

Package: lipidomeR (via r-universe)

August 28, 2024

Type Package

Title Integrative Visualizations of the Lipidome

Version 0.1.2

Description Create lipidome-wide heatmaps of statistics with the 'lipidomeR'. The 'lipidomeR' provides a streamlined pipeline for the systematic interpretation of the lipidome through publication-ready visualizations of regression models fitted on lipidomics data. With 'lipidomeR', associations between covariates and the lipidome can be interpreted systematically and intuitively through heatmaps, where lipids are categorized by the lipid class and are presented on two-dimensional maps organized by the lipid size and level of saturation. This way, the 'lipidomeR' helps you gain an immediate understanding of the multivariate patterns in the lipidome already at first glance. You can create lipidome-wide heatmaps of statistical associations, changes, differences, variation, or other lipid-specific values. The heatmaps are provided with publication-ready quality and the results behind the visualizations are based on rigorous statistical models.

License GPL-3

Encoding UTF-8

LazyData true

URL <https://tommi-s.github.io/>

RoxygenNote 7.0.2

Depends R (>= 3.5.0)

Imports BiocManager, dplyr, ggplot2, knitr, limma, reshape2, shadowtext, stringr, tableone, tidyr

NeedsCompilation no

Author Tommi Suviavaal [aut, cre]
(<https://orcid.org/0000-0002-2583-4912>)

Maintainer Tommi Suviavaal <TSUV0001@RegionH.DK>

Repository CRAN

Date/Publication 2020-03-15 10:20:02 UTC

Contents

| | |
|---|-----------|
| cancerlipidome | 2 |
| compute_F_test_with_limma | 4 |
| compute_models_with_limma | 5 |
| compute_post_hoc_test_with_limma | 6 |
| heatmap_lipidome | 7 |
| heatmap_lipidome_from_limma | 9 |
| humanlipidome | 12 |
| install_Bioconductor_dependencies | 13 |
| liverlipidome | 14 |
| map_lipid_names | 17 |
| Index | 18 |

| | |
|----------------|--|
| cancerlipidome | <i>Levels of lipids in benign and malignant breast tumors in humans.</i> |
|----------------|--|

Description

This data set contains levels of 409 named lipids in 118 human breast tumor tissue samples.

Usage

```
data( cancerlipidome )
```

Format

A long-format data frame with 48262 rows and 7 variables:

ID Participant number

Group Diagnosis of the type tumor: benign, cancer, or metastasis

Race Ethnic background of the participant

Stage Diagnosis of the stage of the tumor

Type Sub-type of the breast tumor. IDC: Invasive Ductal Carcinoma

Lipid_Name Name of the lipid. The names are in the format 'XY(C:D)', where 'XY' is the abbreviation of the lipid class, 'C' is the total number of carbon atoms in the fatty-acid chains, and 'D' is the total number of double-bonds in the fatty acid chains.

Lipid_Level Measured level of the lipid.

Source

This data is available at the NIH Common Fund's National Metabolomics Data Repository (NMDR) website, the Metabolomics Workbench, <https://www.metabolomicsworkbench.org>, where it has been assigned Project ID PR000742. The data can be accessed directly via its Project DOI: [10.21228/M8RX01](https://doi.org/10.21228/M8RX01). This work was supported by NIH grant, U2C- DK119886.

References

Purwaha, P., et al. Unbiased lipidomic profiling of triple-negative breast cancer tissues reveals the association of sphingomyelin levels with patient disease-free survival. *Metabolites* 8, 41 (2018) ([doi: 10.3390/metabo8030041](https://doi.org/10.3390/metabo8030041))

