Package 'NOISeq'

March 26, 2013

Type Package

Title Exploratory analysis and differential expression for RNA-seq data

Version 1.1.5

Date 2012-08-29

Author Sonia Tarazona, Pedro Furio-Tari, Alberto Ferrer and Ana Conesa

Maintainer Sonia Tarazona <starazona@cipf.es>

Depends R (>= 2.13.0), methods, Biobase (>= 2.13.11)

biocViews
Bioinformatics, RNAseq, DifferentialExpression,Visualization, HighThroughputSequencing

Description Analysis of RNA-seq expression data or other similar kind of data. Exploratory plots to evualuate saturation, count distribution, expression per chromosome, type of detected features, features length, etc. Differential expression between two experimental conditions with no parametric assumptions.

License Artistic-2.0

LazyLoad yes

R topics documented:

Biodetection 2
CD 3
CountsBio 3
Data2Save 4
Data_Exploration 5
degenes 6
Differential expression plots 7
DLBio 9
example 10
Exploratory_Plots 10
Marioni 11
myCounts 11
noiseq 12
Normalization 14
Output 15
readData 16
Saturation 17

Index 19

Biodetection *Biodetection class*

Description

Biodetection class generated from dat() function with type="biodetection". This object contains the percentage of each biological class (e.g. biotype) in the genome (i.e. in the whole set of features provided), the corresponding percentage detected by the sample and the percentage of the biotype within the sample.

Usage

```
## S4 method for signature 'Biodetection'
explo.plot(object)
## S4 method for signature 'Biodetection'
dat2save(object)
```

Arguments

object	Object generated from dat() function.

Slots/List Components

Objects of this class contain (at least) the following list components:

dat: List containing the information generated by dat() function. This list has the following elements:

table,table2: Table(s) containing the results for the selected sample(s).

params: Maximum values for the Y-axes scale in the plot generated from these data (explo.plot).

samples: Names of the sample(s) evaluated by dat().

Methods

This class has an specific show method in order to work and print a summary of the elements which are contained and a dat2save method to save the relevant information in an object cleanly. It also has an explo.plot method to plot the data contained in the object.

Author(s)

Sonia Tarazona

CD *CD class*

Description

CD class generated from dat() function with type="cd". This object contains the cumulative distributions of reads for the selected samples.

Usage

```
## S4 method for signature 'CD'
explo.plot(object)
## S4 method for signature 'CD'
dat2save(object)
```

Arguments

object	Object generated from dat() function.

Slots/List Components

Objects of this class contain (at least) the following list components:

dat: Data.frame containing the information generated by dat() function: percentage of detected features and cumulative percentage of reads associated to these features in selected samples.

Methods

This class has an specific show method in order to work and print a summary of the elements which are contained and a dat2save method to save the relevant information in an object cleanly. It also has an explo.plot method to plot the data contained in the object.

Author(s)

Sonia Tarazona

CountsBio *CountsBio class*

Description

CountsBio class generated from dat() function with type="countsbio". This object contains the count distribution for each biological group and also for increasing sequencing depths.

Usage

```
## S4 method for signature 'CountsBio'
explo.plot(object, toplot = 1, samples = NULL, ylim = NULL)
## S4 method for signature 'CountsBio'
dat2save(object)
```

Arguments

object	Object generated from dat() function.
toplot	This parameter indicates which biological group is to be plotted. It may be a number or a text with the name of the biological group. If toplot=1 (or "global"), a global plot with all the biological groups will be generated.
samples	The samples to be plotted. If NULL, the two first samples are plotted because the plot for this object only admit a maximum of two samples.
ylim	Range for the Y axis. If NULL (default), an appropriate range is computed.

Slots/List Components

Objects of this class contain (at least) the following list components:

dat: List containing the information generated by dat() function. This list has the following elements:

result: List containing for all the biological classes (and also a global class with all of them together) the data to be plotted for each selected sample.

bionum: List containing for all the biological classes (and also a global class with all of them together) the number of features for that group.

depth: List containing for each selected sample the increasing sequencing depths to be plotted.

quart: List containing for each selected sample a matrix summarizing the quartiles of the distributions of counts for each biotype at that total sequencing depth in each selected sample.

