Package ‘glycanr’

October 13, 2022

Title Tools for Analysing N-Glycan Data

Version 0.4.0

Maintainer Ivo Ugrina <`ivougrina@gmail.com`>

Language en-US

Description Useful utilities in N-glycan data analysis. This package tries to fill the gap in N-glycan data analysis by providing easy to use functions for basic operations on data (see <`https://en.wikipedia.org/wiki/Glycomics`> for more details on Glycomics). At the moment 'glycanr' is mostly oriented to data obtained by UPLC (Ultra Performance Liquid Chromatography) and LCMS (Liquid chromatography–mass spectrometry) analysis of Plasma and IgG glycome.

Depends R (>= 3.1.2)

Imports ggplot2, tidyr (>= 0.3.1), dplyr (>= 0.4.3), coin

License MIT + file LICENSE

LazyData true

Suggests knitr, markdown, preprocessCore, testthat

VignetteBuilder knitr

URL `https://github.com/iugrina/glycanr`

BugReports `https://github.com/iugrina/glycanr/issues`

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Ivo Ugrina [aut, cre, cph],
Lucija Klaric [aut],
Alyce Russell [aut],
Frano Vuckovic [aut]

Repository CRAN

Date/Publication 2021-03-29 14:40:02 UTC

R topics documented:

`glyco.outliers` *Discover outliers in glycan data*

Description

Returns outliers within every glycan structure

Usage

```
glyco.outliers(data, group = NULL, outlier.function = NULL, alpha = 1.5)
```

Arguments

`data`	data frame in long format containing glycan measurements
`group`	a possible grouping parameter on which stratification of `data` should be conducted. It should be a name of one of the columns in dataframe `data` and of type `factor`.
`outlier.function`	a function that checks for outliers in a vector. Receives one parameter representing a vector and returns logical vector indicating outliers.
`alpha`	If outlier.function parameter is set to NULL outliers are calculated as those points that are greater than upper quartile + alpha * IQR (interquartile range) or lower than lower quartile - alpha * IQR (interquartile range). If parameter outlier.function is not NULL parameter alpha is not used.

Details

Input data frame should have at least the following three columns:
- gid - representing a unique name of a sample
- glycan - representing glycan names
- value - representing measured values

Value

Returns a data.frame with outliers

Author(s)

Ivo Ugrina

Examples

```
data(mpiu)

glyco.outliers(mpiu)

# outliers per plate
glyco.outliers(mpiu, group="Plate")
```

`glyco.plot` *Plot data representing Glycans in boxplot or violin mode using ggplot2*

Description

This function constructs standard plots in exploratory analysis of N-Glycans.

Usage

```
glyco.plot(
  data,
  collapse = TRUE,
  violin = FALSE,
  group = NULL,
  all = TRUE,
  p.adjust.method = "holm",
  print.p.values = TRUE,
  log.transform = FALSE,
  glyco.names = NULL
)
```

Arguments

`data`	data frame which holds columns representing Glycans. These column names must start with 'GP'.
`collapse`	should Glycans be presented in one facet (default) or with more facets (one per Glycan).
`violin`	should Glycans be presented in a boxplot (default) or violin format.
`group`	this a possible grouping parameter on which stratification of `data` should be conducted. It should be a name of one of the columns in dataframe `data` and of type `factor`.
`all`	should all of the variables (default) be presented in the plot or only those that have significant p-values. This variable is meaningful only when `group` is not `NULL` since the testing of differences is conducted between different groups represented by `group` variable. If `group` has only 2 levels then Mann-Whitney-Wilcoxon (`wilcox.test`) test is conducted. Otherwise, Kruskal-Wallis test is conducted (`kruskal.test`). Obtained p-values are adjusted to multiple testing with `p.adjust`.
`p.adjust.method`	method used for adjustment of p-values to multiple testing. Variable p.adjust.method must be an element of `p.adjust.methods`.
`print.p.values`	should p-values be printed on plots
`log.transform`	should Glycans be log transform prior to plotting.
`glyco.names`	names of columns that represent glycan data. If `NULL` all columns starting with 'GP' in their names will be used

Value

Returns a list consisting of p-values, adjusted p-vales and the plot.

Author(s)

Ivo Ugrina

Examples

```
devAskNewPage(TRUE)
exampleData <- data.frame(ID=1:100, GP1=runif(100),
  GP2=rexp(100,0.2), GP3=rgamma(100, 3),
  Plate=factor(sample(1:2,100,replace=TRUE)))
glyco.plot(exampleData)
glyco.plot(exampleData, group='Plate', collapse=FALSE, log=TRUE)
```

`ildt` *Derived traits for Glycan peaks in IgG for LCMS*

Description

Calculates values of derived traits for Glycan peaks in IgG for LCMS

Usage

```
ildt(data = NULL, method = "2014", print.exp.names = FALSE)
```

Arguments

`data`	data frame that holds columns representing Glycans.
`method`	year of the derived traits definition. By default 2014.
`print.exp.names`	If `TRUE` return expected column names representing glycans.

