

Package ‘isomiRs’

April 15, 2020

Version 1.14.0

Date 2019-04-03

Type Package

Title Analyze isomiRs and miRNAs from small RNA-seq

Description Characterization of miRNAs and isomiRs, clustering and differential expression.

biocViews miRNA, RNASeq, DifferentialExpression, Clustering, ImmunoOncology

BugReports <https://github.com/lpantano/isomiRs/issues>

Suggests knitr, org.Mm.eg.db, targetscan.Hs.eg.db, pheatmap, BiocStyle, testthat

Depends R (>= 3.5), DiscrMiner, SummarizedExperiment

Imports AnnotationDbi, assertive.sets, BiocGenerics, Biobase, broom, cluster, cowplot, DEGREport, DESeq2, IRanges, dplyr, GenomicRanges, gplots, ggplot2, gtools, gridExtra, grid, grDevices, graphics, GGally, limma, methods, RColorBrewer, readr, reshape, rlang, stats, stringr, S4Vectors, tidyr, tibble

LazyData true

Maintainer Lorena Pantano <lorena.pantano@gmail.com>

License MIT + file LICENSE

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 6.1.1

Encoding UTF-8

git_url <https://git.bioconductor.org/packages/isomiRs>

git_branch RELEASE_3_10

git_last_commit 1e5d3d5

git_last_commit_date 2019-10-29

Date/Publication 2020-04-14

Author Lorena Pantano [aut, cre],
Georgia Escaramis [aut]

R topics documented:

isomiRs-package	2
counts	3
dat286.long	3
design	4
ego	4
findTargets	5
gene_ex_rse	5
isoAnnotate	6
isoCounts	6
isoDE	7
IsomirDataSeq-class	8
IsomirDataSeqFromFiles	9
isoNetwork	11
isoNorm	12
isoPlot	13
isoPlotNet	14
isoPlotPosition	14
isoPLSDA	15
isoPLSDAplot	16
isoSelect	17
isoTop	18
ma_ex	19
mirData	19
mirna2targetscan	20
mirna_ex_rse	20
mirTritation	21
updateIsomirDataSeq	21
Index	22

isomiRs-package	<i>isomiRs</i>
-----------------	----------------

Description

Characterization of miRNAs and isomiRs, clustering and differential expression.

Author(s)

Maintainer: Lorena Pantano <lorena.pantano@gmail.com>

Authors:

- Georgia Escaramis

See Also

Useful links:

- Report bugs at <https://github.com/lpantano/isomiRs/issues>

counts	<i>Accessors for the count matrix of a IsomirDataSeq object.</i>
--------	--

Description

The counts slot holds the count data as a matrix of non-negative integer count values, one row for each isomiR, and one column for each sample. The normalized matrix can be obtained by using the parameter `norm=TRUE`.

Usage

```
counts.IsomirDataSeq(object, norm = FALSE)

## S4 method for signature 'IsomirDataSeq'
counts(object, norm = FALSE)

## S4 replacement method for signature 'IsomirDataSeq,matrix'
counts(object) <- value
```

Arguments

object	A IsomirDataSeq object.
norm	Boolean, return log2-normalized counts.
value	An integer matrix.

Value

`base::matrix` with raw or normalized count data.

Author(s)

Lorena Pantano

Examples

```
data(mirData)
head(counts(mirData))
```

dat286.long	<i>Data frame containing mirna from Argyropoulos's paper</i>
-------------	--

Description

Argyropoulos, Christos, et al. "Modeling bias and variation in the stochastic processes of small RNA sequencing." Nucleic Acids Research (2017).

Usage

```
dat286.long
```

Format

mirna expression data in long format.

design	<i>Accessors for the 'design' slot of a IsomirDataSeq object.</i>
--------	---

Description

The design holds the R formula which expresses how the counts depend on the variables in colData. See [IsomirDataSeq](#) for details.

Usage

```
## S4 method for signature 'IsomirDataSeq'
design(object)

## S4 replacement method for signature 'IsomirDataSeq,formula'
design(object) <- value
```

Arguments

object	A IsomirDataSeq object.
value	A formula to pass to DESeq2.