Methods

This class has an specific show method in order to work and print a summary of the elements which are contained and a dat2save method to save the relevant information in an object cleanly. It also has an explo.plot method to plot the data contained in the object.

Author(s)

Sonia Tarazona

Data2Save -

Description

-

Value

The dat2save() function takes the object generated by dat() and creates a new one more understandable.

Author(s)

Sonia Tarazona

See Also

readData, addData, dat, explo.plot.

Examples

```
## Load the input object with the expression data and the annotations
data(myCounts)

## Generating data for the plot "biodetection" and samples in columns 3 and 4 of expression data
mydata2plot = dat(mydata, type = "biodetection", k = 0, selection = c(3,4))

## Save the relevant information cleanly
aux <- dat2save(mydata2plot)
```

Data_Exploration *Exploration of expression data.*

Description

Take the expression data and the feature annotations to generate the results that will be used for the exploratory plots (explo.plot) or saved by the user to perform other analyses.

Usage

```
dat(input, type = c("biodetection","cd","countsbio","DLbio","saturation"),
                selection=c(1,2), k = 0, ndepth = 5, newdetections = TRUE)
```

Arguments

input	Object of eSet class with expression data and optional annotation.
type	Type of plot for which the data are to be generated. It can be one of: "biodetection","cd","countsbio","DLbio","saturation". newdetections = TRUE
selection	Vector containing the number or names of the columns (samples) from the expression data to be plotted. Depending on the chosen type, a different maximum number of samples is allowed. For "biodetection", the number of samples may be 1 or 2; for "cd", 2; for "countsbio", "DLbio" and "saturation", the data may be generated for all the available samples.
k	A feature is considered to be detected if the corresponding number of read counts is > k. By default, k = 0. This parameter is used by all types except "cd".
ndepth	Number of different sequencing depths to be simulated and plotted. By default, ndepth = 5. This parameter is only used by types "countsbio","DLbio" and "saturation".
newdetections	If TRUE, a second Y-axis is drawn for new detections per million of new sequenced reads. This parameter is only used by type "saturation". By default, newdetections = TRUE. If samples to be plotted are more than two, this option is disabled.

Value

dat() function returns an S4 object to be used by explo.plot() or to be converted into a more friendly formatted object by the dat2save() function.

Author(s)

Sonia Tarazona

See Also

readData,addData,dat2save,explo.plot

Examples

```
## Load the input object with the expression data and the annotations
data(myCounts)

## Generating data for the plot "biodetection" and samples in columns 3 and 4 of expression data
mydata2plot = dat(mydata, type = "biodetection", k = 0, selection = c(3,4))

## Generating the corresponding plot
explo.plot(mydata2plot)
```

---

degenes *Recover the differencially expressed features*

Description

Recover differencially expressed features for given the threshold from noiseq output object.

Usage

```
degenes(object, q = 0.9, M = NULL)
```

Arguments

object	Object of class Output.
q	Value for the probability threshold.
M	String indicating some options for the differentially expressed genes to select. It can take one of these values: "up" (up-regulated in condition 1), "down" (down-regulated in condition 1), or NULL (all differentially expressed genes).

Value

A matrix containing the differencially expressed features is returned.

Author(s)

Sonia Tarazona

References

Bullard J.H., Purdom E., Hansen K.D. and Dudoit S. (2010) Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments. *BMC Bioinformatics* 11(1):94+.

Mortazavi A., Williams B.A., McCue K., Schaeer L. and Wold B. (2008) Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nature Methods* 5(7):621-628.

Robinson M.D. and Oshlack A. (2010) A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* 11(3):R25+.

Marioni, J.C. and Mason, C.E. and Mane, S.M. and Stephens, M. and Gilad, Y. (2008) RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, **18**: 1509–1517.

See Also

readData, noiseq.