Details

Calculates derived traits from basic glycan peaks. User can choose which definition of the derived traits he will use (see references for different versions/definitions of derived traits).

Value

Returns the data frame with derived traits

Author(s)

Ivo Ugrina

References

Jennifer E. Huffman et al. (2014) "Comparative Performance of Four Methods for High-throughput Glycosylation Analysis of Immunoglobulin G in Genetic and Epidemiological Research*" doi: 10.1074/mcp.M113.037465

`ildt.translate`	*Translate names between computer readable and human readable for derived traits of IgG with LCMS*

Description

Translates names between computer readable and human readable for derived traits of IgG with LCMS

Usage

```
ildt.translate(orignames, to = "inverse", method = "2014")
```

Arguments

`orignames`	vector; type string
`to`	type of translation. If `inverse` is used everything will be translated. For `computer` names will be translated to computer readable, and for `human` names will be translated to human readable.
`method`	year of the derived traits definition. By default 2014.

Details

User can choose which definition of the derived traits he will use (see references for different versions/definitions of derived traits).

Value

Returns a character vector with original and translated names

Author(s)

Ivo Ugrina

References

Jennifer E. Huffman et al. (2014) "Comparative Performance of Four Methods for High-throughput Glycosylation Analysis of Immunoglobulin G in Genetic and Epidemiological Research*" doi: 10.1074/mcp.M113.037465

`iudt` *Derived traits for Glycan peaks in IgG for UPLC*

Description

Calculates values of derived traits for Glycan peaks in IgG for UPLC

Usage

```
iudt(data = NULL, method = "2014", print.exp.names = FALSE)
```

Arguments

`data`	data frame that holds columns representing Glycans. These column names should start with 'GP'.
`method`	year of the derived traits definition. By default 2014.
`print.exp.names`	If `TRUE` return expected column names representing glycans.

Details

Calculates derived traits from basic glycan peaks. User can choose which definition of the derived traits he will use (see references for different versions/definitions of derived traits).

Value

Returns the data frame with derived traits

Author(s)

Ivo Ugrina, Frano Vučković

References

Jennifer E. Huffman et al. (2014) "Comparative Performance of Four Methods for High-throughput Glycosylation Analysis of Immunoglobulin G in Genetic and Epidemiological Research*" doi: 10.1074/mcp.M113.037465

`iudt.translate` *Translate names between computer readable and human readable for derived traits of IgG with UPLC*

Description

Translates names between computer readable and human readable for derived traits of IgG with UPLC

Usage

```
iudt.translate(orignames, to = "inverse", method = "2014")
```

Arguments

`orignames`	vector; type string
`to`	type of translation. If `inverse` is used everything will be translated. For `computer` names will be translated to computer readable, and for `human` names will be translated to human readable.
`method`	year of the derived traits definition. By default 2014.

Details

User can choose which definition of the derived traits he will use (see references for different versions/definitions of derived traits).

Value

Returns a character vector with original and translated names

Author(s)

Ivo Ugrina

References

Jennifer E. Huffman et al. (2014) "Comparative Performance of Four Methods for High-throughput Glycosylation Analysis of Immunoglobulin G in Genetic and Epidemiological Research*" doi: 10.1074/mcp.M113.037465

`mediannorm` *Median Normalization of glycan data*

Description

Returns glycans normalized with Median Normalization approach.

Usage

```
mediannorm(d, grouping = FALSE)
```

Arguments

`d`	data frame in long format containing glycan measurements
`grouping`	should data be normalized per groups

Details

Input data frame should have at least the following three columns:
- gid - representing a unique name of a sample
- glycan - representing glycan names
- value - representing measured values
and if the grouping argument is `TRUE` it should also have column:
- groups - representing groupings (e.g. IgG1, IgG2 and IgG4)

Value

Returns a data.frame with original glycan values substituted by normalized ones

Author(s)

Ivo Ugrina, Lucija Klarić

Examples

```
data(mpiu)
mpiun <- mediannorm(mpiu)
head(mpiun)
```

`medianquotientnorm` *Median Quotient Normalization of glycan data*

Description

Returns glycans normalized with Median Quotient Normalization approach.

Usage

```
medianquotientnorm(d, grouping = FALSE)
```

Arguments

`d`	data frame in long format containing glycan measurements
`grouping`	should data be normalized per groups

Details

Input data frame should have at least the following three columns:
- gid - representing a unique name of a sample
- glycan - representing glycan names
- value - representing measured values
and if the grouping argument is `TRUE` it should also have column:
- groups - representing groupings (e.g. IgG1, IgG2 and IgG4)

Value

Returns a data.frame with original glycan values substituted by normalized ones

Author(s)

Ivo Ugrina, Lucija Klarić

References

Dieterle F,Ross A, Schlotterbeck G, Senn H.:
Probabilistic Quotient Normalization as Robust Method to Account for Diluition of Complex Biological Mixtures. Application in 1H NMR Metabolomics.
Anal Chem 2006;78:4281-90.
doi: 10.1021/ac051632c

Examples

```
data(mpiu)
mpiun <- medianquotientnorm(mpiu)
head(mpiun)
```

`mpiu` *Multiple plates IgG UPLC data example*

Description

This is an example of the data obtained by UPLC while analysing IgG.

