Package 'metaprotr'

October 13, 2022

Title Metaproteomics Post-Processing Analysis

Version 1.2.2

Date 2021-01-28

Author Aaron Millan-Oropeza [aut, cre],
Catherine Juste [aut, ctb],
Ariane Bassignani [aut, ctb],
Céline Henry [aut, ctb]

Maintainer Aaron Millan-Oropeza <aaron.ibt@gmail.com>

Description
Set of tools for descriptive analysis of metaproteomics data generated from high-throughput
mass spectrometry instruments. These tools allow to cluster peptides and proteins abundance,
expressed as spectral counts, and to manipulate them in groups of metaproteins. This information
can be represented using multiple visualization functions to portray the global metaproteome
landscape and to differentiate samples or conditions, in terms of abundance of metaproteins,
taxonomic levels and/or functional annotation. The provided tools allow to implement flexible
analytical pipelines that can be easily applied to studies interested in metaproteomics analysis.

License GPL-3

Encoding UTF-8

URL https://forgemia.inra.fr/pappso/metaprotr

LazyData true

Depends R (>= 3.5.0)

Imports ade4, dendextend, dplyr, ggforce, ggrepel, reshape2, stringr,
tidyverse

RoxygenNote 7.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2021-02-05 08:10:02 UTC

R **topics documented:**

add_kegg	*add_kegg*

Description

Integrates a database containing the functional annotation of the identified metaproteins into a list
defined as "spectral_count_object". The proteins from the "spectral_count_object" must contain
taxonomic information. The functional annotation was obtained from the Kyoto Encyclopedia of
Genes and Genomes (KEGG) Orthology database. This database contains the molecular functions
represented in terms of functional orthologs (KO terms). Check KEGG for more details.

Usage

```
add_kegg(
  spectral_count_object,
  annotation_db,
  taxonomic_db,
  metaproteome_origin,
  protein_file,
  peptide_file,
  text_to_filter = "HUMAN",
  taxonomic_levels_allowed = 1
)
```

Arguments

spectral_count_object

List defined as "spectral_count_object" containing the abundance of the elements (groups, subgroups or peptides) expressed as spectral counts and organized by taxonomic levels. The format of this object is similar to that generated from the function "crumble_taxonomy".

annotation_db   Dataframe containing the functional annotation of the proteins. This dataframe must contain two variables: i) "gene_name": indicating the same protein names to those present in the variable "Accession" from the "peptides_proteins", third dataframe in the list defined as "spectral_count_object"; and, ii) "ko": indicating the KEGG Orthology code assigned to a given protein. An example can be found in this repository.

taxonomic_db   Dataframe containing the taxonomic information for each protein. The first column must contain the same identifiers of those present in the column "Accession" from the dataframe "peptides_proteins" of the "metaproteome_object". Two additional columns have to be present: i) one named "organism" containing the name of the strain assigned to a given protein; and ii) the other named "species.genus.family.order.class.phylum.superkingdom". The taxonomic classification can be obtained from a tool of sequences aligment and must be ordered as follows: species, genus, family, order, class, phylum and superkingdom. The characters inside must be concatenated by a comma without spaces (ex. "Streptococcus anginosus,Streptococcus,Streptococcaceae,Lactobacillales,Bacilli,Firmicutes,Bacteria"). An example can be found in this repository.

metaproteome_origin

List defined as "metaproteome_object" generated from the function 'load_protspeps'.

protein_file   Character indicating the location of a txt file containing the list of proteins generated in X!TandemPipeline using an adapted iterative approach described by Bassignani, 2019. Separation between columns should be indicated by tabulation. For more details regarding data input check format examples.

peptide_file   Character indicating the location of a txt file containing peptides abundances expressed as spectral counts. This file is generated from X!TandemPipeline using an adapted iterative approach described by Bassignani, 2019. Separation between columns should be indicated by tabulation. For more details regarding data input check format examples.

text_to_filter   Character containig a part of text to be searched in the "Description" of the
                  protein file. All the elements containing this character will be removed. The
                  default value was set to "HUMAN".

taxonomic_levels_allowed
                  Numeric value indicating the maximal number of taxonomic levels allowed per
                  spectral group or subgroup (in function of the type of spectral data). The default
                  value is set to 1.

Value

A list defined as "spectral_count_object" with the functional annotation added to the identified
proteins. A new column is added to the dataframe "peptides_proteins". Two quality control plot are
also generated, one with the number of taxonomic entities per spectral level and another with the
number of KO terms per spectral level.

Examples

```
## Not run:

# Download functional and taxonmical annotation db: https://zenodo.org/record/3997093#.X0UYI6Zb_mE
meta99_full_taxo <- read.csv2("full_taxonomy_MetaHIT99.tsv", header= TRUE, sep="\t")
kegg_db <- read.csv2("hs_9_9_igc_vs_kegg89.table", header = TRUE, sep = "\t")

# Files with spectral abundance and proteins list from X!Tandempipeline
protein_file <- "your/specific/location/protein_list.txt"
peptide_file <- "your/specific/location/peptide_counting.txt"
metadata_file <- "your/location/metadata.csv"

metaproteome_origin <- load_protspeps(protein_file, peptide_file, metadata_file)

SCsgp_species <- crumble_taxonomy(SC_subgroups, "species")

SCsgp_species_annot <- add_kegg(
  SCsgp_species,
  kegg_db,
  meta99_full_taxo,
  metaproteome_origin,
  protein_file,
  peptide_file,
  text_to_filter = "HUMAN"
)


## End(Not run)
```

