

# Package: taxodist (via r-universe)

June 2, 2026

**Title** Taxonomic Distance and Phylogenetic Lineage Computation

**Version** 0.5.0

**LazyData** true

**Maintainer** Rodrigo Fonseca Villa <rodrigo03.villa@gmail.com>

**Description** Computes phylogenetic distances between any two taxa using hierarchical lineage data retrieved from The Taxonomicon <<http://taxonomicon.taxonomy.nl>>, a comprehensive curated classification of all life based on Systema Naturae 2000 (Brands, 1989 <<http://taxonomicon.taxonomy.nl>>). Given any two taxon names, retrieves their full lineages, identifies the most recent common ancestor (MRCA), and computes a dissimilarity index based on lineage depth. Outputs native dist objects, enabling direct integration with the R statistical ecosystem for hierarchical clustering, principal coordinate analysis (PCoA), and multivariate ecological analyses. Supports individual distance queries, pairwise distance matrices, clade filtering, and lineage utilities.

**Language** en-US

**License** GPL (>= 3)

**URL** <https://github.com/rodrigorsqrt3/taxodist>

**BugReports** <https://github.com/rodrigorsqrt3/taxodist/issues>

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**SystemRequirements** internet access

**Depends** R (>= 4.1.0)

**Imports** httr (>= 1.4.0), rvest (>= 1.0.0), stringr (>= 1.4.0), purrr (>= 0.3.0), cli (>= 3.0.0), utils, stats, graphics

**Suggests** testthat (>= 3.0.0), mockery, knitr, rmarkdown, xml2, ape, vegan, webmockr

**VignetteBuilder** knitr  
**Config/testthat/edition** 3  
**Config/pak/sysreqs** libicu-dev libxml2-dev libssl-dev  
**Repository** <https://rodrigorsqrt3.r-universe.dev>  
**Date/Publication** 2026-06-02 19:25:34 UTC  
**RemoteUrl** <https://github.com/rodrigorsqrt3/taxodist>  
**RemoteRef** HEAD  
**RemoteSha** d8b2f4ae329d4b6460121e3b486bf617c3adc41c

## Contents

taxodist-package . . . . .	3
cache_info . . . . .	4
check_coverage . . . . .	5
clear_cache . . . . .	5
closest_relative . . . . .	6
compare_lineages . . . . .	6
distance_matrix . . . . .	7
filter_clade . . . . .	8
focal_distances . . . . .	8
get_lineage . . . . .	9
get_lineage_by_id . . . . .	10
get_taxonomicon_id . . . . .	11
is_member . . . . .	12
lineage_depth . . . . .	12
load_cache . . . . .	13
mrca . . . . .	14
plot.taxodist_cluster . . . . .	14
plot.taxodist_ord . . . . .	15
print.taxodist_focal . . . . .	16
print.taxodist_path . . . . .	16
print.taxodist_result . . . . .	17
save_cache . . . . .	17
shared_clades . . . . .	18
summary.taxodist_ord . . . . .	19
taxo_cluster . . . . .	19
taxo_distance . . . . .	20
taxo_heatmap . . . . .	21
taxo_ordinate . . . . .	22
taxo_path . . . . .	23
taxo_search . . . . .	24
taxobase . . . . .	25

## Index

26

## Description

taxodist computes phylogenetic distances between any two taxa using hierarchical lineage data retrieved from The Taxonomicon (taxonomy.nl), a comprehensive curated classification of all life based on Systema Naturae 2000.

### Core functions:

- `get_lineage()` — retrieve the full lineage of any taxon
- `taxo_distance()` — compute the tree metric distance between two taxa
- `mrca()` — find the most recent common ancestor
- `distance_matrix()` — compute all pairwise distances for a set of taxa
- `closest_relative()` — find the closest relative among candidates
- `compare_lineages()` — print a side-by-side lineage comparison
- `shared_clades()` — list clades shared between two taxa
- `is_member()` — test clade membership
- `filter_clade()` — filter taxa by clade membership
- `check_coverage()` — check Taxonomicon coverage for a list of taxa
- `lineage_depth()` — get the lineage depth of a taxon
- `clear_cache()` — clear the session lineage cache

### Mathematical background:

The distance metric is based on the depth of the most recent common ancestor (MRCA):

$$d(A, B) = \frac{1}{\text{depth}(\text{MRCA}(A, B))}$$

The deeper the shared ancestor, the smaller the distance. This metric ensures that taxa sharing the same MRCA are always equidistant from any third taxon, regardless of lineage depth below the split — a key biological correctness property absent from Jaccard-based approaches.

