Package 'EnImpute'

January 12, 2019

Type Package

Title An ensemble learning method for imputing single cell RNA sequencing data

Version 1.0

Author Xiao-Fei Zhang

Maintainer Xiao-Fei Zhang <zhangxf@mail.ccnu.edu.cn>

Description EnImpute is an ensemble learning method for imputing single cell RNA sequencing data. It ensembles results from multiple individual imputation methods. The current implementation of EnImpute integrates seven state-of-the-art methods: ALRA, DCA, DrImpute, MAGIC, SAVER, scImpute and Seurat.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Depends R (>= 3.4), DrImpute, Rmagic, rsvd, SAVER, Seurat, scImpute, stats

NeedsCompilation no

R topics documented:

`baron` *Human pancreatic islet data*

Description

This is the Human pancreatic islet dataset (GSM2230757). The raw data contains 20,125 genes and 1,937 cells. Here we use the reference and downsampled datasets generated by Huang et al (2018) which contain 2,284 genes and 1,076 cells (available at https://github.com/mohuangx/SAVER-paper/tree/master/SAVER-data). For details about the approach to generate the reference and downsampled datasets, please refer to Huang et al (2018). This data is an object of class list of length two. count.ref is the reference count matrix and count.samp is the downsampled count matrix.

Usage

```
baron
```

Format

An object of class `list` of length 2.

Author(s)

Xiao-Fei zhang, <zhangxf@mail.ccnu.edu.cn>

References

Baron, Maayan, et al (2016). A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346-360.

Mo Huang et al (2018). SAVER: gene expression recovery for single-cell RNA sequencing. *Nature Methods*, 15:539-542.

Xiao-Fei Zhang et al (2018), EnImpute: An ensemble learning method for imputing single cell RNA sequencing data. submmited

Examples

```
data("baron")
```

`EnImpute` *Run EnImpute on a raw read count matrix*

Description

This function is implemented to run EnImpute on a raw read count matrix. EnImpute is an ensemble learning-based method for imputing dropout values in scRNA-seq data. The current implementation of EnImpute integrates seven state-of-the-art methods: Adaptively-thresholded low rank approximation (ALRA), Deep count autoencoder network (DCA), DrImpute, Markov affinity-based graph imputation of cells (MAGIC), Single-cell analysis via expression recovery (SAVER), scImpute and Seurat. EnImpute first runs the seven individual imputation methods, and then uses the trimmed mean of the imputed values generated by different individual methods as a consensus result. This function depends on the follwing R package: DrImpute, Rmagic, rsvd, SAVER, Seurat, scImpute, stats. These R packages will be automatically installed along with EnImpute. EnImpute also depends on the following two Python packages: dca and MAGIC. Before installing the R package EnImpute, please install the two Python packages following the corresponding readme files, and check whether they can be run from the command line.

Usage

```
EnImpute(count, scale.factor = 10000, trim = 0.3, ALRA = TRUE,
  DCA = TRUE, DrImpute = TRUE, MAGIC = TRUE, SAVER = TRUE,
  scImpute = TRUE, Seurat = TRUE, ALRA.k = 0, ALRA.q = 10,
  DCA.normtype = "zheng", DCA.type = "zinb-conddisp", DCA.l2 = 0,
  DCA.l1 = 0, DCA.l2enc = 0, DCA.l1enc = 0, DCA.ridge = 0,
  DCA.gradclip = 5, DCA.activation = "relu", DCA.hiddensize = "64,32,64",
  DCA.hyper = FALSE, DCA.hypern = 1000, DrImpute.ks = 10:15,
  DrImpute.dists = c("spearman", "pearson"), DrImpute.method = "mean",
  DrImpute.cls = NULL, MAGIC.k = 10, MAGIC.alpha = 15, MAGIC.t = "auto",
  MAGIC.npca = 20, MAGIC.t.max = 20, MAGIC.knn.dist.method = "euclidean",
  MAGIC.n.jobs = 1, SAVER.do.fast = TRUE, SAVER.ncores = 2,
  SAVER.size.factor = NULL, SAVER.npred = NULL, SAVER.null.model = FALSE,
  SAVER.mu = NULL, scImpute.drop_thre = 0.5, scImpute.Kcluster = 10,
  scImpute.labeled = FALSE, scImpute.labels = NULL,
  scImpute.genelen = NULL, scImpute.ncores = 1, Seurat.genes.use = NULL,
  Seurat.genes.fit = NULL, Seurat.gram = TRUE)
```