---

add_taxonomy   *add_taxonomy*

Description

Integrates the database containing the taxonomic classification of the identified proteins in a "metaproteome_object". The taxonomic classification is previously obtained by aligment algorithms and must include seven taxonomic levels assigned to a given protein: species, genus, family, order, class, phylum and superkingdom

Usage

```
add_taxonomy(metaproteome_object, taxonomic_database)
```

Arguments

metaproteome_object

> List defined as "metaproteome_object" containing proteins and peptides abundances. The format of this object is similar to that generated from the function "load_protspeps".

taxonomic_database

> Dataframe containing the taxonomic information for each protein. The first column must contain the same identifiers of those present in the column "Accession" from the dataframe "peptides_proteins" of the "metaproteome_object". Two additional columns have to be present: i) one named "organism" containing the name of the strain assigned to a given protein; and ii) the other named "species.genus.family.order.class.phylum.superkingdom". The taxonomic classification can be obtained from a tool of sequences aligment and must be ordered as follows: species, genus, family, order, class, phylum and superkingdom. The characters inside must be concatenated by a comma (ex."Streptococcus anginosus,Streptococcus,Streptococcaceae,Lactobacillales,Bacilli,Firmicutes,Bacteria"). An example can be found in this repository.

Value

A "metaproteome_object", which is a list of six elements with format similar to that generated from the function "load_protspeps". An additional column containing the taxonomic annotation is added to the dataframe named "peptides_proteins".

Examples

```
## Not run:

# Download taxonmical annotation db: https://zenodo.org/record/3997093#.X0UYI6Zb_mE
meta99_full_taxo <- read.csv2("MetaHIT99_best_hit_taxo_complete.tsv", header = TRUE, sep="\t")

# Files with spectral abundance and proteins list from X!Tandempipeline
protein_file <- "your/specific/location/protein_list.txt"
peptide_file <- "your/specific/location/peptide_counting.txt"
metadata_file <- "your/location/metadata.csv"

metaproteome <- load_protspeps(proteins_file, peptides_file, metadata_file)

metaproteome_taxo <- add_taxonomy(metaproteome, meta99_full_taxo)
```

```
## End(Not run)
```

---

crumble_taxonomy   *crumble_taxonomy*

Description

Generates a list of four elements defined as "spectral_count_object" containing taxonomic classi-
fication. The first element is a dataset that contains the spectral counts abundance organized by a
provided taxonomic level. The possible taxonomic levels are: species, genus, family, order, class,
phylum or superkingdom.

Usage

```
crumble_taxonomy(spectral_count_object, taxonomic_level, filter_rate = 1)
```

Arguments

spectral_count_object

        List defined as "spectral_count_object" containing dataframes with abundance
        expressed as spectral counts and organized by peptides, subgroups or groups.
        The format of this object is similar to that generated with the function "getsc_specific".
        Taxonomy must be added previously with "add_taxonomy" function.

taxonomic_level

        Character indicating the taxonomic level to which the spectral abundance will
        be arranged in the samples of the "spectral_count_object". The possible op-
        tions are: "species", "genus", "family", "order", "class", "phylum" or "superk-
        ingdom".

filter_rate   Numeric value between 0 and 1 that indicates the minimal rate of consensual an-
        notation desired by the user within each level of the spectral category (subgroup
        or group). This rate is defined as the ratio between the number of the most fre-
        quent annotation entity ("species", "genus", "family", "order", "class", "phylum"
        or "superkingdom") divided by the total number of entities within each level of
        the spectral category under study (subgroup or group). The default value is set
        to 1, if 100 % of consensus is desired.

Value

A list of four elements defined as "spectral_count_object", the first element is a dataframe with
abundance expressed as spectral counts of entities (peptides, subgroups or groups) organized by
the provided taxonomic level. The second element is a dataframe that contains the experiment
information. The third element is a dataframe containing the information of peptides with their
associated proteins. And the fourth element is a character indicating the type of object generated.

Examples

```
data(fecal_waters)

superkingdom_fecalwaters <- crumble_taxonomy(fecal_waters, "superkingdom")

phylum_fecalwaters <- crumble_taxonomy(fecal_waters, "phylum")

class_fecalwaters <- crumble_taxonomy(fecal_waters, "class")

order_fecalwaters <- crumble_taxonomy(fecal_waters, "order")

family_fecalwaters <- crumble_taxonomy(fecal_waters, "family")

genus_fecalwaters <- crumble_taxonomy(fecal_waters, "genus")

species_fecalwaters <- crumble_taxonomy(fecal_waters, "species")
```

---

export_ipath3	*export_ipath3*

Description

Exports the KEGG Orthology (KO) terms in the adapted format to be used in the tool iPATH3. The exported data is obtained from a "spectral_count_object" containing the functional annotation of the identified proteins from one condition or sample.

