Package 'Linnorm'

November 12, 2026

Type Package

Title Linear model and normality based normalization and transformation method (Linnorm)

Version 2.36.0

Date 2023-10-12

Author Shun Hang Yip <shunyip@bu.edu>

Maintainer Shun Hang Yip <shunyip@bu.edu>

Description Linnorm is an algorithm for normalizing and transforming RNA-seq, single cell RNA-seq, ChIP-seq count data or any large scale count data. It has been independently reviewed by Tian et al. on Nature Methods (https://doi.org/10.1038/s41592-019-0425-8). Linnorm can work with raw count, CPM, RPKM, FPKM and TPM.

Depends R(>= 4.1.0)

License MIT + file LICENSE

Imports Rcpp (>= 0.12.2), RcppArmadillo (>= 0.8.100.1.0), fpc, vegan, mclust, apcluster, ggplot2, ellipse, limma, utils, statmod, MASS, igraph, grDevices, graphics, fastcluster, ggdendro, zoo, stats, amap, Rtsne, gmodels

LinkingTo Rcpp, RcppArmadillo

Suggests BiocStyle, knitr, rmarkdown, markdown, gplots, RColorBrewer, moments, testthat, matrixStats

VignetteBuilder knitr

biocViews ImmunoOncology, Sequencing, ChIPSeq, RNASeq, DifferentialExpression, GeneExpression, Genetics, Normalization, Software, Transcription, BatchEffect, PeakDetection, Clustering, Network, SingleCell

NeedsCompilation yes

LazyData false

URL https://doi.org/10.1093/nar/gkx828

RoxygenNote 7.2.3

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

git_branch RELEASE_3_23

git_last_commit 09293ac

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

Islam2011 . . . 2
LIHC . . . 3
LinearRegression . . . 3
LinearRegressionFP . . . 4
Linnorm . . . 5
Linnorm.Cor . . . 7
Linnorm.DataImput . . . 9
Linnorm.HClust . . . 11
Linnorm.HVar . . . 13
Linnorm.limma . . . 15
Linnorm.Norm . . . 16
Linnorm.PCA . . . 18
Linnorm.SGenes . . . 20
Linnorm.tSNE . . . 21
RnaXSim . . . 23
SEQC . . . 25

Index **26**

`Islam2011` *scRNA-seq data from Islam et al. 2011*

Description

GEO accession GSE29087: 92 single cells (48 mouse embryonic stem cells, 44 mouse embryonic fibroblasts and 4 negative controls) were analyzed by single-cell tagged reverse transcription (STRT).

Usage

```
data(Islam2011)
```

Format

A matrix with 22936 rows (genes) and 96 columns (samples). The first 48 columns are ES cells, the following 44 columns are mouse embryonic fibroblasts and the remaining 4 columns and negative controls. Data is in raw counts format.

References

Islam et al. (2011) Genome Res 2011 Jul;21(7):1160-7. PMID: 21543516

`LIHC` *Partial RNA-seq data from TCGA LIHC (Liver Hepatocellular Carcinoma)*

Description

TPM Expression data

Usage

```
data(LIHC)
```

Format

A matrix with 25914 rows (genes) and 10 columns (samples). The first 5 columns are Tumor samples, the remaining 5 columns are adjacent Normal samples. They are paired samples from 5 individuals. Data is in TPM format.

References

https://tcga-data.nci.nih.gov/

`LinearRegression` *One Pass Linear Regression.*

Description

This function performs Linear Regression on the input data with a one pass algorithm implemented in C++. This is for users who only need m and c from the y=mx + c equation. Compared to the lm function, this function is much faster.

Usage

```
LinearRegression(x, y)
```

Arguments

`x`	Numeric vector. x values.
`y`	Numeric vector. corresponding y values.

Details

This function calculates m and c in the linear equestion, y = mx + c.

Value

This function returns a list with one object, "coefficients". The first element in this object is c; the second element is m in the y = mx + c equation.

Examples

```
x <- 1:10
y <- 1:10
results <- LinearRegression(x,y)
m <- results$coefficients[[2]]
c <- results$coefficients[[1]]
```

LinearRegressionFP *One Pass Linear Regression with fixed point.*

Description

This function performs Linear Regression on the input data with a fixed point. It uses a one pass algorithm implemented in C++. This is for users who only need m and c from the y=mx + c equation. Compared to the lm function, this function is much faster.

Usage

```
LinearRegressionFP(x, y, x1, y1)
```

Arguments

x	Numeric vector. x values.
y	Numeric vector. corresponding y values.
x1	Numeric. x coordinate of the fixed point.
y1	Numeric. y coordinate of the fixed point.

Details

This function calculates m and c in the linear equestion, y = mx + c.

Value

This function returns a list with one object, "coefficients". The first element in this object is c; the second element is m in the y = mx + c equation.

Examples

```
x <- 1:10
y <- 1:10
x1 <- 1
y1 <- 2
results <- LinearRegressionFP(x,y, x1, y1)
m <- results$coefficients[[2]]
c <- results$coefficients[[1]]
```

Linnorm *Linnorm Normalizing Transformation Function*

Description

This function performs the Linear model and normality based transformation method (Linnorm) for (sc)RNA-seq expression data or large scale count data.

Usage

```
Linnorm(
  datamatrix,
  RowSamples = FALSE,
  spikein = NULL,
  spikein_log2FC = NULL,
  showinfo = FALSE,
  perturbation = "Auto",
  Filter = FALSE,
  minNonZeroPortion = 0.75,
  L_F_p = 0.3173,
  L_F_LC_Genes = "Auto",
  L_F_HC_Genes = 0.01,
  BE_F_p = 0.3173,
  BE_F_LC_Genes = "Auto",
  BE_F_HC_Genes = 0.01,
  BE_strength = 0.5,
  max_F_LC = 0.75,
  DataImputation = FALSE,
  ...
)
```

Arguments

datamatrix	The matrix or data frame that contains your dataset. Raw Counts, CPM, RPKM, FPKM or TPM are supported. Undefined values such as NA are not supported. It is not compatible with log transformed datasets.

