Package ‘cellity’

January 15, 2026

Type Package

Title Quality Control for Single-Cell RNA-seq Data

Version 1.39.0

Date 2016-02-22

Author Tomislav Illicic, Davis McCarthy

Maintainer Tomislav Ilicic <ti243@cam.ac.uk>

Description A support vector machine approach to identifying and filtering low quality cells from single-cell RNA-seq datasets.

License GPL (>= 2)

Depends R (>= 3.3)

Imports AnnotationDbi, e1071, ggplot2, graphics, grDevices, grid, mvoutlier, org.Hs.eg.db, org.Mm.eg.db, robustbase, stats, topGO, utils

Suggests BiocStyle, caret, knitr, testthat, rmarkdown

VignetteBuilder knitr

LazyData true

biocViews ImmunoOncology, RNASeq, QualityControl, Preprocessing, Normalization, Visualization, DimensionReduction, Transcriptomics, GeneExpression, Sequencing, Software, SupportVectorMachine

RoxygenNote 5.0.1

git_url https://git.bioconductor.org/packages/cellity

git_branch devel

git_last_commit 6d83db0

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2026-01-15

Contents

cellity-package 2
assess_cell_quality_PCA 2
assess_cell_quality_SVM 3
extract_features 4
extra_human_genes 5
extra_mouse_genes 6
feature_generation 6
feature_info 7
mES1_features 7
mES1_labels 8
multiplot 9
normalise_by_factor 9
param_mES_all 10
param_mES_common 10
plot_pca 11
sample_counts 12
sample_stats 12
simple_cap 13
sum_prop 13
training_mES_features 14
training_mES_labels 14
uni.plot 15

Index **16**

`cellity-package` *Quality Control for Single-Cell RNA-seq Data*

Description

cellity provides a support vector machine and PCA approaches to identifying and filtering low quality cells from single-cell RNA-seq datasets.

`assess_cell_quality_PCA` *ASSESS CELL QUALITY USING PCA AND OUTLIER DETECTION*

Description

ASSESS CELL QUALITY USING PCA AND OUTLIER DETECTION

Usage

```
assess_cell_quality_PCA(features, file = "")
```

Arguments

features	Input dataset containing features (cell x features)
file	Output_file where plot is saved

Details

This function applies PCA on features and uses outlier detection to determine which cells are low and which are high quality

Value

Returns a dataframe indicating which cell is low or high quality (0 or 1 respectively)

Examples

```
data(training_mES_features)
training_mES_features_all <- training_mES_features[[1]]
training_quality_PCA_allF <- assess_cell_quality_PCA(training_mES_features_all)
```

---

assess_cell_quality_SVM

Assess quality of a cell - SVM version

Description

Assess quality of a cell - SVM version

Usage

```
assess_cell_quality_SVM(training_set_features, training_set_labels,
  ensemble_param, test_set_features)
```

Arguments

training_set_features	A training set containing features (cells x features) for prediction
training_set_labels	Annotation of each individual cell if high or low quality (1 or 0 respectively)
ensemble_param	Dataframe of parameters for SVM
test_set_features	Dataset to predict containing features (cells x features)

Details

This function takes a traning set + annotation to predict a test set. It requires that hyper-parameters have been optimised.

Value

Returns a dataframe indicating which cell is low or high quality (0 or 1 respectively)

data.frame with decision on quality of cells

Examples

```
data(param_mES_all)
data(training_mES_features)
data(training_mES_labels)
data(mES1_features)
data(mES1_labels)
mES1_features_all <- mES1_features[[1]]
training_mES_features_all <- training_mES_features[[1]]
mES1_quality_SVM <- assess_cell_quality_SVM( training_mES_features_all,
training_mES_labels[,2], param_mES_all, mES1_features_all)
```

extract_features *Extracts biological and technical features for given dataset*

Description

Extracts biological and technical features for given dataset

Usage

```
extract_features(counts_nm, read_metrics, prefix = "", output_dir = "",
  common_features = NULL, GO_terms = NULL, extra_genes = NULL,
  organism = "mouse")
```