Value

design for the experiment

Examples

```
data(mirData)
design(mirData) <- formula(~ 1)
```

ego	<i>enrichResult class</i>
-----	---------------------------

Description

enrichResult class

Usage

```
ego
```

Format

enrichResult class with the output of: `ego <- enrichGO(row.names(assay(gene_ex_rse, "norm")), org.Mm.eg.db, "ENSEMBL", ont = "BP")`

 findTargets

Find miRNAs target using mRNA/miRNA expression

Description

This function creates a matrix with rows (genes) and columns (mirnas) with values indicating if miRNA-gene pair is target according putative targets and negative correlation of the expression of both molecules.

Usage

```
findTargets(mirna_rse, gene_rse, target, summarize = "group",
            min_cor = -0.6)
```

Arguments

mirna_rse	SummarizedExperiment with miRNA information. See details.
gene_rse	SummarizedExperiment with gene information. See details.
target	Data.frame with two columns: gene and miRNA.
summarize	Character column name in colData(rse) to use to group samples and compare between miRNA/gene expression.
min_cor	Numeric cutoff for correlation value that will be use to consider a miRNA-gene pair as valid.

Value

mirna-gene matrix

Examples

```
data(isoExample)
mirna_ma <- data.frame(gene = names(gene_ex_rse)[1:20],
                      mir = names(mirna_ex_rse))
corMat <- findTargets(mirna_ex_rse, gene_ex_rse, mirna_ma)
```

 gene_ex_rse

Data frame containing gene expression data

Description

Data frame containing gene expression data

Usage

```
gene_ex_rse
```

Format

gene expression data with 18 samples: example of a time series data

isoAnnotate	Annotate the rawData of the <i>IsomirDataSeq</i> object
-------------	---

Description

Get the sequence and the name information for each isomiR, and the importance value (`isomir_reads/mirna_reads`) for each sample.

Usage

```
isoAnnotate(ids)
```

Arguments

`ids` Object of class *IsomirDataSeq*.

Details

`edit_mature_position` represents the position at the mature sequence + nucleotide at reference + nucleotide at isomiR.

Value

[data.frame](#) with the sequence, isomir name, and importance for each sample and isomiR.

Examples

```
data(mirData)
head(isoAnnotate(mirData))
```

isoCounts	Create count matrix with different summarizing options
-----------	--

Description

This function collapses isomiRs into different groups. It is a similar concept than how to work with gene isoforms. With this function, different changes can be put together into a single miRNA variant. For instance all sequences with variants at 3' end can be considered as different elements in the table or analysis having the following naming `hsa-miR-124a-5p.iso.t3:AAA`.

Usage

```
isoCounts(ids, ref = FALSE, iso5 = FALSE, iso3 = FALSE,
  add = FALSE, snv = FALSE, seed = FALSE, all = FALSE, minc = 1,
  mins = 1, merge_by = NULL)
```

Arguments

ids	Object of class IsomirDataSeq .
ref	Differentiate reference miRNA from rest.
iso5	Differentiate trimming at 5 miRNA from rest.
iso3	Differentiate trimming at 3 miRNA from rest.
add	Differentiate additions miRNA from rest.
snv	Differentiate nt substitution miRNA from rest.
seed	Differentiate changes in 2-7 nts from rest.
all	Differentiate all isomiRs.
minc	Int minimum number of isomiR sequences to be included.
mins	Int minimum number of samples with number of sequences bigger than minc counts.
merge_by	Column in coldata to merge samples into a single column in counts. Useful to combine technical replicates.

Details

You can merge all isomiRs into miRNAs by calling the function only with the first parameter `isoCounts(ids)`. You can get a table with isomiRs altogether and the reference miRBase sequences by calling the function with `ref=TRUE`. You can get a table with 5' trimming isomiRs, miRBase reference and the rest by calling with `isoCounts(ids, ref=TRUE, iso5=TRUE)`. If you set up all parameters to `TRUE`, you will get a table for each different sequence mapping to a miRNA (i.e. all isomiRs).

Examples for the naming used for the isomiRs are at http://seqcluster.readthedocs.org/mirna_annotation.html#mirna-annotation.

Value

[IsomirDataSeq](#) object with new count table. The count matrix can be access with `counts(ids)`.

Examples

```
data(mirData)
ids <- isoCounts(mirData, ref=TRUE)
head(counts(ids))
# taking into account isomiRs and reference sequence.
ids <- isoCounts(mirData, ref=TRUE, minc=10, mins=6)
head(counts(ids))
```

Description

This function does differential expression analysis with [DESeq2::DESeq2-package](#) using the specific formula. It will return a [DESeq2::DESeqDataSet](#) object.