RowSamples	Logical. In the datamatrix, if each row is a sample and each column is a feature, set this to TRUE so that you don't need to transpose it. Linnorm works slightly faster with this argument set to FALSE, but it should be negligable for smaller datasets. Defaults to FALSE.
spikein	character vector. Names of the spike-in genes in the datamatrix. Defaults to NULL.
spikein_log2FC	Numeric vector. Log 2 fold change of the spike-in genes. Defaults to NULL.
showinfo	Logical. Show algorithm running information. Defaults to FALSE.
perturbation	Integer >=2 or "Auto". To search for an optimal minimal deviation parameter (please see the article), Linnorm uses the iterated local search algorithm which perturbs away from the initial local minimum. The range of the area searched in each perturbation is exponentially increased as the area get further away from the initial local minimum, which is determined by their index. This range is calculated by 10 * (perturbation ^ index). Defaults to "Auto".
Filter	Logical. Should Linnorm filter the dataset in the end results? Defaults to FALSE.
minNonZeroPortion	Double >= 0.01, <= 0.95. Minimum non-Zero Portion Threshold. Genes not satisfying this threshold will be removed. For exmaple, if set to 0.75, genes without at least 75 percent of the samples being non-zero will be removed. Defaults to 0.75.
L_F_p	Double >= 0, <= 1. Filter genes with standard deviation and skewness less than this p value before applying Linnorm algorithm. Defaults to 0.3173.
L_F_LC_Genes	Double >= 0.01, <= 0.95 or Character "Auto". Filter this portion of the lowest expressing genes before applying Linnorm algorithm. It can be determined automatically by setting to "Auto". Defaults to "Auto".
L_F_HC_Genes	Double >= 0.01, <= 0.95. Filter this portion of the highest expressing genes before applying Linnorm algorithm. Defaults to 0.01.
BE_F_p	Double >=0, <= 1. Filter genes with standard deviation and skewness less than this p value before applying Linnorm's batch effect normalization algorithm. Defaults to 0.3173.
BE_F_LC_Genes	Double >= 0.01, <= 0.95 or Character "Auto". Filter this portion of the lowest expressing genes before applying Linnorm's batch effect normalization algorithm. It can be determined automatically by setting to "Auto". Defaults to "Auto".
BE_F_HC_Genes	Double >= 0.01, <= 0.95. Filter this portion of the highest expressing genes before applying Linnorm's batch effect normalization algorithm. Defaults to 0.01.
BE_strength	Double >0, <= 1. Before Linnorm transformation, how strongly should Linnorm normalize batch effects? Defaults to 0.5.
max_F_LC	Double >=0, <= 0.95. When L_F_LC or B_F_LC is set to auto, this is the maximum threshold that Linnorm would assign. Defaults to 0.75.
DataImputation	Logical. Perform data imputation on the dataset after transformation. Defaults to FALSE.
...	place holder for any new arguments.

Details

This function normalizes and transforms the input dataset using the Linnorm algorithm.

Value

This function returns a transformed data matrix.

If Filter is set to True, this function will output a list with the following objects:

- Linnorm: The full non-filtered transformed data matrix.
- Keep_Features: The features that survived filtering, users may use it to filter the data.

Examples

```
#Obtain example matrix:
data(LIHC)
#Transformation:
transformedExp <- Linnorm(LIHC)
```

Linnorm.Cor *Linnorm-gene correlation network analysis.*

Description

This function first performs Linnorm transformation on the dataset. Then, it will perform correlation network analysis on the dataset.

Usage

```
Linnorm.Cor(
  datamatrix,
  RowSamples = FALSE,
  input = "Raw",
  method = "pearson",
  MZP = 0.5,
  sig.q = 0.05,
  plotNetwork = TRUE,
  plotNumPairs = 5000,
  plotdegree = 0,
  plotname = "networkplot",
  plotformat = "png",
  plotVertexSize = 1,
  plotFontSize = 1,
  plot.Pos.cor.col = "red",
  plot.Neg.cor.col = "green",
  vertex.col = "cluster",
  plotlayout = "kk",
  clusterMethod = "cluster_edge_betweenness",
  ...
)
```

