

Package ‘yatah’

March 1, 2020

Title Yet Another TAXonomy Handler

Version 0.1.0

Description Provides functions to manage taxonomy when lineages are described with strings and ranks separated with special patterns like ``|*__` or ``;*__`.

License GPL-3

URL <https://github.com/abichat/yatah>, <https://abichat.github.io/yatah>

BugReports <https://github.com/abichat/yatah/issues>

Depends R (>= 2.10)

Imports ape, purrr, stats, stringr

Suggests dplyr, ggtree, knitr, rmarkdown, spelling, testthat (>= 2.1.0)

VignetteBuilder knitr

biocViews

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.0.2.9000

NeedsCompilation no

Author Antoine Bichat [aut, cre] (<<https://orcid.org/0000-0001-6599-7081>>)

Maintainer Antoine Bichat <a.bichat@yahoo.fr>

Repository CRAN

Date/Publication 2020-03-01 15:50:02 UTC

R topics documented:

.allchr	2
.ranks	2
abundances	3
all_clades	3
depth	4
error_lineage	4
is_clade	4
is_lineage	5
is_rank	6
last_clade	6
last_rank	7
taxtable	8
taxtree	8

Index **10**

.allchr	<i>Characters allowed in lineages</i>
---------	---------------------------------------

Description

Characters allowed in lineages

Usage

.allchr

Format

An object of class character of length 1.

.ranks	<i>Ranks</i>
--------	--------------

Description

Named vector of ranks

Usage

.ranks

Format

An object of class character of length 8.

abundances	<i>Abundance table for 199 samples.</i>
------------	---

Description

A dataset containing the abundances of 1585 lineages among 199 patients.

Usage

```
abundances
```

Format

A data.frame with 1585 rows and 200 variables:

lineages lineage (string)

XXX abundance of each lineage in the sample XXX (double)

Source

[Zeller et al., 2014](#), [Pasolli et al., 2017](#)

all_clades	<i>Extract all clades present in the lineages</i>
------------	---

Description

Extract all clades present in the lineages

Usage

```
all_clades(lineage, simplify = TRUE)
```

Arguments

lineage string. Vector of lineages.

simplify logical. Should the output be a vector or a dataframe?

Value

The clades present in the lineage. Vector of ordered strings or data.frame.

Examples

```
lineage1 <- "k__Bacteria|p__Verrucomicrobia|c__Verrucomicrobiae"
lineage2 <- "k__Bacteria|p__Firmicutes|c__Clostridia"
all_clades(c(lineage1, lineage2))
all_clades(c(lineage1, lineage2), simplify = FALSE)
```

depth	<i>Common depth</i>
-------	---------------------

Description

Throw an error if depth is not the same across lineages.

Usage

```
depth(lineage)
```

Arguments

lineage	string. Vector of lineages.
---------	-----------------------------

error_lineage	<i>Throw error if the string is not a lineage</i>
---------------	---

Description

Throw error if the string is not a lineage

Usage

```
error_lineage(string)
```

Arguments

string	string to be tested as lineage.
--------	---------------------------------

is_clade	<i>Test if a lineage belongs to a clade</i>
----------	---

Description

Test if a lineage belongs to a clade

Usage

```
is_clade(
  lineage,
  clade,
  rank = c(".", "kingdom", "phylum", "class", "order", "family", "genus", "species",
    "strain")
)
```

Arguments

lineage	string. Vector of lineages.
clade	string.
rank	string. One of c("kingdom", "phylum", "class", "order", "family", "genus", "species", "strain") with partial matching.

Details

If rank is set to ., clade is looked for among all ranks.

Value

logical.

Examples

```
lineage1 <- "k__Bacteria|p__Verrucomicrobia|c__Verrucomicrobiae"
lineage2 <- "k__Bacteria|p__Firmicutes|c__Clostridia"
is_clade(c(lineage1, lineage2), clade = "Verrucomicrobia", rank = "phylum")
is_clade(c(lineage1, lineage2), clade = "Clostridia")
```

is_lineage	<i>Test if a string is a lineage</i>
------------	--------------------------------------

Description

Test if a string is a lineage

Usage

```
is_lineage(string)
```

Arguments

string	string to be tested as lineage.
--------	---------------------------------

Value

A logical.