### Data source:

All lineage data is sourced from The Taxonomicon (taxonomy.nl), based on Systema Naturae 2000 by S.J. Brands (1989 onwards). Please cite this resource when using taxodist in published work.

### Author(s)

**Maintainer:** Rodrigo Fonseca Villa <rodrigo03.villa@gmail.com> ([ORCID](#))

### References

Brands, S.J. (1989 onwards). Systema Naturae 2000. Amsterdam, The Netherlands. Retrieved from The Taxonomicon, <http://taxonomicon.taxonomy.nl>.

**See Also**

Useful links:

- <https://github.com/rodrigosqrt3/taxodist>
- Report bugs at <https://github.com/rodrigosqrt3/taxodist/issues>

---

cache\_info

*Inspect the current taxodist lineage cache*

---

**Description**

Reports the number of cached entries, their total memory footprint, and the names of all taxa whose lineages are stored in the current session. Useful for understanding what has already been retrieved before running further computations.

**Usage**

```
cache_info()
```

**Value**

Invisibly returns a list with elements:

`n_lineages` Integer. Number of cached lineages.

`n_ids` Integer. Number of cached taxon IDs.

`taxa` Character vector of taxa with cached lineages.

`size_bytes` Numeric. Total memory used by the cache.

**See Also**

[save\\_cache\(\)](#), [load\\_cache\(\)](#), [clear\\_cache\(\)](#)

**Examples**

```
get_lineage("Tyrannosaurus")
get_lineage("Velociraptor")
cache_info()
```

---

check_coverage	<i>Check whether a taxon is covered by The Taxonomicon</i>
----------------	------------------------------------------------------------

---

**Description**

Queries The Taxonomicon for a taxon name and returns a logical indicating whether the taxon was found. Useful for pre-screening a list of names before running distance computations.

**Usage**

```
check_coverage(taxa, verbose = FALSE)
```

**Arguments**

taxa	A character vector of one or more taxon names.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

**Value**

A named logical vector. TRUE indicates the taxon was found, FALSE indicates it was not.

**Examples**

```
check_coverage(c("Tyrannosaurus", "Velociraptor", "Fakeosaurus"))
```

---

clear_cache	<i>Clear the taxodist lineage cache</i>
-------------	-----------------------------------------

---

**Description**

Clears all cached lineages stored in the current R session. Useful when you suspect cached data is stale or want to force fresh retrieval.

**Usage**

```
clear_cache()
```

**Value**

Invisibly returns NULL.

**Examples**

```
clear_cache()
```

---

closest_relative	<i>Find the closest relative of a taxon among a set of candidates</i>
------------------	-----------------------------------------------------------------------

---

**Description**

Given a query taxon and a vector of candidate taxa, returns the candidate with the smallest phylogenetic distance to the query.

**Usage**

```
closest_relative(taxon, candidates, verbose = FALSE)
```

**Arguments**

taxon	A character string giving the query taxon name.
candidates	A character vector of candidate taxon names to compare against.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

**Value**

A data frame with columns taxon (candidate name) and distance (tree metric distance), sorted by distance ascending. Returns NULL if the query taxon cannot be found.

**Examples**

```
closest_relative("Tyrannosaurus",
  c("Velociraptor", "Triceratops", "Brachiosaurus", "Allosaurus"))
```

---

compare_lineages	<i>Compare lineages of two taxa side by side</i>
------------------	--------------------------------------------------

---

**Description**

Prints the lineages of two taxa aligned at their most recent common ancestor, making the point of divergence easy to identify.

**Usage**

```
compare_lineages(taxon_a, taxon_b, verbose = FALSE)
```

**Arguments**

taxon_a	A character string giving the first taxon name.
taxon_b	A character string giving the second taxon name.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

**Value**

Invisibly returns a list with elements lineage\_a, lineage\_b, and mrca\_depth.