Usage

```
mpiu
```

Format

An object of class `data.frame` with 13680 rows and 4 columns.

`mpiunorm` *Normalized mpiu data*

Description

This is an example of the normalized mpiu data. It is intended to be used for unit tests.

Usage

```
mpiunorm
```

Format

An object of class `data.frame` with 68400 rows and 5 columns.

`phdt` *Derived traits for Glycan peaks in PLASMA for HPLC*

Description

Calculates values of derived traits for Glycan peaks in Plasma for HPLC

Usage

```
phdt(data = NULL, method = "2011", print.exp.names = FALSE)
```

Arguments

`data`	data frame that holds columns representing Glycans. These column names should start with 'GP'.
`method`	year of the derived traits definition. By default 2011.
`print.exp.names`	If `TRUE` return expected column names representing glycans.

Details

Calculates derived traits from basic glycan peaks. User can choose which definition of the derived traits he will use (see references for different versions/definitions of derived traits).

Value

Returns the data frame with derived traits

Author(s)

Ivo Ugrina, Lucija Klarić

References

Lu et al. (2011) "Screening Novel Biomarkers for Metabolic Syndrome by Profiling Human Plasma N-Glycans in Chinese Han and Croatian Populations" doi: 10.1021/pr2004067 Irena Trbojevic-Akmacic et al. "Plasma N-glycome composition associates with chronic low back pain" doi: 10.1016/j.bbagen.2018.07.003

`quantilenorm` *Quantile Normalization of glycan data*

Description

Returns glycans normalized with Quantile Normalization approach.

Usage

```
quantilenorm(d, grouping = FALSE, transpose = FALSE)
```

Arguments

`d`	data frame in long format containing glycan measurements
`grouping`	should data be normalized per groups
`transpose`	transpose the data prior to normalization

Details

Input data frame should have at least the following three columns:
- gid - representing a unique name of a sample
- glycan - representing glycan names
- value - representing measured values
and if the grouping argument is `TRUE` it should also have column:
- groups - representing groupings (e.g. IgG1, IgG2 and IgG4)

Value

Returns a data.frame with original glycan values substituted by normalized ones

Author(s)

Ivo Ugrina, Lucija Klarić

References

Bolstad, B. M., Irizarry R. A., Astrand, M, and Speed, T. P.:
A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance.
Bioinformatics 19(2), p. 185-193, 2003.
doi: 10.1093/bioinformatics/19.2.185

Examples

```
data(mpiu)
if(requireNamespace("preprocessCore", quietly=TRUE)){
  mpiun <- quantilenorm(mpiu)
  head(mpiun)

  # transpose (change) subjects and measurements
  mpiunt <- quantilenorm(mpiu, transpose=TRUE)
  head(mpiunt)
}
```

`refpeaknorm` *Reference Peak Normalization of glycan data*

Description

Returns glycans normalized with Reference Peak Normalization approach.

Usage

```
refpeaknorm(d, grouping = FALSE, peak = NULL)
```

Arguments

`d`	data frame in long format containing glycan measurements
`grouping`	should data be normalized per groups
`peak`	glycan name to use as the reference peak. If `NULL` peak with maximal value (summed through all samples) is used

Details

Input data frame should have at least the following three columns:
- gid - representing a unique name of a sample
- glycan - representing glycan names
- value - representing measured values
and if the grouping argument is `TRUE` it should also have column:
- groups - representing groupings (e.g. IgG1, IgG2 and IgG4)

Value

Returns a data.frame with original glycan values substituted by normalized ones

Author(s)

Ivo Ugrina, Lucija Klarić

Examples

```
data(mpiu)
mpiun <- refpeaknorm(mpiu)
head(mpiun)
```

`tanorm` *Total Area Normalization of glycan data*

Description

Returns glycans normalized with Total Area Normalization approach.

Usage

```
tanorm(d, grouping = FALSE)
```

Arguments

`d`	data frame in long format containing glycan measurements
`grouping`	should data be normalized per groups

Details

Input data frame should have at least the following three columns:
- gid - representing a unique name of a sample
- glycan - representing glycan names
- value - representing measured values

and if the grouping argument is `TRUE` it should also have column:
- groups - representing groupings (e.g. IgG1, IgG2 and IgG4)

Value

Returns a data.frame with original glycan values substituted by normalized ones

Author(s)

Ivo Ugrina

Examples

```
data(mpiu)
mpiun <- tanorm(mpiu)
head(mpiun)
```

Index