Arguments

datamatrix	The matrix or data frame that contains your dataset. Raw Counts, CPM, RPKM, FPKM, TPM or Linnorm transformed data are supported. Undefined values such as NA are not supported.
RowSamples	Logical. In the datamatrix, if each row is a sample and each column is a feature, set this to TRUE so that you don't need to transpose it. Linnorm works slightly faster with this argument set to TRUE, but it should be negligable for smaller datasets. Defaults to FALSE.
input	Character. "Raw" or "Linnorm". In case you have already transformed your dataset with Linnorm, set input into "Linnorm" so that you can put the Linnorm transformed dataset into the "datamatrix" argument. Defaults to "Raw".
method	Character. "pearson", "kendall" or "spearman". Method for the calculation of correlation coefficients. Defaults to "pearson".
MZP	Double >=0, <= 1. Minimum non-Zero Portion Threshold for this function. Genes not satisfying this threshold will be removed for correlation calculation. For exmaple, if set to 0.3, genes without at least 30 percent of the samples being non-zero will be considered for this study. Defaults to 0.5.
sig.q	Double >=0, <= 1. Only gene pairs with q values less than this threshold will be included in the "Results" data frame. Defaults to 0.05.
plotNetwork	Logical. Should the program output the network plot to a file? An "igraph" object will be included in the output regardless. Defaults to TRUE.
plotNumPairs	Integer >= 50. Number of gene pairs to be used in the network plot. Defaults to 5000.
plotdegree	Integer >= 0. In the network plot, genes (vertices) without at least this number of degree will be removed. Defaults to 0.
plotname	Character. Name of the network plot. File extension will be appended to it. Defaults to "networkplot".
plotformat	Character. "pdf" or "png". Network plot output format. Defaults to "png".
plotVertexSize	Double >0. Controls vertex Size in the network plot. Defaults to 1.
plotFontSize	Double >0. Controls font Size in the network plot. Defaults to 1.
plot.Pos.cor.col	Character. Color of the edges of positively correlated gene pairs. Defaults to "red".
plot.Neg.cor.col	Character. Color of the edges of negatively correlated gene pairs. Defaults to "green".
vertex.col	Character. "cluster" or a color. This controls the color of the vertices. Defaults to "cluster".
plotlayout	Character. "kk" or "fr". "kk" uses Kamada-Kawai algorithm in igraph to assign vertex and edges. It scales edge length with correlation strength. However, it can cause overlaps between vertices. "fr" uses Fruchterman-Reingold algorithm in igraph to assign vertex and edges. It prevents overlatps between vertices better than "kk", but edge lengths are not scaled to correlation strength. Defaults to "kk".

clusterMethod	Character. "cluster_edge_betweenness", "cluster_fast_greedy", "cluster_infomap", "cluster_label_prop", "cluster_leading_eigen", "cluster_louvain", "cluster_optimal", "cluster_spinglass" or "cluster_walktrap". These are clustering functions from the igraph package. Defaults to "cluster_edge_betweenness".
...	arguments that will be passed into Linnorm's transformation function.

Details

This function performed gene correlated study in the dataset by using Linnorm transformation

Value

This function will output a list with the following objects:

- Results: A data frame containing the results of the analysis, showing only the significant results determined by "sig.q" (see below).
- Cor.Matrix: The resulting correlation matrix between each gene.
- q.Matrix: A matrix of q values of each of the correlation coefficient from Cor.Matrix.
- Cluster: A data frame that shows which gene belongs to which cluster.
- igraph: The igraph object for users who want to draw the network plot manually.
- Linnorm: Linnorm transformed data matrix.

The "Results" data frame has the following columns:

- Gene1: Name of gene 1.
- Gene2: Name of gene 2.
- Cor: Correlation coefficient between the two genes.
- p.value: p value of the correlation coefficient.
- q.value: q value of the correlation coefficient.

Examples

```
data(Islam2011)
#Analysis on Islam2011 embryonic stem cells
results <- Linnorm.Cor(Islam2011[,1:48], plotNetwork=FALSE)
```

Linnorm.DataImput *Linnorm Data Imputation Function. (In development)*

Description

This function performs data imputation for (sc)RNA-seq expression data or large scale count data. It will treat every zero count in the dataset as missing data and replace them with predicted values.

Usage

```
Linnorm.DataImput(
  datamatrix,
  RowSamples = FALSE,
  showinfo = FALSE,
  MZP = 0.25,
  LC_F = "Auto",
  max_LC_F = 0.75,
  FG_Recov = 0.5,
  method = "euclidean",
  VarPortion = 0.75,
  ...
)
```

Arguments

datamatrix	The matrix or data frame that contains your dataset. It is only compatible with log transformed datasets.
RowSamples	Logical. In the datamatrix, if each row is a sample and each column is a feature, set this to TRUE so that you don't need to transpose it. Defaults to FALSE.
showinfo	Logical. Show algorithm running information. Defaults to FALSE.
MZP	Double >=0, <= 1. Minimum non-Zero Portion Threshold for this function. Genes not satisfying this threshold will be removed. For exmaple, if set to 0.3, genes without at least 30 percent of the samples being non-zero will be removed. Defaults to 0.25.
LC_F	Double >= 0.01, <= 0.95 or Character "Auto". Filter this portion of the lowest expressing genes. It can be determined automatically by setting to "Auto". Defaults to "Auto".
max_LC_F	Double >=0, <= 0.95. When LC_F is set to auto, this is the maximum threshold that Linnorm would assign. Defaults to 0.75.
FG_Recov	Double >=0, <= 1. In the low count gene filtering algorithm, recover this portion of genes that are filtered. Defaults to 0.5.
method	Character. Method for calculating the distance matrix. This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary", "pearson", "correlation", "spearman" or "kendall". Any unambiguous substring can be given. Defaults to "euclidean".
VarPortion	Double >0, <=0.95. Portion of the variance from PCA to be used for data imputation. Defaults to 0.5.
...	place holder for any new arguments.

Details

This function performs data imputation on the dataset. It first generates a distance matrix using principal components from PCA. Then, by default, using the distance matrix as weight, it predicts missing values from each gene using inverse euclidean distance weighted mean.

Value

This function returns a data matrix.

Examples

```
#Obtain example matrix:
data(Islam2011)
#Transformation:
Transformed <- Linnorm(Islam2011)
#Data imputation
DataImput <- Linnorm.DataImput(Transformed)
```

`Linnorm.HClust` *Linnorm-hierarchical clustering analysis.*

Description

This function first performs Linnorm transformation on the dataset. Then, it will perform hierarchical clustering analysis.

Usage

```
Linnorm.HClust(
  datamatrix,
  RowSamples = FALSE,
  MZP = 0,
  DataImputation = TRUE,
  input = "Raw",
  method_hclust = "ward.D",
  method_dist = "pearson",
  Group = NULL,
  num_Clust = 0,
  Color = "Auto",
  ClustRect = TRUE,
  RectColor = "red",
  fontsize = 0.5,
  linethickness = 0.5,
  plot.title = "Hierarchical clustering",
  ...
)
```

Arguments

`datamatrix`	The matrix or data frame that contains your dataset. Raw Counts, CPM, RPKM, FPKM, TPM or Linnorm transformed data are supported. Undefined values such as NA are not supported.