Examples

```
# Import the data set.
data( cancerlipidome )

# Convert the data into wide format, where each lipid is one column and
# each sample is one row.
cancerlipidome.wide <-
  tidyr::pivot_wider(
    data = cancerlipidome,
    names_from = Lipid_Name,
    values_from = Lipid_Level
  )
# Inspect the data frame.
# View( cancerlipidome.wide )
# Create a mapping of the lipid names.
names.mapping <-
  map_lipid_names( x = unique( cancerlipidome$"Lipid_Name" ) )
# Compute the regression models.
result.limma <-
  compute_models_with_limma(
    x = cancerlipidome.wide,
    dependent.variables = names.mapping$"Name",
    independent.variables = c( "Group" )
  )

# Create a figure of all lipids and factors.
figure.output <-
  heatmap_lipidome_from_limma(
    x = result.limma$"model",
    names.mapping = names.mapping,
    axis.x.carbons = FALSE,
    class.facet = "row",
    plot.all = TRUE,
    plot.individual = FALSE,
    print.figure = TRUE,
    scales = "free",
    space = "free"
  )

# Create individual figures for each factor.
figure.output <-
  heatmap_lipidome_from_limma(
    x = result.limma$"model",
    names.mapping = names.mapping,
    axis.x.carbons = FALSE,
    class.facet = "wrap",
```

```

    omit.class = "PA",
    plot.all = FALSE,
    plot.individual = TRUE,
    print.figure = FALSE,
    scales = "free",
    space = "free"
  )
# Print the figure of differences between cancer and benign tumors.
print( figure.output[[ "GroupCancer" ]] )

```

```
compute_F_test_with_limma
```

Compute an F-test for a multi-level factor independent variable

Description

Use this to achieve analysis of variance (ANOVA) or analysis of covariance (ANCOVA). The F-test is based on a model produced by the function [compute_models_with_limma](#). To use this function, first call the function [compute_models_with_limma](#).

Usage

```
compute_F_test_with_limma(x, p.adj.threshold = 0.05, print.table = FALSE)
```

Arguments

| | |
|------------------------------|---|
| <code>x</code> | (Required) list of models for which the test will be done. The F-test will be computed for the first independent variable that was specified in the <code>independent.variables</code> argument to the function compute_models_with_limma . |
| <code>p.adj.threshold</code> | (Optional) numeric value specifying the threshold for statistical significance of difference after correction for multiple testing. |
| <code>print.table</code> | (Optional) TRUE or FALSE: Print the results of the test? |

Value

List `x` supplemented by the results of the F-test.

See Also

[compute_models_with_limma](#) for the model computation step that is required prior to calling this function.

[compute_post_hoc_test_with_limma](#) for the pairwise post-hoc comparisons that may follow the F-test done with this function.

 compute_models_with_limma

Compute lipid-specific regression models

Description

Use this function to computing multiple regression models that can be directly supplied to the visualization functions of the 'lipidomeR'.

Usage

```
compute_models_with_limma(
  x,
  dependent.variables,
  independent.variables,
  random.effect = NULL,
  formula = NULL,
  F.test = FALSE,
  print.table1 = FALSE,
  scale.dependent.variables = TRUE,
  scale.independent.variables = FALSE,
  verbose = TRUE
)
```

Arguments

| | |
|------------------------------------|---|
| <code>x</code> | (Required) data matrix. |
| <code>dependent.variables</code> | (Required) vector of names of dependent variables. These should be the names of the lipids. |
| <code>independent.variables</code> | (Required) vector of names of the independent variables. These should be the names of the variables defining the experiment design. |
| <code>random.effect</code> | (Optional) name of a single variable specifying the random effect for a random-effects model. For instance, ID specifies a random effect as in <code>limma::duplicateCorrelation(..., block = x\$'ID')</code> . |
| <code>formula</code> | (Optional) character string of model formula in the format accepted by the function <code>model.matrix</code> and starting with <code>~</code> . Variables mentioned in the formula should be included in the <code>independent.variables</code> argument. For instance, <code>Group * Treatment</code> . |
| <code>F.test</code> | (Optional) TRUE or FALSE: Should an F-test for analysis of variance (ANOVA) or analysis of covariance (ANCOVA) be computed? |
| <code>print.table1</code> | (Optional) TRUE or FALSE: Should a summary table of the independent variables be printed? |

`scale.dependent.variables`
 (Optional) TRUE or FALSE: Should dependent variables be scaled to zero-mean and unit-variance prior to model fitting?

`scale.independent.variables`
 (Optional) TRUE or FALSE: Should independent variables be scaled to zero-mean and unit-variance prior to model fitting?

`verbose`
 (Optional) TRUE or FALSE: Print messages from the model fitting?

