Package 'dbnorm'

November 29, 2019

Type Package

Title Drift across-batch normalization and visualization

Version 0.1.0

Author Nasim Bararpour

Maintainer Nasim Bararpour <nasimbararpour@gmail.com>

Description The dbnorm contains several functions applicable in a large-scale metabolomics as well as other high throughput experiments. Notably, it includes distinct functions for pre-processing of data and estimation of missing values, conventional functions for batch effect correction based on statistical models, as well as functions using advanced statistical tools to generate several diagnosis plots to inform users about their data structure. Several statistical models are implemented in the dbnorm such as two-stage procedure model as described by Giordan (2013) or empirical Bayes methods in two setting of parametric and non-parametric as described by Johnson et al.(2007), in order to give users the flexibility to choose one of those models which better fits to their data. By including advanced statistical tools, the dbnorm package allows user to inspect the structure and quality of multidimensional datasets both in macroscopic view, at the batch levels, and microscopically, at the level of features.

License LGPL(>= 2)

Encoding UTF-8

Imports ber, ggfortify, factoextra, ggplot2, NormalizeMets, sva,MASS, base(>= 3.5.1)

Suggests limma,installr,impute,Biobase,pcaMethods,tibble,knitr, rmarkdown, processx, backports, fs, Rcpp,BiocParallel,genefilter,stats

Remotes r-lib/testthat

biocViews Software

LazyData yes

RoxygenNote 6.1.1

R topics documented:

`ACDdbnorm` *Adjusted coefficient of determination for a data normalized for across batch signal drift*

Description

This function gives a quick notification about the performance of the statistical models implemented in the `dbnorm` package such as *Giordan (2013)* and/or empirical Bayes methods in two setting of parametric and non-parametric as described by *Johnson et al.(2007)* and in *sva* package by *Leek et al.(2012)*. It calculates adjusted coefficient of determination or *Adjusted R-Squared* for each variable estimated in a regression model for its dependency to the batch level in the raw data and treated data via either of those models. Immediately, the performance of applied models are presented by two scores calculated based on the total degree of variability and maximum variability explained by the batch level for each variables. Which respectively notify the overall performance of a model and the consistency of model performance for all detected variables (features), facilitating quick comparison of the models for selecting one of those models, which is more appropriate to the data structure. This function is suggested for less than *2k* features.
install.packages (c("ggplot2", "NormalizeMets","ggfortify", "factoextra","MASS","ber"))
source("https://bioconductor.org/biocLite.R")
biocLite(c("pcaMethods","impute","sva","limma","genefilter"))

Usage

```
ACDdbnorm(m)
```

Arguments

`m`	A data frame in which rows define the independent experiments (samples) and columns the features (variables).Batch order must be framed in the first column.

Details

Zero and NA values are not allowed. Optionally missing value can be imputed by `emvf` or `emvd`, functions implemented in `dbnorm` package. Input data must be normalized beforehand.

Value

A two columns matrix, for each applied model, consisting of the name of the variables (features) with the corresponding Adjusted R-squared value in *csv* format saved in the working directory. In parallel, two distinct bar plots for the scores given to the total variability and maximum variability with respect to each treatment algorithm compiled into *pdf* together with the corresponding exact value presented in the table and saved in *csv*.

References

M.Giordan (2013) *https://link.springer.com/article/10.1007/s12561-013-9081-1*
Johnson et al. (2007) *http://www.ncbi.nlm.nih.gov/pubmed/16632515*
Leek et al. (2012) *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3307112/*

Examples

```
batch<- rep(gl(3,7,labels = c(1:3)),1)
y<- matrix(rnorm(2100),nrow=21)
m<-data.frame(batch,y)
```

`dbnormBer` *Clustering and regression analysis of data normalization via ber-model and visualization*

Description

This function allows you to adjust the data for across batch signal drift or batch effect using two-stage procedure approach as described by *M. Giordan (2013)* and includes advanced statistical tools to inspect the structure and quality of high throughput experiment both macroscopically and microscopically at the level of sample sets and metabolic feature, respectively. Notably, using this function users perform unsupervised clustering analysis on the raw and the treated dataset. In parallel, *Adjusted- R squared* value for each variable (feature) estimated by regression model is calculated, which demonstrate the dependency of variable (feature) to the batch level in either of those datasets. In addition, for quick notification about the performance of the applied model two scores are reported, which are calculated based on the total degree of variability and the maximum variability. These scores notify respectively the overall performance of the model and the consistency of model performance for all detected variables (features).
install.packages (c("ggplot2", "NormalizeMets","ggfortify", "factoextra","MASS","ber"))
source("https://bioconductor.org/biocLite.R")
biocLite(c("pcaMethods","impute","limma","genefilter"))

Usage

```
dbnormBer(f)
```

Arguments

`f`	A data frame in which rows define the independent experiments (samples) and columns the features (variables).Batch order must be framed in the first column.