Arguments

`counts_nm` Gene expression counts dataframe (genes x cells). Either normalised by library size or TPM values

`read_metrics` Dataframe with mapping statistics produced by python pipeline

`prefix` Prefix of outputfiles

`output_dir` Output directory of files

`common_features` Subset of features that are applicable within one species, but across cell types

`GO_terms` DataFrame with gene ontology term IDs, that will be used in feature extraction

`extra_genes` Additional genes used for feature extraction

`organism` The target organism to generate the features for

Details

This function takes a combination of gene counts and mapping statistics to extract biological and technical features, which than can be used for quality data analysis

Value

a list with two elements, one providing all features, and one providing common features.

Examples

```
data(sample_counts)
data(sample_stats)
sample_counts_nm <- normalise_by_factor(sample_counts, colSums(sample_counts))
sample_features <- extract_features(sample_counts_nm, sample_stats)
```

`extra_human_genes` *Additional human genes that are used in feature extraction*

Description

This list contains human genes that are used for feature extraction of biological features

Usage

```
extra_human_genes
```

Format

a list containing vectors of genes. Name indicates which GO category.

Value

NULL, but makes available a list with metadata

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

`extra_mouse_genes` *Additional mouse genes that are used in feature extraction*

Description

This list contains mouse genes that are used for feature extraction of biological features

Usage

```
extra_mouse_genes
```

Format

a list containing vectors of genes. Name indicates which GO category.

Value

NULL, but makes available a list with metadata

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

`feature_generation` *Helper Function to create all features*

Description

Helper Function to create all features

Usage

```
feature_generation(counts_nm, read_metrics, GO_terms, extra_genes, organism)
```

Arguments

`counts_nm`	Gene expression counts dataframe (genes x cells). Either normalised by library size or TPM values
`read_metrics`	Dataframe with mapping statistics produced by python pipeline
`GO_terms`	DataFrame with gene ontology term IDs, that will be used in feature extraction
`extra_genes`	Additional genes used for feature extraction
`organism`	The target organism to generate the features for

Value

Returns the entire set of features in a data.frame

`feature_info` *Information which genes and GO categories should be included as features. Also defines which features are cell-type independent (common features)*

Description

This list contains metadata information that is used to extract features from in the function extract_features

Usage

```
feature_info
```

Format

a list with 2 elements (GO_terms,common_features).

Value

NULL, but makes available a list with metadata

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

`mES1_features` *Real test dataset containing all and common features from the paper (mES1)*

Description

This list contains 2 dataframes where each contains features per cell (cell X features) that can be used for classification.

Usage

```
mES1_features
```

Format

a list with 2 elements (all_features, common_features).

Value

NULL, but makes available a list with 2 dataframes

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

`mES1_labels` *Real test dataset containing annotation of cells*

Description

This data frame has 2 columns: First showing cell names, the second indicating if cell is of low (0) or high (1) quality

Usage

```
mES1_labels
```

Format

a dataframe with 2 columns (cell_names, label).

Value

NULL, but makes available a dataframe with cell annotations

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

`multiplot` *Internal multiplot function to combine plots onto a grid*

Description

Internal multiplot function to combine plots onto a grid

Usage

```
multiplot(..., plotlist = NULL, file, cols = 6, layout = NULL)
```

Arguments

`...`	individual plots to combine into a single plot
`plotlist`	a vector with names of plots to use in the plot
`file`	string giving filename to which pdf of plots is to be saved
`cols`	integer giving number of columns for the plot
`layout`	matrix defining the layout for the plots

Value

a plot object

`normalise_by_factor` *Internal function to normalize by library size*

Description

Internal function to normalize by library size

Usage

```
normalise_by_factor(counts, norm_factor)
```

Arguments

`counts`	matrix of counts
`norm_factor`	vector of normalisation factors

Value

a matrix with normalized gene counts

Examples

```
data(sample_counts)
data(sample_stats)
sample_counts_nm <- normalise_by_factor(sample_counts, colSums(sample_counts))
```

param_mES_all *Parameters used for SVM classification*

Description

This data frame has 3 columns: gamma, cost, class.weights and is optimised for all features and our training data

Usage

```
param_mES_all
```

Format

a dataframe with 3 columns (gamma, cost, class.weights).