RowSamples	Logical. In the datamatrix, if each row is a sample and each column is a feature, set this to TRUE so that you don't need to transpose it. Linnorm works slightly faster with this argument set to TRUE, but it should be negligable for smaller datasets. Defaults to FALSE.
MZP	Double >=0, <= 1. Minimum non-Zero Portion Threshold for this function. Genes not satisfying this threshold will be removed from HVG anlaysis. For exmaple, if set to 0.3, genes without at least 30 percent of the samples being non-zero will be removed. Defaults to 0.
DataImputation	Logical. Perform data imputation on the dataset after transformation. Defaults to TRUE.
input	Character. "Raw" or "Linnorm". In case you have already transformed your dataset with Linnorm, set input into "Linnorm" so that you can input the Linnorm transformed dataset into the "datamatrix" argument. Defaults to "Raw".
method_hclust	Charcter. Method to be used in hierarchical clustering. (From hclust fastcluster: the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median" or "centroid".) Defaults to "ward.D".
method_dist	Charcter. Method to be used in hierarchical clustering. (From Dist amap: the distance measure to be used. This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary", "pearson", "correlation", "spearman" or "kendall". Any unambiguous substring can be given.) Defaults to "pearson".
Group	Character vector with length equals to sample size. Each character in this vector corresponds to each of the columns (samples) in the datamatrix. If this is provided, sample names will be colored according to their group. Defaults to NULL.
num_Clust	Integer >= 0. Number of clusters in hierarchical clustering. No cluster will be highlighted if this is set to 0. Defaults to 0.
Color	Character vector. Color of the groups/clusters in the plot. This vector must be as long as num_Clust, or Group if it is provided. Defaults to "Auto".
ClustRect	Logical. If num_Clust > 0, should a rectangle be used to highlight the clusters? Defaults to TRUE.
RectColor	Character. If ClustRect is TRUE, this controls the color of the rectangle. Defaults to "red".
fontsize	Numeric. Font size of the texts in the figure. Defualts to 0.5.
linethickness	Numeric. Controls the thickness of the lines in the figure. Defaults to 0.5.
plot.title	Character. Set the title of the plot. Defaults to "Hierarchical clustering".
...	arguments that will be passed into Linnorm's transformation function.

Details

This function performs PCA clustering using Linnorm transformation.

Value

It returns a list with the following objects:

- Results: If num_Clust > 0, this outputs a named vector that contains the cluster assignment information of each sample. Else, this outputs a number 0.
- plot: Plot of hierarchical clustering.
- Linnorm: Linnorm transformed data matrix.

Examples

```
#Obtain example matrix:
data(Islam2011)
#Example:
HClust.results <- Linnorm.HClust(Islam2011, Group=c(rep("ESC",48), rep("EF",44), rep("NegCtrl",4)))
```

`Linnorm.HVar` *Linnorm-Hvar pipeline for highly variable gene discovery.*

Description

This function first performs Linnorm transformation on the dataset. Then, it will perform highly variable gene discovery.

Usage

```
Linnorm.HVar(
  datamatrix,
  RowSamples = FALSE,
  spikein = NULL,
  spikein_log2FC = NULL,
  log.p = FALSE,
  sig.value = "p",
  sig = 0.05,
  MZP = 0.25,
  FG_Recov = 0.5,
  plot.title = "Mean vs SD plot",
  ...
)
```

Arguments

`datamatrix`	The matrix or data frame that contains your dataset. Raw Counts, CPM, RPKM, FPKM or TPM are supported. Undefined values such as NA are not supported. It is not compatible with log transformed datasets.
`RowSamples`	Logical. In the datamatrix, if each row is a sample and each column is a feature, set this to TRUE so that you don't need to transpose it. Linnorm works slightly faster with this argument set to TRUE, but it should be negligable for smaller datasets. Defaults to FALSE.

spikein	character vector. Names of the spike-in genes in the datamatrix. Defaults to NULL.
spikein_log2FC	Numeric vector. Log 2 fold change of the spike-in genes. Defaults to NULL.
log.p	Logical. Output p/q values in log scale. Defaults to FALSE.
sig.value	Character. "p" or "q". Use p or q value for highlighting significant genes. Defaults to "p".
sig	Double >0, <= 1. Significant level of p or q value for plotting. Defaults to 0.05.
MZP	Double >=0, <= 1. Minimum non-Zero Portion Threshold for this function. Genes not satisfying this threshold will be removed from HVG anlaysis. For exmaple, if set to 0.3, genes without at least 30 percent of the samples being non-zero will be removed. Defaults to 0.25.
FG_Recov	Double >=0, <= 1. In the low count gene filtering algorithm, recover this portion of genes that are filtered. Defaults to 0.5.
plot.title	Character. The plot's title. Defaults to "Mean vs SD plot".
...	arguments that will be passed into Linnorm's transformation function.

Details

This function discovers highly variable gene in the dataset using Linnorm transformation.

Value

This function will output a list with the following objects:

- Results: A matrix with the results.
- plot: Mean vs Standard Deviation Plot which highlights significant genes.
- Linnorm: Linnorm transformed data matrix.

The Results matrix has the following columns:

- Transformed.Avg.Exp: Average expression of non-zero Linnorm transformed data.
- Transformed.SD: Standard deviation of non-zero Linnorm transformed data.
- Normalized.Log2.SD.Fold.Change: Normalized log2 fold change of the gene's standard deviation.
- p.value: p value of the statistical test.
- q.value: q value/false discovery rate/adjusted p value of the statistical test.