**Examples**

```
compare_lineages("Tyrannosaurus", "Velociraptor")
compare_lineages("Tyrannosaurus", "Triceratops")
```

---

distance_matrix	<i>Compute pairwise taxonomic distances for a set of taxa</i>
-----------------	---------------------------------------------------------------

---

**Description**

Given a vector of taxon names, computes all pairwise phylogenetic distances and returns a symmetric distance matrix. Lineages are cached after first retrieval to minimise redundant network requests.

**Usage**

```
distance_matrix(taxa, verbose = FALSE, progress = TRUE)
```

**Arguments**

taxa	A character vector of taxon names.
verbose	Logical. If TRUE, prints progress for each pair. Default FALSE.
progress	Logical. If TRUE, shows a progress bar. Default TRUE.

**Value**

A symmetric numeric matrix of class "dist" containing pairwise distances. Row and column names are set to the input taxon names. Taxa that could not be found are included with NA distances.

**See Also**

[taxo\\_distance\(\)](#), [closest\\_relative\(\)](#)

**Examples**

```
theropods <- c("Tyrannosaurus", "Velociraptor", "Spinosaurus",
              "Allosaurus", "Carnotaurus")
mat <- distance_matrix(theropods)
print(mat)
```

---

filter_clade	<i>Filter a vector of taxa to those belonging to a given clade</i>
--------------	--------------------------------------------------------------------

---

**Description**

Given a vector of taxon names and a clade name, returns only those taxa whose lineage includes the specified clade.

**Usage**

```
filter_clade(taxa, clade, verbose = FALSE)
```

**Arguments**

taxa	A character vector of taxon names.
clade	A character string giving the clade to filter by.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

**Value**

A character vector of taxa that are members of the specified clade.

**Examples**

```
taxa <- c("Tyrannosaurus", "Triceratops", "Velociraptor",
          "Brachiosaurus", "Homo")
filter_clade(taxa, "Theropoda")
filter_clade(taxa, "Dinosauria")
```

---

focal_distances	<i>Compute distances from a focal taxon to a community of taxa</i>
-----------------	--------------------------------------------------------------------

---

**Description**

Given a focal taxon and a vector of community taxa, retrieves all lineages and computes the taxonomic distance from the focal taxon to each member of the community. Returns a sorted data frame that also reports the most recent common ancestor (MRCA) for each pair, making it easy to interpret *why* a taxon is close or distant.

**Usage**

```
focal_distances(focal, community, verbose = FALSE, progress = TRUE)
```

**Arguments**

focal	A character string giving the focal taxon name.
community	A character vector of community taxon names to compare against. The focal taxon may be included — it will receive a distance of 0.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.
progress	Logical. If TRUE, shows a progress bar. Default TRUE.

**Value**

A data frame of class "taxodist\_focal" with columns:

taxon	Character. Community taxon name.
distance	Numeric. Taxonomic distance to the focal taxon.
mrca	Character. Name of the most recent common ancestor.
mrca_depth	Integer. Depth of the MRCA node.

Rows are sorted by distance ascending (closest relatives first). Returns NULL if the focal taxon cannot be found.

**See Also**

[closest\\_relative\(\)](#), [distance\\_matrix\(\)](#), [taxo\\_distance\(\)](#)

**Examples**

```
community <- c("Velociraptor", "Triceratops", "Brachiosaurus",
              "Allosaurus", "Homo", "Quercus")
focal_distances("Tyrannosaurus", community)
```

---

get\_lineage

*Retrieve the full taxonomic lineage of a taxon by name*

---

**Description**

A convenience wrapper that combines [get\\_taxonomic\\_id\(\)](#) and [get\\_lineage\\_by\\_id\(\)](#) into a single call. Given a taxon name, returns its complete lineage from root to tip.

**Usage**

```
get_lineage(taxon, clean = TRUE, verbose = FALSE)
```

**Arguments**

taxon	A character string giving the taxon name.
clean	Logical. If TRUE (default), removes philosophical root nodes and cleans formatting markers.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

**Value**

A character vector of clade names ordered from root to tip, or NULL if the taxon cannot be found.