Examples

```
## Load the object mynoiseq generated by computing differential expression probability with noiseq() on Marioni's data:
data(noiseq)

## Third, use degenes() function to extract differentially expressed features:
mynoiseq.deg = degenes(mynoiseq, q = 0.8, M = NULL)
```

Differential expression plots

Plot to compare expression values for two conditions or to represent differential expression statistics

---

Description

Function to represent three possible plots: one to compare the expression values in each condition, one also to compare the expression values in each condition but according to the chromosome and position within the chromosome, and the other one with the (M,D) statistics. In all three plots, the differentially expressed features may be highlighted.

Usage

```
DE.plot(output, q = NULL, graphic = c("MD","expr","chrom","distr"), pch = 20, cex = 0.5, col = 1, pch.sel =
```

Arguments

output	Object of class Output.
q	Probability of differential expression threshold to determine differentially expressed features.
graphic	String indicating which kind of plot is to be generated. If "expr", the feature expression values are depicted. If "MD", the values for the (M,D) statistics when comparing both conditions are used. If "chrom", the feature expression values are depicted across their positions in the chromosomes (if chromosome information has been provided).

pch, cex, col,... Graphical parameters as in any other R plot. See par. They do not apply for graphic="chrom".

pch.sel, cex.sel, col.sel
pch, cex and col, respectively, to represent differentially expressed features. They do not apply for graphic="chrom".

log.scale If TRUE, log2(data+K) values are depicted instead of the expression data in the Output object. K is an appropriate constant to avoid negative values. It does not apply for graphic="MD".

chromosomes Character vector indicating the chromosomes to be plotted. If NULL, all chromosomes are plotted. It only applies for graphic="chrom" and graphic="distr". For graphic="chrom", the chromosomes are plotted in the given order. In some cases (e.g. chromosome names are character strings), it is very convenient to specify the order although all chromosomes are being plotted. For graphic="distr", the chromosomes are plotted according to the number of features they contain (from the highest number to the lowest).

join If FALSE, each chromosome is depicted in a separate line. If TRUE, all the chromosomes are depicted in the same line, consecutively (useful for prokaryote organisms). It only applies for graphic="chrom".

Author(s)

Sonia Tarazona

References

Bullard J.H., Purdom E., Hansen K.D. and Dudoit S. (2010) Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments. *BMC Bioinformatics* 11(1):94+.

Mortazavi A., Williams B.A., McCue K., Schaeer L. and Wold B. (2008) Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nature Methods* 5(7):621-628.

Robinson M.D. and Oshlack A. (2010) A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* 11(3):R25+.

Marioni, J.C. and Mason, C.E. and Mane, S.M. and Stephens, M. and Gilad, Y. (2008) RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, **18**: 1509–1517.

See Also

readData, noiseq, degenes.

Examples

```
## We load the object generated after running noiseq on Marioni's data
data(noiseq)

## Third, plot the expression values for all genes and highlighting the differentially expressed genes
DE.plot(mynoiseq, q = 0.8, graphic = "expr", log.scale = TRUE)
DE.plot(mynoiseq, q = 0.8, graphic = "MD", ylim = c(0,50))
DE.plot(mynoiseq, chromosomes = c(1,2), log.scale = TRUE,join = FALSE, q = 0.8, graphic = "chrom")
DE.plot(mynoiseq, chromosomes = NULL, q = 0.8, graphic = "distr")
```

DLBio *DLBio class*

Description

DLBio class generated from dat() function with type="DLbio". This object contains the median length of the detected features for each biotype at increasing sequencing depths.

Usage

```
## S4 method for signature 'DLBio'
explo.plot(object, samples = NULL, toplot = "protein_coding",ylim = NULL)
## S4 method for signature 'DLBio'
dat2save(object)
```

Arguments

object	Object generated from dat() function.
toplot	This parameter indicates which biological group is to be plotted. It may be a number or a text with the name of the biological group.
samples	The samples to be plotted. If NULL, the two first samples are plotted.
ylim	Range for Y axis. If NULL (default), an appropriate range is computed.

Slots/List Components

Objects of this class contain (at least) the following list components:

dat: List containing the information generated by dat() function. This list has the following elements:

result: A list with as many elements as biological groups. Each biological group is another list with all the selected samples, and for each sample, the median length of the detected features at each sequencing depth.

bionum: A list with as many elements as biological groups, and for each one, the number of features in that biological group.

depth: A list with as many elements as selected samples. For each one, the increasing sequencing depths to be plotted. The last sequencing depth is the real one and the others are simulated from it using the multinomial distribution.

biolength: A list with as many elements as biological groups. For each one, the median length of the features in that group.