Examples

```
data(Islam2011)
results <- Linnorm.HVar(Islam2011)
```

Linnorm.limma *Linnorm-limma pipeline for Differentially Expression Analysis*

Description

This function first performs Linnorm transformation on the dataset. Then, it will perform limma for DEG analysis. Please cite both Linnorm and limma when you use this function for publications.

Usage

```
Linnorm.limma(
  datamatrix,
  design = NULL,
  RowSamples = FALSE,
  MZP = 0,
  output = "DEResults",
  noINF = TRUE,
  robust = FALSE,
  ...
)
```

Arguments

datamatrix	The matrix or data frame that contains your dataset. Raw Counts, CPM, RPKM, FPKM or TPM are supported. Undefined values such as NA are not supported. It is not compatible with log transformed datasets.
design	A design matrix required for limma. Please see limma's documentation or our vignettes for more detail.
RowSamples	Logical. In the datamatrix, if each row is a sample and each column is a feature, set this to TRUE so that you don't need to transpose it. Linnorm works slightly faster with this argument set to TRUE, but it should be negligable for smaller datasets. Defaults to FALSE.
MZP	Double >=0, <= 1. Minimum non-Zero Portion Threshold for this function. Genes not satisfying this threshold will be removed from HVG anlaysis. For exmaple, if set to 0.3, genes without at least 30 percent of the samples being non-zero will be removed. Defaults to 0.
output	Character. "DEResults" or "Both". Set to "DEResults" to output a matrix that contains Differential Expression Analysis Results. Set to "Both" to output a list that contains both Differential Expression Analysis Results and the transformed data matrix.
noINF	Logical. Prevent generating INF in the fold change column by adding the estimated count of one. If it is set to FALSE, zero or INF will be generated if one of the conditions has zero expression. Defaults to TRUE.
robust	Logical. In the eBayes function of Limma, run with robust setting with TRUE or FALSE. Defaults to FALSE.
...	arguments that will be passed into Linnorm's transformation function.

Details

This function performs both Linnorm and limma for users who are interested in differential expression analysis.

Value

If output is set to "DEResults", this function will output a matrix with Differntial Expression Analysis Results with the following columns:

- logFC: Log 2 Fold Change
- XPM: Average Expression. If input is raw count or CPM, this column has the CPM unit. If input is RPKM, FPKM or TPM, this column has the TPM unit.
- t: moderated t-statistic
- P.Value: p value
- adj.P.Val: Adjusted p value. This is also called False Discovery Rate or q value.
- B: log odds that the feature is differential

If output is set to Both, this function will output a list with the following objects:

- DEResults: Differntial Expression Analysis Results as described above.
- Linnorm: Linnorm transformed data matrix.

Examples

```
#Obtain example matrix:
data(LIHC)
#Create limma design matrix (first 5 columns are tumor, last 5 columns are normal)
designmatrix <- c(rep(1,5),rep(2,5))
designmatrix <- model.matrix(~ 0+factor(designmatrix))
colnames(designmatrix) <- c("group1", "group2")
rownames(designmatrix) <- colnames(LIHC)
#DEG analysis
DEGResults <- Linnorm.limma(LIHC, designmatrix)
```

`Linnorm.Norm` *Linnorm Normalization Function*

Description

This function performs batch effect and library size difference normalization on the input dataset.

Usage

```
Linnorm.Norm(
  datamatrix,
  RowSamples = FALSE,
  spikein = NULL,
  spikein_log2FC = NULL,
  showinfo = FALSE,
  output = "XPM",
  minNonZeroPortion = 0.3,
  BE_F_p = 0.3173,
  BE_F_LC_Genes = "Auto",
  BE_F_HC_Genes = 0.01,
  BE_strength = 0.5,
  max_F_LC = 0.75
)
```

Arguments

datamatrix	The matrix or data frame that contains your dataset. Raw Counts, CPM, RPKM, FPKM or TPM are supported. Undefined values such as NA are not supported. It is not compatible with log transformed datasets.
RowSamples	Logical. In the datamatrix, if each row is a sample and each column is a feature, set this to TRUE so that you don't need to transpose it. Linnorm works slightly faster with this argument set to TRUE, but it should be negligable for smaller datasets. Defaults to FALSE.
spikein	character vector. Names of the spike-in genes in the datamatrix. Defaults to NULL.
spikein_log2FC	Numeric vector. Log 2 fold change of the spike-in genes. Defaults to NULL.
showinfo	Logical. Show algorithm running information. Defaults to FALSE.
output	character. "Raw" or "XPM". Output's total count will be approximately the median of the inputs' when set to "Raw". Output CPM (if input is raw counts or CPM) or TPM (if input is RPKM FPKM or TPM) when set to "XPM".
minNonZeroPortion	Double >=0, <= 1. Minimum non-Zero Portion Threshold. Genes not satisfying this threshold will be removed. For exmaple, if set to 0.75, genes without at least 75 percent of the samples being non-zero will be removed. Defaults to 0.75.
BE_F_p	Double >=0, <= 1. Filter genes with standard deviation and skewness less than this p value before applying Linnorm's batch effect normalization algorithm. Defaults to 0.3173.
BE_F_LC_Genes	Double >= 0.01, <= 0.95 or Character "Auto". Filter this portion of the lowest expressing genes before applying Linnorm's batch effect normalization algorithm. It can be determined automatically by setting to "Auto". Defaults to "Auto".
BE_F_HC_Genes	Double >=0, <= 1. Filter this portion of the highest expressing genes before applying Linnorm's batch effect normalization algorithm. Defaults to 0.01.