Usage

```
export_ipath3(
  spectral_count_object,
  type_export,
  target_variable,
  sample_condition,
  hexadecimal_color,
  taxonomic_levels = NULL,
  force = FALSE
)
```

Arguments

spectral_count_object

List defined as "spectral_count_object" containing protein abundance expressed as spectral counts by a taxonomic level. The functional annotation must be added to this object. The format of this object is similar to that generated from the function "add_kegg".

type_export	Character indicating the type of export to be used. The possible options are: i) "all" that selects all the KO terms from a given sample or a given condition; and, ii) "selection" that extracts the KO terms present in selected taxonomic entities (one or more).
target_variable	
	Character indicating the column name from metadata containing the condition or sample to be analyzed.
sample_condition	
	Atomic vector indicating the sample from which the functional information will be extracted.
hexadecimal_color	
	Character indicating the color to be used in iPATH3, this value must be indicated in hexadecimal format (eg. #ff0000).
taxonomic_levels	
	Optional vector indicating the taxonomic levels from which the KO terms will be extracted. This option is needed only if the type of export is "selection".
force	Logic value set at FALSE by default in order to ask permission to create a file in the workstation of the user.

Value

A csv file containing the KO terms present in a given sample or condition. The content of this file can be inserted directly in the tool iPATH3. The width of the lines in iPATH3 will be displayed by the percentage of spectra in the selected sample or condition. In this way, KO terms belonging to a given taxonomic level are represented in three intervals based on their abundace: i) below 2 percent, i) between 2 to 10 percent, or i) above 10 percent.

Examples

```
data(species_annot_fw)

export_ipath3(
   species_annot_fw,
   "all",
   "SampleID",
   "Q1_prot",
   "#840AA3"
)

taxonomic_entities <- c("Bacteroides caccae", "Coprococcus catus", "Merdimonas faecis")
export_ipath3(
   species_annot_fw,
   "selection",
   "SC_name",
   "FW2",
   "#28c1df",
   taxonomic_entities
)
```

---

export_robject   *export_robject*

Description

Exports one of the dataframes present in a "metaproteome_object" or in "spectral_count_object". The export extensions can be RDATA or RDS.

Usage

```
export_robject(entry_object, data_exported, format_data, force = FALSE)
```

Arguments

entry_object   A "metaproteome_object" or a "spectral_count_object" with similar format to that generated with the functions "load_protspeps" or "getsc_specific", respectively.

data_exported   Character indicating the type of data to be exported from a "metaproteome_object" or a "spectral_count_object". The possible options are: i) "proteins", ii) "peptides", iii) "pepProts", iv) "spectral", v) "spectral_percent", vi) "metadata".

format_data   Character indicating the file extension, this can be either "RDATA" or "RDS".

force   Logic value set at FALSE by default in order to ask permission to create object in the workstation of the user.

Value

A file with the extension "RDATA" or "RDS" containing the information from the selected dataframe from a "metaproteome_object" or a "spectral_count_object".

Examples

```
data(fecal_waters)
export_robject(fecal_waters, "pepProts", "rdata")

data(species_fw)
export_robject(species_fw, "spectral", "rds")
```

---

export_vennlists   *export_vennlists*

Description

Exports as csv files the elements (groups, subgroups, peptides or taxonomic levels) generated from the function "plot_venn".

Usage

```
export_vennlists(venn_lists_object, output_repo = NULL, force = FALSE)
```

Arguments

venn_lists_object

List defined as "venn_lists_object" containing the elements (peptides, subgroups, groups or taxonomic elements) generated with the function "plot_venn".

output_repo   Character indicating the path of a previously created directory where the lists will be exported. This parameter is optional.

force   Logic value set at FALSE by default in order to ask permission to create csv files in the workstation of the user.

Value

csv files containing the elements present on each logic section (specific and intersections) from the list defined as "venn_lists_object".

Examples

```
data(venn_methods)

export_vennlists(venn_methods)
```

---

fecal_waters   *fecal waters*

Description

Data containing the abundance of 474 metaproteins expressed in spectral counts. Data generated from an Orbitrap Fusion Lumos Tribrid Mass Spectrometer. The dataset contains the metaproteomes from three extraction methods: i) "Q" for Qiagen, ii) "FW" for fecal waters, and iii) "Q_FW" for the mixture of Qiagen and fecal waters. Data generated in the context of the project Microbiome Rapid Access (Université Paris-Saclay).

Usage

```
data(fecal_waters)
```

Format

A list of four elements defined as "spectral_count_object", generated with the function "getsc_specific"

SC_subgroups dataframe with 9 samples containing 474 subgroups or metaproteins with abundance expressed as spectral counts

metadata information related to the 9 samples from the experiment

peptides_proteins information related to each of the 1557 identified peptides

type_object character indicating the type of object

```
filter_shared                    filter_shared
```

Description

Keeps the elements from a "spectral_count_object" that are common to all levels of a provided explanatory variable. This variable MUST correspond to the name of ONE column of the dataframe "metadata" (eg. conditions or samples). This function allows to identify the elements that are common to several conditions or samples.

Usage

```
filter_shared(spectral_count_object, metadata_feature)
```

Arguments

```
spectral_count_object
```
> List containing dataframes with proteomics elements whose abundance is expressed as spectral counts and are organized by peptides, subgroups, groups or taxonomic levels. The format of this object is similar to that generated from the functions "getsc_specific" and "crumble_taxonomy".

```
metadata_feature
```
> Character indicating the name of one explanatory variable (ONE column name) of the dataframe "metadata".

Value

A list defined as "spectral_count_object" in which the elements (groups, subgroups, peptides or taxonomic levels) of a given variable (condition or samples) from metadata will have at least ONE spectra per variable.

Examples

```
data(fecal_waters)
data(species_fw)

common_elements_per_sample <- filter_shared(fecal_waters, "SampleID")

common_elements_per_condition <- filter_shared(species_fw, "Condition")
```

---

filter_text   *filter_text*

Description

Matches the entities containing a given chain of characters inside an explanatory variable (column name) of the dataframe "peptides_proteins" from a "spectral_count_object". Based on the user's decision, the peptides, subgroups, groups or taxonomic levels containig the provided chain of characters will be kept or discarded in a newly generated object.

