Package 'M3Drop'

April 12, 2018

Version 1.4.0

Date 2016-07-15

Title Michaelis-Menten Modelling of Dropouts in single-cell RNASeq

Author Tallulah Andrews <tallulandrews@gmail.com>

Maintainer Tallulah Andrews <tallulandrews@gmail.com>

Depends R (>= 3.3), numDeriv

Imports RColorBrewer, gplots, bbmle, statmod, grDevices, graphics, stats

Suggests ROCR, knitr, M3DExampleData

VignetteBuilder knitr

biocViews RNASeq, Sequencing, Transcriptomics, GeneExpression, Software, DifferentialExpression, DimensionReduction, FeatureExtraction

Collate basics.R Plotting_fxns.R Curve_fitting.R Extremes.R Normalization.R Brennecke_implementation.R Threeway_ProportionalArea_VennDiagrams.R

Description This package fits a Michaelis-Menten model to the pattern of dropouts in single-cell RNASeq data. This model is used as a null to identify significantly variable (i.e. differentially expressed) genes for use in downstream analysis, such as clustering cells.

URL https://github.com/tallulandrews/M3Drop

BugReports https://github.com/tallulandrews/M3Drop/issues

License GPL (>=2)

NeedsCompilation no

R topics documented:

`BrenneckeGetVariableGenes` *Identify Highly Variable Genes*

Description

Implements the method of Brennecke et al. (2013) to identify highly variable genes.

Usage

```
BrenneckeGetVariableGenes(expr_mat, spikes=NA, suppress.plot=FALSE, fdr=0.1, minBiolDisp=0.5)
```

Arguments

`expr_mat`	a numeric matrix of normalized or raw (not log-transformed) expression values, columns = samples, rows = genes.
`spikes`	a vector of gene names of row numbers of spike-in genes which are subject to only technical variance.
`suppress.plot`	Whether to make the plot or just calculate the requisite values.
`fdr`	Use FDR to identify significantly highly variable genes.
`minBiolDisp`	Minimum percentage of variance due to biological factors.

Details

Identifies significantly highly variable genes as detailed in Brennecked et al [1]. If spike-ins are provided they are used fit a function to the relationship between gene expression and variance due to technical factors. If spike-ins are not provided then all genes are used in the fitting.

Value

Vector of names of highly variable genes.

References

Brennecke et al. (2013) Accounting for technical noise in single-cell RNA-seq experiments. Nature Methods 10, 1093-1095. doi:10.1038/nmeth.2645

Examples

```
library(M3DExampleData)
HVG <- BrenneckeGetVariableGenes(Mmus_example_list$data)
HVG_spike <- BrenneckeGetVariableGenes(Mmus_example_list$data, spikes=5550:5600)
```

`Fitting_Dropout_Models` *Fit functions to the dropouts vs expression distribution.*

Description

Fits the modified Michaelis-Menten equation (MM), a logistic regession (logistic), or a double exponential (ZIFA) function to the relationship between mean expression and dropout-rate (proportion of zero values).

Usage

```
bg__fit_MM(p, s)
bg__fit_logistic(p, s)
bg__fit_ZIFA(p, s)
```

Arguments

`p`	a vector of dropout rates for each gene.
`s`	a vector of mean expression values for each gene. Must be the same order & length as p.

Details

Fits one of different models to the relationship between dropout rate and mean expression. The three models are: `bg__fit_MM` : the Michaelis-Menten function

$$P = 1 - \frac{S}{S + K}$$

(see: [1]). Fit using `mle2` using normally distributed error. `bg__fit_logistic` : a logistic regression between P and log base 10 of S (used by [2]). Fit using `glm` (excludes genes where S == 0). `bg__fit_ZIFA` : a double exponential

$$P = e^{\lambda S^2}$$

(used by [3]). Fit using `lm` after log-transformation (genes were P == 0 are assigned a value of one tenth of the smallest P which is not 0).