Examples

```
is_lineage("k__Bacteria|p__Firmicutes|c__Clostridia|o__Clostridiales")
```

is_rank	<i>Test if a lineage goes down to a specified rank</i>
---------	--

Description

Test if a lineage goes down to a specified rank

Usage

```
is_rank(
  lineage,
  rank = c("kingdom", "phylum", "class", "order", "family", "genus", "species",
           "strain")
)
```

Arguments

lineage	string. Vector of lineages.
rank	string. One of c("kingdom", "phylum", "class", "order", "family", "genus", "species", "strain") with partial matching.

Value

logical.

Examples

```
lineage1 <- "k__Bacteria|p__Verrucomicrobia|c__Verrucomicrobiae"
lineage2 <- "k__Bacteria|p__Firmicutes|c__Clostridia"
is_rank(c(lineage1, lineage2), "class")
is_rank(c(lineage1, lineage2), "order")
```

last_clade	<i>Extract the last clade of a lineage</i>
------------	--

Description

Extract the last clade of a lineage

Usage

```
last_clade(lineage, same = TRUE)
```

Arguments

lineage	string. Vector of lineages.
same	logical. Does the lineage have the same depth? Default to TRUE.

Value

A string. The last clades of the given lineages.

Examples

```
lineage1 <- "k__Bacteria|p__Verrucomicrobia|c__Verrucomicrobiae"  
lineage2 <- "k__Bacteria|p__Firmicutes|c__Clostridia"  
last_clade(c(lineage1, lineage2))
```

last_rank	<i>Extract the last rank of a lineage</i>
-----------	---

Description

Extract the last rank of a lineage

Usage

```
last_rank(lineage, same = TRUE)
```

Arguments

lineage	string. Vector of lineages.
same	logical. Does the lineage have the same depth? Default to TRUE.

Value

A string. The last rank of the given lineages.

Examples

```
lineage1 <- "k__Bacteria|p__Verrucomicrobia|c__Verrucomicrobiae"  
lineage2 <- "k__Bacteria|p__Firmicutes|c__Clostridia"  
last_rank(c(lineage1, lineage2))
```

taxtable	<i>Taxonomic table</i>
----------	------------------------

Description

Compute taxonomic table from lineages.

Usage

```
taxtable(lineage)
```

Arguments

lineage string. Vector of lineages.

Details

Duplicated lineages are removed.

Value

A data.frame with columns corresponding to different ranks.

Examples

```
lineage1 <- "k__Bacteria|p__Verrucomicrobia|c__Verrucomicrobiae"
lineage2 <- "k__Bacteria|p__Firmicutes|c__Clostridia"
lineage3 <- "k__Bacteria|p__Firmicutes|c__Bacilli"
taxtable(c(lineage1, lineage2, lineage3))
```

taxtree	<i>Taxonomic tree</i>
---------	-----------------------

Description

Compute taxonomic tree from taxonomic table.

Usage

```
taxtree(table, collapse = TRUE, lineage_length = 1, root = "")
```

Arguments

table dataframe.
collapse logical. Should node with one child be vanished? Default to TRUE.
lineage_length double. Lineage length from the root to the leaves. Default to 1.
root character. Name of the root if there is no natural root.

Value

A phylo object.

Examples

```
lineage1 <- "k__Bacteria|p__Verrucomicrobia|c__Verrucomicrobiae"  
lineage2 <- "k__Bacteria|p__Firmicutes|c__Clostridia"  
lineage3 <- "k__Bacteria|p__Firmicutes|c__Bacilli"  
table <- taxtable(c(lineage1, lineage2, lineage3))  
taxtree(table)
```

Index

*Topic **datasets**

.allchr, 2

.ranks, 2

abundances, 3

.allchr, 2

.ranks, 2

abundances, 3

all_clades, 3

depth, 4

error_lineage, 4

is_clade, 4

is_lineage, 5

is_rank, 6

last_clade, 6

last_rank, 7

taxtable, 8

taxtree, 8