Usage

```
filter_text(spectral_count_object, pepsprots_feature, text_to_filter, decision)
```

Arguments

spectral_count_object

> List containing dataframes with proteomics elements whose abundance is expressed as spectral counts and are organized by peptides, subgroups, groups or taxonomic levels. The format of this object is similar to that generated from the functions "getsc_specific" and "crumble_taxonomy".

pepsprots_feature

> Character indicating the name of one explanatory variable (ONE column name) of the dataframe "peptides_proteins".

text_to_filter   Character containig the text to be searched in the "pepsprots_feature" content.

decision   Character indicating wether the elements containing the matched text will be kept or dirscarted. The two allowed option are: "keep" or "discard".

Value

A list defined as "spectral_count_object" with or without the elements (peptides, subgroups, groups, taxonomic items) that matched the provided text in a given variable of the "peptides_proteins" dataframe.

Examples

```
data(fecal_waters)
data(species_fw)

cysteine_alkylations <- filter_text(fecal_waters, "Modifs", "57.02146", "keep")

exclude_medimonas <- filter_text(species_fw, "organism", "Merdimonas faecis BR31", "discard")
```

filter_unshared   *filter_unshared*

Description

Keeps the elements from a "spectral_count_object" that are specific to one level of a provided explanatory variable. This variable MUST correspond to the name of ONE column of the dataframe "metadata" (eg. conditions or samples). This function allows to identify the elements that are specific to one condition or one sample.

Usage

```
filter_unshared(spectral_count_object, metadata_feature)
```

Arguments

spectral_count_object

> List containing dataframes with proteomics elements whose abundance is expressed as spectral counts and are organized by peptides, subgroups, groups or taxonomic levels. The format of this object is similar to that generated from the functions "getsc_specific" and "crumble_taxonomy".

metadata_feature

> Character indicating the name of one explanatory variable (ONE column name) of the dataframe "metadata".

Value

A list defined as "spectral_count_object" with the specific elements per sample or condition, having at least one spectra.

Examples

```
data(fecal_waters)
data(species_fw)
```

```
specific_elements_per_sample <- filter_unshared(fecal_waters, "SampleID")

specific_elements_per_condition <- filter_unshared(species_fw, "Condition")
```

---

getsc_specific   *getsc_specific*

Description

Returns the abundances, expressed as spectral counts (SC), of the different peptides, subgroups
(also referred as metaprotein) or groups within the samples of the experiment. The abundance
corresponds to the sum of SC of the specific peptides present in a given subgroup or group. See
X!TandemPipeline for more details concerning the grouping algorithm.

Usage

```
getsc_specific(metaproteome_object, type_SCspecific)
```

Arguments

metaproteome_object

> List defined as "metaproteome_object" with dataframes having a similar format
> to that generated from "load_protspeps" function. It contains metaproteomics
> data such as peptide and protein abundances and sample information.

type_SCspecific

> Character indicating the type of data to be returned. The possible options are: i)
> "sc_specific_peptides" for peptides, or ii) "sc_groups" for groups, or iii) "sc_subgroups"
> for subgroups. For more details of the identification algorithm (peptides, sub-
> groups, groups) check X!TandemPipeline.

Value

A list of four elements defined as "spectral_count_object". The first element is a dataframe orga-
nized in function of peptides OR subgroups (also referred as metaproteins) OR groups. The entities
of this dataframe have their abundance expressed as SC from specific peptides. The second ele-
ment is a dataframe of metadata containing the experiment information. The third element is a
dataframe containing the information of peptides with their associated proteins. The fourth element
is a character indicating the type of object generated.

Examples

```
## Not run:

# From a given "metaproteome_object" add the taxonomic classification
```

```
metaproteome <- load_protspeps(proteins_file, peptides_file, metadata_file)
metaproteome_taxo <- add_taxonomy(metaproteome, meta99_full_taxo)

# Organize proteomics data by peptides OR subgroups OR groups

SC_specific_peptides <- getsc_specific(metaproteome_taxo, 'sc_specific_peptides')
SC_specific_groups <- getsc_specific(metaproteome_taxo, 'sc_groups')
SC_specific_subgroups <- getsc_specific(metaproteome_taxo, 'sc_subgroups')


## End(Not run)
```

---

identify_differences   *identify_differences*

Description

Shows the most differential taxonomic elements between two conditions or samples from a list
defined as "spectral_count_object" with taxonomic classification. These elements are those with an
absolute log2 (condition + 1 / reference + 1) > 3. If a given condition has several replicates the
mean value is taken into account.

Usage

```
identify_differences(
  spectral_count_object,
  target_variable,
  list_conditions,
  filter_ratio = 3,
  force = FALSE
)
```

Arguments

spectral_count_object

> List defined as "spectral_count_object" containing dataframes with abundance
> expressed as spectral counts and organized by taxonomic levels. The format of
> this object is similar to that generated from the function "crumble_taxonomy".

target_variable

> Character indicating the variable name containing the conditions or samples to
> be compared. This value corresponds to the name of one column from metadata.

list_conditions

> Atomic vector indicating two conditions to be compared. The first element will
> be considered as the reference.

filter_ratio   Numeric value indicating the fold change filter to be considered for the pairwise
> comparison. The minimal value can be a fold change of 1.25. The default value
> is set at 3.

force   Logic value set at FALSE by default in order to ask permission to create an
> object in the workstation of the user.