Usage

```
isoDE(ids, formula = NULL, ...)
```

Arguments

<code>ids</code>	Object of class <code>IsomirDataSeq</code> .
<code>formula</code>	Formula used for DE analysis.
<code>...</code>	Options to pass to <code>isoCounts()</code> including <code>ref</code> , <code>iso5</code> , <code>iso3</code> , <code>add</code> , <code>subs</code> and <code>seed</code> parameters.

Details

First, this function collapses all isomiRs in different types. Read more at `isoCounts()` to know the different options available to collapse isomiRs.

After that, `DESeq2::DESeq2-package` is used to do differential expression analysis. It uses the count matrix and design experiment stored at (`counts(ids)` and `colData(ids)`) `IsomirDataSeq` object to construct a `DESeq2::DESeqDataSet` object.

Value

`DESeq2::DESeqDataSet` object. To get the differential expression isomiRs, use `DESeq2::results()` from `DESeq2` package. This allows to ask for different contrast without calling again `isoDE()`. Read results manual to know how to access all the information.

Examples

```
data(mirData)
ids <- isoCounts(mirData, minc=10, mins=6)
dds <- isoDE(mirData, formula=~condition)
```

`IsomirDataSeq-class` *Class that contains all isomiRs annotation for all samples*

Description

The `IsomirDataSeq` is a subclass of `SummarizedExperiment`. used to store the raw data, intermediate calculations and results of an miRNA/isomiR analysis. This class stores all raw isomiRs data for each sample, processed information, summary for each isomiR type, raw counts, normalized counts, and table with experimental information for each sample.

Details

`IsomirDataSeqFromFiles` creates this object using `seqbuster` output files.

Methods for this objects are `isomiRs::counts()` to get count matrix and `isomiRs::isoSelect()` for miRNA/isomiR selection. Functions available for this object are `isomiRs::isoCounts()` for count matrix creation, `isomiRs::isoNorm()` for normalization, `isomiRs::isoDE()` for differential expression and `isomiRs::isoPLSDA()` for clustering. `isomiRs::isoPlot()` helps with basic expression plot.

`metadata` contains one list:

- `rawData` is a [data.frame](#) with the information of each sequence found in the data and the counts for each sample.

The naming of isomiRs follows these rules:

- `miRNA name`
- `type:ref` if the sequence is the same than the miRNA reference. `iso` if the sequence has variations.
- `iso_5p` tag: indicates variations at 5 position. The naming contains two words: `direction-nucleotides`, where `direction` can be UPPER CASE NT (changes upstream of the 5 reference position) or LOWER CASE NT (changes downstream of the 5 reference position). `0` indicates no variation, meaning the 5 position is the same than the reference. After `direction`, it follows the nucleotide/s that are added (for upstream changes) or deleted (for downstream changes).
- `iso_3p` tag: indicates variations at 3 position. The naming contains two words: `direction-nucleotides`, where `direction` can be LOWER CASE NT (upstream of the 3 reference position) or UPPER CASE NT (downstream of the 3 reference position). `0` indicates no variation, meaning the 3 position is the same than the reference. After `direction`, it follows the nucleotide/s that are added (for downstream changes) or deleted (for upstream changes).
- `iso_add` tag: indicates nucleotides additions at 3 position. The naming contains two words: `direction-nucleotides`, where `direction` is UPPER CASE NT (upstream of the 5 reference position). `0` indicates no variation, meaning the 3 position has no additions. After `direction`, it follows the nucleotide/s that are added.
- `iso_snv` tag: indicates nucleotides substitutions along the sequences. The naming contains three words: `position-nucleotide@isomiR-nucleotide@reference`.
- `iso_snv_seed` tag: same than `iso_snv` tag, but only if the change happens between nucleotide 2 and 8.

In general nucleotides in UPPER case mean insertions respect to the reference sequence, and nucleotides in LOWER case mean deletions respect to the reference sequence.

Examples

```
path <- system.file("extra", package="isomiRs")
fn_list <- list.files(path, full.names = TRUE)
de <- data.frame(row.names=c("f1" , "f2"),
                 condition = c("newborn", "newborn"))
ids <- IsomirDataSeqFromFiles(fn_list, coldata=de)

head(counts(ids))
```

IsomirDataSeqFromFiles

Loads miRNA annotation from seqbuster tool or pre-processed data.

