Package ‘schtools’

January 16, 2023

Title Schloss Lab Tools for Reproducible Microbiome Research

Version 0.4.0

Date 2023-01-15

Description A collection of useful functions and example code created and used by the Schloss Lab for reproducible microbiome research. Perform common tasks like read files created by mothur <https://mothur.org/>, tidy up your microbiome data, and format R Markdown documents for publication. See the website <http://www.schlosslab.org/schtools/> for more information, documentation, and examples.

License MIT + file LICENSE

URL http://www.schlosslab.org/schtools/, https://github.com/SchlossLab/schtools

BugReports https://github.com/SchlossLab/schtools/issues

Imports dplyr, formatR, glue, rlang, stringr, tidyr

Suggests covr, cowplot, ggplot2, ggtext, here, hexSticker, hrbrthemes, knitr, purrr, readr, rmarkdown, showtext, testthat

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Author Kelly Sovacool [aut, cre] (<https://orcid.org/0000-0003-3283-829X>),
Nick Lesniak [aut] (<https://orcid.org/0000-0001-9359-5194>),
Patrick Schloss [aut],
Sarah Lucas [ctb] (<https://orcid.org/0000-0003-1676-5801>),
Courtney Armour [ctb] (<https://orcid.org/0000-0002-5250-1224>),
Jacqueline Moltzau [ctb] (<https://orcid.org/0000-0003-1398-5980>),
Andrew Hannah [ctb],
Nielson Baxter [ctb],
Alyxandria Schubert [ctb],
Kathryn Iverson [ctb]

Maintainer Kelly Sovacool <sovacool@umich.edu>

Repository CRAN

Date/Publication 2023-01-16 07:50:02 UTC

R topics documented:

`calc_relabun` *Calculate OTU relative abundances from a shared file*

Description

Calculate OTU relative abundances from a shared file

Usage

```
calc_relabun(abs_abun_dat)
```

Arguments

`abs_abun_dat`	a data frame from reading in a shared file. Should contain a `Group` column for sample names, `Otu` columns for absolute counts of each OTU, and rows as each sample.

Value

a new data frame with OTU relative abundances in long format.

Author(s)

Kelly Sovacool <`sovacool@umich.edu`>

Examples

```
shared_dat <- readr::read_tsv(system.file("extdata", "test.shared",
  package = "schtools"
))
shared_dat %>% calc_relabun()
```

`close_enough` *Check whether two numeric vectors are close enough for gov't work.*

Description

This is like `dplyr::near()` except with much less precision.

Usage

```
close_enough(x, y, tol = 10^-3)
```

Arguments

`x`	a numeric vector
`y`	another numeric vector
`tol`	tolerance (default: `10^-3`.)

Value

`TRUE` if all numbers are near enough within the tolerance, otherwise `FALSE`

Author(s)

Kelly Sovacool <`sovacool@umich.edu`>

Examples

```
close_enough(0.0004, 0)
close_enough(0.8887, 0.8884)
close_enough(1, 2)
```

`format_number` *Format human-readable numbers.*

Description

Pastes formatted `x` if numeric, otherwise `x` unmodified. Circumvents R's automatic scientific notation. If a number is nearly whole (see `is_nearly_whole()`), it is rounded to have zero decimal places. Otherwise, numbers >= 1 are rounded to 1 decimal place; numbers < 1 are rounded to have 2 significant digits.

Usage

```
format_number(x, nsmall = 1, signif_precise = 2)
```

Arguments

`x`	inline code
`nsmall`	number of digits after the decimal point to round to when `x` is not nearly whole but `x >= 1`.
`signif_precise`	number of significant digits to use when `x` is not nearly whole

Value

formatted `x` if numeric, otherwise `x` unmodified.

Author(s)

Pat Schloss <`pschloss@umich.edu`>

Kelly Sovacool <`sovacool@umich.edu`>

Examples

```
format_number(0.0256)
format_number(.Machine$double.eps^0.5)
format_number(100000.08)
format_number(1.00000000000000000001)
format_number("this is a string")
```

`get_wildcards_tbl` *Get the Snakemake wildcards as a tibble*

Description

Get the Snakemake wildcards as a tibble

Usage

```
get_wildcards_tbl()
```

Value

a tibble of wildcards, with columns as names and rows as values

Author(s)

Kelly Sovacool <`sovacool@umich.edu`>

`inline_hook` *Inline hook for knitr to paste human-readable numbers and nice lists.*

Description

Inline hook for knitr to paste human-readable numbers and nice lists.