Methods

This class has an specific show method in order to work and print a summary of the elements which are contained and a dat2save method to save the relevant information in an object cleanly. It also has an explo.plot method to plot the data contained in the object.

Author(s)

Sonia Tarazona

example *Example of objects used and created by the NOISeq package*

Description

This is a quick view of the objects generated by the package. To take a look, see the usage information. These objects have been created from Marioni's reduce dataset (only chromosomes I to IV).

Usage

```
# To load the object myCounts generated by the readData() function from R objects containing expression data,
data(myCounts)

# To load the object generated after running the noiseq() function to compute differential expression:
data(noiseq)
```

References

Marioni, J.C. and Mason, C.E. and Mane, S.M. and Stephens, M. and Gilad, Y. (2008) RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, **18**: 1509–1517.

Exploratory_Plots *Exploratory plots for sequencing data.*

Description

Different types of plots showing the biological classification for detected features, the comparison of count distributions, the distribution of counts or the length for detected features, the saturation over different sequencing depths, etc.

Value

The explo.plot() function takes the object generated by dat() and draws the corresponding plot.

Author(s)

Sonia Tarazona

See Also

readData, addData, dat.

Examples

```
## Load the input object with the expression data and the annotations
data(myCounts)

## Generating data for the plot "biodetection" and samples in columns 3 and 4 of expression data
mydata2plot = dat(mydata, type = "biodetection", k = 0, selection = c(3,4))

## Generating the corresponding plot
explo.plot(mydata2plot)
```

Marioni *Marioni's dataset*

Description

This is a reduced version for the RNA-seq count data from Marioni et al. (2008) along with additional annotation such as gene biotype, gene length, chromosome, start position and end position for genes in chromosomes I to IV. The expression data consists of 10 samples from kidney and liver tissues. There are five technical replicates (lanes) per tissue.

Usage

```
data(Marioni)
```

References

Marioni, J.C. and Mason, C.E. and Mane, S.M. and Stephens, M. and Gilad, Y. (2008) RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, **18**: 1509–1517.

---

myCounts *Class myCounts*

Description

This is the main class which contains the information needed to do the different analyses.

Extends

Class eSet (package 'Biobase').

Quick View

This object will contain the expression data and further information needed to do the exploratory analysis or the normalization such as the length, biotypes, chromosomes and positions for each feature.

Internally, the data is stored as follows:

As myCounts derives from eSet, we have used the slot assayData to store all the expression data, phenoData to store the factors with the conditions, featureData which will contain the variables Length, Biotype, Chromosome, Start Position, End Position for each feature. It has been used the slot experimentData derived from MIAME-class which will contain the type of replicates (biological replicates, technical replicates or no replicates at all).

Author(s)

Sonia Tarazona

See Also

If you need further information to know the methods that can be used, see eSet, AnnotatedDataFrame-class, MIAME-class.

---

noiseq *Differential expression method*

Description

noiseq computes differential expression between two experimental conditions from read count data (e.g. RNA-seq).

Usage

noiseq(input, k = 0.5, norm = c("rpkm","uqua","tmm","n"), replicates = c("technical","biological","no"), factor=NULL, conditions=NULL, pnr = 0.2, nss = 5, v = 0.02, lc = 1)

Arguments

input	Object of eSet class coming from readData function or other R packages such as DESeq.
factor	A string indicating the name of factor whose levels are the conditions to be compared.
conditions	A vector containing the two conditions to be compared by the differential expression algorithm (needed when the factor contains more than 2 different conditions).
replicates	In this argument, the type of replicates to be used is defined. Technical, biological or none. By default, technical replicates option is chosen.
k	Counts equal to 0 are replaced by k. By default, k = 0.5.
norm	Normalization method. It can be one of "rpkm" (default), "uqua" (upper quartile), "tmm" (trimmed mean of M) or "n" (no normalization).
lc	Length correction is done by dividing expression by length^lc. By default, lc = 1.