Description

This function parses output of seqbuster tool to allow isomiRs/miRNAs analysis of samples in different groups such as characterization, differential expression and clustering. It creates an [isomiRs::IsomirDataSeq](#) object.

Usage

```
IsomirDataSeqFromFiles(files, coldata, rate = 0.2, canonicalAdd = TRUE,
  uniqueMism = TRUE, uniqueHits = FALSE, design = ~1L,
  minHits = 1L, header = TRUE, skip = 0, quiet = TRUE, ...)
```

```
IsomirDataSeqFromRawData(rawdata, coldata, design = ~1L, pct = 0.1,
  whitelist = NULL, ...)
```

Arguments

files	files with the output of seqbuster tool
coldata	data frame containing groups for each sample
rate	minimum counts fraction to consider a mismatch a real mutation
canonicalAdd	boolean only keep A/T non-template addition. All non-template nucleotides at the 3' end will be removed if they contain C/G nts.
uniqueMism	boolean only keep mutations that have a unique hit to one miRNA molecule. For instance, if the sequence map to two different miRNAs, then it would be removed.
uniqueHits	boolean whether filtering ambiguous sequences or not.
design	a formula to pass to DESeq2::DESeqDataSet
minHits	Minimum number of reads in the sample to consider it in the final matrix.
header	boolean to indicate files contain headers
skip	skip first line when reading files
quiet	boolean indicating to print messages while reading files. Default FALSE.
...	arguments provided to SummarizedExperiment. including rowData.
rawdata	data.frame stored in metadata slot of IsomirDataSeq object.
pct	numeric used to remove isomiRs with an importance lower than this value. Importance is calculated by dividing the isomiR count by the total counts of the miRNA to which it maps.
whitelist	character vector with sequences to keep even if the filtering step would have removed them. They have to match the seq column in the table.

Details

This function parses the output of http://seqcluster.readthedocs.org/mirna_annotation.html for each sample to create a count matrix for isomiRs, miRNAs or isomiRs grouped in types (i.e all sequences with variations at 5' but ignoring any other type). It creates [isomiRs::IsomirDataSeq](#) object (see link to example usage of this class) to allow visualization, queries, differential expression analysis and clustering. To create the [isomiRs::IsomirDataSeq](#), it parses the isomiRs files, and generates an initial matrix having all isomiRs detected among samples. As well, it creates a summary for each isomiR type (trimming, addition and substitution) to visualize general isomiRs distribution.

Value

[IsomirDataSeq](#) class object.

Examples

```

path <- system.file("extra", package="isomiRs")
fn_list <- list.files(path, full.names = TRUE)
de <- data.frame(row.names=c("f1" , "f2"),
                 condition = c("newborn", "newborn"))
ids <- IsomirDataSeqFromFiles(fn_list, coldata=de)

head(counts(ids))
IsomirDataSeqFromRawData(metadata(ids)[["rawData"]], de)

```

isoNetwork

*Clustering miRNAs-genes pairs in similar pattern expression***Description**

Clustering miRNAs-genes pairs

Usage

```

isoNetwork(mirna_rse, gene_rse, summarize = NULL, target = NULL,
           org = NULL, enrich = NULL, genename = "ENSEMBL", min_cor = -0.6,
           min_fc = 0.5)

```

Arguments

mirna_rse	SummarizedExperiment with miRNA information. See details.
gene_rse	SummarizedExperiment with gene information. See details.
summarize	Character column name in colData(rse) to use to group samples and compare between miRNA/gene expression.
target	Matrix with miRNAs (columns) and genes (rows) target prediction (1 if it is a target, 0 if not).
org	AnnotationDb object. For example:(org.Mm.eg.db)
enrich	The output of clusterProfiler of similar functions.
genename	Character keytype of the gene names in gene_rse object.
min_cor	Numeric cutoff to consider a miRNA to regulate a target.
min_fc	Numeric cutoff to consider as the minimum log2FoldChange between groups to be considered in the analysis.

Details

This function will correlate miRNA and gene expression data using a specific metadata variable to group samples and detect pattern of expression that will be annotated with GO terms. mirna_rse and gene_rse can be created using the following code:

```
mi_rse = SummarizedExperiment(assays=SimpleList(norm=mirna_matrix), colData, metadata=list(sign=mirna_keep))
```

where, mirna_matrix is the normalized counts expression, colData is the metadata information and mirna_keep the list of miRNAs to be used by this function.