Details

Zero and NA values are not allowed. Optionally missing value can be imputed by functions implemented in `dbnorm` package. Input must be normalized and transformed beforehand.

Value

An adjusted dataset in *csv* format together with a series of graphical displays such as **PCA** score plot, **Scree** plot, and and plot of *Adjusted-R squared* value for the variables (features) compiled into *pdf*. In addition, the *Adjusted- R squared* value for each variables for the raw and corrected datasets separately saved in *csv*. Besides, a table of *Sum* and *Maximum Adjusted-R squared* presented in *csv*. **RLA** plot visualized in the *Viewer* panel in the *rstudio* console.

References

M.Giordan (2013) *https://link.springer.com/article/10.1007/s12561-013-9081-1*

Examples

```
batch<- rep(gl(3,7,labels = c(1:3)),1)
y<- matrix(rnorm(2100),nrow=21)
f<-data.frame(batch,y)
```

`dbnormNPcom` *Clustering and regression analysis of data normalization via ComBat-NonParametric model and visualization*

Description

This function allows you to adjust the data for across batch signal drift or batch effect using Non-Parametric Empirical Bayes approach as described by *Johnson et al.(2007)* and in sva package. *dbnormNPcom* includes advanced statistical tools to inspect the structure and quality of high throughput experiment both macroscopically and microscopically at the sample batch level and metabolic feature level, respectively. Notably, using this function users perform unsupervised clustering analysis of the raw data and the treated dataset. In parallel, *Adjusted-R squared* value for each variable (feature) estimated by regression model is calculated, which demonstrate the dependency of variable (feature) to the batch level in either of those datasets. In addition, for quick notification about the performance of the applied model two scores are reported, which are calculated based on the total degree of variability and the maximum variability. These scores notify respectively the overall performance of the model and the consistency of model performance for all detected variables (features).
install.packages (c("ggplot2", "NormalizeMets","ggfortify", "factoextra"))
source("https://bioconductor.org/biocLite.R")
biocLite(c("pcaMethods","impute","sva","limma","genefilter"))

Usage

```
dbnormNPcom(f)
```

Arguments

`f`	A data frame in which rows define the independent experiments (samples) and columns the features (variables).Batch order must be framed in the first column.

Details

Zero and NA values are not allowed. Optionally missing value can be imputed by `emvf` or `emvd`, functions implemented in `dbnorm` package. Input must be normalized and transformed beforehand.

Value

An adjusted dataset in *csv* format together with a series of graphical displays such as **PCA** score plot, **Scree** plot, and and plot of *Adjusted-R squared* value for the variables (features) compiled into *pdf*. In addition, the *Adjusted- R squared* value for each variables for the raw and corrected datasets separately saved in *csv*. Besides, a table of *Sum* and *Maximum Adjusted-R squared* presented in *csv*. **RLA** plot visualized in the *Viewer* panel in the *rstudio* console.

References

Johnson et al.(2007) *http://www.ncbi.nlm.nih.gov/pubmed/16632515*
Leek et al. (2012) *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3307112/*

Examples

```
batch<- rep(gl(3,7,labels = c(1:3)),1)
y<- matrix(rnorm(2100),nrow=21)
f<-data.frame(batch,y)
```

`dbnormPcom` *Clustering and regression analysis of data normalization via ComBat-Parametric model and visualization*

Description

This function allows you to adjust the data for across batch signal drift or batch effect using Parametric Empirical Bayes approach as described by *Johnson et al.(2007)* and via *sva* package as explained by *Leek et al.(2012)*. *dbnormPcom* includes advanced statistical tools to inspect the structure and quality of high throughput experiment both macroscopically and microscopically at the sample batch level and metabolic feature level, respectively. Notably, using this function users perform unsupervised clustering analysis of the raw data and the treated dataset. In parallel, *Adjusted- R squared* value for each variable (feature) estimated by regression model is calculated, which demonstrate the dependency of variable (feature) to the batch level in either of those datasets. In addition, for quick notification about the performance of the applied model two scores are reported, which are calculated based on the total degree of variability and the maximum variability. These scores notify respectively the overall performance of the model and the consistency of model performance for all detected variables (features).
install.packages (c("ggplot2", "NormalizeMets","ggfortify", "factoextra"))
source("https://bioconductor.org/biocLite.R")
biocLite(c("pcaMethods","impute","sva","limma","genefilter"))

Usage

```
dbnormPcom(f)
```

Arguments

`f`	A data frame in which rows define the independent experiments (samples) and columns the features (variables).Batch order must be framed in the first column.