Value

List of regression results that can be directly supplied as an argument to the function [heatmap_lipidome_from_limma](#) and other visualization functions of the `lipidomeR`.

See Also

[heatmap_lipidome_from_limma](#) for visualizing the output of this function.

compute_post_hoc_test_with_limma

Compute pairwise post-hoc comparisons for a multi-level factor

Description

Use this function to achieve the post-hoc comparisons between the multiple levels of an independent variable. These comparisons follow analysis of variance (ANOVA) or analysis of covariance (ANCOVA). The pairwise comparisons are based on a result of an F-test produced by the function [compute_F_test_with_limma](#). To use this function, first call the functions [compute_models_with_limma](#) and [compute_F_test_with_limma](#) consecutively.

Usage

```
compute_post_hoc_test_with_limma(
  x,
  p.adj.threshold = NULL,
  remap.level.names = FALSE
)
```

Arguments

x (Required) list of models for which the test will be done. The pairwise comparisons will be computed using the levels of the first independent variable that was specified in the `independent.variables` argument to the function [compute_models_with_limma](#). The comparisons will be computed for the lipids that had an F-test result of statistical significance from the function [compute_F_test_with_limma](#).

p.adj.threshold

(Optional) numeric value specifying the threshold of statistical significance in the pairwise comparisons after a correction for multiple testing. We recommend to leave this argument unfilled, leading to the same threshold to be used as in the preceding F-test.

remap.level.names

(Optional) TRUE or FALSE: Should the levels of the factor independent variable be re-mapped? This feature can be used to solve a problem with the factor levels, which may occur in some versions of the limma package. We recommend to keep this argument unchanged from the default value.

Value

List `x` supplemented by the results of the pairwise post-hoc comparisons.

See Also

[compute_models_with_limma](#) for the model computation step that is required prior to calling this function.

[compute_F_test_with_limma](#) for the F-test step that is required prior to calling this function.

heatmap_lipidome

Create 'lipidomeR' heatmaps of arbitrary lipid-specific values.

Description

Use this function to create a heatmap of any lipid-specific values. Note: Use the function [heatmap_lipidome_from_limma](#) to create heatmaps of model statistics.

Usage

```
heatmap_lipidome(  
  x,  
  names.mapping,  
  axis.x.carbons = TRUE,  
  class.facet = "row",  
  fill.direction = "increasing",  
  fill.limits = c(0, 40),  
  fill.midpoint = 20,  
  melt.value.name = "CV",  
  melt.variable.name = NULL,  
  melt.x = TRUE,  
  range.min.N.carbons = 5,  
  range.min.N.double.bonds = 5,  
  scale.fill.log = NULL,  
  scales = "free_y",  
  space = "free",
```

```

    x.names = "row.names",
    x.variables = NULL
  )

```

Arguments

- x** (Required) named vector of numeric values to create a figure of. Names need to match to the argument names.mapping through the function `map_lipid_names`. Alternatively, a data frame can be supplied. In that case, set `melt.x = TRUE`.
- names.mapping** (Required) mapping of lipid names from the `map_lipid_names` function.
- axis.x.carbons** (Optional) TRUE or FALSE: Should the lipid size (i.e., number of carbon atoms in the fatty acid chain) be presented on the x-axis or y-axis?
- class.facet** (Optional) character string with possible values 'col', 'row' or 'wrap': Present lipid classes as panels organized into columns, rows or into a wrapped layout spanning over multiple rows and columns. The alternative 'wrap' is only available with `plot.individual = TRUE`.
- fill.direction** (Optional) TRUE or FALSE: Should color fill be in an increasing direction?
- fill.limits** (Optional) numeric vector of length two, indicating the limits of the fill scale.
- fill.midpoint** (Optional) numeric value specifying the midpoint of the fill scale.
- melt.value.name** (Optional) character string, specifying the name of the variable that will be shown as fill in the heatmap.
- melt.variable.name** (Optional) character string, specifying the name of of the variable that will be used to creating faceted sub-heatmaps.
- melt.x** (Optional) TRUE or FALSE: Should the argument x be molten by the function `melt` prior to plotting? Set `melt.x = TRUE`, if you want to plot a data frame instead of a vector of values. In that case, each column of the data frame will be plotted as an individual facet.
- range.min.N.carbons** (Optional) numeric value to specify the minimum range of the axis showing the lipid size (number of carbon atoms in the fatty acid chains). This value can be increased from the default value to improve readability in situations, where there are lipid classes with little or no variation in the lipid size.
- range.min.N.double.bonds** (Optional) numeric value to specify the minimum range of the axis showing the lipid saturation (number of double bonds in the fatty acid chains). This value can be increased from the default value to improve readability in situations, where there are lipid classes with little or no variation in the lipid saturation.
- scale.fill.log** (Optional) numeric value specifying the base of the logarithm, which will be used to creating a logarithmic scale for the fill scale of the plot.
- scales** (Optional) character string with possible values 'fixed', 'free', 'free_x' or 'free_y'. This argument specifies, whether the axes in multiple sub-heatmaps will be in the same scale ('fixed') or in a scale specific to each sub-figure. See the function `facet_grid` for details.