pnr — Percentage of the total reads used to simulated each sample when no replicates are available. By default, pnr = 0.2.

nss — Number of samples to simulate for each condition (nss>= 2). By default, nss = 5.

v — Variability in the simulated sample total reads. By default, v = 0.02. Sample total reads is computed as a random value from a uniform distribution in the interval [(pnr-v)*sum(counts), (pnr+v)*sum(counts)]

Value

The function returns an object of class Output

Author(s)

Sonia Tarazona

References

Bullard J.H., Purdom E., Hansen K.D. and Dudoit S. (2010) Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments. *BMC Bioinformatics* 11(1):94+.

Mortazavi A., Williams B.A., McCue K., Schaeer L. and Wold B. (2008) Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nature Methods* 5(7):621-628.

Robinson M.D. and Oshlack A. (2010) A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* 11(3):R25+.

Marioni, J.C. and Mason, C.E. and Mane, S.M. and Stephens, M. and Gilad, Y. (2008) RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, **18**: 1509–1517.

See Also

readData.

Examples

```
## Load the input object from Marioni's data as returned by readData()
data(myCounts)

## Computing differential expression probability on RPKM-normalized data by NOISeq-real using factor "Tissue"
mynoiseq = noiseq(mydata, k = 0.5, norm = "rpkm", replicates = "technical", factor="Tissue", pnr = 0.2, nss = 5, v = 0.

## Computing differential expression probability on Upper Quartile normalized data by NOISeq-real using factor "Tissu
mynoiseq.uqua = noiseq(mydata, k = 0.5, norm = "uqua", replicates = "technical", factor="TissueRun", conditions = c(
pnr = 0.2, nss = 5, v = 0.02, lc = 1)
```

Normalization *Normalization methods*

Description

Normalization procedures such as RPKM (Mortazavi et al., 2008), Upper Quartile (Bullard et al., 2010) and TMM (Trimmed Mean of M) (Robinson and Oshlack, 2010). These normalization functions are used within the noiseq function but may be also used by themselves to normalize a dataset.

Usage

uqua(datos, long = 1000, lc = 1, k = 0)
rpkm(datos, long = 1000, lc = 1, k = 0)
tmm(datos, long = 1000, lc = 1, k = 0, refColumn = 1, logratioTrim = 0.3, sumTrim = 0.05, doWeighting = TRU

Arguments

datos	Matrix containing the read counts for each sample.
long	Numeric vector containing the length of the features. If long == 1000, no length correction is applied.
lc	Correction factor for length normalization. This correction is done by dividing the counts vector by length^lc. By default, lc = 1.
k	Counts equal to 0 are changed to k in order to avoid indeterminations when applying logarithms, for instance. By default, k = 0.5
refColumn	Column to use as reference (only needed for tmm function).
logratioTrim	Amount of trim to use on log-ratios ("M" values) (only needed for tmm function).
sumTrim	Amount of trim to use on the combined absolute levels ("A" values) (only needed for tmm function).
doWeighting	Logical, whether to compute (asymptotic binomial precision) weights (only needed for tmm function).
Acutoff	Cutoff on "A" values to use before trimming (only needed for tmm function).

Details

tmm normalization method was taken from *edgeR* package (Robinson et al., 2010).

Although Upper Quartile and TMM methods themselves do not correct for the length of the features, these functions in NOISeq allow users to combine the normalization procedures with an additional length correction whenever the length information is available.

Author(s)

Sonia Tarazona

References

Bullard J.H., Purdom E., Hansen K.D. and Dudoit S. (2010) Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments. *BMC Bioinformatics* 11(1):94+.

Mortazavi A., Williams B.A., McCue K., Schaeer L. and Wold B. (2008) Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nature Methods* 5(7):621-628.

Robinson M.D. and Oshlack A. (2010) A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* 11(3):R25+.

Robinson M.D., McCarthy D.J. and Smyth G.K. (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26(1):139-140.