**Examples**

```
get_lineage("Tyrannosaurus")
get_lineage("Homo sapiens")
get_lineage("Quercus robur")
```

---

get\_lineage\_by\_id      *Retrieve the full taxonomic lineage of a taxon*

---

**Description**

Given a Taxonomicon numeric ID, retrieves and parses the complete hierarchical lineage from root (Natura) to the taxon itself. The lineage is returned as a character vector ordered from root to tip.

**Usage**

```
get_lineage_by_id(taxon_id, clean = TRUE, verbose = FALSE)
```

**Arguments**

taxon_id	A numeric or character string giving the Taxonomicon ID. Obtain this with <a href="#">get_taxonomicon_id()</a> .
clean	Logical. If TRUE (default), removes philosophical root nodes above Biota (i.e., Natura, actualia, Mundus, naturalia) and strips dagger and superscript markers from names.
verbose	Logical. If TRUE, prints status messages. Default FALSE.

**Details**

Lineage data is sourced from The Taxonomicon, which is based on Systema Naturae 2000 (Brands, S.J., 1989 onwards). The depth of lineages in The Taxonomicon substantially exceeds that of other programmatic sources such as the Open Tree of Life, particularly for well-studied clades such as Dinosauria, where intermediate clades at the level of superfamilies, tribes, and named subclades are fully resolved.

**Value**

A character vector of clade names from root to tip, or NULL if retrieval fails.

**See Also**

[get\\_lineage\(\)](#), [taxo\\_distance\(\)](#)

### Examples

```
id <- get_taxonomicon_id("Tyrannosaurus")
lin <- get_lineage_by_id(id)
print(lin)
```

---

get\_taxonomicon\_id      *Find the Taxonomicon ID for a taxon name*

---

### Description

Queries The Taxonomicon (taxonomy.nl) to retrieve the internal numeric identifier for a given taxon name. The search filters out non-biological entities such as astronomical objects that may share the same name.

### Usage

```
get_taxonomicon_id(taxon, verbose = FALSE)
```

### Arguments

taxon	A character string giving the taxon name to search for. Typically a genus name (e.g., "Tyrannosaurus") but species and higher ranks are also supported.
verbose	Logical. If TRUE, prints status messages during retrieval. Default is FALSE.

### Details

The function queries the static search endpoint at `taxonomicon.taxonomy.nl/TaxonList.aspx` and parses the resulting HTML to extract the taxon ID from the hierarchy link. When multiple matches exist (e.g., a genus name shared with an astronomical object), biological entries are prioritised by filtering for entries annotated as dinosaur, reptile, archosaur, animal, plant, fungus, or bacterium.

### Value

A character string containing the Taxonomicon numeric ID, or NULL if the taxon is not found.

### See Also

[get\\_lineage\(\)](#), [taxo\\_distance\(\)](#)

### Examples

```
get_taxonomicon_id("Tyrannosaurus") # returns "50841"
get_taxonomicon_id("Homo")
get_taxonomicon_id("Quercus")
```

---

is_member	<i>Test whether one taxon is nested within another</i>
-----------	--------------------------------------------------------

---

**Description**

Returns TRUE if taxon is a member of clade — i.e., if the clade name appears in the taxon's lineage.

**Usage**

```
is_member(taxon, clade, verbose = FALSE)
```

**Arguments**

taxon	A character string giving the taxon name to test.
clade	A character string giving the clade name to test membership in.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

**Value**

A logical value, or NULL if the taxon cannot be found.

**Examples**

```
is_member("Tyrannosaurus", "Theropoda") # TRUE
is_member("Triceratops", "Theropoda") # FALSE
is_member("Homo", "Amniota") # TRUE
```

---

lineage_depth	<i>Get the lineage depth of a taxon</i>
---------------	-----------------------------------------

---

**Description**

Returns the number of nodes in the lineage of a taxon, from root to tip. This reflects how deeply nested the taxon is within the taxonomic hierarchy.

**Usage**

```
lineage_depth(taxon, verbose = FALSE)
```

**Arguments**

taxon	A character string giving the taxon name.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

**Value**

An integer giving the lineage depth, or NULL if the taxon cannot be found.

**Examples**

```
lineage_depth("Tyrannosaurus") # deep - many intermediate clades
lineage_depth("Biota")         # shallow - near root
```

---

load_cache	<i>Load a previously saved taxodist cache from disk</i>
------------	---------------------------------------------------------

---

**Description**

Restores lineage data saved with [save\\_cache\(\)](#) into the current session cache, avoiding network requests for taxa already retrieved in a previous session.