| | |
|-------------|---|
| space | (Optional) character string with possible values 'fixed', 'free', 'free_x' or 'free_y'. This argument specifies, whether the sub-heatmaps will be of identical size ('fixed') or not. |
| x.names | (Optional) character string specifying the name of the variable in the argument x, which will be used to matching the values to the argument names.mapping. Use this argument only together with melt.x = TRUE. |
| x.variables | (Optional) character vector specifying the names of the variables, which will be included as individual facets in the figure. Use this argument only together with melt.x = TRUE. |

heatmap_lipidome_from_limma

Create 'lipidomeR' heatmaps of model statistics

Description

Use this function to creating heatmaps of model statistics from the output of the [compute_models_with_limma](#) function or related functions ([compute_F_test_with_limma](#) and [compute_post_hoc_test_with_limma](#)).

Usage

```
heatmap_lipidome_from_limma(
  x,
  names.mapping = NULL,
  axis.x.carbons = TRUE,
  baseline.adjusted = FALSE,
  class.facet = "row",
  class.subset = NULL,
  F.test = FALSE,
  omit.class = NULL,
  omit.factor = NULL,
  order.factor = FALSE,
  p.val.thresholds = c(0.01, 0.05, 0.1),
  p.val.labels = c(8, 4, 3),
  p.val.label.bg.size = 2,
  p.val.label.size = 1,
  p.adj.method = "BH",
  plot.individual = FALSE,
  plot.all = TRUE,
  print.figure = TRUE,
  print.formula = TRUE,
  formula.width = 110,
  legend.key.size.multiplier = 2,
  range.min.N.carbons = 5,
  range.min.N.double.bonds = 5,
  scales = "fixed",
```

```

shadowtext = FALSE,
space = "free",
survival = FALSE,
verbose = FALSE,
wrap.contrast.name = TRUE
)

```

Arguments

- x** (Required) list of output from the [compute_models_with_limma](#) function or related functions.
- names.mapping** (Optional) mapping of lipid names from the [map_lipid_names](#) function.
- axis.x.carbons** (Optional) TRUE or FALSE: Should the lipid size (i.e., number of carbon atoms in the fatty acid chain) be presented on the x-axis or y-axis?
- baseline.adjusted** TRUE or FALSE: Is the model object that is specified as argument **x** a baseline-adjusted regression model? (Note: this model type is not yet provided in the package.)
- class.facet** (Optional) character string with possible values 'col', 'row' or 'wrap': Present lipid classes as panels organized into columns, rows or into a wrapped layout spanning over multiple rows and columns. The alternative 'wrap' is only available with `plot.individual = TRUE`.
- class.subset** (Optional) character vector specifying a subset of the lipid classes (e.g., `c('PC', 'SM', 'TG')`). Same outcome can be achieved with the argument `omit.class`.
- F.test** (Optional) TRUE or FALSE: Should the result of an F-test be visualized instead of individual model coefficients? Result from the function [compute_F_test_with_limma](#) needs to be provided as argument **x** for the [heatmap_lipidome_from_limma](#) function along with the argument `F.test = TRUE`.
- omit.class** (Optional) character vector of lipid classes omitted from the visualization (e.g., `c('PC', 'SM', 'TG')`).
- omit.factor** (Optional) character vector of lipid classes omitted from the visualization (e.g., `c('Time')`).
- order.factor** (Optional) TRUE or FALSE: Sort the factor panels alphabetically?
- p.val.thresholds** (Optional) numeric vector with increasing values of highlighting thresholds for multiple-testing-corrected p-values. For instance, `c(0.01, 0.05, 0.1)` gives distinct highlighting of associations in the ranges $p < 0.01$, $0.01 \geq p < 0.05$, and $0.05 \geq p < 0.1$. If the number of these categories is changed from the default value, also the argument `p.val.labels` needs to be modified to match it.
- p.val.labels** (Optional) numeric vector of point characters for the p-value highlighting categories specified in the argument `p.val.thresholds`. The values need to be compatible with the function [scale_shape_manual](#). If the number of these labels is changed from the default value, also the argument `p.val.thresholds` needs to be modified to match it.