Value

Barplots (pdf) and a csv file with the defferential taxonomic elements between TWO conditions or sample. These elements are those that fulfill the ratio log2 (condition + 1 / reference + 1) > filter_ratio.

Examples

```
data(species_fw)
identify_differences(species_fw, "Methods", c("S", "S_EF"))

identify_differences(species_fw, "Methods", c("EF", "S_EF"), filter_ratio = 1.3)
```

---

inspect_sample_elements

inspect_sample_elements

Description

Displays a graph that indicates the number of common elements from a "spectral_count_object" (peptides, subgroups, groups or taxonomic entities) per sample. This function is useful to distinguish heterogeneity between samples in an experimental design.

Usage

```
inspect_sample_elements(spectral_count_object, force = FALSE)
```

Arguments

spectral_count_object
                 List defined as "spectral_count_object" containing dataframes with abundance
                 expressed as spectral counts. The spectral data can be organized by peptides,
                 subgroups, groups or taxonomic levels.

force   Logic value set at FALSE by default in order to ask permission to create a pdf
                 file in the workstation of the user.

Value

Barplots (pdf) ilustrating the common spectral elements (peptides, subgroups, groups, taxonomic elements) per sample in a "spectral_count_object".

Examples

```
data(fecal_waters)
inspect_sample_elements(fecal_waters)
```

---

load_protspeps   *load_protspeps*

Description

Loads three files: i) peptides abundances expressed as spectral counts, ii) proteins information, and iii) metadata of the mass spectrometry samples. Combines the three files into a "metaproteome_object", a list containing these dataframes.

Usage

```
load_protspeps(protein_file, peptide_file, metadata_file)
```

Arguments

protein_file	Character indicating the location of a txt file containing the list of proteins generated in X!TandemPipeline using an adapted iterative approach described by Bassignani, 2019. Separation between columns should be indicated by tabulation. For more details regarding data input check format examples.
peptide_file	Character indicating the location of a txt file containing peptides abundances expressed as spectral counts. This file is generated from X!TandemPipeline using an adapted iterative approach described by Bassignani, 2019. Separation between columns should be indicated by tabulation. For more details regarding data input check format examples.
metadata_file	Character indicating the location of a csv file containing the samples information. The following columns names MUST be present: "SC_name" (sample ids assigned by the user), "msrunfile" (name of samples as indicated in mass spectrometry files and in the columns of peptide_file) and "SampleID" (codes indicating the experimental group). Additional columns containing complementary information can be added by the user (ex. replicates, order of injection, etc.). Separation between columns should be indicated by tabulation. For more details regarding data input check format examples.

Value

A "metaproteome_object", which is a list of six elements containing: 1) dataframe of the protein identifiers, 2) dataframe of the peptide identifiers, 3) dataframe containing the information of peptides with their associated proteins, 4) dataframe of metadata containing the experiment information, 5) dataframe of spectral counts per peptide on each sample, 6) character indicating the type of object generated.

Examples

```
## Not run:

protein_file <- "location/peptides_abundances.csv"
peptide_file <- "location/proteins_list.csv"
metadata <- "location/metadata.csv"
metaproteome <- load_protspeps(protein_file, peptide_file, metadata_file)


## End(Not run)
```

---

plot_dendocluster   *plot_dendocluster*

Description

Draws a dendogram where samples are clustered based on the number of elements present on each
sample from a "spectral_count_object". This graph is constructed based on Spearman correlations
transformed into distances and plotted with the logic of the package dendextend.

Usage

```
plot_dendocluster(
  spectral_count_object,
  target_variable,
  file_title,
  hclust_method = "ward.D",
  correlation_method = "spearman",
  force = FALSE
)
```

Arguments

spectral_count_object

>   List defined as "spectral_count_object" containing dataframes with abundance
>   expressed as spectral counts from peptides, subgroups, groups or taxonomic
>   levels. The format of this object is similar to that generated from the functions
>   "getsc_specific" and "crumble_taxonomy".

target_variable

>   Character indicating the name of one column from metadata. The different lev-
>   els in this column will be represented as different colors in the final dendogram.

file_title   Character indicating the name of the generated file.

hclust_method   Character indicating the agglomeration method to be used for the hierarchical
                  clustering. The possible methods are discribed on hclust. The default method is
                  "ward.D".

correlation_method

> Character indicating the correlation coeficient to be computed. The possible options are discribed in the function cor. The default value is "spearman".

force
> Logic value set at FALSE by default in order to ask permission to create a pdf file in the workstation of the user.

Value

A dendogram plot (pdf) indicating the number of elements per sample.

Examples

```
data(fecal_waters)
str(fecal_waters$metadata)

plot_dendocluster(fecal_waters, "Condition", "title_dendogram")

plot_dendocluster(fecal_waters, "Condition", "title_dendogram",
   hclust_method = "mcquitty")

plot_dendocluster(fecal_waters, "Condition", "title_dendogram_groups",
   correlation_method = "pearson")
```

---

plot_fulltaxo   *plot_fulltaxo*

Description

Provides the number of taxonomic entities per sample in the different taxonomic levels. The taxonomic levels are: species, genus, family, order, class, phylum and superkingdom.