Value

Named list including: K,fitted_err/B0,B1/lambda,fitted_err : the fitted parameters predictions : predicted values of p for each gene SSr/SAr : sum of squared/absolute residuals model : vector of string descriptors of the fit

References

[1] Keener, J.; Sneyd, J. (2008). Mathematical Physiology: I: Cellular Physiology (2 ed.). Springer. ISBN 978-0-387-75846-6 [2] Kharchenko, PV; Silberstein, L; Scadden, DT. (2014) Bayesian approach to single-cell differential expression analysis. Nature Methods. 11:740-742 [3] Pierson, E; Yau, C. (2015) ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biology. 16:241 doi:10.1186/s13059-015-0805-z

Examples

```
#  library(M3DExampleData)
#  gene_info = bg__calc_variables(Mmus_example_list$data)
#  MM_fit = bg__fit_MM(gene_info$p, gene_info$s)
#  logistic_fit = bg__fit_logistic(gene_info$p, gene_info$s)
#  ZIFA_fit = bg__fit_ZIFA(gene_info$p, gene_info$s)
```

`M3DropCleanData` *Filter Expression Data*

Description

Filters and normalizes a given expression matrix. Removes low quality cells and undetected genes, and normalizes counts to counts per million.

Usage

```
M3DropCleanData(expr_mat, labels = NA, is.counts = TRUE, suppress.plot = FALSE, pseudo_genes = NA
```

Arguments

`expr_mat`	a numeric matrix of raw or normalized (not log-transformed) expression values, columns = samples/cells, rows = genes.
`labels`	a vector of length equal to the number of columns of expr_mat with names or group IDs for each cell.
`is.counts`	logical, whether the provided data is unnormalized read/fragment counts.
`suppress.plot`	logical, whether to plot the distribution of number of detected genes per cell.
`pseudo_genes`	a vector of gene names of known pseudogenes which will be removed from the cleaned data.
`min_detected_genes`	minimum number of genes/cell for a cell to be included in the cleaned data.

Details

Retains genes detected (expression>0) in more than 3 cells and with mean normalized expression >= 10^-5. If min_detected_genes is defined all cells not reaching the threshold are removed. Otherwise, fits a normal distribution to the distribution of detected genes/cell and removes those cells with significantly few detected genes (FDR 5%). This fit is plotted for visual inspection. If is.counts==TRUE then each column is converted to counts per million (ignoring ERCC spike-ins if present).

Value

list with elements: data, the normalized filtered expression matrix; and labels, labels of the remaining cells.

Examples

```
  library(M3DExampleData)
  # Remove all cells with < 2000 detected genes and convert to cpm
  cpm <- M3DropCleanData(Mmus_example_list$data, Mmus_example_list$labels,
is.counts=TRUE, min_detected_genes=2000)
  # Removes cells with significantly few detected genes (FDR=5%)
  filtered_only <- M3DropCleanData(Mmus_example_list$data, Mmus_example_list$labels,
is.counts=FALSE)
```

M3DropDifferentialExpression

Differentially Expressed Genes.

Description

Use Michaelis-Menten curve to find differentially expressed (DE) genes.

Usage

```
M3DropDifferentialExpression(expr_mat, mt_method="bon", mt_threshold=0.05, suppress.plot=FALSE)
```

Arguments

`expr_mat`	a numeric matrix of normalized (not log-transformed) expression values, columns = samples, rows = genes.
`mt_method`	the multiple testing method used in p.adjust
`mt_threshold`	the threshold for identifying significantly DE genes.
`suppress.plot`	logical, whether to plot the fitted curve and highlight DE genes.

Details

Fits a Michaelis-Menten function to the dropout-rate (if not provided) of the provided expression matrix. Identifies genes where the gene-specific K calculated as (S = mean expression, P = dropout rate):

$$K = \frac{S * P}{1 - P}$$

is significantly larger than the K fitted to the entire dataset. Combines standard errors of the fitted K, the gene-specific dropout rate and the gene-specific average expression using error propagation rules. Determines the signficance of the gene-specific K using a Z-test of the log-transformed Ks with the propagated error then applies the specified multiple testing correction to identify DE genes. Plots the dropout rate vs gene expression with the fitted MM curve and highlights in purple the significantly DE genes.

Value

`M3Drop_Differential Expression` : a data.frame of significantly differentially expressed genes with columns: Gene, p.value, q.value

Examples

```
library(M3DExampleData)
Normalized_data <- M3DropCleanData(Mmus_example_list$data,
                labels = Mmus_example_list$labels,
                is.counts=TRUE, min_detected_genes=2000)
DE_genes <- M3DropDifferentialExpression(Normalized_data$data,
mt_method="fdr", mt_threshold=0.01)
```

`M3DropDropoutModels` *Fit functions to the dropouts vs expression distribution.*

Description

Fits the modified Michaelis-Menten equation (MM), a logistic regession (logistic), or a double exponential (ZIFA) function to the relationship between mean expression and dropout-rate (proportion of zero values).