Usage

```
inline_hook(x)
```

Arguments

`x`	just about anything

Value

a string where each element in `x` is separated by a comma and numbers are in a human-readable format.

Author(s)

Kelly Sovacool <`sovacool@umich.edu`>

Pat Schloss <`pschloss@umich.edu`>

Examples

```
inline_hook(c(1.2993992, 0.03, 1000))
inline_hook(c("cats", "dogs"))
```

`is_nearly_whole` *Checks whether a number is near to a whole number*

Description

Checks whether a number is near to a whole number

Usage

```
is_nearly_whole(x)
```

Arguments

`x`	a numeric

Value

`TRUE` or `FALSE`

Author(s)

Kelly Sovacool <`sovacool@umich.edu`>

Examples

```
is_nearly_whole(.Machine$double.eps^0.5)
is_nearly_whole(.Machine$double.eps^0.6)
is_nearly_whole(1)
```

`is_nondesc` *Check whether all elements given are sorted in non-descending order*

Description

Check whether all elements given are sorted in non-descending order

Usage

```
is_nondesc(...)
```

Arguments

`...`	anything!

Value

`TRUE` if the elements are sorted in non-descending order, otherwise `FALSE`

Author(s)

Kelly Sovacool <`sovacool@umich.edu`>

Examples

```
is_nondesc(1, 2, 3)
is_nondesc(c(1, 2), 3)
is_nondesc(6, 4, 1)
is_nondesc("a", "b", "c")
is_nondesc(c("z", "y"))
```

`load_deps` *Install & load packages*

Description

Install & load packages

Usage

```
load_deps(...)
```

Arguments

`...`	package names to install & load

Author(s)

Kelly Sovacool <`sovacool@umich.edu`>

`log_snakemake` *Save output, messages, warnings, and errors to the Snakemake log file*

Description

This function checks whether a log file was specified in the Snakemake rule. If so, it directs any output, messages, warnings, or errors to the rule-specific log file. See the Snakemake documentation on log files and R scripts for more details.

Usage

```
log_snakemake(quiet = TRUE)
```

Arguments

`quiet`	Silence messages about the status of the snakemake object and log file (default: `TRUE`).

Author(s)

Kelly Sovacool <`sovacool@umich.edu`>

Examples

```
# The Snakemake object doesn't exist, so nothing happens
log_snakemake(quiet = FALSE)
```

`parse_tax` *Convert taxonomy strings into dataframe of labels based on taxonomic classification*

Description

Convert taxonomy strings into dataframe of labels based on taxonomic classification

Usage

```
parse_tax(dat)
```

Arguments

`dat`	dataframe from mothur taxonomy file with columns `OTU`, `Size`, and `Taxonomy`

Value

a wide dataframe with taxonomic labels

Author(s)

Nick Lesniak, <`nlesniak@umich.edu`>

Examples

```
taxonomy_filepath <- system.file("extdata",
  "test.taxonomy",
  package = "schtools"
)
taxonomy_tbl <- read_tax(taxonomy_filepath)
head(taxonomy_tbl)
```

`paste_oxford_list` *Create a prose string from a list or vector*

Description

The word 'and' is inserted before the last element and an Oxford comma is used.

Usage

```
paste_oxford_list(x)
```

Arguments

`x`	a list or vector

Value

a string where each element in `x` is separated by a comma

Author(s)

Pat Schloss <`pschloss@umich.edu`>

Kelly Sovacool <`sovacool@umich.edu`>

Examples

```
paste_oxford_list(1:3)
paste_oxford_list(c("cats", "dogs", "turtles"))
```

`pool_taxon_counts` *Pool OTU counts at a particular taxonomic level*

Description

Enables comparing analyses at different taxonomic resolutions, as seen in doi:10.1128/mbio.03161-21. Implementation adapted from here.