**Usage**

```
load_cache(file)
```

**Arguments**

file                    Path to an .rds file created by [save\\_cache\(\)](#).

**Value**

Invisibly returns NULL.

**See Also**

[save\\_cache\(\)](#), [clear\\_cache\(\)](#)

**Examples**

```
tmp <- tempfile(fileext = ".rds")
save_cache(tmp)
load_cache(tmp)
unlink(tmp)
```

---

mrca	<i>Compute the most recent common ancestor of two taxa</i>
------	------------------------------------------------------------

---

**Description**

Retrieves the lineages of two taxa and returns the name of their most recent common ancestor (MRCA) — the deepest node shared by both lineages.

**Usage**

```
mrca(taxon_a, taxon_b, verbose = FALSE)
```

**Arguments**

taxon_a	A character string giving the first taxon name.
taxon_b	A character string giving the second taxon name.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

**Value**

A character string giving the name of the MRCA, or NULL if either taxon cannot be found or no common ancestor exists.

**Examples**

```
mrca("Tyrannosaurus", "Velociraptor") # "Tyrannoraptora"
mrca("Tyrannosaurus", "Triceratops")  # "Dinosauria"
mrca("Tyrannosaurus", "Homo")         # "Amniota"
```

---

plot.taxodist_cluster	<i>Plot method for taxodist_cluster objects</i>
-----------------------	-------------------------------------------------

---

**Description**

Plots the hierarchical clustering dendrogram produced by `taxo_cluster()`.

**Usage**

```
## S3 method for class 'taxodist_cluster'
plot(
  x,
  main = "Taxonomic Clustering",
  xlab = "",
  sub = paste("Method:", if (!is.null(x$clust)) x$clust$method else "None"),
  ...
)
```

**Arguments**

x	A taxodist_cluster object from <code>taxo_cluster()</code> .
main	Plot title. Default: "Taxonomic Clustering".
xlab	X-axis label. Default: "".
sub	Subtitle. Default: clustering method used.
...	Additional arguments passed to <code>stats::plot.hclust()</code> .

**Value**

Invisibly returns x. Called for side effects (plotting).

**Examples**

```
taxa <- c("Tyrannosaurus", "Velociraptor", "Triceratops", "Brachiosaurus")
cl <- taxo_cluster(taxa)
plot(cl)
```

---

plot.taxodist\_ord      *Plot method for taxodist\_ord objects*

---

**Description**

Plots the PCoA ordination produced by `taxo_ordinate()`.

**Usage**

```
## S3 method for class 'taxodist_ord'
plot(
  x,
  main = "Taxonomic Ordination (PCoA)",
  xlab = "PC1",
  ylab = "PC2",
  labels = if (!is.null(x$points)) rownames(x$points) else NULL,
  ...
)
```

**Arguments**

x	A taxodist_ord object from <code>taxo_ordinate()</code> .
main	Plot title. Default: "Taxonomic Ordination (PCoA)".
xlab	X-axis label. Default: "PC1".
ylab	Y-axis label. Default: "PC2".
labels	Character vector of labels. Default: rownames of points.
...	Additional arguments passed to <code>graphics::text()</code> .

**Value**

Invisibly returns x. Called for side effects (plotting).

**Examples**

```
taxa <- c("Tyrannosaurus", "Velociraptor", "Triceratops", "Brachiosaurus")
ord <- taxo_ordinate(taxa)
plot(ord)
```

---

`print.taxodist_focal` *Print method for taxodist\_focal objects*

---

**Description**

Print method for taxodist\_focal objects

**Usage**

```
## S3 method for class 'taxodist_focal'
print(x, ...)
```

**Arguments**

x                    A taxodist\_focal object from `focal_distances()`.  
...                  Additional arguments (ignored).

**Value**

Invisibly returns x. Called for side effects (printing).

---

`print.taxodist_path` *Print method for taxodist\_path objects*

---

**Description**

Print method for taxodist\_path objects

**Usage**

```
## S3 method for class 'taxodist_path'
print(x, ...)
```

**Arguments**

x                    A taxodist\_path object from [taxo\\_path\(\)](#).  
...                   Additional arguments (ignored).