Usage

```
plot_fulltaxo(spectral_count_object, force = FALSE)
```

Arguments

spectral_count_object

> List defined as "spectral_count_object" containing dataframes with format similar to that generated with the function "getsc_specific". This object contains abundances expressed as spectral counts from peptides, subgroups (metaproteins) or groups. Taxonomy must have previously been added with the function "add_taxonomy".

force
> Logic value set at FALSE by default in order to ask permission to create a pdf and a csv file in the workstation of the user.

Value

Bar plots (pdf) and csv file with the number of taxonomic species, genus, family, order, class, phylum and superkingdom per sample. An additional csv file is generated providing the rate of assignment. The rate of assigment corresponds to the ratio between the number of the most frequent annotation ("species", "genus", "family", "order", "class", "phylum" or "superkingdom") and the total number of elements within each level of the spectral category under study (subgroup or group). The csv file is generated only when the "spectral_count_object" is organized by subgroup or by group.

Examples

```
data(fecal_waters)
plot_fulltaxo(fecal_waters)
```

---

plot_intensities   *plot_intensities*

Description

Draws violin plots containing the abundance intensities expressed as spectral counts per level (peptides, subgroups, groups or taxonomic entities) in provided samples or conditions from a "spectral_count_object". If the provided conditions have several replicates the mean value is taken into account.

Usage

```
plot_intensities(
  spectral_count_object,
  target_variable,
  image_title = NULL,
  force = FALSE
)
```

Arguments

spectral_count_object

List defined as "spectral_count_object" containing dataframes with abundance expressed as spectral counts from peptides, subgroups, groups or taxonomic levels. The format of this object is similar to that generated from the functions "getsc_specific" and "crumble_taxonomy".

target_variable

Character indicating the name of one column from metadata, the column must contain the conditions to be displayed.

image_title	Character indicating the title to be displayed in the generated image.
force	Logic value set at FALSE by default in order to ask for permission to create a pdf file in the workstation of the user.

Value

Violin plots (pdf) indicating the spectral counts of the different levels (peptides, subgroups, groups or taxonomic entities) per sample or condition.

Examples

```
data(fecal_waters)
plot_intensities(fecal_waters, "SC_name", "Title to display inside the plot")

data(species_fw)
plot_intensities(species_fw, "Condition", "Abundance per condition")
```

---

plot_intensities_ratio

plot_intensities_ratio

Description

Generates a scatter plot of the log2 (ratio + 1) between two conditions considering the spectral counts of each entity (peptides, subgroups, groups or taxonomic levels) from a "spectral_count_object". If a given condition has several replicates the mean value is taken into account.

Usage

```
plot_intensities_ratio(
  spectral_count_object,
  target_variable,
  list_conditions,
  force = FALSE
)
```

Arguments

spectral_count_object

List defined as "spectral_count_object" containing dataframes with abundance expressed as spectral counts organized by peptides, subgroups, groups or taxonomic levels. The format of this object is similar to that generated from the functions "getsc_specific" and "crumble_taxonomy".

target_variable

> Character indicating the variable name containing the conditions to be compared. This value corresponds to the name of one column from metadata.

list_conditions

> Atomic vector indicating two conditions to A be compared. The first element is considered as the reference (denominator) for ratio calculations.

force   Logic value set at FALSE by default in order to ask permission to create object in the workstation of the user.

Value

A scatter plot (pdf) indicating the log2 (ratio + 1) of the entities (peptides, soubgroups, groups, taxonomic) between the two conditions provided.

Examples

```
data(fecal_waters)

plot_intensities_ratio(fecal_waters, "Methods", c("EF", "S"))

plot_intensities_ratio(fecal_waters, "SC_name", c("Q1", "Q2"))
```

---

  plot_pca   *plot_pca*

Description

Performs a Principal Components Analysis (PCA) from the spectral counts of the entities (peptides, subgroups, groups or taxonomic elements) in a "spectral_count_object" with or without taxonomy. PCA decomposition of high dimensional data allows to observe global effects in two dimensions. For more details of the used function check dudi.pca from ade4.

Usage

```
plot_pca(spectral_count_object, colors_var, pc_components, force = FALSE)
```

Arguments

spectral_count_object

> List described as "spectral_count_object" containing dataframes with abundance expressed as spectral counts from peptides, subgroups, groups or taxonomic levels. The format of this object is similar to that generated from the functions "getsc_specific" and "crumble_taxonomy". The PCA projections will be applied to these observations.

`colors_var`	Character indicating the name of one column from metadata. The samples will be represented in different colors in function of the levels of this variable (ex. conditions).
`pc_components`	Two numeric values indicating two principal components to be analyzed.
`force`	Logic value set as FALSE by default in order to ask permission to create a file in the workstation of the user.

Value

A pdf file containing the results of PCA applied to the two provided principal components. Including a bar plot indicating the percentage of variance per principal component.

Examples

```
data(fecal_waters)
plot_pca(fecal_waters, "Methods", c(1, 2))

data(species_fw)
plot_pca(species_fw, "Methods", c(1, 3))

data(species_annot_fw)
plot_pca(species_annot_fw, "Condition", c(1, 2))
```

---

plot_pietaxo *plot_pietaxo*

Description

Generates a pie chart with taxonomic distribution of one selected sample or condition. If the provided condition has several replicates the mean value is taken into account.

