

# Package: yatah (via r-universe)

September 10, 2024

**Title** Yet Another TAXonomy Handler

**Version** 1.0.0

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

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

**Depends** R (>= 2.10)

**Imports** ape, lifecycle, purrr, stats, stringr

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

**VignetteBuilder** knitr

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Repository** <https://abichat.r-universe.dev>

**RemoteUrl** <https://github.com/abichat/yatah>

**RemoteRef** HEAD

**RemoteSha** d5476361fb39818eeeb2129c2d180101e266372b

## Contents

abundances . . . . .	2
all_ranks . . . . .	3
get_all_clades . . . . .	3
get_clade . . . . .	4

get_last_clade . . . . .	4
get_last_rank . . . . .	5
is_clade . . . . .	6
is_lineage . . . . .	6
is_rank . . . . .	7
taxtable . . . . .	8
taxtree . . . . .	8
trim_common . . . . .	9
trim_rank . . . . .	10
trim_void . . . . .	10
<b>Index</b>	<b>12</b>

---

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 ([doi:10.15252/msb.20145645](https://doi.org/10.15252/msb.20145645)), Pasolli et al., 2017 ([doi:10.1038/nmeth.4468](https://doi.org/10.1038/nmeth.4468)).

## Examples

```
dim(abundances)
abundances[1:5, 1:7]
```

---

all_ranks	<i>Ranks handled by yatah</i>
-----------	-------------------------------

---

**Description**

Ranks handled by yatah

**Usage**

```
all_ranks
```

**Format**

An object of class character of length 8.

**Examples**

```
all_ranks
```

---

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

---

**Description**

Extract all clades present in the lineages

**Usage**

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

**Arguments**

lineage	string. Vector of lineages.
simplify	logical. Should the output be a vector or a dataframe?

**Details**

If a clade correspond to different ranks (e.g. Actinobacteria is both a phylum and a clade), it will be displayed only one time when `simplify` is set to `TRUE`. It is also the case for different clades with same name and same rank when `simplify` is set to `FALSE`.

**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"
lineage3 <- "k__Bacteria|p__Actinobacteria|c__Actinobacteria"
get_all_clades(c(lineage1, lineage2, lineage3))
get_all_clades(c(lineage1, lineage2, lineage3), simplify = FALSE)
```

---

get_clade	<i>Extract the clade of a desired rank in a lineage</i>
-----------	---

---

**Description**

Extract the clade of a desired rank in a lineage

**Usage**

```
get_clade(lineage, rank = yatah::all_ranks, same = TRUE)
```

**Arguments**

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

**Value**

A string.

**Examples**

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

---

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

---

**Description**

Extract the last clade of a lineage

**Usage**

```
get_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"  
get_last_clade(c(lineage1, lineage2))
```

---

*get\_last\_rank*                      *Extract the last rank of a lineage*

---

**Description**

Extract the last rank of a lineage

**Usage**

```
get_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"  
get_last_rank(c(lineage1, lineage2))
```

---

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(".", yatah::all_ranks))
```

**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.
--------	---------------------------------

**Details**

Alphanumeric character, hyphen, dots, square brackets and non-consecutive underscores are allowed in clades names.

**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 = yatah::all_ranks)
```

```
is_at_least_rank(lineage, rank = yatah::all_ranks)
```

**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")
is_at_least_rank(c(lineage1, lineage2), "phylum")
is_at_least_rank(c(lineage1, lineage2), "order")
```

---

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)
```

---

trim\_common

*Trim lineages until the shallowest common rank.*

---

**Description**

Trim lineages until the shallowest common rank.

**Usage**

```
trim_common(lineage, remove_void = TRUE, only_tail = TRUE)
```

**Arguments**

lineage            string. Vector of lineages.  
remove\_void        Should void ranks be removed? Default to TRUE.  
only\_tail          Logical to be passed to trim\_void(). Used only if remove\_void is set to TRUE.

**Value**

The trimmed lineages, with same depth.

**Examples**

```
lineage1 <- "k__Bacteria|p__Verrucomicrobia|c__Verrucomicrobiae"  
lineage2 <- "k__Bacteria|p__Firmicutes"  
lineage3 <- "k__Bacteria|p__|c__Clostridia"  
trim_common(c(lineage1, lineage2, lineage3), remove_void = FALSE)  
trim_common(c(lineage1, lineage2, lineage3), only_tail = FALSE)
```

---

trim_rank	<i>Trim lineages until a specified rank</i>
-----------	---

---

**Description**

Trim lineages until a specified rank

**Usage**

```
trim_rank(lineage, rank = yatah::all_ranks, same = TRUE)
```

**Arguments**

lineage	string. Vector of lineages.
rank	string. One of c("kingdom", "phylum", "class", "order", "family", "genus", "species", "strain") with partial matching.
same	logical. Does the lineage have the same depth? Default to TRUE.

**Details**

Returns NA if a lineage is not as deep as the specified rank.

**Value**

The trimmed lineages. Depth could be different among them.

**Examples**

```
lineage1 <- "k__Bacteria|p__Verrucomicrobia|c__Verrucomicrobiae"
lineage2 <- "k__Bacteria|p__Firmicutes|c__Clostridia"
trim_rank(c(lineage1, lineage2), rank = "phylum")
trim_rank(c(lineage1, lineage2), rank = "genus")
```

---

trim_void	<i>Trim void ranks in lineages</i>
-----------	------------------------------------

---

**Description**

Trim void ranks in lineages

**Usage**

```
trim_void(lineage, same = TRUE, only_tail = FALSE)
```

**Arguments**

lineage	string. Vector of lineages.
same	logical. Does the lineage have the same depth? Default to TRUE.
only_tail	Logical. If FALSE (default), void ranks amid lineages and subranks are removed. If TRUE, only final void ranks are removed.

**Details**

If there is a void rank amid a lineage, deeper ranks will be removed. See the example with lineage3.

**Value**

The trimmed lineages. Depth could be different among them.

**Examples**

```
lineage1 <- "k__Bacteria|p__Verrucomicrobia|c__Verrucomicrobiae|o__|f__"
lineage2 <- "k__Bacteria|p__Firmicutes|c__"
lineage3 <- "k__Bacteria|p__|c__Verrucomicrobiae|o__|f__"
trim_void(c(lineage1, lineage2, lineage3), same = FALSE)
trim_void(c(lineage1, lineage2, lineage3), same = FALSE, only_tail = TRUE)
```

# Index

## \* datasets

- abundances, [2](#)
- all\_ranks, [3](#)

- abundances, [2](#)
- all\_ranks, [3](#)

- get\_all\_clades, [3](#)
- get\_clade, [4](#)
- get\_last\_clade, [4](#)
- get\_last\_rank, [5](#)

- is\_at\_least\_rank (is\_rank), [7](#)
- is\_clade, [6](#)
- is\_lineage, [6](#)
- is\_rank, [7](#)

- taxtable, [8](#)
- taxtree, [8](#)
- trim\_common, [9](#)
- trim\_rank, [10](#)
- trim\_void, [10](#)