Usage

```
M3DropDropoutModels(expr_mat, xlim=NA, suppress.plot=FALSE)
```

Arguments

`expr_mat`	a numeric matrix of normalized (not log-transformed) expression values, columns = samples, rows = genes.
`xlim`	limits for x-axis of plot.
`suppress.plot`	logical, whether to plot fit curves or not.

Details

Plots the dropout-rate (P) vs average gene expression (S) for all genes. Fits three different models and adds the fitted curves to the plot. The three models are: MMfit : the Michaelis-Menten function

$$P = 1 - \frac{S}{S + K}$$

(see: [1]). LogiFit : a logistic regression between P and log base 10 of S (used by [2]). ExpoFit : a double exponential

$$P = e^{\lambda S^2}$$

(used by [3]).

Value

Invisibly, a list of output from each fit (MMfit, LogiFit, ExpoFit).

References

[1] Keener, J.; Sneyd, J. (2008). Mathematical Physiology: I: Cellular Physiology (2 ed.). Springer. ISBN 978-0-387-75846-6 [2] Kharchenko, PV; Silberstein, L; Scadden, DT. (2014) Bayesian approach to single-cell differential expression analysis. Nature Methods. 11:740-742 [3] Pierson, E; Yau, C. (2015) ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biology. 16:241 doi:10.1186/s13059-015-0805-z

Examples

```
library(M3DExampleData)
M3DropDropoutModels(Mmus_example_list$data)
```

`M3DropExpressionHeatmap`

Plot Heatmap of Gene Expression

Description

Plots a customized heatmap of scaled log expression values.

Usage

```
M3DropExpressionHeatmap(genes, expr_mat, cell_labels=NA, interesting_genes=NA, key_genes=genes,
```

Arguments

`genes`	a character vector of gene names to be plot.
`expr_mat`	a numeric matrix of normalized (not log-transformed) expression values, columns = samples, rows = genes.
`cell_labels`	factor of labels for each cell in the expression matrix that will be used to coloured in a top bar of the heatmap.
`interesting_genes`	list of vectors of gene names that will be used to colour the bar to the left of the heatmap.
`key_genes`	a character vector of gene names to be labelled on the heatmap.
`key_cells`	a character vector of cells to be labelled on the heatmap. Unlabelled cells will be assigned a numerical index

Details

Modifies the gplots function heatmap.2 to replace the row dendrogram with a legend of the colours used in the columns colour bar (cell_labels) and use a custom colour scalling. Expression is displayed as Z-scores of log transformed expression (adding a pseudocount of 1) coloured blue-white-red centered at 0 and binned in the range [-2,2].

Value

Invisibly, output from heatmap.2 call.

Examples

```
library(M3DExampleData)
M3DropExpressionHeatmap(head(rownames(Mmus_example_list$data),20),Mmus_example_list$data, cell_labels = Mm
```

M3DropGetExtremes *Get outliers from MM curve.*

Description

Identifies outliers left and right of a fitted Michaelis-Menten curve.

Usage

```
M3DropGetExtremes(expr_mat, fdr_threshold=0.1, percent=NA, v_threshold=c(0.05,0.95), suppress.p
```

Arguments

expr_mat	a numeric matrix of normalized (not log-transformed) expression values, columns = samples, rows = genes.
fdr_threshold	the threshold for identifying significant outliers after multiple testing correction.
percent	identify this percentage of data that is most extreme in each direction.
v_threshold	restrict to this range of dropout rates to avoid poorly fit regions of the data.
suppress.plot	logical, whether to plot the fitted Michaelis-Menten curve and highlight to identified most extreme outliers.

Details

Fits a Michaelis-Menten function to the dropout-rate of the provided data, then identifies the most extreme left and/or right outliers from the curve. Horizontal residuals are calculated as :

$$\log_{10} S - \log_{10} \frac{K * (1 - P)}{P}$$

. Extreme left[right] outliers are identified either as the `percent` smallest[largest] horizontal residuals. If `percent` is undefined (default) a normal distribution is fitted to the horizontal residuals and a Z-test is used to identify significant outliers after FDR multiple testing correction.

Only genes with dropout rates within v_threshold will be considered to avoid the skewing of residuals due to the exponential parts of the MM curve near P = 0 & P = 1.

`M3DropGetExtremes` identifies both left and right residuals using the provided thresholds in each direction. Eg. will return the `percent` smallest and `percent` largest residuals. It also plots the fitted MM curve and highlights the left and right extreme outliers unless suppress.plot=TRUE .