BE_strength	Double >0, <= 1. How strongly should Linnorm normalize batch effects? Defaults to 0.5.
max_F_LC	Double >=0, <= 0.95. When L_F_LC or B_F_LC is set to auto, this is the maximum threshold that Linnorm would assign. Defaults to 0.75.

Details

This function normalizes the input dataset using the Linnorm algorithm.

Value

This function returns a normalized data matrix.

Examples

```
#Obtain example matrix:
data(LIHC)
#Normalization:
normalizedExp <- Linnorm(LIHC)
```

Linnorm.PCA *Linnorm-PCA Clustering pipeline for subpopulation Analysis*

Description

This function first performs Linnorm transformation on the dataset. Then, it will perform Principal component analysis on the dataset and use k-means clustering to identify subpopulations of cells.

Usage

```
Linnorm.PCA(
  datamatrix,
  RowSamples = FALSE,
  input = "Raw",
  MZP = 0,
  HVar_p_value = 0.5,
  DataImputation = TRUE,
  num_PC = 3,
  num_center = c(1:20),
  Group = NULL,
  Coloring = "kmeans",
  pca.scale = FALSE,
  kmeans.iter = 2000,
  plot.title = "PCA K-means clustering",
  ...
)
```

Arguments

datamatrix	The matrix or data frame that contains your dataset. Raw Counts, CPM, RPKM, FPKM or TPM are supported. Undefined values such as NA are not supported. It is not compatible with log transformed datasets.
RowSamples	Logical. In the datamatrix, if each row is a sample and each column is a feature, set this to TRUE so that you don't need to transpose it. Linnorm works slightly faster with this argument set to TRUE, but it should be negligable for smaller datasets. Defaults to FALSE.
input	Character. "Raw" or "Linnorm". In case you have already transformed your dataset with Linnorm, set input into "Linnorm" so that you can put the Linnorm transformed dataset into the "datamatrix" argument. Defaults to "Raw".
MZP	Double >=0, <= 1. Minimum non-Zero Portion Threshold for this function. Genes not satisfying this threshold will be removed from the analysis. For exmaple, if set to 0.3, genes without at least 30 percent of the samples being non-zero will be removed. Defaults to 0.
HVar_p_value	Double >=0, <= 1. Highly variable feature p value threshold to be used for tSNE. Defaults to 0.5.
DataImputation	Logical. Perform data imputation on the dataset after transformation. Defaults to TRUE.
num_PC	Integer >= 2. Number of principal componenets to be used in K-means clustering. Defaults to 3.
num_center	Numeric vector. Number of clusters to be tested for k-means clustering. fpc, vegan, mclust and apcluster packages are used to determine the number of clusters needed. If only one number is supplied, it will be used and this test will be skipped. Defaults to c(1:20).
Group	Character vector with length equals to sample size. Each character in this vector corresponds to each of the columns (samples) in the datamatrix. In the plot, the shape of the points that represent each sample will be indicated by their group assignment. Defaults to NULL.
Coloring	Character. "kmeans" or "Group". If Group is not NULL, coloring in the PCA plot will reflect each sample's group. Otherwise, coloring will reflect k means clustering results. Defaults to "Group".
pca.scale	Logical. In the prcomp(for Principal component analysis) function, set the "scale." parameter. It signals the function to scale unit variances in the variables before the analysis takes place. Defaults to FALSE.
kmeans.iter	Numeric. Number of iterations in k-means clustering. Defaults to 2000.
plot.title	Character. Set the title of the plot. Defaults to "PCA K-means clustering".
...	arguments that will be passed into Linnorm's transformation function.

Details

This function performs PCA clustering using Linnorm transformation.

Value

It returns a list with the following objects:

- k_means: Output of kmeans(for K-means clustering) from the stat package. Note: It contains a "cluster" object that indicates each sample's cluster assignment.
- PCA: Output of prcomp(for Principal component analysis) from the stat package.
- plot: Plot of PCA clustering.
- Linnorm: Linnorm transformed data matrix.

Examples

```
#Obtain example matrix:
data(Islam2011)
#Example:
PCA.results <- Linnorm.PCA(Islam2011)
```

Linnorm.SGenes *Linnorm model stable gene selection tool.*

Description

For datasets without spike-ins and for users who do not wish to rely on spike-ins, we provide this model stable gene selection tool.

Usage

```
Linnorm.SGenes(
  datamatrix,
  RowSamples = FALSE,
  showinfo = FALSE,
  minNonZeroPortion = 0.75,
  F_p = 0.3173,
  F_LC_Genes = "Auto",
  F_HC_Genes = 0.01,
  max_F_LC = 0.75
)
```

Arguments

datamatrix	The matrix or data frame that contains your dataset. Raw Counts, CPM, RPKM, FPKM or TPM are supported. Undefined values such as NA are not supported. It is not compatible with log transformed datasets.
RowSamples	Logical. In the datamatrix, if each row is a sample and each column is a feature, set this to TRUE so that you don't need to transpose it. Linnorm works slightly faster with this argument set to TRUE, but it should be negligable for smaller datasets. Defaults to FALSE.

showinfo Logical. Show algorithm running information. Defaults to FALSE.

minNonZeroPortion
Double >=0, <= 1. Minimum non-Zero Portion Threshold. Genes not satisfying this threshold will be removed. For exmaple, if set to 0.75, genes without at least 75 percent of the samples being non-zero will be removed. Defaults to 0.75.

F_p Double >=0, <= 1. Filter genes with standard deviation and skewness less than this p value. Defaults to 0.3173.

F_LC_Genes Double >= 0.01, <= 0.95 or Character "Auto". Filter this portion of the lowest expressing genes. It can be determined automatically by setting to "Auto". Defaults to "Auto".