Examples

```
## Simulate some count data and the features length
datasim = matrix(sample(0:100, 2000, replace = TRUE), ncol = 4)
lengthsim = sample(100:1000, 500)

## RPKM normalization
myrpkm = rpkm(datasim, long = lengthsim, lc = 1, k = 0)

## Upper Quartile normalization, dividing normalized data by the square root of the features length and replacing counts
myuqua = uqua(datasim, long = lengthsim, lc = 0.5, k = 1)

## TMM normalization with no length correction
mytmm = tmm(datasim, long = 1000, lc = 1, k = 0)
```

Output *Output class of NOISeq*

Description

Output object containing the results from differential expression analysis by NOISeq.

Slots/List Components

Objects of this class contain (at least) the following list components:

comparison: String indicating the two experimental conditions being compared and the sense of the comparison.

factor: String indicating the factor chosen to compute the differential expression.

k: Value to replace zeroes in orden to avoid inderminations when computing logarithms.

lc: Correction factor for length normalization. Counts are divided by length^lc.

method: Normalization method chosen. It can be one of "rpkm" (default), "uqua" (Upper Quartile), "tmm" (Trimmed Mean of M) or "n" (no normalization).

replicates: Type of replicates: "technical" for technical replicates and "biological" for biological ones.

results: R data frame containing the differential expression results, where each row corresponds to a feature. The columns are: Expression values for each condition to be used by NOISeq (the

columns names are the levels of the factor); differential expression statistics (columns "M" and "D"); probability of differential expression ("prob"); "ranking", which is a summary statistic of "M" and "D" values equal to -sign(M)*sqrt(M^2 + D^2), than can be used for instance in gene set enrichment analysis; "length" of each feature (if provided); chromosome where the feature is ("Chrom"), if provided; start and end position of the feature within the chromosome ("GeneStart", "GeneEnd"), if provided.

nss: Number of samples to be simulated for each condition (only when there are not replicates available).

pnr: Percentage of the total sequencing depth to be used in each simulated replicate (only when there are not replicates available). If, for instance, pnr = 0.2 , each simulated replicate will have 20% of the total reads of the only available replicate in that condition.

v: Variability of the size of each simulated replicate (only used by NOISeq-sim).

Methods

This class has an specific show method in order to work and print a summary of the elements which are contained.

Author(s)

Sonia Tarazona

---

readData *Create an object of eSet class*

Description

Create an object of eSet class to be used by NOISeq functions by taking R matrix or data.frame objects.

Usage

readData(data, factors, length = NULL, biotype = NULL, chromosome = NULL)
addData(data, length = NULL, biotype = NULL, chromosome = NULL, factors = NULL)

Arguments

data	The matrix or data.frame containing the counts (or expression data) for each feature and condition. Rows are features and columns are conditions.
length	Optional argument. Vector containing the length of each feature. The names of the vector must be the feature names or ids with the same type of identifier used in data.
factors	A data.frame containing the different conditions of each sample included in the data object.
biotype	Optional argument. Vector containing the biological classification of each feature. The names of the vector must be the feature names or ids with the same type of identifier used in data.
chromosome	Optional argument. A matrix or data.frame containing the chromosome, start position and end position of each feature. The row names must be the feature names or ids with the same type of identifier used in data.

Value

It returns an object of eSet class myCounts with all the information defined and ready to be used.

Author(s)

Sonia Tarazona

References

Marioni, J.C. and Mason, C.E. and Mane, S.M. and Stephens, M. and Gilad, Y. (2008) RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, **18**: 1509–1517.

Examples

```
# Load an object containing the information explained above
data(Marioni)

# Create the object with the data
mydata <- readData(data=mycounts, biotype=mybiotypes, chromosome=mychroms, factors=myfactors)

# Add length annotation to the existing data object
mydata <- addData(mydata, length=mylength)
```

Saturation *Saturation class*

Description

Saturation class generated from dat() function with type="biodetection". This object contains the number of detected features per biotype at increasing sequencing depths and also the new detections when increasing the sequencing depth in one million of reads.