Usage

```
plot_pietaxo(
  spectral_count_object,
  target_variable,
  sampling,
  filter_percent = 1,
  force = FALSE
)
```

Arguments

spectral_count_object

> list defined as "spectral_count_object" containing dataframes with spectral counts abundance of the samples organized by taxonomy (species, genus, family, order, class, phylum or superkingdom). This object is generated with the function "crumble_taxonomy".

target_variable

> Character indicating the name of one column from metadata. This column must contain the identifiers of the sample or condition to be plotted.

sampling

> Character indicating the name of sample or condition to be plotted. This character must be present in the "target_variable".

filter_percent

> Optional numeric value between 0 and 99 that sets the minimal percentage of spectral counts at which the taxonomic elements will be displayed. The elements whose values are lower than this number will be gathered and displayed as "others". The default value is set at 1.

force

> Logic value set at FALSE by default in order to ask permission to create files in the workstation of the user.

Value

A pie chart (pdf) and a csv file with the taxonomic distribution of one sample or one condition. In the csv file, all the elements have their: a) spectral counts, b) percentage of spectral count in the sample, c) the taxonomic name and d) the taxonomic elements that were assiged as 'others' in function of the filter provided.

Examples

```
data(species_fw)

plot_pietaxo(species_fw, "Methods", "S")

plot_pietaxo(species_fw, "SC_name", "Q1")
```

---

plot_stackedtaxo   *plot_stackedtaxo*

Description

Generates stacked barplots of the spectral counts distributions among the different taxonomic entities ("species", "genus", "family", "order", "class", "phylum" or "superkingdom") within the samples or conditions of a "spectral_count_object" with taxonomy. If the provided conditions have several replicates the mean value is taken into account.

Usage

```
plot_stackedtaxo(
  spectral_count_object,
  target_variable,
  bars_data,
  filter_percent = 1,
  force = FALSE
)
```

Arguments

`spectral_count_object`

List defined as "spectral_count_object" containing dataframes with spectral counts abundance organized by taxonomy (species, genus, family, order, class, phylum or superkingdom). This object is generated with the function "crumble_taxonomy".

`target_variable`

Character indicating the name of one column from metadata. The stacked barplots will be ordered by the levels of this variable.

`bars_data`   Character indicating the type of labels to be displayed in the stacked bars. The possible options are "percent" or "numbers".

`filter_percent` Optional numeric value between 0 and 99 that sets the minimal percentage of spectral counts at which the taxonomic elements will be displayed. The elements whose values are lower than this number will be gathered and displayed as "others". The default value is set at 1.

`force`   Logic value set at FALSE by default in order to ask permission to create a pdf in the workstation of the user.

Value

Barplots (pdf) of the taxonomic distribution of the samples present in a "spectral_count_object" with taxonomic levels.

Examples

```
data(species_fw)

plot_stackedtaxo(species_fw, 'SampleID', 'percent', 2)

plot_stackedtaxo(species_fw, 'SC_name', 'numbers')
```

---

plot_venn *plot_venn*

Description

Generates a Venn diagram comparing up to 3 conditions. The lists of elements for each condition are also returned as a "venn_lists_object".

Usage

```
plot_venn(
  spectral_count_object,
  target_variable,
  list_conditions,
  force = FALSE
)
```

Arguments

spectral_count_object

List defined as "spectral_count_object" containing dataframes with abundance expressed as spectral counts from peptides, subgroups, groups or taxonomic levels. The format of this object is similar to that generated from the functions "getsc_specific" and "crumble_taxonomy".

target_variable

Character indicating the name of the explanatory variable that contains the conditions to be compared. This value corresponds to the name of one column from the metadata dataframe.

list_conditions

Atomic vector indicating the conditions to be compared. The provided elements (2 or 3) must be present in the variable indicated as "target_variable".

force   Logic value set at FALSE by default in order to ask permission to create a pdf file in the workstation of the user.

Value

A Venn diagram (pdf) and a list defined as "venn_list_object" containing the elements (peptides, soubgroups, groups or taxonomic levels) for each logical section of the Venn diagram (specific and intersections).

Examples

```
data(fecal_waters)
venn_QFW1_Q1 <- plot_venn(fecal_waters, "SC_name", c("Q1_FW1", "Q1"))

data(species_fw)
```

```
venn_all <- plot_venn(species_fw, "Methods", c("S_EF", "S", "EF"))
```

---

remove_element   *remove_element*

Description

Removes elements from a "spectral_count_object". These elements can be: i) samples, ii) peptides, iii) proteins, iv) soubgroups, v) groups, vi) sequences, vii) species, viii) genus, ix) family, x) order, xi) class, xii) phylum or xiii) superkingdom.

Usage

```
remove_element(spectral_count_object, target_variable, list_elements)
```

Arguments

spectral_count_object

List defined as "spectral_count_object" containing dataframes with abundance expressed as spectral counts from peptides, subgroups, groups or taxonomic levels. The format of this object is similar to that generated from the functions "getsc_specific" and "crumble_taxonomy."

target_variable

Character indicating the variable that contains the elements to be removed. The options are : i) "peptides", ii) "proteins", iii) "soubgroups", iv) "groups", v) "sequences", vi) "species", vii) "genus", viii) "family", ix) "order", x) "class", xi) "phylum" or xii) "superkingdom". To select xiii) "samples", it should be indicated the name of ONE column from metadata.

list_elements   Atomic vector indicating the elements to be removed. For "samples", indicate the element(s) present in the provided variable from metadata. For "peptides", "proteins", "subgroups" and "groups" provide the X!Tandem nomenclature. For "sequences", provide the peptide sequences expressed as aminoacids. For any taxonomic level, provide the taxonomic entities.