Usage

```
pool_taxon_counts(otu_shared_dat, otu_tax_dat, taxon_level)
```

Arguments

`otu_shared_dat`	data frame created from a shared file at the OTU level.
`otu_tax_dat`	data frame created from a taxonomy file at the OTU level. Must be from the same dataset as the shared file.
`taxon_level`	taxonomic level to pool OTUs into. Options: "kingdom", "phylum", "class", "order", "family", "genus". This should be the name of a column in `otu_tax_dat` as a character string.

Value

a shared data frame with the OTUs at the specified `taxon_level` and a corresponding taxonomy dataframe with new OTU numbers.

Author(s)

Kelly Sovacool, `<sovacool@umich.edu>`

Pat Schloss `<pschloss@umich.edu>`

Examples

```
tax_dat <- read_tax(system.file("extdata", "test.taxonomy",
  package = "schtools"
))
shared_dat <- readr::read_tsv(system.file("extdata", "test.shared",
  package = "schtools"
))
pool_taxon_counts(shared_dat, tax_dat, "genus")
pool_taxon_counts(shared_dat, tax_dat, "family")
pool_taxon_counts(shared_dat, tax_dat, "phylum")
```

`read_dist` *Read in lower left triangular matrix from file*

Description

Read in lower left triangular matrix from file

Usage

```
read_dist(dist_filename)
```

Arguments

`dist_filename`	filename of lower left triangular matrix (.dist)

Value

distance matrix as a tibble

Author(s)

Nick Lesniak, <`nlesniak@umich.edu`>

Examples

```
dist_filepath <- system.file("extdata",
  "sample.final.thetayc.0.03.lt.ave.dist",
  package = "schtools"
)
dist_tbl <- read_dist(dist_filepath)
head(dist_tbl)
```

`read_tax` *Read in a taxonomy file and parse it to a wide dataframe*

Description

Read in a taxonomy file and parse it to a wide dataframe

Usage

```
read_tax(taxonomy_filename, sep = "\t")
```

Arguments

`taxonomy_filename`	filename of taxonomy file
`sep`	Character that separates fields of the taxonomy file. (Default: `\t`).

Value

dataframe of taxonomic labels, formatted by `parse_tax()`

Author(s)

Nick Lesniak, <`nlesniak@umich.edu`>

Kelly Sovacool, <`sovacool@umich.edu`>

Examples

```
taxonomy_filepath <- system.file("extdata",
  "test.taxonomy",
  package = "schtools"
)
taxonomy_tbl <- read_tax(taxonomy_filepath)
head(taxonomy_tbl)
```

`schtools` *schtools: Schloss Lab Tools for Reproducible Microbiome Research*

Description

A collection of useful functions and example code created and used by the Schloss Lab for reproducible microbiome research. Perform common tasks like read files created by mothur, tidy up your microbiome data, and format rmarkdown documents for publication. See the website for more information, documentation, and examples.

`set_knitr_opts` *Set knitr chunk options & inline hook*

Description

Call this function in the setup chunk of your R Markdown files.

Usage

```
set_knitr_opts()
```

Author(s)

Pat Schloss <`pschloss@umich.edu`>

Kelly Sovacool <`sovacool@umich.edu`>

`theme_lucas` *Sarah's go-to theme for ggplot2*

Description

Requires the `hrbrthemes` package and the `PT Sans` and `PT Sans Narrow` fonts from Google Fonts.

Usage

```
theme_lucas()
```

Value

list of ggproto objects

Author(s)

Sarah Lucas <`salucas@umich.edu`>

Examples

```
library(ggplot2)
library(showtext)

# run once to download the PT Sans fonts
font_add_google(name = "PT Sans", family = "PT Sans")
font_add_google(name = "PT Sans Narrow", family = "PT Sans Narrow")
showtext_auto()

# make a plot with theme_lucas()
ggplot(mtcars) +
  aes(x = mpg, y = wt, color = cyl) +
  geom_point() +
  theme_lucas()
```

`theme_sovacool` *Kelly's go-to theme for ggplot2*

Description

Uses `ggplot2::theme_bw()` and removes margins.

Usage

```
theme_sovacool()
```

Value

list of ggproto objects

Author(s)

Kelly Sovacool <`sovacool@umich.edu`>

Examples

```
library(ggplot2)
ggplot(mtcars) +
  aes(x = mpg, y = wt, color = cyl) +
  geom_point() +
  theme_sovacool()
```

Index