Value

`M3DropGetExtremes` List containing elements left and right, vectors of the names of the extreme genes to the left and right of the curve respectively.

Examples

```
library(M3DExampleData)
extreme_gene_lists <- M3DropGetExtremes(Mmus_example_list$data, fdr_threshold=0.1)
extreme_gene_lists <- M3DropGetExtremes(Mmus_example_list$data, percent=0.01)
```

M3DropGetHeatmapCellClusters

Extracts cell clusters from heatmap output

Description

Extracts the clustering correponding to the given number of clusters from heatmap output.

Usage

```
M3DropGetHeatmapCellClusters(heatout, k)
```

Arguments

heatout	Output from a gene-expression heatmap.
k	Number of clusters.

Details

Traverses down the dendrogram and cuts at the first point where there are at least k clusters.

Value

A vector of cluster labels for each cell.

Examples

```
library(M3DExampleData)
genes <- rownames(Mmus_example_list$data)[1:20]
heatmap_out <- M3DropExpressionHeatmap(genes, Mmus_example_list$data)
clusters <- M3DropGetHeatmapCellClusters(heatmap_out, k=5)
```

M3DropGetMarkers *Identify marker genes*

Description

Calculates area under the ROC curve for each gene to predict the best group of cells from all other cells.

Usage

```
M3DropGetMarkers(expr_mat, labels)
```

Arguments

expr_mat	a numeric matrix of normalized expression values, columns = samples, rows = genes.
labels	a vector of group ids for each cell/sample.

Details

Uses the ROCR package to calculate the AUC for each gene for the group with the highest average rank. Significant is calculated using a Wilcox rank-sum test.

Value

A dataframe with a row for each gene and columns: AUC, Group (which label this gene had the highest average rank for), and pval (uncorrected p-value of prediction).

Examples

```
library(M3DExampleData)
marker_gene_table <- M3DropGetMarkers(Mmus_example_list$data, Mmus_example_list$labels)
```

`M3DropTestShift` *Test for horizontal shift.*

Description

Tests whether a given set of genes are significantly shifted to the left or right of the Michaelis-Menten curve.

Usage

```
M3DropTestShift(expr_mat, genes_to_test, name="", background=rownames(expr_mat), suppress.plot=
```

Arguments

`expr_mat`	a numeric matrix of normalized (not log-transformed) expression values, columns = samples, rows = genes.
`genes_to_test`	vector of gene names to test.
`name`	string used to title the plot.
`background`	vector of gene names to test against. (default = all genes)
`suppress.plot`	logical, whether to the fitted Michaelis-Menten curve and highlight the given set of genes to test.

Details

Fits a Michaelis-Menten function to the dropout-rate of the provided data, then tests whether a given set of genes (eg. pseudogenes) is significantly shifted left or right of the curve. Horizontal residuals are calculated as :

$$\log_{10} S - \log_{10} \frac{K * (1 - P)}{P}$$

. Uses a Wilcox rank-sum test/Mann-Whitney U test to compare the residuals for the given genes to the residuals for all genes.

Value

A one row dataframe with columns: sample (median horizontal residual of genes in the test set), pop (median horizontal residual of genes in the background set), p.value

Examples

```
library(M3DExampleData)
gene_set <- c("Dppa2","Tdgf1","Rnf130","Tet1","Uhrf1","Pttg1","Zfp600","Stat1")
shift_output <- M3DropTestShift(Mmus_example_list$data, gene_set)
```

M3DropThreeSetVenn *Three-way Venn Diagram*

Description

Plot an area-proportional three-set Venn Diagram with labels.

Usage

```
M3DropThreeSetVenn(set1, set2, set3, names)
```

Arguments

set1	a vector of items in the first set.
set2	a vector of items in the second set.
set3	a vector of items in the third set.
names	a vector of names of each set

Details

Approximates area-proportional three-set Venn Diagram with code by David J. States (available at: http://tolstoy.newcastle.edu.au/R/help/03a/1115.html). Then places labels within each circle and overlap-region using code by Tallulah Andrews.

Value

None

Examples

```
SetA <- c(1:20)
SetB <- c(15:30)
SetC <- c(5,10,15,20,25,30,35,40,45,50,55,60)
M3DropThreeSetVenn(SetA, SetB, SetC, names=c("A","B","C"))
```

Index