Arguments

count	raw read count matrix. The rows correspond to genes and the columns correspond to cells.
scale.factor	scale factor used to re-scale the imputed results generated by different individual imputation methods. Default is 10000.
trim	specifies the fraction (between 0 and 0.5) of observations to be trimmed from each end before the mean is computed. Default is 0.3.
ALRA	a boolean variable that defines whether to impute the raw data using the ALRA method. Default is TRUE.
DCA	a boolean variable that defines whether to impute the raw data using the DCA method. Default is TRUE. If the Python package dca has not been installed correctly, please set "DCA=FALSE".
DrImpute	a boolean variable that defines whether to impute the raw data using the DrImpute method. Default is TRUE.
MAGIC	a boolean variable that defines whether to impute the raw data using the MAGIC method. Default is TRUE. If the Python package magic has not been installed correctly, please set "MAGIC=FALSE".
SAVER	a boolean variable that defines whether to impute the raw data using the SAVER method. Default is TRUE.
scImpute	a boolean variable that defines whether to impute the raw data using the scImpute method. Default is TRUE.
Seurat	a boolean variable that defines whether to impute the raw data using the Seurat method. Default is TRUE.
ALRA.k	the rank of the rank-k approximation in ALRA. Set to 0 for automated choice of k. Default is 0.
ALRA.q	the number of power iterations in randomized SVD used by ALRA. Default is 10.
DCA.normtype	a string variable specifying the type of size factor estimation in DCA. Possible values: "deseq", "zheng". Default is "zheng".

DCA.type	a string variable specifying type of autoencoder in DCA. Possible values: "normal", "poisson", "nb", "nb-shared", "nb-conddisp", "nb-fork", "zinb", "zinb-shared", "zinb-conddisp", "zinb-fork". Default is "zinb-conddisp".
DCA.l2	a real number specifying the L2 regularization coefficient in DCA. Default is 0.
DCA.l1	a real number specifying the L1 regularization coefficient in DCA. Default is 0.
DCA.l2enc	a real number specifying the encoder-specific L2 regularization coefficient in DCA. Default is 0.
DCA.l1enc	a real number specifying the encoder-specific L1 regularization coefficient in DCA. Default is 0.
DCA.ridge	a real number specifying the L2 regularization coefficient for dropout probabilities in DCA. Default is 0.
DCA.gradclip	a real number specifying the Clip grad values in DCA. Default is 5.
DCA.activation	a string value specifying the activation function of hidden unit in DCA. Default is "relu".
DCA.hiddensize	a string vector specifying the size of hidden layers in DCA. Default is "64,32,64".
DCA.hyper	a logical value specifying whether hyperparameter search is performed in DCA.
DCA.hypern	an integer specifying the number of samples drawn from hyperparameter distributions during optimization in DCA. Default is 1000.
DrImpute.ks	an integer vector specifying the number of cell clustering groups in DrImpute. Default is 10:15.
DrImpute.dists	a string vector specifying the distance metrics in DrImpute. Default is c("spearman", "pearson").
DrImpute.method	a string specifying the method used for imputation in DrImpute. Use "mean" for mean imputation, "med" for median imputation.
DrImpute.cls	a matrix specifying the clustering information manually provided by users in DrImpute. The rows represent different clusterings, and the columns represent cells. Default is NULL, which means the user do not provide the clustering information.
MAGIC.k	an integer specifying the number of nearest neighbors on which to build kernel in MAGIC. Default is 10.
MAGIC.alpha	an integer specifying the decay rate of kernel tails in MAGIC. Default is 15.
MAGIC.t	an integer specifying the diffusion time for the Markov Affinity Matrix in MAGIC. Default is "auto". For detail about the approach to set paramter t automatically, please refer to the reference.
MAGIC.npca	an integer specifying the number of PCA components in MAGIC. Default is 20.
MAGIC.t.max	an integer specifying the maximum value of t to test for automatic t selection in MAGIC. Default is 20.
MAGIC.knn.dist.method	a string value specifying the metric for building kNN graph in MAGIC. Recommended values: "euclidean", "cosine". Default is "euclidean".
MAGIC.n.jobs	an integer specifying the number of jobs used for computation in MAGIC. If -1 all CPUs are used. If 1 is given, no parallel computing code is used at all. For n.jobs below -1, (n.cpus + 1 + n.jobs) are used. Thus for n.jobs = -2, all CPUs but one are used.
SAVER.do.fast	a boolean variable specifying whether the prediction step is approximated in SAVER. Default is TRUE.

`SAVER.ncores` number of cores to use in SAVER. Default is 1.

`SAVER.size.factor`
: a vector of cell size specifying the normalization factors in SAVER. If the data is already normalized or normalization is not desired, set size.factor = 1. Default uses mean library size normalization.

`SAVER.npred` number of genes for regression prediction in SAVER. Select the top npred genes in terms of mean expression for regression prediction. Default is all genes.

`SAVER.null.model`
: a boolean variable specifying whether to use mean gene expression as prediction in SAVER. Default is FALSE

`SAVER.mu` matrix of prior means in SAVER.

`scImpute.drop_thre`
: a number (between 0 and 1) specifying the threshold on dropout probability in scImpute. Default is 0.5.

`scImpute.Kcluster`
: an integer specifying the number of cell subpopulations in scImpute. Default is 10.

`scImpute.labeled`
: a boolean variable indicating whether cell type information is given in scImpute. Default is FALSE.