Value

A list defined as "spectral_count_object" without the elements provided in the second argument of the function.

Examples

```
data(fecal_waters)
data(species_fw)

data_selected_samples <- remove_element(fecal_waters, "Methods", c("S_EF", "EF"))
```

```
data_selected_peptides <- remove_element(fecal_waters, "peptides", c("pepa3c417", "pepd4664a1"))

data_selected_proteins <- remove_element(species_fw, "proteins", c("a3.a9.a1", "a5.b81.a1"))

data_selected_subgroups <- remove_element(species_fw, "subgroups", c("a3.a9", "b73.a5"))

data_selected_groups <- remove_element(species_fw, "groups", c("a3", "b34", "c231"))

data_selected_sequences <- remove_element(species_fw, "sequences", c("AQLNFGGTIENVVIRDEFPLEK"))
```

---

select_element   *select_element*

Description

Keeps specific elements from a "spectral_count_object". These elements can be: i) samples, ii)
peptides, iii) proteins, iv) soubgroups, v) groups, vi) sequences, vii) species, viii) genus, ix) family,
x) order, xi) class, xii) phylum or xiii) superkingdom.

Usage

```
select_element(spectral_count_object, target_variable, list_elements)
```

Arguments

spectral_count_object

         List defined as "spectral_count_object" containing dataframes with abundance
         expressed as spectral counts from peptides, subgroups, groups or taxonomic
         levels. The format of this object is similar to that generated from the functions
         "getsc_specific" and "crumble_taxonomy."

target_variable

         Character indicating the variable that contains the elements to be kept. The
         options are : i) "peptides", ii) "proteins", iii) "soubgroups", iv) "groups", v)
         "sequences", vi) "species", vii) "genus", viii) "family", ix) "order", x) "class",
         xi) "phylum" or xii) "superkingdom". To select xiii) "samples", it should be
         indicated the name of ONE column from metadata.

list_elements   Atomic vector indicating the elements to be kept For "samples", indicate the
         element(s) present in the provided variable from metadata. For "peptides", "pro-
         teins", "subgroups" and "groups" provide the X!Tandem nomenclature. For "se-
         quences", provide the peptide sequences expressed as aminoacids. For any tax-
         onomic level, provide the taxonomic entities.

Value

A list defined as "spectral_count_object" with the elements provided in the second argument of the
function.

Examples

```
data(fecal_waters)
data(species_fw)

data_selected_samples <- select_element(fecal_waters, "Methods", c("S_EF", "EF"))

data_selected_peptides <- select_element(fecal_waters, "peptides", c("pepa3c417", "pepd4664a1"))

data_selected_proteins <- select_element(species_fw, "proteins", c("a3.a9.a1", "a5.b81.a1"))

data_selected_subgroups <- select_element(species_fw, "subgroups", c("a3.a9", "b73.a5"))

data_selected_groups <- select_element(species_fw, "groups", c("a3", "b34", "c231"))

data_selected_sequences <- select_element(species_fw, "sequences", c("AQLNFGGTIENVVIRDEFPLEK"))
```

---

species_annot_fw   *species_annot_fw*

Description

Data containing the abundance of 15 species expressed in spectral counts and with functional annotation. Data generated from an Orbitrap Fusion Lumos Tribrid Mass Spectrometer. The dataset contains the metaproteomes from three extraction methods: i) "Q" for Qiagen, ii) "FW" for fecal waters, and iii) "Q_FW" for the mixture of Qiagen and fecal waters. Data generated in the context of the project Microbiome Rapid Access (Université Paris-Saclay).

Usage

```
data(species_annot_fw)
```

Format

A list of four elements defined as "spectral_count_object" with functional annotation, generated with the function "add_kegg"

SC_subgroups dataframe with 9 samples containing 15 especies with abundance expressed as spectral counts

metadata information related to the 9 samples from the experiment

peptides_proteins information related to each of the 1315 identified peptides

type_object character indicating the type of object

species_fw	*species_fw*

Description

Data containing the abundance of 17 species expressed in spectral counts. Data generated from an Orbitrap Fusion Lumos Tribrid Mass Spectrometer. The dataset contains the metaproteomes from three extraction methods: i) "Q" for Qiagen, ii) "FW" for fecal waters, and iii) "Q_FW" for the mixture of Qiagen and fecal waters. Data generated in the context of the project Microbiome Rapid Access (Université Paris-Saclay).

Usage

```
data(species_fw)
```

Format

A list of four elements defined as "spectral_count_object" with taxonomic annotation generated with the function "crumble_taxonomy"

SC_subgroups dataframe with 9 samples containing 17 especies with abundance expressed as spectral counts

metadata information related to the 9 samples from the experiment

peptides_proteins information related to each of the 1557 identified peptides

type_object character indicating the type of object

venn_methods	*venn methods*

Description

Data containing the subgroups for each logical section of the Venn diagram (specific and intersections) from two methods of extraction. The extraction methods are: i) "S" for Qiagen, and ii) "S_EF" for the mixture of Qiagen and fecal waters.

Usage

```
data(venn_methods)
```

Format

A list of six elements defined as "venn_lists_object", generated with the function "plot_venn"

S_EF 385 subgroups found in the method 'S_EF'

S 242 subgroups found the method 'S'

intersection 201 subgroups in common between the methods 'S' and 'S_EF'

Specific in S_EF 184 specific subgroups found in the method 'S_EF'

S 41 specific subgroups found in the method 'S'

type_object character indicating the type of object

Index