**Value**

Invisibly returns x. Called for side effects (printing).

---

`print.taxodist_result`    *Print method for taxodist distance results*

---

**Description**

Print method for taxodist distance results

**Usage**

```
## S3 method for class 'taxodist_result'  
print(x, ...)
```

**Arguments**

x                    A list returned by [taxo\\_distance\(\)](#).  
...                   Additional arguments (ignored).

**Value**

Invisibly returns x. Called for side effects (printing).

---

`save_cache`                    *Save the taxodist lineage cache to disk*

---

**Description**

Serialises the current session cache to an `.rds` file so it can be restored in a future session with [load\\_cache\(\)](#). Useful for reproducibility and for avoiding repeated network requests.

**Usage**

```
save_cache(file)
```

**Arguments**

file                    Path to the `.rds` file to write.

**Value**

Invisibly returns NULL.

**See Also**

[load\\_cache\(\)](#), [clear\\_cache\(\)](#)

**Examples**

```
tmp <- tempfile(fileext = ".rds")
save_cache(tmp)
unlink(tmp)
```

---

shared\_clades

*List all clades shared between two taxa*

---

**Description**

Returns the vector of clade names forming the shared trunk of two taxa's lineages, from root down to (and including) their MRCA.

**Usage**

```
shared_clades(taxon_a, taxon_b, verbose = FALSE)
```

**Arguments**

taxon_a	A character string giving the first taxon name.
taxon_b	A character string giving the second taxon name.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

**Value**

A character vector of shared clade names ordered from root to MRCA, or NULL if either taxon cannot be found.

**Examples**

```
shared_clades("Tyrannosaurus", "Velociraptor")
shared_clades("Tyrannosaurus", "Homo")
```

---

summary.taxodist\_ord *Summary method for taxodist\_ord objects*

---

### Description

Prints a summary of the PCoA ordination, including the goodness-of-fit and the proportion of variance explained by each principal coordinate.

### Usage

```
## S3 method for class 'taxodist_ord'
summary(object, ...)
```

### Arguments

object            A taxodist\_ord object from [taxo\\_ordinate](#).  
 ...              Additional arguments (ignored).

### Value

Invisibly returns a data frame containing the eigenvalues and variance explained by each dimension.

---

taxo\_cluster            *Cluster taxa by taxonomic distance*

---

### Description

Computes pairwise taxonomic distances and performs hierarchical clustering.

### Usage

```
taxo_cluster(taxa, method = "average", ...)
```

### Arguments

taxa                A character vector of taxon names, or a dist object from [distance\\_matrix\(\)](#).  
 method             Clustering method passed to [stats::hclust\(\)](#). Default "average" (UPGMA), which works well with taxonomic distances.  
 ...                Additional arguments passed to [distance\\_matrix\(\)](#) (e.g. verbose, progress).

### Value

An object of class "taxodist\_cluster" — a list with:

hclust The [stats::hclust\(\)](#) result.  
 dist The underlying distance matrix.

**See Also**

[taxo\\_ordinate\(\)](#), [distance\\_matrix\(\)](#)

**Examples**

```
taxa <- c("Tyrannosaurus", "Velociraptor", "Triceratops", "Brachiosaurus")
cl <- taxo_cluster(taxa)
if (!is.null(cl$hclust)) {
  plot(cl)
}
```

---

taxo\_distance

*Compute the phylogenetic distance between two taxa*

---

**Description**

Given two taxon names, retrieves their lineages from The Taxonomicon and computes a taxonomic distance based on the depth of their most recent common ancestor (MRCA):

**Usage**

```
taxo_distance(taxon_a, taxon_b, verbose = FALSE)
```

**Arguments**

taxon_a	A character string giving the first taxon name.
taxon_b	A character string giving the second taxon name.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

**Details**

$$d(A, B) = \frac{1}{\text{depth}(\text{MRCA}(A, B))}$$

The deeper the shared ancestor, the smaller (closer to zero) the distance. This metric ensures that taxa diverging at the same node are always equidistant from any third taxon, regardless of lineage depth differences below the split.