`scImpute.labels`
: a character vector specifying the cell type in scImpute. Only needed when `labeled = TRUE`. Default is NULL

`scImpute.genelen`
: an integer vector giving the length of each gene in scImpute. Default is NULL.

`scImpute.ncores`
: an integer specifying the number of cores used for parallel computation in scImpute. Default is 1.

`Seurat.genes.use`
: a vector of genes that can be used for building the models in Seurat. Default use the high variable gene detected by the `FindVariableGenes` in the Seurat package.

`Seurat.genes.fit`
: a vector of genes to impute values for. Default is all genes.

`Seurat.gram` a logical value specifying whether the Gram matrix is precomputed in Seurat. Default is TRUE.

Value

a list with the following components

`count.EnImpute.log`
: Imputed count matrix generated by EnImpute (log scale).

`count.EnImpute.exp`
: Imputed count matrix generated by EnImpute (exp scale).

`count.imputed.individual.exp`
: Imputed count matrices generated by different individual imputation methods (exp scale).

`Methods.used` The individual methods used by EnImpute.

Author(s)

Xiao-Fei Zhang <zhangxf@mail.ccnu.edu.cn>

References

[1] George C. Linderman, et al. Zero-preserving imputation of scrna-seq data using low-rank approximation. bioRxiv, 2018.
[2] Gokcen Eraslan et al. Single cell rna-seq denoising using a deep count autoencoder. bioRxiv, page 00681, 2018.
[3] Il-Youp Kwak et al. Drimpute: Imputing dropout events in single cell rna sequencing data. BMC Bioinformatics, 19:220, 2018.
[4] David van Dijk et al. Recovering gene interactions from single-cell data using data diffusion. Cell, 174:1-14, 2018.
[5] Mo Huang et al. Saver: gene expression recovery for single-cell rna sequencing. Nature Methods, 15:539-542, 2018.
[6] Wei Vivian Li and Jingyi Jessica Li. An accurate and robust imputation method scimpute for singlecell rna-seq data. Nature Communications, 9(1):997, 2018.
[7] Rahul Satija et al. Spatial reconstruction of single-cell gene expression data. Nature Biotechnology, 33(5):495-502, 2015.
[8] Xiao-Fei Zhang et al, EnImpute: imputing dropout events in single cell RNA sequencing data via ensemble learning, 2019.

Examples

```
data("baron")
baron_imputation_result = EnImpute(baron$count.samp)

# do not use SAVER and scImpute as base imputation methods
data("baron")
baron_imputation_result = EnImpute(baron$count.samp, SAVER=FALSE, scImpute=FALSE)

# data("manno")
# manno_imputation_result = EnImpute(manno$count.samp)

# data("zeisel")
# zeisel_imputation_result = EnImpute(zeisel$count.samp)
```

---

manno *Human ventral midbrain data*

Description

This is the Human ventral midbrain dataset (GSE76381). The raw data contains 19,531 genes and 1,977 cells. Here we use the reference and downsampled datasets generated by Huang et al (2018) which contain 2,059 genes and 947 cells (available at https://github.com/mohuangx/SAVER-paper/tree/master/SAVER-data). For details about the approach to generate the reference and downsampled datasets, please refer to Huang et al (2018). This data is an object of class list of length two. count.ref is the reference count matrix and count.samp is the downsampled count matrix.

Usage

```
manno
```

Format

An object of class `list` of length 2.

Author(s)

Xiao-Fei zhang, <zhangxf@mail.ccnu.edu.cn>

References

La Manno, Gioele, et al (2016). Molecular diversity of midbrain development in mouse, human, and stem cells. Cell 167(2): 566-580.

Mo Huang et al (2018). SAVER: gene expression recovery for single-cell RNA sequencing. *Nature Methods*, 15:539-542.

Xiao-Fei Zhang et al (2018), EnImpute: An ensemble learning method for imputing single cell RNA sequencing data. submmited

Examples

```
data("manno")
```

---

zeisel *Mouse brain single-cell RNA-seq dataset*

Description

This is the mouse cortex and hippocampus dataset (http://linnarssonlab.org/cortex/). The raw data contains 19,972 genes and 3,005 cells. Here we use the reference and downsampled datasets generated by Huang et al (2018) which contain 3,529 genes and 1,799 cells (available at https://github.com/mohuangx/SAVER-paper/tree/master/SAVER-data). For details about the approach to generate the reference and downsampled datasets, please refer to Huang et al (2018). This data is an object of class list of length two. count.ref is the reference count matrix and count.samp is the downsampled count matrix.

Usage

```
zeisel
```

Format

An object of class `list` of length 2.

Author(s)

Xiao-Fei zhang, <zhangxf@mail.ccnu.edu.cn>

References

Amit Zeisel et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* 347(6226):1138-1142.

Mo Huang et al (2018). SAVER: gene expression recovery for single-cell RNA sequencing. *Nature Methods*, 15:539-542.

Xiao-Fei Zhang et al (2018), EnImpute: An ensemble learning method for imputing single cell RNA sequencing data. submmited

Examples

```
data("zeisel")
```

Index