| | |
|---|---|
| <code>p.val.label.bg.size</code> | (Optional) numeric value to scale the size of the colored background of the symbols that indicate values with statistical significance. The background is used to ensure that the white symbols are visible also when the color of a heatmap rectangle is of a bleak color. |
| <code>p.val.label.size</code> | (Optional) numeric value to scale the size of the symbols that indicate values with statistical significance. |
| <code>p.adj.method</code> | (Optional) name of the method to correct p-values for multiple testing. Accepted values are as in the function <code>stats::p.adjust()</code> . |
| <code>plot.individual</code> | (Optional) TRUE or FALSE: Create specific figures for each coefficient of the model? For models with many independent variables, this may take some time to complete. |
| <code>plot.all</code> | (Optional) TRUE or FALSE: Create a combined figure of all the coefficients of the model? For models with many independent variables, this may lead to too busy a figure. |
| <code>print.figure</code> | (Optional) TRUE or FALSE: Print created figure(s) to the standard output? Note that when <code>plot.individual = TRUE</code> , multiple figures will be printed consecutively. |
| <code>print.formula</code> | (Optional) TRUE or FALSE: Add the model formula as a title to the figure? We recommend to use <code>print.formula = TRUE</code> to make the result more understandable and reproducible. |
| <code>formula.width</code> | (Optional) numeric value to specify the width of a line in the model formula. Relevant only if <code>print.formula = TRUE</code> . |
| <code>legend.key.size.multiplier</code> | (Optional) numeric value to scale the size of the figure (key) legends. |
| <code>range.min.N.carbons</code> | (Optional) numeric value to specify the minimum range of the axis showing the lipid size (number of carbon atoms in the fatty acid chains). This value can be increased from the default value to improve readability in situations, where there are lipid classes with little or no variation in the lipid size. |
| <code>range.min.N.double.bonds</code> | (Optional) numeric value to specify the minimum range of the axis showing the lipid saturation (number of double bonds in the fatty acid chains). This value can be increased from the default value to improve readability in situations, where there are lipid classes with little or no variation in the lipid saturation. |
| <code>scales</code> | (Optional) character string with possible values 'fixed', 'free', 'free_x' or 'free_y'. This argument specifies, whether the axes in multiple sub-heatmaps will be in the same scale ('fixed') or in a scale specific to each sub-figure. See the function <code>facet_grid</code> for details. |
| <code>shadowtext</code> | (Optional) TRUE or FALSE: Should an alternative method of statistical significance highlighting be used in the figure? If TRUE, highlighting will be based on the function <code>geom_shadowtext</code> . |
| <code>space</code> | (Optional) character string with possible values 'fixed', 'free', 'free_x' or 'free_y'. This argument specifies, whether the sub-heatmaps will be of identical size ('fixed') or not. See the function <code>facet_grid</code> for details. |

survival (Optional) TRUE or FALSE: Is the model object that is specified as argument `x` a survival model? (Note: this model type is not yet provided in the package.)
verbose (Optional) TRUE or FALSE: Print information about the progress of the function call?
wrap.contrast.name (Optional) TRUE or FALSE: Wrap the name of a contrast to multiple lines to ensure readability?