**Value**

A named list of class "taxodist\_result" with the following elements:

distance	Numeric. The distance between the two taxa. Returns 0 if one taxon is an ancestor of the other.
mrca	Character. The name of the most recent common ancestor.

mrca\_depth Integer. The depth of the MRCA node.  
 depth\_a Integer. The lineage depth of taxon A.  
 depth\_b Integer. The lineage depth of taxon B.  
 taxon\_a Character. Name of the first taxon.  
 taxon\_b Character. Name of the second taxon.

Returns NULL if either taxon cannot be found.

## References

Brands, S.J. (1989 onwards). Systema Naturae 2000. Amsterdam, The Netherlands. Retrieved from The Taxonomicon, <http://taxonomicon.taxonomy.nl>.

## See Also

[mrca\(\)](#), [distance\\_matrix\(\)](#), [get\\_lineage\(\)](#)

## Examples

```

# Distance between two theropods
taxo_distance("Tyrannosaurus", "Velociraptor")

# Distance between very distantly related taxa
taxo_distance("Tyrannosaurus", "Quercus")

# Distance between two oviraptorid genera
taxo_distance("Nomingia", "Huanansaurus")

```

---

taxo\_heatmap

*Plot a taxonomic heatmap*

---

## Description

Computes pairwise taxonomic distances and plots a heatmap with hierarchical clustering dendrograms on the margins. Darker/hotter colors typically represent smaller distances (closer relatives).

## Usage

```
taxo_heatmap(taxa, ...)
```

## Arguments

taxa A character vector of taxon names, or a dist object from [distance\\_matrix](#).  
 ... Additional arguments passed to [heatmap](#).

**Value**

Invisibly returns the underlying `dist` object. Called primarily for its side effect (plotting).

**Examples**

```
taxa <- c("Tyrannosaurus", "Velociraptor", "Homo", "Panthera", "Quercus")
taxo_heatmap(taxa)
```

---

taxo_ordinate	<i>Ordinate taxa in taxonomic distance space</i>
---------------	--------------------------------------------------

---

**Description**

Computes pairwise taxonomic distances and applies classical multidimensional scaling (PCoA) to project taxa into a low-dimensional space.

**Usage**

```
taxo_ordinate(taxa, k = 2, ...)
```

**Arguments**

taxa	A character vector of taxon names, or a <code>dist</code> object from <a href="#">distance_matrix()</a> .
k	Number of dimensions. Default 2.
...	Additional arguments passed to <a href="#">distance_matrix()</a> .

**Value**

An object of class "taxodist\_ord" — a list with:

- `points` A matrix of coordinates (taxa x k dimensions).
- `dist` The underlying distance matrix.
- `GOF` Goodness-of-fit from [stats::cmdscale\(\)](#).
- `eig` The eigenvalues computed during PCoA.

**See Also**

[taxo\\_cluster\(\)](#), [distance\\_matrix\(\)](#)

**Examples**

```
taxa <- c("Tyrannosaurus", "Velociraptor", "Triceratops", "Brachiosaurus")
ord <- taxo_ordinate(taxa)
if (!is.null(ord$points)) {
  plot(ord$points, type = "n")
  text(ord$points, labels = rownames(ord$points))
}
```

---

taxo_path	<i>Get the taxonomic path between two taxa</i>
-----------	------------------------------------------------

---

### Description

Returns the full node-by-node path from one taxon up to their most recent common ancestor (MRCA) and back down to the other taxon. The result is a data frame with one row per node, making it easy to inspect, filter, or pipe into other functions.

### Usage

```
taxo_path(taxon_a, taxon_b, verbose = FALSE)
```

### Arguments

taxon_a	A character string giving the first taxon name.
taxon_b	A character string giving the second taxon name.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

### Value

A data frame of class "taxodist\_path" with columns:

node Character. The clade or taxon name at this step.

depth Integer. The depth of this node in the full lineage of its side (or the shared lineage for the MRCA).

direction Character. One of "a" (ascending from taxon A to MRCA), "mrca" (the shared ancestor), or "b" (descending from MRCA to taxon B).

Returns NULL if either taxon cannot be found.