Usage

```
## S4 method for signature 'Saturation'
explo.plot(object, toplot = 1, samples = NULL,ylim = NULL, yrightlim = NULL)
## S4 method for signature 'Saturation'
dat2save(object)
```

Arguments

object	Object generated from dat() function.
toplot	This parameter indicates which biological group is to be plotted. It may be a number or a text with the name of the biological group. If toplot=1 (or "global"), a global plot with all the biological groups will be generated.
samples	The samples to be plotted. If NULL, all the samples are plotted for Saturation object.
ylim	Range for Y axis (on the left-hand side of the plot). If NULL (default), an appropriate range is computed.
yrightlim	Range for Y right-axis (on the right-hand side of the plot). If NULL (default), an appropriate range is computed.

Slots/List Components

Objects of this class contain (at least) the following list components:

dat: List containing the information generated by dat() function. This list has the following elements:

saturation: List containing for all the biological classes (and also a global class with all of them together) the saturation data to be plotted for each selected sample (in Y left axis).

bionum: List containing for all the biological classes (and also a global class with all of them together) the number of features for that group.

depth: List containing for each selected sample the increasing sequencing depths to be plotted.

newdet: List containing for all the biological classes (and also a global class with all of them together) the new detection data to be plotted for each selected sample (in Y right axis).

Methods

This class has an specific show method in order to work and print a summary of the elements which are contained and a dat2save method to save the relevant information in an object cleanly. It also has an explo.plot method to plot the data contained in the object.

Author(s)

Sonia Tarazona

Index

∗Topic **classes**
  Biodetection, 2
  CD, 3
  CountsBio, 3
  DLBio, 9
  myCounts, 11
  Output, 15
  Saturation, 17
∗Topic **datasets**
  example, 10
  Marioni, 11

addData, *5, 6*, *10*
addData *(*readData*)*, 16

Biodetection, 2
Biodetection-class *(*Biodetection*)*, 2

CD, 3
CD-class *(*CD*)*, 3
CountsBio, 3
CountsBio-class *(*CountsBio*)*, 3

dat, *5*, *10*
dat *(*Data_Exploration*)*, 5
dat2save *(*Data2Save*)*, 4
dat2save,Biodetection-method *(*Biodetection*)*, 2
dat2save,CD-method *(*CD*)*, 3
dat2save,CountsBio-method *(*CountsBio*)*, 3
dat2save,DLBio-method *(*DLBio*)*, 9
dat2save,Saturation-method *(*Saturation*)*, 17
Data2Save, 4
Data_Exploration, 5
DE.plot *(*Differential expression plots*)*, 7
degenes, 6, *8*
Differential expression plots, 7
DLBio, 9
DLBio-class *(*DLBio*)*, 9

example, 10
explo.plot, *5*
explo.plot *(*Exploratory_Plots*)*, 10
explo.plot,Biodetection-method *(*Biodetection*)*, 2
explo.plot,CD-method *(*CD*)*, 3
explo.plot,CountsBio-method *(*CountsBio*)*, 3
explo.plot,DLBio-method *(*DLBio*)*, 9
explo.plot,Saturation-method *(*Saturation*)*, 17
Exploratory_Plots, 10

Marioni, 11
mybiotypes *(*Marioni*)*, 11
mychroms *(*Marioni*)*, 11
myCounts, 11, *17*
mycounts *(*Marioni*)*, 11
myCounts-class *(*myCounts*)*, 11
mydata *(*example*)*, 10
myfactors *(*Marioni*)*, 11
mylength *(*Marioni*)*, 11
mynoiseq *(*example*)*, 10

noiseq, *7, 8*, 12
Normalization, 14

Output, *6–8*, *13*, 15
Output-class *(*Output*)*, 15

par, *8*

readData, *5–8*, *10*, *12, 13*, 16
rpkm *(*Normalization*)*, 14

Saturation, 17
Saturation-class *(*Saturation*)*, 17
show,Biodetection-method *(*Biodetection*)*, 2
show,CD-method *(*CD*)*, 3
show,CountsBio-method *(*CountsBio*)*, 3
show,DLBio-method *(*DLBio*)*, 9
show,Output-method *(*Output*)*, 15
show,Saturation-method *(*Saturation*)*, 17

tmm *(*Normalization*)*, 14

uqua *(*Normalization*)*, 14