Value

list with network information

Examples

```
# library(org.Mm.eg.db)
# library(clusterProfiler)
data(isoExample)
# ego <- enrichGO(row.names(assay(gene_ex_rse, "norm")),
#                 org.Mm.eg.db, "ENSEMBL", ont = "BP")
data <- isoNetwork(mirna_ex_rse, gene_ex_rse,
                  summarize = "group", target = ma_ex,
                  enrich = ego)
isoPlotNet(data, minGenes = 5)
```

isoNorm

Normalize count matrix

Description

This function normalizes raw count matrix using `DESeq2::rlog()` function from [DESeq2::DESeq2-package](#).

Usage

```
isoNorm(ids, formula = NULL, maxSamples = 50)
```

Arguments

<code>ids</code>	Object of class IsomirDataSeq .
<code>formula</code>	Formula that will be used for normalization.
<code>maxSamples</code>	Maximum number of samples to use with <code>DESeq2::rlog()</code> , if not <code>limma::voom()</code> is used.

Value

[IsomirDataSeq](#) object with the normalized count matrix in a slot. The normalized matrix can be access with `counts(ids, norm=TRUE)`.

Examples

```
data(mirData)
ids <- isoCounts(mirData, minc=10, mins=6)
ids <- isoNorm(mirData, formula=~condition)
head(counts(ids, norm=TRUE))
```

`isoPlot`*Plot the amount of isomiRs in different samples*

Description

This function plot different isomiRs proportion for each sample. It can show trimming events at both side, additions and nucleotides changes.

Usage

```
isoPlot(ids, type = "iso5", column = NULL, use = NULL, nts = FALSE)
```

Arguments

<code>ids</code>	Object of class IsomirDataSeq .
<code>type</code>	String (iso5, iso3, add, snv, all) to indicate what isomiRs to use for the plot. See details for explanation.
<code>column</code>	String indicating the column in <code>colData</code> to color samples.
<code>use</code>	Character vector to only use these isomiRs for the plot. The id used is the rownames that comes from using <code>isoCounts</code> with all the arguments on TRUE.
<code>nts</code>	Boolean to indicate whether plot positions of nucleotides changes when showing single nucleotides variants.

Details

There are four different values for `type` parameter. To plot trimming at 5' or 3' end, use `type="iso5"` or `type="iso3"`. Get a summary of all using `type="all"`. In this case, it will plot 3 positions at both side of the reference position described at miRBase site. Each position refers to the sequences that start/end before or after the miRBase reference. The color indicates the sample group. The size of the point is proportional to the abundance considering the total as all the sequences in the sample. The position at y is the different sequences considering the total as all sequences with changes for the specific isomiR showed.

Same logic applies to `type="add"` and `type="snv"`. However, when `type="add"`, the plot will refer to addition events from the 3' end of the reference position. Note that this additions don't match to the precursor sequence, they are non-template additions. In this case, only 3 positions after the 3' end will appear in the plot. When `type="snv"`, it will appear one position for each nucleotide in the reference miRNA. Points will indicate isomiRs with nucleotide changes at the given position. When `type="all"` a color coordinate map will show the abundance of each isomiR type in a single plot. Note the position is relatively to the sequence not the miRNA.

Value

[ggplot2::ggplot\(\)](#) Object showing different isomiRs changes at different positions.

Examples

```
data(mirData)
isoPlot(mirData)
```

 isoPlotNet

Functional miRNA / gene expression profile plot

Description

Plot analysis from [isoNetwork\(\)](#). See that function for an example of the figure.

Usage

```
isoPlotNet(obj, minGenes = 2)
```

Arguments

obj	Output from isoNetwork() .
minGenes	Minimum number of genes per term to be kept.

Value

Network ggplot.

 isoPlotPosition

Plot nucleotides changes at a given position

Description

This function plot different isomiRs proportion for each sample at a given position focused on the nucleotide change that happens there.

Usage

```
isoPlotPosition(ids, position = 1L, column = NULL)
```

Arguments

ids	Object of class IsomirDataSeq .
position	Integer indicating the position to show.
column	String indicating the column in colData to color samples.