Value

List of 'lipidomeR' heatmap figure(s).

See Also

[compute_models_with_limma](#) for computing the argument `x` for this function.

humanlipidome

Concentrations of lipids in a human plasma reference sample

Description

This data set contains concentrations of 403 named lipids in the 'National Institute of Standards and Technology Human Plasma Standard Reference Material' (NIST SRM 1950).

Usage

```
data( humanlipidome )
```

Format

A data frame with 403 rows and 2 variables:

Name Name of the lipid. The names are in the format 'XY(C:D)', where 'XY' is the abbreviation of the lipid class, 'C' is the total number of carbon atoms in the fatty-acid chains, and 'D' is the total number of double-bonds in the fatty acid chains.

Concentration Concentration of the lipid (umol/mL)

Source

This data is available at the NIH Common Fund's National Metabolomics Data Repository (NMDR) website, the 'Metabolomics Workbench', <https://www.metabolomicsworkbench.org>, where it has been assigned Project ID PR000004. The data can be accessed directly via its Project DOI: [10.21228/M8MW26](https://doi.org/10.21228/M8MW26). This work was supported by NIH grant, U2C- DK119886.

References

Quehenberger, O. et al. Lipidomics reveals a remarkable diversity of lipids in human plasma. *J Lipid Res.* 51, 3299-3305 (2010) (doi: [10.1194/jlr.M009449](https://doi.org/10.1194/jlr.M009449))

Examples

```

# Load the data set.
data( humanlipidome )
# Transform the concentrations into log-10 scale.
humanlipidome$"Concentration_log10_umol_per_mL" <-
  log10( humanlipidome$"Concentration" )
# Enumerate the lipid names into values.
names.mapping <- map_lipid_names( x = humanlipidome$"Name" )
# Create the lipidomeR heatmap of lipid concentrations.
heatmap_lipidome(
  x = humanlipidome[ , c( "Name", "Concentration_log10_umol_per_mL" ) ],
  names.mapping = names.mapping,
  class.facet = "wrap",
  x.names = "Name",
  fill.limits =
    range(
      x = humanlipidome$"Concentration_log10_umol_per_mL",
      na.rm = TRUE
    ),
  fill.midpoint =
    sum(
      range(
        x = humanlipidome$"Concentration_log10_umol_per_mL",
        na.rm = TRUE
      )
    ) / 2,
  melt.value.name = "Concentration_umol_per_mL_log10",
  scales = "free"
)

```

```
install_Bioconductor_dependencies
```

Install the Bioconductor dependencies for the 'lipidomeR'

Description

Run this function, if you encounter an error with missing packages, such as, [limma](#).

Usage

```
install_Bioconductor_dependencies(pkgs.bioconductor = c("limma"))
```

Arguments

pkgs.bioconductor

Character vector of Bioconductor packages to install.

| | |
|---------------|---|
| liverlipidome | <i>Levels of lipids in the human liver with or without non-alcoholic liver disease (NAFLD).</i> |
|---------------|---|

Description

This data set contains levels of 383 named lipids in 88 liver tissue samples.

Usage

```
data( liverlipidome )
```

Format

A long-format data frame with 33704 rows and 13 variables:

ID Participant number

Diagnosis Diagnosis of the liver: normal, steatosis, non-alcoholic steatohepatitis (NASH), or cirrhosis

Gender Gender of the participant

BMI Body-mass-index (BMI) of the participant

Ethnicity Ethnicity of the participant

Age Age of the participant

AST Aspartate aminotransferase blood test (U/l)

ALT Alanine aminotransferase blood test (U/l)

ALKP Alkaline phosphatase blood test (U/l)

TBIL Total bilirubin blood test (mg/dl)

Glucose Glucose blood test (mg/dl)

Type Sub-type of the breast tumor. IDC: Invasive Ductal Carcinoma

Lipid_Name Name of the lipid. The names are in the format 'XY(C:D)', where 'XY' is the abbreviation of the lipid class, 'C' is the total number of carbon atoms in the fatty-acid chains, and 'D' is the total number of double-bonds in the fatty acid chains.