F_HC_Genes Double >=0, <= 1. Filter this portion of the highest expressing genes. Defaults to 0.01.

max_F_LC Double >=0, <= 0.95. When F_LC is set to auto, this is the maximum threshold that Linnorm would assign. Defaults to 0.75.

Details

This function selects stable genes from the dataset using the Linnorm's algorithm.

Value

This function returns a data matrix that contains stable genes only.

Examples

```
#Obtain example matrix:
data(Islam2011)
#Transformation:
StableGenes <- Linnorm.SGenes(Islam2011)
```

Linnorm.tSNE *Linnorm t-SNE Clustering pipeline for subpopulation Analysis*

Description

This function first performs Linnorm transformation on the dataset. Then, it will perform t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction on the dataset and use k-means clustering to identify subpopulations of cells.

Usage

```
Linnorm.tSNE(
  datamatrix,
  RowSamples = FALSE,
  input = "Raw",
  MZP = 0,
```

```
  HVar_p_value = 0.5,
  num_PC = 3,
  num_center = c(1:20),
  Group = NULL,
  Coloring = "kmeans",
  kmeans.iter = 2000,
  plot.title = "t-SNE K-means clustering",
  ...
)
```

Arguments

datamatrix	The matrix or data frame that contains your dataset. Raw Counts, CPM, RPKM, FPKM or TPM are supported. Undefined values such as NA are not supported. It is not compatible with log transformed datasets.
RowSamples	Logical. In the datamatrix, if each row is a sample and each column is a feature, set this to TRUE so that you don't need to transpose it. Linnorm works slightly faster with this argument set to TRUE, but it should be negligable for smaller datasets. Defaults to FALSE.
input	Character. "Raw" or "Linnorm". In case you have already transformed your dataset with Linnorm, set input into "Linnorm" so that you can put the Linnorm transformed dataset into the "datamatrix" argument. Defaults to "Raw".
MZP	Double >=0, <= 1. Minimum non-Zero Portion Threshold for this function. Genes not satisfying this threshold will be removed from the analysis. For exmaple, if set to 0.3, genes without at least 30 percent of the samples being non-zero will be removed. Defaults to 0.
HVar_p_value	Double >=0, <= 1. Highly variable feature p value threshold to be used for tSNE. Defaults to 0.5.
num_PC	Integer >= 2. Number of principal componenets to be used in K-means clustering. Defaults to 3.
num_center	Numeric vector. Number of clusters to be tested for k-means clustering. fpc, vegan, mclust and apcluster packages are used to determine the number of clusters needed. If only one number is supplied, it will be used and this test will be skipped. Defaults to c(1:20).
Group	Character vector with length equals to sample size. Each character in this vector corresponds to each of the columns (samples) in the datamatrix. In the plot, the shape of the points that represent each sample will be indicated by their group assignment. Defaults to NULL.
Coloring	Character. "kmeans" or "Group". If Group is not NA, coloring in the plot will reflect each sample's group. Otherwise, coloring will reflect k means clustering results. Defaults to "Group".
kmeans.iter	Numeric. Number of iterations in k-means clustering. Defaults to 2000.
plot.title	Character. Set the title of the plot. Defaults to "t-SNE K-means clustering".
...	arguments that will be passed into Linnorm's transformation function.

Details

This function performs t-SNE K-means clustering using Linnorm transformation.

Value

It returns a list with the following objects:

- k_means: Output of kmeans(for K-means clustering) from the stat package. Note: It contains a "cluster" object that indicates each sample's cluster assignment.
- tSNE: Output from Rtsne.
- plot: Plot of t-SNE K-means clustering.
- Linnorm: Linnorm transformed data matrix.

Examples

```
#Obtain example matrix:
data(Islam2011)
#Example:
tSNE.results <- Linnorm.tSNE(Islam2011)
```

RnaXSim *This function simulates an RNA-seq dataset based on a given distribution.*

Description

This function simulates an RNA-seq dataset based on a given distribution.

Usage

```
RnaXSim(
  datamatrix,
  distribution = "NB",
  NumRep = 5,
  NumDiff = 2000,
  NumFea = 20000,
  showinfo = FALSE,
  DEGlog2FC = "Auto",
  MaxLibSizelog2FC = 0.5
)
```

Arguments

datamatrix	Matrix. The matrix or data frame that contains your dataset. Each row is a feature (or Gene) and each column is a sample (or replicate). Raw Counts, CPM, RPKM, FPKM or TPM are supported. Undefined values such as NA are not supported. It is not compatible with log transformed datasets.This program assumes that all columns are replicates of the same sample.
distribution	Character: Defaults to "Poisson". This parameter controls the output distribution of the simulated RNA-seq dataset. It can be one of "Gamma" (Gamma distribution), "Poisson" (Poisson distribution), "LogNorm" (Log Normal distribution) or "NB" (Negative Binomial distribution).
NumRep	Integer: The number of replicates. This is half of the number of output samples. Defaults to 3.
NumDiff	Integer: The number of Differentially Changed Features. Defaults to 2000.
NumFea	Integer: The number of Total Features. Defaults to 20000.
showinfo	Logical: should we show data information on the console? Defaults to FALSE.
DEGlog2FC	"Auto" or Double: log 2 fold change threshold that defines differentially expressed genes. If set to "Auto," DEGlog2FC is defined at the level where ANOVA can get a q value of 0.05 with the average expression, where the data values are log1p transformed. Defaults to "Auto".
MaxLibSizelog2FC	Double: The maximum library size difference from the mean that is allowed, in terms of log 2 fold change. Set to 0 to prevent program from generating library size differences. Defaults to 0.5.

