

R documentation

of ‘DigitalDLSorter-class.Rd’

September 3, 2020

DigitalDLSorter-class

The DigitalDLSorter Class.

Description

The DigitalDLSorter object is the core of digitalDLSorter. This object stores the different intermediate data resulting from running pipeline from real single-cell data to the trained Deep Neural Network, including the data on which to carry out the process of deconvolution. Only it is used in the case of building new deconvolution models. For deconvoluting bulk samples using pre-trained models, see [deconvDigitalDLSorter](#) function.

Details

This object uses other classes to store the different type of data produced during the process:

- `SingleCellExperiment` class for single-cell RNA-seq data, using sparse matrix from the `Matrix` package (`dgCMatrix` class) to store the matrix of counts.
- `ZinbParams` class with the estimated parameters for the simulation of new single-cell profiles.
- `SummarizedExperiment` class for storing bulk RNA-seq data. In this case, it is possible to load all data in memory or the use of HDF5 files as back-end by `DelayedArray` and `HDF5Array` packages. See [generateBulkSamples](#) for details.
- [ProbMatrixCellTypes](#) class for the composition cell matrices built during the process. See `?ProbMatrixCellTypes` for details.
- [DigitalDLSorterDNN](#) class for storing the trained Deep Neural Network. This step is performed by `keras`. See [DigitalDLSorterDNN](#) for details.

Slots

`single.cell.real` Real single-cell data stored in a `SingleCellExperiment` object. The counts matrix is stored as a `dgCMatrix` object to optimize the amount of used memory.

`zinb.params` `ZinbParams` object with estimated parameters for the simulation of new single-cell expression profiles.

`single.cell.final` Final single-cell expression profiles used for simulating bulk RNA-seq profiles with known cell composition.

`prob.cell.types` `ProbMatrixCellTypes` class with the cell composition matrix built for the simulation of bulk RNA-seq profiles. The entries determine the proportion of single-cell types that will constitute the simulated bulk samples.

`bulk.sim` A list with two elements: train and test simulated bulk RNA-seq. This data are stored as a `SummarizedExperiment` object. We recommend the use of HDF5 file as a back-end due to the large amount of memory that they occupy.

`final.data` The final data that will be used for training and testing the Deep Neural Network. As in the previous slot, it is a list with two items, train and test. With respect to train counts matrix, it can be the train bulk RNA-seq samples, the train scRNA-seq samples or a combination of both. In the case of test counts matrix, RNA-seq data from bulk and single-cell will be combined. Moreover, data is scaled and shuffled for training.

`trained.model` `DigitalDLSorterDNN` object with the trained model, different metrics obtained during the training and evaluation metrics from the application of the model on test data. After executing `calculateEvalMetrics`, it is also possible to find the results of the model evaluation.

`deconv.data` Optional slot where is possible to store new bulk samples for its deconvolution. It is a list whose name is the name of the data provided. It is possible to store more than one dataset to make predictions. See `deconvDigitalDLSorterObj` for details.

`deconv.results` Slot where the results from the deconvolution process over `deconv.data` data are stored. It is a list whose name is the name of the data from which they come.

`project` Name of the project.

`version` Version of DigitalDLSorter this object was built under.

The package can be used in two ways: to build new models of deconvolution from scRNA-seq data or to deconvolute bulk RNA-seq samples using pre-trained models integrated into the package. If you want to build new models, see `loadRealSCProfiles` or `loadFinalSCProfiles` functions. If you want to use pre-trained models, see `deconvDigitalDLSorter` function.

Index

calculateEvalMetrics, [2](#)

deconv.data, [2](#)

deconvDigitalDLSorter, [1](#), [2](#)

deconvDigitalDLSorterObj, [2](#)

DigitalDLSorter
 (*DigitalDLSorter-class*), [1](#)

DigitalDLSorter-class, [1](#)

DigitalDLSorterDNN, [1](#), [2](#)

generateBulkSamples, [1](#)

loadFinalSCProfiles, [2](#)

loadRealSCProfiles, [2](#)

ProbMatrixCellTypes, [1](#)