Lipid_Level Measured level of the lipid.

Source

This data is available at the NIH Common Fund's National Metabolomics Data Repository (NMDR) website, the 'Metabolomics Workbench', <https://www.metabolomicsworkbench.org>, where it has been assigned Project ID PR000633. The data can be accessed directly via its Project DOI: [10.21228/M8MW26](https://doi.org/10.21228/M8MW26). This work was supported by NIH grant, U2C- DK119886.

References

Gorden, D. Lee, et al. Biomarkers of NAFLD Progression: a Lipidomics Approach to an Epidemic. *J Lip Res.* 56(3) 722-36 (2015) (doi: [10.1194/jlr.P056002](https://doi.org/10.1194/jlr.P056002))

Examples

```
# Load the data set.
data( liverlipidome )
# Convert the data into wide format, where each lipid is one column and
# each sample is one row.
liverlipidome.wide <-
  tidyr::pivot_wider(
    data = liverlipidome,
    names_from = Lipid_Name,
    values_from = Lipid_Level
  )
# Create a mapping of the lipid names.
names.mapping <-
  map_lipid_names( x = unique( liverlipidome$"Lipid_Name" ) )
# Compute the regression models.
result.limma <-
  compute_models_with_limma(
    x = liverlipidome.wide,
    dependent.variables = names.mapping$"Name",
    independent.variables = c( "Diagnosis" ),
    F.test = TRUE # Compute an F-test for a factor variable.
  )
# Compute the F-test.
result.limma <- compute_F_test_with_limma( x = result.limma )
# Print a figure of the F-test.

figure.output <-
  heatmap_lipidome_from_limma(
    x = result.limma,
    names.mapping = names.mapping,
    F.test = TRUE,
    axis.x.carbons = FALSE,
    class.facet = "wrap",
    plot.all = FALSE,
    plot.individual = TRUE,
    scales = "free",
    space = "free"
  )

# Compute pairwise post-hoc comparisons between the factor levels for
# the dependent variables (i.e., lipids) with a significant F-test result.
result.limma <-
  compute_post_hoc_test_with_limma(
    x = result.limma,
    remap.level.names = TRUE
  )
# Print a figure of all post-hoc comparisons.

figure.output <-
  heatmap_lipidome_from_limma(
    x = result.limma$"result.post.hoc.test",
    names.mapping = names.mapping,
```

```

    axis.x.carbons = FALSE,
    plot.all = TRUE,
    plot.individual = FALSE,
    scales = "free",
    space = "free"
  )

# Specify the contrasts of the post-hoc comparison that will be included
# in the figure.
contrasts.included <-
  c( "DiagnosisSteatosis", "DiagnosisNASH", "DiagnosisCirrhosis" )
# Get the omitted contrasts based on the above definition.
contrasts.omitted <-
  colnames( result.limma$result.post.hoc.test$"p.value" )[
    !(
      colnames( result.limma$result.post.hoc.test$"p.value" ) %in%
      contrasts.included
    )
  ]
# Find dependent variables (i.e., lipids) that have any significant
# difference.
has.any.significant <-
  apply(
    X =
      result.limma$result.post.hoc.test$"p.value"[
        ,
        contrasts.included
      ],
    MAR = 2,
    FUN = p.adjust,
    method = "BH"
  )
has.any.significant <-
  rownames(
    has.any.significant[
      apply(
        X = has.any.significant < 0.05,
        MAR = 1,
        FUN = any
      ),
    ]
  )
# Include in the figure only lipid classes that have at least four
# significant differences.
classes.included <-
  names(
    which(
      table(
        names.mapping[
          make.names( has.any.significant ), "Class"
        ]
      ) > 4
    )
  )

```

```
)
classes.omitted <- unique( names.mapping$"Class" )
classes.omitted <-
  classes.omitted[ !( classes.omitted ) %in% classes.included ]
# Print a figure of the selected post-hoc-comparisons.
figure.output <-
  heatmap_lipidome_from_limma(
    x = result.limma$"result.post.hoc.test",
    names.mapping = names.mapping,
    axis.x.carbons = FALSE,
    omit.class = classes.omitted,
    omit.factor = contrasts.omitted,
    plot.all = TRUE,
    plot.individual = FALSE,
    scales = "free",
    space = "free"
  )
```

| | |
|-----------------|---|
| map_lipid_names | <i>Enumerate lipid names into values about lipid class, size and saturation</i> |
|-----------------|---|

Description

Use this function to create a mapping of the lipids to values, which can be used to creating 'lipidomeR' heatmaps.