Details

It shows the nucleotides changes at the given position for each sample in each group. The color indicates the sample group. The size of the point is proportional to the number of total counts of isomiRs with changes. The position at y is the supporting the change. Note the position is relatively to the sequence not the miRNA.

Value

[ggplot2::ggplot\(\)](#) Object showing nucleotide changes at a given position.

Examples

```
data(mirData)
isoPlotPosition(mirData)
```

isoPLSDA

*Partial Least Squares Discriminant Analysis for IsomirDataSeq***Description**

Use PLS-DA method with the normalized count data to detect the most important features (miRNAs/isomiRs) that explain better the group of samples given by the experimental design. It is a supervised clustering method with permutations to calculate the significance of the analysis.

Usage

```
isoPLSDA(ids, group, validation = NULL, learn = NULL, test = NULL,
  tol = 0.001, nperm = 400, refinement = FALSE, vip = 1.2)
```

Arguments

ids	Object of class IsomirDataSeq
group	Column name in <code>colData(ids)</code> to use as variable to explain.
validation	Type of validation, either NULL or "learntest". Default NULL.
learn	Optional vector of indexes for a learn-set. Only used when validation="learntest". Default NULL.
test	Optional vector of indices for a test-set. Only used when validation="learntest". Default NULL
tol	Tolerance value based on maximum change of cumulative R-squared coefficient for each additional PLS component. Default tol=0.001.
nperm	Number of permutations to compute the PLD-DA p-value based on R2 magnitude. Default nperm=400.
refinement	Logical indicating whether a refined model, based on filtering out variables with low VIP values.
vip	Variance Importance in Projection threshold value when a refinement process is considered. Default vip=1.2 .

Details

Partial Least Squares Discriminant Analysis (PLS-DA) is a technique specifically appropriate for analysis of high dimensionality data sets and multicollinearity (*Perez-Enciso, 2013*). PLS-DA is a supervised method (i.e. makes use of class labels) with the aim to provide a dimension reduction strategy in a situation where we want to relate a binary response variable (in our case young or old status) to a set of predictor variables. Dimensionality reduction procedure is based on orthogonal transformations of the original variables (miRNAs/isomiRs) into a set of linearly uncorrelated latent variables (usually termed as components) such that maximizes the separation between the different classes in the first few components (*Xia, 2011*). We used sum of squares captured by the model (R2) as a goodness of fit measure.

We implemented this method using the `DiscrMiner::DiscrMiner-package` into `isoPLSDA()` function. The output p-value of this function will tell about the statistical significant of the group separation using miRNA/isomiR expression data.

Read more about the parameters related to the PLS-DA directly from `DiscrMiner::plsDA()` function.

Value

A `base::list` with the following elements: `R2Matrix` (R-squared coefficients of the PLS model), `components` (of the PLS, similar to PCs in a PCA), `vip` (most important isomiRs/miRNAs), `group` (classification of the samples), `p.value` and `R2PermutationVector` obtained by the permutations.

If the option `refinement` is set to `TRUE`, then the following elements will appear: `R2RefinedMatrix` and `componentsRefinedModel` (R-squared coefficients of the PLS model only using the most important miRNAs/isomiRs). As well, `p.valRefined` and `R2RefinedPermutationVector` with p-value and R2 of the permutations where samples were randomized. And finally, `p.valRefinedFixed` and `R2RefinedFixedPermutationVector` with p-value and R2 of the permutations where miRNAs/isomiRs were randomized.

References

Perez-Enciso, Miguel and Tenenhaus, Michel. Prediction of clinical outcome with microarray data: a partial least squares discriminant analysis (PLS-DA) approach. *Human Genetics*. 2003.

Xia, Jianguo and Wishart, David S. Web-based inference of biological patterns, functions and pathways from metabolomic data using *MetaboAnalyst*. *Nature Protocols*. 2011.

Examples

```
data(mirData)
# Only miRNAs with > 10 reads in all samples.
ids <- isoCounts(mirData, minc=10, mins=6)
ids <- isoNorm(ids, formula=~condition)
pls.ids = isoPLSDA(ids, "condition", nperm = 2)
cat(paste0("pval:", pls.ids$p.val))
cat(paste0("components:", pls.ids$components))
```

isoPLSDAplot

Plot components from isoPLSDA analysis (pairs plot)

Description

Plot the most significant components that come from `isoPLSDA()` analysis together with the density of the samples scores along those components.

