast and intuitive batch effect removal tool for single-cell data. BBKNN is originally used in the 'scanpy' python package, and now can be used with 'Seurat' seamlessly.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0), methods, utils

LinkingTo Rcpp (>= 1.0.8)

Imports dplyr, glmnet, Matrix, Rcpp, RcppAnnoy, reticulate, Rtsne, Seurat, SeuratObject, tidytable, uwot

LazyData true

RoxygenNote 7.1.2

URL `https://github.com/ycli1995/bbknnR`,
`https://github.com/Teichlab/bbknn`,
`https://bbknn.readthedocs.io/en/latest/`

BugReports `https://github.com/ycli1995/bbknnR/issues`

Suggests knitr, rmarkdown, testthat (>= 3.0.0), patchwork

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation yes

Author Yuchen Li [aut, cre]

Maintainer Yuchen Li <`liyuchen_1995@outlook.com`>

Repository CRAN

Date/Publication 2022-05-27 08:00:12 UTC

R topics documented:

`panc8_small` *A small example version of the pancreas scRNA-seq dataset*

Description

A subsetted version of the pancreas scRNA-seq dataset to test BBKNN

Usage

```
panc8_small
```

Format

A Seurat object with the following slots filled

assays Currently only contains one assay ("RNA" - scRNA-seq expression data)

counts - Raw expression data

- data - Normalized expression data
- scale.data - Scaled expression data
- var.features - names of the current features selected as variable
- meta.features - Assay level metadata such as mean and variance

meta.data Cell level metadata

active.assay Current default assay

active.ident Current default idents

graphs Empty

reductions Dimensional reductions: currently PCA

version Seurat version used to create the object

commands Command history

Source

SeuratData `https://github.com/satijalab/seurat-data`

RidgeRegression *Perform ridge regression on scaled expression data*

Description

Perform ridge regression on scaled expression data, accepting both technical and biological categorical variables. The effect of the technical variables is removed while the effect of the biological variables is retained. This is a preprocessing step that can aid BBKNN integration.

Usage

```
RidgeRegression(object, ...)

## Default S3 method:
RidgeRegression(
  object,
  latent_data,
  batch_key,
  confounder_key,
  lambda = 1,
  seed = 42,
  verbose = TRUE,
  ...
)

## S3 method for class 'Seurat'
RidgeRegression(
  object,
  batch_key,
  confounder_key,
  assay = NULL,
  features = NULL,
  lambda = 1,
  run_pca = TRUE,
  npcs = 50,
  reduction.name = "pca",
  reduction.key = "PC_",
  replace = FALSE,
  seed = 42,
  verbose = TRUE,
  ...
)
```

Arguments

`object`	An object
`...`	Arguments passed to other methods

`latent_data` Extra data to regress out, should be cells x latent data

`batch_key` Variables to regress out as technical effects. Must be included in column names of latent_data

`confounder_key` Variables to to retain as biological effects. Must be included in column names of latent_data

`lambda` A user supplied lambda sequence. pass to `glmnet`

`seed` Set a random seed. By default, sets the seed to 42. Setting NULL will not set a seed.

`verbose` Whether or not to print output to the console

`assay` Name of Assay ridge regression is being run on

`features` Features to compute ridge regression on. If features=NULL, ridge regression will be run using the variable features for the Assay.

`run_pca` Whether or not to run pca with regressed expression data (TRUE by default)

`npcs` Total Number of PCs to compute and store (50 by default)

`reduction.name` Dimensional reduction name (pca by default)

`reduction.key` Dimensional reduction key, specifies the string before the number for the dimension names (PC by default)

`replace` Whether or not to replace original scale.data with regressed expression data (TRUE by default)

Value

Returns a Seurat object.

References

Park, Jong-Eun, et al. "A cell atlas of human thymic development defines T cell repertoire formation." Science 367.6480 (2020): eaay3224.

`RunBBKNN` *Perform batch balanced KNN*

Description

Batch balanced KNN, altering the KNN procedure to identify each cell's top neighbours in each batch separately instead of the entire cell pool with no accounting for batch. The nearest neighbours for each batch are then merged to create a final list of neighbours for the cell. Aligns batches in a quick and lightweight manner.

