SingleCellData

Release 1.0.0

Edwin Vans

Jun 27, 2020

Contents

1 SingleCell Class

1.1 SingleCellData

A Python package that contains a class for managing single-cell RNA-seq datasets. See https://edwinv87.github.io/singlecelldata/ for more information and documentation.

`class singlecelldata.SingleCell`(*dataset*, *data*, *celldata=None*, *genedata=None*)

A python class for managing single-cell RNA-seq datasets.

`dataset`

A string for the name of the dataset.

Type str

`data`

The main dataframe or assay for storing the gene expression counts. The shape of this dataframe is (d x n) where d is the number of genes (features) and n is the number of cells (samples).

Type Pandas Dataframe

`celldata`

The dataframe or assay used to store more data (metadata) about cells. The shape of this dataframe is (n x m), where m is number of columns representing different types of information about the cells, such as cell types etc.

Type Pandas Dataframe

`genedata`

The dataframe or assay used to store more data (metadata) about genes. The shape of this dataframe is (d x m), where m is number of columns representing different types of information about the genes such as gene names etc.

Type Pandas Dataframe

`dim`

Variable representing the dimensionality of the data assay. It is (d, n).

Type tuple

`addCellData`(*col_data*, *col_name*)

Adds a column in the celldata dataframe.

Parameters

- **col_data** (*List or Numpy array*) – The data to be added to the celldata dataframe. The size of the List or Numpy array should be equal to n.
- **col_name** (*str*) – The name of the data column.

addGeneData(*col_data*, *col_name*)

Adds a column in the genedata dataframe.

Parameters

- **col_data** (*List or Numpy array*) – The data to be added to the genedata dataframe. The size of the List or Numpy array should be equal to d.
- **col_name** (*str*) – The name of the data column.

checkCellData(*column*)

Checks whether a column exists in the celldata dataframe.

Parameters **column** (*str*) – The name of the column.

Returns True if column exists in the dataframe, False otherwise.

Return type bool

checkGeneData(*column*)

Checks whether a column exists in the genedata dataframe.

Parameters **column** (*str*) – The name of the column.

Returns True if column exists in the dataframe, False otherwise.

Return type bool

getCellData(*column*)

Returns data stored in the celldata dataframe.

Parameters **column** (*str*) – The name of the data column.

Returns A n-dimensional array containing cell data by the column name.

Return type Numpy array

Raises **ValueError** – If column does not exist in the celldata dataframe.

getCounts()

Returns a Numpy array of the counts/data in the data dataframe. This method is called from the class instance and requires no input arguments.

Returns A (d x n) array of gene expression counts/data.

Return type Numpy array

getDistinctCellTypes(*column*)

Returns the unique cell type information stored in the celldata dataframe.

Parameters **column** (*str*) – This parameter is the column name of the cell labels in the celldata assay.

Returns Containing unique values in the celldata dataframe under the column passed into this function.

Return type Numpy array

getGeneData(*column*)

Returns data stored in the genedata dataframe.

Parameters **column** (*str*) – The name of the data column.

Returns A d-dimensional array containing gene data by the column name.

Return type Numpy array

Raises **`ValueError`** – If column does not exist in the genedata dataframe.

`getNumericCellLabels``(`*column*`)`

Returns the numeric (int) cell labels from the celldata assay which contains the string or int cell labels. This method is useful when computing Rand Index or Adjusted Rand Index after clustering.

Parameters **`column`** (*`str`*) – This parameter is the column name of the string or int cell labels in the celldata assay.

Returns Array containing the integer representation of data in the celldata dataframe under the column passed into this function.

Return type Numpy array (int)

`isSpike``(`*spike_type*, *gene_names_column*`)`

Prints a message if spike-ins are detected in the dataset. Creates a filter to remove spike-ins from the dataset when counts/data is returned using getCounts() method.

Parameters **`spike_type`** (*`str`*) – A string representing the type of spike-in.

`print``()`

Prints a summary of the single-cell dataset.

`removeCellData``(`*column*`)`

Removes a column from the celldata dataframe. First checks whether the column exists in the celldata dataframe.

Parameters **`column`** (*`str`*) – The name of the data column.

`removeGeneData``(`*column*`)`

Removes a column from the genedata dataframe. First checks whether the column exists in the genedata dataframe.

Parameters **`column`** (*`str`*) – The name of the data column.

`setCounts``(`*new_counts*`)`

Sets the new counts values in the data dataframe.

Parameters **`new_counts`** (*`Numpy array`*) – A numpy array with the shape = dim, representing new count values. The data dataframe will be updated with the new count values.

Python Module Index