Usage

```
isoPLSDAplot(pls, n = 2)
```

Arguments

<code>pls</code>	Output from <code>isoPLSDA()</code> function.
<code>n</code>	Number of components to plot.

Details

The function `isoPLSDAplot` helps to visualize the results from `isoPLSDA()`. It will plot the samples using the significant components (t1, t2, t3 ...) from the PLS-DA analysis and the samples score distribution along the components. It uses `GGally::ggpairs()` for the plot.

Value

`GGally::ggpairs()` plot showing the scores for each sample using isomiRs/miRNAs expression to explain variation.

`base::data.frame` object with a first column referring to the sample group, and the following columns referring to the score that each sample has for each component from the PLS-DA analysis.

Examples

```
data(mirData)
# Only miRNAs with > 10 reads in all samples.
ids <- isoCounts(mirData, minc=10, mins=6)
ids <- isoNorm(ids, formula=~condition)
pls.ids <- isoPLSDA(ids, "condition", nperm = 2)
isoPLSDAplot(pls.ids)
```

isoSelect

Method to select specific miRNAs from an IsomirDataSeq object.

Description

This method allows to select a miRNA and all its isomiRs from the count matrix.

Usage

```
isoSelect.IsomirDataSeq(object, mirna, minc = 10)
```

```
## S4 method for signature 'IsomirDataSeq'
isoSelect(object, mirna, minc = 10)
```

Arguments

<code>object</code>	A <code>IsomirDataSeq</code> object.
<code>mirna</code>	String referring to the miRNA to show.
<code>minc</code>	Minimum number of isomiR reads needed to be included in the table.

Value

`S4Vectors::DataFrame` with count information. The row.names show the isomiR names, and each of the columns shows the counts for this isomiR in that sample. Mainly, it will return the count matrix only for isomiRs belonging to the miRNA family given by the `mirna` parameter. IsomiRs need to have counts bigger than `minc` parameter at least in one sample to be included in the output. Annotation of isomiRs follows these rules:

- miRNA name
- mismatches

- additions
- 5 trimming events
- 3 trimming events

Author(s)

Lorena Pantano

Examples

```
data(mirData)
# To select isomiRs from let-7a-5p miRNA
# and with 10000 reads or more.
isoSelect(mirData, mirna="hsa-let-7a-5p", minc=10000)
```

isoTop

Heatmap of the top expressed isomiRs

Description

This function creates a heatmap with the top N isomiRs/miRNAs. It uses the matrix under counts(ids) to get the top expressed isomiRs/miRNAs using the average expression value and plot a heatmap with the raw counts for each sample.

Usage

```
isoTop(ids, top = 20)
```

Arguments

ids Object of class [IsomirDataSeq](#).

top Number of isomiRs/miRNAs used.

Value

PCA of the top expressed miRNAs

Examples

```
data(mirData)
isoTop(mirData)
```

ma_ex	<i>Data frame containing gene-mirna relationship</i>
-------	--

Description

Data frame containing gene-mirna relationship

Usage

```
ma_ex
```

Format

A data frame with rows same as gene_ex_rse and columns same as mirna_ex_rse.

mirData	<i>Example of IsomirDataSeq with human brain miRNA counts data</i>
---------	--

Description

This data set is the object return by [IsomirDataSeqFromFiles](#). It contains miRNA count data from 14 samples: 7 control individuals (pc) and 7 patients with Parkinson's disease in early stage (Pantano et al, 2016). Use colData to see the experiment design.

Usage

```
data("mirData")
```

Format

a [IsomirDataSeq](#) class.

Author(s)

Lorena Pantano, 2018-04-27

Source

Data is available from GEO dataset under accession number GSE97285

Every sample was analyzed with seqbuster tool, see http://seqcluster.readthedocs.org/mirna_annotation.html for more details. You can get same files running the small RNA-seq pipeline from <https://github.com/bcbio/bcbio-nextgen>.

bcbio_nextgen was used for the full analysis.

See raw-data.R to know how to recreate the object. This script is inside "extra" folder of the package.

References

Pantano L, Friedlander MR, Escaramis G, Lizano E et al. Specific small-RNA signatures in the amygdala at premotor and motor stages of Parkinson's disease revealed by deep sequencing analysis. *Bioinformatics* 2016 Mar 1;32(5):673-81. PMID: 26530722

mirna2targetscan *Find targets in targetscan database*

Description

From a list of miRNA names, find their targets in targetscan.Hs.eg.db annotation package.