Details

Zero and NA values are not allowed. Optionally missing value can be imputed by functions implemented in `dbnorm` package. Input must be normalized and transformed beforehand.

Value

An adjusted dataset in *csv* format together with a series of graphical displays such as **PCA** score plot, **Scree** plot, and and plot of *Adjusted-R squared* value for the variables (features) compiled into *pdf*. In addition, the *Adjusted- R squared* value for each variables for the raw and corrected datasets separately saved in *csv*. Besides, a table of *Sum* and *Maximum Adjusted-R squared* presented in *csv*. **RLA** plot visualized in the *Viewer* panel in the *rstudio* console.

References

Johnson et al. (2007) *http://www.ncbi.nlm.nih.gov/pubmed/16632515*
Leek et al. (2012) *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3307112/*

Examples

```
batch<- rep(gl(3,7,labels = c(1:3)),1)
y<- matrix(rnorm(2100),nrow=21)
f<-data.frame(batch,y)
```

`emvd` *Estimation of missing value data-based*

Description

It returns to a matrix of data in which missing values are estimated by the lowest detected value in the entire experiment. By this function, all NA values are replaced by Zero values, that of being ultimately replaced by the lowest value detected in the experiment. Ultimately, data matrix is transposed to restore original structure.

Usage

```
emvd(m)
```

Arguments

`m`	An array or a matrix

Details

empty entries are not allowed

Value

A matrix with estimated missing value.

Examples

```
m<- data.frame(x1=c(50,NA,6,10,30),x2=c(2,8,NA,15,0))
```

emvf *Estimation of missing value feature-based*

Description

It returns to a matrix of data in which missing values are estimated. By this function, all Zero values are first replaced by NA values, which are then replaced by the lowest detected value on the column margin. This function is applicable in all sort of high-throughput experiments. In a metabolomics workflow, it is suggested for a targeted approach where variables (features) measured at specific mass to charge transition.

Usage

```
emvf(m)
```

Arguments

m	An array or a matrix

Details

empty entries are not allowed

Value

A matrix with estimated missing value.

Examples

```
m<- data.frame(x1=c(50,NA,6,10,30),x2=c(2,8,NA,15,0))
```

profplotber *Visualization of analytical heterogeneity on the profile of variables (features) in ber- corrected data*

Description

profplotber allows you to adjust the data for batch effect using two-stage procedure approach as describes by *Giordan (2013)* and informs you about the presence of across batch signal drift or batch effect in the treated data determined by the shifted *probability density function* plots (*pdf* plots) of variables (features) detected in an experiment.
install.packages (c("ggplot2","ggfortify", "factoextra","MASS","ber"))

Usage

```
profplotber(m)
```

Arguments

m	A data frame in which rows define the independent experiments (samples) and columns the features (variables).Batch order must be framed in the first column.

Details

Zero and NA values are not allowed. Optionally missing value can be imputed by functions implemented in dbnorm package. Input must be normalized and transformed beforehand.

Value

An adjusted dataset in *csv* format together with the series of profile plot for the variables(features) in the sample sets analyzed in the entire experiment provided by the **Scatter** plot,**Violin** plot and **Density** plot compiled into *pdf*.

References

M.Giordan (2013) *https://link.springer.com/article/10.1007/s12561-013-9081-1*

Examples

```
batch<- rep(gl(3,7,labels = c(1:3),1))
y<- matrix(rnorm(2100),nrow=21)
m<-data.frame(batch,y)
```

---

profplotnpcom — *Visualization of analytical heterogeneity on the profile of variables (features) in ComBat-Non Parametric- corrected data*

Description

Visualization of analytical heterogeneity on the profile of variables (features) in ComBat-Non Parametric-corrected data

Usage

```
profplotnpcom(m)
```

Arguments

m	A data frame in which rows define the independent experiments (samples) and columns the features (variables).Batch order must be framed in the first column.

Details

Zero and NA values are not allowed. Optionally missing value can be imputed by functions implemented in dbnorm package. Input must be normalized and transformed beforehand.

Value

Original dataset in *csv* format together with the series of profile plot of the feature (variable) in all samples analyzed in the entire experiment provided by **Scatter** plot,**Violin** plot and **Density** plot compiled into a *pdf*.