Usage

```
map_lipid_names(x)
```

Arguments

x (Required) character vector of lipid names. The names are expected in the format 'XY(C:D)', where 'XY' is the abbreviation of the lipid class, 'C' is the total number of carbon atoms in the fatty-acid chains, and 'D' is the total number of double-bonds in the fatty acid chains.

Value

Data frame with lipid names in various formats for R and the enumerated values of lipid class (Class), lipid size (N. carbons) and lipid unsaturation (N. double. bonds).

See Also

[heatmap_lipidome_from_limma](#) for creating 'lipidomeR' heatmaps of regression model results based on the output of this function.

[heatmap_lipidome](#) for creating 'lipidomeR' heatmaps of any lipid-specific values based on the output of this function.

Index

- * **NASH**
 - liverlipidome, [14](#)
 - * **NIST**
 - humanlipidome, [12](#)
 - * **blood**
 - humanlipidome, [12](#)
 - * **breast**
 - cancerlipidome, [2](#)
 - * **cancer**
 - cancerlipidome, [2](#)
 - * **carcinoma**
 - cancerlipidome, [2](#)
 - * **concentration**
 - humanlipidome, [12](#)
 - * **datasets**
 - cancerlipidome, [2](#)
 - humanlipidome, [12](#)
 - liverlipidome, [14](#)
 - * **data**
 - cancerlipidome, [2](#)
 - humanlipidome, [12](#)
 - liverlipidome, [14](#)
 - * **diagnosis**
 - cancerlipidome, [2](#)
 - liverlipidome, [14](#)
 - * **disease**
 - liverlipidome, [14](#)
 - * **ductal**
 - cancerlipidome, [2](#)
 - * **human**
 - cancerlipidome, [2](#)
 - humanlipidome, [12](#)
 - liverlipidome, [14](#)
 - * **invasive**
 - cancerlipidome, [2](#)
 - * **lipidome**
 - cancerlipidome, [2](#)
 - humanlipidome, [12](#)
 - liverlipidome, [14](#)
 - * **lipidomics**
 - cancerlipidome, [2](#)
 - humanlipidome, [12](#)
 - liverlipidome, [14](#)
 - * **lipids**
 - cancerlipidome, [2](#)
 - humanlipidome, [12](#)
 - liverlipidome, [14](#)
 - * **liver**
 - liverlipidome, [14](#)
 - * **molecule**
 - cancerlipidome, [2](#)
 - humanlipidome, [12](#)
 - liverlipidome, [14](#)
 - * **non-alcoholic**
 - liverlipidome, [14](#)
 - * **plasma**
 - humanlipidome, [12](#)
 - * **reference**
 - humanlipidome, [12](#)
 - * **steatohepatitis**
 - liverlipidome, [14](#)
 - * **tissue**
 - cancerlipidome, [2](#)
 - liverlipidome, [14](#)
 - * **tumor**
 - cancerlipidome, [2](#)
- cancerlipidome, [2](#)
- compute_F_test_with_limma, [4](#), [6](#), [7](#), [9](#), [10](#)
- compute_models_with_limma, [4](#), [5](#), [6](#), [7](#), [9](#), [10](#), [12](#)
- compute_post_hoc_test_with_limma, [4](#), [6](#), [9](#)
- facet_grid, [8](#), [11](#)
- geom_shadowtext, [11](#)
- heatmap_lipidome, [7](#), [17](#)

heatmap_lipidome_from_limma, [6](#), [7](#), [9](#), [10](#),
[17](#)
humanlipidome, [12](#)
install_Bioconductor_dependencies, [13](#)
liverlipidome, [14](#)
map_lipid_names, [8](#), [10](#), [17](#)
melt, [8](#)
model.matrix, [5](#)
scale_shape_manual, [10](#)