Usage

```
mirna2targetscan(mirna, species = "hsa", org = NULL, keytype = NULL)
```

Arguments

mirna	Character vector with miRNA names as in miRBase 21.
species	hsa or mmu supported right now.
org	AnnotationDb object. For example:(org.Mm.eg.db)
keytype	Character mentioning the gene id to use. For example, ENSEMBL.

Value

[data.frame](#) with 4 columns:

- miRFamily
- Seedmatch
- PCT
- entrezGene

Examples

```
library(targetscan.Hs.eg.db)
mirna2targetscan(c("hsa-miR-34c-5p"))
```

mirna_ex_rse *Data frame containing mirna expression data*

Description

Data frame containing mirna expression data

Usage

```
mirna_ex_rse
```

Format

mirna expression data with 18 samples: example of a time series data

mirTritation	<i>Data frame containing mirna from Argyropoulos's paper</i>
--------------	--

Description

Argyropoulos, Christos, et al. "Modeling bias and variation in the stochastic processes of small RNA sequencing." *Nucleic Acids Research* (2017).

Usage

```
mirTritation
```

Format

mirna expression data in long format. Train and test data to use with isoCorrect

updateIsomirDataSeq	<i>Update IsomirDataSeq object from version < 1.7</i>
---------------------	--

Description

In version 1.9 IsomirDataSeq object changed their internal structure to save space and speed up loading and downstream functions.

Usage

```
updateIsomirDataSeq(object)
```

Arguments

object [IsomirDataSeq](#).

Details

This function will update to the current structure.

Index

*Topic **datasets**

- dat286.long, 3
- ego, 4
- gene_ex_rse, 5
- ma_ex, 19
- mirna_ex_rse, 20
- mirTritation, 21

- base::data.frame, 17
- base::list, 16
- base::matrix, 3

- counts, 3
- counts, IsomirDataSeq-method (counts), 3
- counts.IsomirDataSeq (counts), 3
- counts<-, IsomirDataSeq, matrix-method (counts), 3

- dat286.long, 3
- data.frame, 6, 9, 20
- DESeq2::DESeq2-package, 7, 8, 12
- DESeq2::DESeqDataSet, 7, 8, 10
- DESeq2::results(), 8
- DESeq2::rlog(), 12
- design, 4
- design, IsomirDataSeq-method (design), 4
- design<-, IsomirDataSeq, formula-method (design), 4
- DiscrMiner::DiscrMiner-package, 16
- DiscrMiner::plsDA(), 16

- ego, 4

- findTargets, 5

- gene_ex_rse, 5
- GGally::ggpairs(), 17
- ggplot2::ggplot(), 13, 14

- isoAnnotate, 6
- isoCounts, 6
- isoCounts(), 8
- isoDE, 7
- isoDE(), 8
- IsomirDataSeq, 4, 6–8, 10, 12–15, 17–19, 21

- IsomirDataSeq (IsomirDataSeq-class), 8
- IsomirDataSeq-class, 8
- IsomirDataSeqFromFiles, 8, 9, 19
- IsomirDataSeqFromRawData (IsomirDataSeqFromFiles), 9
- isomiRs (isomiRs-package), 2
- isomiRs-package, 2
- isomiRs::counts(), 8
- isomiRs::isoCounts(), 8
- isomiRs::isoDE(), 8
- isomiRs::IsomirDataSeq, 9, 10
- isomiRs::isoNorm(), 8
- isomiRs::isoPlot(), 8
- isomiRs::isoPLSDA(), 8
- isomiRs::isoSelect(), 8
- isoNetwork, 11
- isoNetwork(), 14
- isoNorm, 12
- isoPlot, 13
- isoPlotNet, 14
- isoPlotPosition, 14
- isoPLSDA, 15
- isoPLSDA(), 16, 17
- isoPLSDAplot, 16
- isoSelect, 17
- isoSelect, IsomirDataSeq-method (isoSelect), 17
- isoSelect.IsomirDataSeq (isoSelect), 17
- isoTop, 18

- limma::voom(), 12

- ma_ex, 19
- mirData, 19
- mirna2targetscan, 20
- mirna_ex_rse, 20
- mirTritation, 21

- S4Vectors::DataFrame, 17

- updateIsomirDataSeq, 21