### See Also

[mrca\(\)](#), [shared\\_clades\(\)](#), [compare\\_lineages\(\)](#)

### Examples

```
taxo_path("Tyrannosaurus", "Velociraptor")  
taxo_path("Tyrannosaurus", "Homo")
```

---

taxo_search	<i>Search The Taxonomicon for a taxon name</i>
-------------	------------------------------------------------

---

### Description

Queries The Taxonomicon database and returns a data frame of all available biological entries matching the search string. This is particularly useful for exploring homonyms, historical ranks, or taxonomic synonyms before computing distances.

### Usage

```
taxo_search(taxon, verbose = FALSE)
```

### Arguments

taxon	A character string giving the taxon name to search for.
verbose	Logical. If TRUE, prints status messages. Default FALSE.

### Value

A data frame of class "data.frame" with columns:

id Character. The numeric Taxonomicon ID.

name Character. The full taxon description, including rank and author.

Returns NULL if no matches are found.

### See Also

[get\\_lineage\(\)](#), [taxo\\_distance\(\)](#)

### Examples

```
taxo_search("Bacteria")
taxo_search("Nereis")
taxo_search("Tyrannosaurus")
```

---

taxobase

*Reference Taxonomic Dataset (taxobase)*

---

### Description

A pre-computed dataset containing 50 taxonomic clades spanning across the tree of life, fetched from The Taxonomicon. This provides a offline fallback containing lineages, distance matrices, and search queries for demonstrating the package.

### Usage

```
data(taxobase)
```

### Format

A structured list containing the following pre-computed components:

**taxa** Character vector of 1,000 candidate taxonomic genera.

**found\_taxa** Character vector of verified clades found in The Taxonomicon.

**coverage** Named logical vector indicating database presence of each taxon.

**matrix** A pre-computed taxonomic 'dist' object for verified clades.

**pairwise** A pre-computed distance list between Tyrannosaurus and Homo.

**lineage\_homo** Full taxonomy tree structure list for 'Homo'.

**lineage\_tyrannosaurus** Full taxonomy tree structure list for 'Tyrannosaurus'.

**closest** Dataframe demonstrating the closest relative query.

**filter** Character vector filtering verified taxa down to Theropoda.

**search** Reference search result dataset for the term 'Bacteria'.

### Source

<http://taxonomicon.taxonomy.nl>

### Examples

```
data(taxobase)
length(taxobase$found_taxa)
head(labels(taxobase$matrix))
```

# Index

## \* datasets

- taxobase, 25
- cache\_info, 4
- check\_coverage, 5
- check\_coverage(), 3
- clear\_cache, 5
- clear\_cache(), 3, 4, 13, 18
- closest\_relative, 6
- closest\_relative(), 3, 7, 9
- compare\_lineages, 6
- compare\_lineages(), 3, 23
- distance\_matrix, 7, 21
- distance\_matrix(), 3, 9, 19–22
- filter\_clade, 8
- filter\_clade(), 3
- focal\_distances, 8
- focal\_distances(), 16
- get\_lineage, 9
- get\_lineage(), 3, 10, 11, 21, 24
- get\_lineage\_by\_id, 10
- get\_lineage\_by\_id(), 9
- get\_taxonomicon\_id, 11
- get\_taxonomicon\_id(), 9, 10
- graphics::text(), 15
- heatmap, 21
- is\_member, 12
- is\_member(), 3
- lineage\_depth, 12
- lineage\_depth(), 3
- load\_cache, 13
- load\_cache(), 4, 17, 18
- mrca, 14
- mrca(), 3, 21, 23
- plot.taxodist\_cluster, 14
- plot.taxodist\_ord, 15
- print.taxodist\_focal, 16
- print.taxodist\_path, 16
- print.taxodist\_result, 17
- save\_cache, 17
- save\_cache(), 4, 13
- shared\_clades, 18
- shared\_clades(), 3, 23
- stats::cmdscale(), 22
- stats::hclust(), 19
- stats::plot.hclust(), 15
- summary.taxodist\_ord, 19
- taxo\_cluster, 19
- taxo\_cluster(), 14, 15, 22
- taxo\_distance, 20
- taxo\_distance(), 3, 7, 9–11, 17, 24
- taxo\_heatmap, 21
- taxo\_ordinate, 19, 22
- taxo\_ordinate(), 15, 20
- taxo\_path, 23
- taxo\_path(), 17
- taxo\_search, 24
- taxobase, 25
- taxodist (taxodist-package), 3
- taxodist-package, 3