Value

NULL, but makes available a dataframe with parameters

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

param_mES_common *Parameters used for SVM classification*

Description

This data frame has 3 columns: gamma, cost, class.weights and is optimised for common features and our training data

Usage

```
param_mES_common
```

Format

a dataframe with 3 columns (gamma, cost, class.weights).

Value

NULL, but makes available a dataframe with parameters

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

`plot_pca` *Plots PCA of all features. Colors high and low quality cells based on outlier detection.*

Description

Plots PCA of all features. Colors high and low quality cells based on outlier detection.

Usage

```
plot_pca(features, annot, pca, col, output_file)
```

Arguments

`features`	Input dataset containing features (cell x features)
`annot`	Matrix annotation of each cell
`pca`	PCA of features
`col`	color code indicating what color high and what low quality cells
`output_file`	where plot is stored

Details

This function plots PCA of all features + most informative features

Value

Plots of PCA

`sample_counts` *Sample gene expression data containing 40 cells*

Description

This data frame contains genes (rows) and cells (columns) showing raw read counts

Usage

```
sample_counts
```

Format

a dataframe with genes x cells

Value

NULL, but makes available a dataframe with raw read counts

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

`sample_stats` *Sample read statistics data containing 40 cells*

Description

This data frame contains read metrics (columns) and cells (rows)

Usage

```
sample_stats
```

Format

a dataframe with cells x metrics

Value

NULL, but makes available a dataframe with read statistics

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

simple_cap *Converts all first letters to capital letters*

Description

Converts all first letters to capital letters

Usage

```
simple_cap(x)
```

Arguments

`x`	string

Value

a character vector in title case

sum_prop *Sums up normalised values of genes to groups.*

Description

Supports TPM and proportion of mapped reads.

Usage

```
sum_prop(counts, genes_interest)
```

Arguments

`counts`	Normalised gene expression count matrix
`genes_interest`	dataframe of genes of interest to merge

Value

a vector of sums per group

`training_mES_features` *Original training dataset containing all and common features from the paper (training mES)*

Description

This list contains 2 dataframes where each contains features per cell (cell X features) that can be used for classification.

Usage

```
training_mES_features
```

Format

a list with 2 elements (all_features, common_features).

Value

NULL, but makes available a list with 2 dataframes

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

`training_mES_labels` *Original training dataset containing annotation of cells*

Description

This data frame has 2 columns: First showing cell names, the second indicating if cell is of low (0) or high (1) quality

Usage

```
training_mES_labels
```

Format

a dataframe with 2 columns (cell_names, label).

Value

NULL, but makes available a dataframe with cell annotations

Author(s)

Tomislav Ilicic & Davis McCarthy, 2015-03-05

Source

Wellcome Trust Sanger Institute

`uni.plot` *Internal function to detect outliers from the mvoultier pacakge Modified slightly so that plots are not printed*

Description

Internal function to detect outliers from the mvoultier pacakge Modified slightly so that plots are not printed

Usage

```
uni.plot(x, symb = FALSE, quan = 1/2, alpha = 0.025)
```

Arguments

`x`	A matrix containing counts
`symb`	Symbols
`quan`	quan
`alpha`	alpha

Value

a list of outlier indicators

Index

assess_cell_quality_PCA, 2
assess_cell_quality_SVM, 3

cellity-package, 2

extra_human_genes, 5
extra_mouse_genes, 6
extract_features, 4

feature_generation, 6
feature_info, 7

mES1_features, 7
mES1_labels, 8
multiplot, 9

normalise_by_factor, 9

param_mES_all, 10
param_mES_common, 10
plot_pca, 11

sample_counts, 12
sample_stats, 12
simple_cap, 13
sum_prop, 13

training_mES_features, 14
training_mES_labels, 14

uni.plot, 15