References

Johnson et al. (2007) *http://www.ncbi.nlm.nih.gov/pubmed/16632515*
Leek et al. (2012) *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3307112/*

Examples

```
batch<- rep(gl(3,7,labels = c(1:3),1))
y<- matrix(rnorm(2100),nrow=21)
m<-data.frame(batch,y)
```

`profplotpcom` *Visualization of analytical heterogeneity on the profile of variables (features) in ComBat-Parametric -corrected data*

Description

profplotpcom allows you to adjust the data for batch effect using Parametric Empirical Bayes approach as described by *Johnson et al.(2007)* and via *sva* package as explained by *Leek et al.(2012)*, and informs you about the presence of across batch signal drift or batch effect in the treated data, determined by the shifted *probability density function* plots (*pdf* plots) of variables (features) detected in an experiment.
install.packages (c("ggplot2","ggfortify"))
source("https://bioconductor.org/biocLite.R")
biocLite(c("pcaMethods","impute","sva","limma","genefilter"))

Usage

```
profplotpcom(m)
```

Arguments

`m`	A data frame in which rows define the independent experiments (samples) and columns the features (variables).Batch order must be framed in the first column.

Details

Zero and NA values are not allowed. Optionally missing value can be imputed functions implemented in `dbnorm` package. Input must be normalized and transformed beforehand.

Value

Original dataset in *csv* format together with the series of profile plot for the variables(features) in the sample sets analyzed in the entire experiment provided by the **Scatter** plot,**Violin** plot and **Density** plot compiled into *pdf*.

References

Johnson et al. (2007) *http://www.ncbi.nlm.nih.gov/pubmed/16632515*
Leek et al. (2012) *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3307112/*

Examples

```
batch<- rep(gl(3,7,labels = c(1:3),1))
y<- matrix(rnorm(2100),nrow=21)
m<-data.frame(batch,y)
```

`profplotraw` *Visualization of analytical heterogeneity on the profile of variables (features) in raw data*

Description

This function informs you about the presence of across batch signal drift or batch effect in the raw data determined by the shifted *probability density function* plots (*pdf* plots) of variables (features) detected in an experiment.
install.packages (c("ggplot2", "ggfortify", "factoextra"))

Usage

```
profplotraw(m)
```

Arguments

`m`	A data frame in which rows define the independent experiments (samples) and columns the features (variables).Batch order must be framed in the first column.

Details

Zero and NA values are not allowed. Optionally missing value can be imputed by functions implemented in `dbnorm` package. Input must be normalized and transformed beforehand.

Value

Original dataset in *csv* format together with the series of profile plot for the variables(features) in the sample sets analyzed in the entire experiment provided by the **Scatter** plot,**Violin** plot and **Density** plot compiled into *pdf*.

Examples

```
batch<- rep(gl(3,7,labels = c(1:3),1))
y<- matrix(rnorm(2100),nrow=21)
m<-data.frame(batch,y)
```

`Visdbnorm` *Visualization of drift across batch normalization*

Description

This function performs batch effect adjustment via three statistical models implemented in the `dbnorm`, namely two-stage procedure two-stage procedure model as described by *Giordan (2013)* and/or empirical Bayes methods in two setting of parametric and non-parametric as described by *Johnson et al.(2007)* and in *sva* package by *Leek et al.(2012)*. Meanwhile, the graphical inferences in the context of unsupervised learning algorithms create visual inspection to inform users about the spatial separation of the sample sets analyzed in the different analytical runs alongside the distribution of variables (features) in the raw and treated datasets. This function is suggested for less than *2k* variables (features).
install.packages (c("ggplot2", "NormalizeMets","ggfortify", "factoextra","MASS","ber"))
source("https://bioconductor.org/biocLite.R")
biocLite(c("pcaMethods","impute","sva","limma","genefilter"))

Usage

```
Visdbnorm(f)
```

Arguments

`f`	A data frame in which rows define the independent experiments (samples) and columns the features (variables).Batch order must be framed in the first column.

Details

Zero and NA values are not allowed. Optionally missing value can be imputed by `emvf` or `emvd`, functions implemented in the **dbnorm** package. Input data must be normalized beforehand.

Value

Three datasets, adjusted by either of applied statistical algorithms prepared in *csv* and together with series of plot such as **PCA** plot and **Scree plot** compiled into *pdf* are saved in the working directory. **RLA** plots are represented in the Viewer panel of *rstudio*.

References

M.Giordan (2013) *https://link.springer.com/article/10.1007/s12561-013-9081-1*
Johnson et al. (2007) *http://www.ncbi.nlm.nih.gov/pubmed/16632515*
Leek et al. (2012) *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3307112/*

Examples

```
batch<- rep(gl(3,7,labels = c(1:3)),1)
y<- matrix(rnorm(2100),nrow=21)
f<-data.frame(batch,y)
```

Index