Value

This function returns a list that contains a matrix of count data in integer raw count and a vector that shows which genes are differentially expressed. In the matrix, each row is a gene and each column is a replicate. The first NumRep (see parameter) of the columns belong to sample 1, and the last NumRep (see parameter) of the columns belong to sample 2. There will be NumFea (see parameter) number of rows. The top NumCorr of genes will be positively or negatively correlated with each other (randomly); and they are evenly separated into groups. Each group is not intended to be correlated to each other, but, by chance, it can happen.

Examples

```
#Obtain example matrix:
data(SEQC)
expMatrix <- SEQC
#Example for Negative Binomial distribution
simulateddata <- RnaXSim(expMatrix, distribution="NB", NumRep=5, NumDiff = 200, NumFea = 2000)
#Example for Poisson distribution
simulateddata <- RnaXSim(expMatrix, distribution="Poisson", NumRep=5, NumDiff = 200, NumFea = 2000)
#Example for Log Normal distribution
simulateddata <- RnaXSim(expMatrix, distribution="LogNorm", NumRep=5, NumDiff = 200, NumFea = 2000)
#Example for Gamma distribution
simulateddata <- RnaXSim(expMatrix, distribution="Gamma", NumRep=5, NumDiff = 200, NumFea = 2000)
```

SEQC *Partial RNA-seq data from SEQC/MAQC-III Sample A*

Description

Raw Count data

Usage

```
data(SEQC)
```

Format

A matrix with 50227 rows (genes) and 10 columns (samples).

References

SEQC/MAQC-III Consortium. A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. Nature biotechnology 32.9 (2014): 903-914.

Index

∗ **Analysis**
Linnorm.PCA, 18
Linnorm.tSNE, 21
∗ **CPM**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.SGenes, 20
Linnorm.tSNE, 21
∗ **Clustering**
Linnorm.HClust, 11
Linnorm.PCA, 18
Linnorm.tSNE, 21
∗ **Component**
Linnorm.PCA, 18
Linnorm.tSNE, 21
∗ **Count**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.SGenes, 20
Linnorm.tSNE, 21
RnaXSim, 23
∗ **Expression**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.SGenes, 20
Linnorm.tSNE, 21
RnaXSim, 23
∗ **FPKM**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.SGenes, 20
Linnorm.tSNE, 21
∗ **Filter**
Linnorm.SGenes, 20
∗ **Gamma**
RnaXSim, 23
∗ **K-means**
Linnorm.PCA, 18
Linnorm.tSNE, 21
∗ **Linear**
LinearRegression, 3
LinearRegressionFP, 4
∗ **Linnorm**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.DataImput, 9
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.SGenes, 20
Linnorm.tSNE, 21
∗ **Log**
RnaXSim, 23
∗ **Negative**
RnaXSim, 23
∗ **PCA**

Linnorm.PCA, 18
Linnorm.tSNE, 21
∗ **Parametric**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.PCA, 18
Linnorm.tSNE, 21
∗ **Poisson**
RnaXSim, 23
∗ **Principal**
Linnorm.PCA, 18
Linnorm.tSNE, 21
∗ **RNA-seq**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.DataImput, 9
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.SGenes, 20
Linnorm.tSNE, 21
RnaXSim, 23
∗ **RPKM**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.SGenes, 20
Linnorm.tSNE, 21
∗ **Raw**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.SGenes, 20
Linnorm.tSNE, 21
RnaXSim, 23
∗ **Regression**
LinearRegression, 3
LinearRegressionFP, 4
∗ **Simulate**
RnaXSim, 23
∗ **Simulation**
RnaXSim, 23
∗ **Stable**
Linnorm.SGenes, 20
∗ **TPM**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.SGenes, 20
Linnorm.tSNE, 21
∗ **cell**
Linnorm.DataImput, 9
∗ **coefficient**
Linnorm.Cor, 7
∗ **constant**
LinearRegression, 3
LinearRegressionFP, 4
∗ **correlation**
Linnorm.Cor, 7
∗ **data**
Linnorm.DataImput, 9
∗ **distribution**
RnaXSim, 23
∗ **embedding**
Linnorm.tSNE, 21
∗ **hierarchical**
Linnorm.HClust, 11
∗ **highly**
Linnorm.HVar, 13
∗ **imputation**
Linnorm.DataImput, 9
∗ **k-means**
Linnorm.PCA, 18
Linnorm.tSNE, 21
∗ **kendall**
Linnorm.Cor, 7
∗ **kmeans**
Linnorm.PCA, 18
Linnorm.tSNE, 21

∗ **limma**
Linnorm.limma, 15
∗ **missing**
Linnorm.DataImput, 9
∗ **neighbor**
Linnorm.tSNE, 21
∗ **normalization**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.tSNE, 21
∗ **pearson**
Linnorm.Cor, 7
∗ **single**
Linnorm.DataImput, 9
∗ **slope**
LinearRegression, 3
LinearRegressionFP, 4
∗ **spearman**
Linnorm.Cor, 7
∗ **stochastic**
Linnorm.tSNE, 21
∗ **t-SNE**
Linnorm.tSNE, 21
∗ **t-distributed**
Linnorm.tSNE, 21
∗ **transformation**
Linnorm, 5
Linnorm.Cor, 7
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.PCA, 18
Linnorm.tSNE, 21
∗ **value**
Linnorm.DataImput, 9
∗ **variable**
Linnorm.HVar, 13
∗ **variance**
Linnorm.HVar, 13

Islam2011, 2

LIHC, 3
LinearRegression, 3
LinearRegressionFP, 4
Linnorm, 5
Linnorm.Cor, 7
Linnorm.DataImput, 9
Linnorm.HClust, 11
Linnorm.HVar, 13
Linnorm.limma, 15
Linnorm.Norm, 16
Linnorm.PCA, 18
Linnorm.SGenes, 20
Linnorm.tSNE, 21

RnaXSim, 23

SEQC, 25