R documentation

of 'generateTrainAndTestBulkProbMatrix.Rd'

September 3, 2020

`generateTrainAndTestBulkProbMatrix`
Generate training and test cell composition matrix.

Description

Generate training and test cell composition matrices for the simulation of bulk samples with known cell composition using single-cell expression profiles. The resulting matrix will determine the proportion of the different cell types that will form the simulated bulk samples.

Usage

```
generateTrainAndTestBulkProbMatrix(
  object,
  cell.type.column,
  prob.design,
  proportions.train = c(10, 5, 20, 15, 10, 40),
  proportions.test = c(10, 5, 20, 15, 10, 40),
  train.freq = 2/3,
  n.cells = 100,
  num.bulk.samples = NULL,
  exclusive.types = NULL,
  verbose = TRUE
)
```

Arguments

`object`	`DigitalDLSorter` object with `single.cell.real` and `zinb.params` slots.
`cell.type.column`	Name or number of the column in cells metadata corresponding with the cell type of each cell.
`prob.design`	`data.frame` with the frequency ranges expected for each cell type present in the experiment. This information can be estimated from literature or from the single-cell experiment itself. This `data.frame` must be built by three columns with specific headers:

- A cell type column with the same name of the cell type column in cells.metadata. If the name of the column is not the same, function returns an error. Cell types must appear on cells.metadata.
- A second column named 'from' with the start frequency for each cell type.
- A third column named 'to' with the final frequency for each cell type.

`proportions.train`
: Vector of five integer numbers that determine the proportions of bulk samples that will be generated by the methods explained in details in train samples. This vector represents proportions, so they must add 100 and none can be less than 1. By default, a majority of random samples without using predefined ranges will be generated.

`proportions.test`
: As `proportions.train` for test samples.

`train.freq`
: Proportion of cells used for training set (2/3 by default).

`n.cells`
: Number of cells that are aggregated in order to simulate one bulk RNA-seq sample (100 by default).

`num.bulk.samples`
: Integer which allows to establish the number of bulk samples that will be generated taking into account training and test data. If it is NULL (by default), approximately 18 more samples will be formed than there are cells in `single.cell.final` slot.

`exclusive.types`
: Vector of cell types which allows to establish cell types that biologically do not make sense to be mixed during the generation of bulk samples. Some samples presents this exclusive cell types. If it is equal to NULL (by default), all cell types will be mixed when generating bulk samples.

`verbose`
: Show informative messages during the execution.

Details

First of all, simulated single-cell profiles are split into training and test subsets (2/3 for training and 1/3 for test by default). Then, to avoid biases due to the composition of bulk samples, proportions for the mixtures (bulk samples) of cell types ($w_1, ..., w_k$, where k is the number of cell types available in single-cell profiles), are randomly generated using five different approaches:

1. Cell proportions are randomly sampled from a truncated uniform distribution with predefined limits according to a priori knowledge of the abundance of each cell type (see `prob.design` argument). This information ban be inferred from the single cell analysis itself or from the literature.
2. A second set is generated by randomly permuting cell type labels from a distribution generated by the previous method.
3. Cell proportions are randomly sampled as by method 1 without replacement.
4. Using the last method for generating proportions, cell types labels are randomly sampled.
5. Cell proportions are randomly sampled from a Dirichlet distribution.

If you want to see the distribution of cell type proportions generated by each method during the process, you can access them with `showProbPlot` function (see examples).

It is important to note that the number of bulk-samples simulated are determined in this step. You can predefine the number of bulk profiles generated using `num.bulk.samples` argument. By default, the number of bulk samples generated depends on the number of single-cell profiles available:

approximately 18 more samples will be formed than there are cells in `single.cell.final`. We recommend set a number of 30000 samples. This number will be split in training and test data: 60% for training and 40% for evaluating the model.

Value

A `DigitalDLSorter` object with `prob.cell.types` slot containing a `ProbMatrixCellTypes` object. For more information about the structure of this class, see `ProbMatrixCellTypes`. The most important element is the cell composition matrix, which is formed by n rows (being n the number of bulk samples that will be generated) and k columns (being k the number of cell types present in the experiment).

References

Torroja, C. y Sánchez-Cabo, F. (2019). digitalDLSorter: A Deep Learning algorithm to quantify immune cell populations based on scRNA-Seq data. Frontiers in Genetics 10, 978. doi: 10.3389/fgene.2019.00978

See Also

`generateBulkSamples`, `ProbMatrixCellTypes`.

Examples

```
## generate a data.frame with frequency ranges of each cell type
probMatrix <- data.frame(
  Cell_types = c("ER+", "HER2+", "ER+ and HER2+", "TNBC",
                 "Stromal", "Monocyte", "Tme", "BGC",
                 "Bmem", "DC", "Macrophage", "TCD8", "Treg"),
  from = c(rep(30, 4), 1, rep(0, 8)),
  to = c(rep(70, 4), 50, rep(15, 8))
)
## Not run:
## Without a predefined number of bulk samples
DDLSChung <- generateTrainAndTestBulkProbMatrix(
  object = DDLSChung,
  cell.type.column = "Cell_type",
  prob.design = probMatrix,
  verbose = TRUE
)

## End(Not run)
## Not run:
## With a predefined number of bulk samples
DDLSChung <- generateTrainAndTestBulkProbMatrix(
  object = DDLSChung,
  cell.type.column = "Cell_type",
  prob.design = probMatrix,
  num.bulk.samples = 1000,
  verbose = TRUE
)

## End(Not run)
```

Index