Usage

```
RunBBKNN(object, ...)

## Default S3 method:
RunBBKNN(
  object,
  batch_list,
  n_pcs = 50L,
  neighbors_within_batch = 3L,
  trim = NULL,
  approx = TRUE,
  use_annoy = TRUE,
  annoy_n_trees = 10L,
  pynndescent_n_neighbors = 30L,
  pynndescent_random_state = 0L,
  use_faiss = TRUE,
  metric = "euclidean",
  set_op_mix_ratio = 1,
  local_connectivity = 1,
  seed = 42,
  verbose = TRUE,
  ...
)

## S3 method for class 'Seurat'
RunBBKNN(
  object,
  batch_key,
  assay = NULL,
  reduction = "pca",
  n_pcs = 50L,
  graph_name = "bbknn",
  set_op_mix_ratio = 1,
  local_connectivity = 1,
  run_TSNE = TRUE,
  TSNE_name = "tsne",
  TSNE_key = "tSNE_",
  run_UMAP = TRUE,
  UMAP_name = "umap",
  UMAP_key = "UMAP_",
  min_dist = 0.3,
  spread = 1,
  seed = 42,
  verbose = TRUE,
  ...
)
```

Arguments

`object` An object

`...` Arguments passed to other methods

`batch_list` A character vector with the same length as nrow(pca)

`n_pcs` Number of dimensions to use. Default is 50.

`neighbors_within_batch`
How many top neighbours to report for each batch; total number of neighbours in the initial k-nearest-neighbours computation will be this number times the number of batches. This then serves as the basis for the construction of a symmetrical matrix of connectivities.

`trim` Trim the neighbours of each cell to these many top connectivities. May help with population independence and improve the tidiness of clustering. The lower the value the more independent the individual populations, at the cost of more conserved batch effect. Default is 10 times neighbors_within_batch times the number of batches. Set to 0 to skip.

`approx` If TRUE, use approximate neighbour finding - RcppAnnoy or pyNNDescent. This results in a quicker run time for large datasets while also potentially increasing the degree of batch correction.

`use_annoy` Only used when approx = TRUE. If TRUE, will use RcppAnnoy for neighbour finding. If FALSE, will use pyNNDescent instead.

`annoy_n_trees` Only used with annoy neighbour identification. The number of trees to construct in the annoy forest. More trees give higher precision when querying, at the cost of increased run time and resource intensity.

`pynndescent_n_neighbors`
Only used with pyNNDescent neighbour identification. The number of neighbours to include in the approximate neighbour graph. More neighbours give higher precision when querying, at the cost of increased run time and resource intensity.

`pynndescent_random_state`
Only used with pyNNDescent neighbour identification. The RNG seed to use when creating the graph.

`use_faiss` If approx = FALSE and the metric is "euclidean", use the faiss package to compute nearest neighbours if installed. This improves performance at a minor cost to numerical precision as faiss operates on float32.

`metric` What distance metric to use. The options depend on the choice of neighbour algorithm. "euclidean", the default, is always available.

`set_op_mix_ratio`
Pass to 'set_op_mix_ratio' parameter for `umap`

`local_connectivity`
Pass to 'local_connectivity' parameter for `umap`

`seed` Set a random seed. By default, sets the seed to 42. Setting NULL will not set a seed.

`verbose` Whether or not to print output to the console `batch_key` Column name in meta.data discriminating between your batches.

`assay` used to construct Graph.

`reduction` Which dimensional reduction to use for the BBKNN input. Default is PCA

`graph_name` Name of the generated BBKNN graph. Default is bbknn.

`run_TSNE` Whether or not to run t-SNE based on BBKNN results.

`TSNE_name` Name to store t-SNE dimensional reduction.

`TSNE_key` Specifies the string before the number of the t-SNE dimension names. tSNE by default.

`run_UMAP` Whether or not to run UMAP based on BBKNN results.

`UMAP_name` Name to store UMAP dimensional reduction.

`UMAP_key` Specifies the string before the number of the UMAP dimension names. tSNE by default.

`min_dist` Pass to ’min_dist’ parameter for `umap`

`spread` Pass to ’spread’ parameter for `umap`

Value

Returns a Seurat object containing a new BBKNN Graph. If run t-SNE or UMAP, will also return corresponded reduction objects.

References

Polański, Krzysztof, et al. "BBKNN: fast batch alignment of single cell transcriptomes." Bioinformatics 36.3 (2020): 964-965.

Index