Package 'taxodist'

November 6, 2026

Title Taxonomic Hierarchy Distances and Lineage Analysis

Version 1.0.0

LazyData true

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

Description Computes distances between taxonomic hierarchy nodes using lineage data retrieved from The Taxonomicon <http://taxonomicon.taxonomy.nl>. For distinct nodes, distance is defined as the reciprocal of the depth of their most recent common ancestor; identical nodes have distance zero. This definition yields an ultrametric within each connected hierarchy. Path length, Wu-Palmer, and rank-based distances are also available, together with the taxonomic diversity and distinctness indices of Clarke and Warwick (1998) <doi:10.1046/j.1365-2664.1998.3540523.x> and their randomization test. Functions are provided for auditable name resolution, online or user-supplied lineage analysis, node ranks, clade membership, pairwise, matrix, and cross-set distance calculation, classification trees with named clades, hierarchical clustering, principal coordinates analysis, portable and comparable JSON analysis bundles, and cache management. Distance matrices are returned as base R 'dist' objects. The distances represent classification depth rather than evolutionary time or phylogenetic branch length.

Language en-US

License GPL (>= 3)

URL https://github.com/rodrigosqrt3/taxodist

BugReports https://github.com/rodrigosqrt3/taxodist/issues

Encoding UTF-8

RoxygenNote 7.3.3

SystemRequirements internet access for online taxonomy retrieval

Depends R (>= 4.1.0)

Imports httr (>= 1.4.0), rvest (>= 1.0.0), stringr (>= 1.4.0), cli (>= 3.0.0), jsonlite (>= 1.8.0), utils, stats, graphics, tools, xml2 (>= 1.3.0)

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

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Rodrigo Fonseca Villa [aut, cre] (ORCID:
<https://orcid.org/0009-0005-2938-2270>)

Repository CRAN

Date/Publication 2026-11-06 12:50:02 UTC

Contents

taxodist-package . . . 3
cache_info . . . 5
check_coverage . . . 5
clear_cache . . . 6
closest_relative . . . 7
compare_bundles . . . 7
compare_lineages . . . 8
cross_distances . . . 9
distance_matrix . . . 10
filter_clade . . . 11
focal_distances . . . 12
get_lineage . . . 13
get_lineage_by_id . . . 14
get_taxonomicon_id . . . 15
is_member . . . 16
lineage_depth . . . 16
lineage_ranks . . . 17
load_cache . . . 18
mrca . . . 19
plot.taxodist_cluster . . . 19
plot.taxodist_ord . . . 20
print.taxodist_bundle . . . 21
print.taxodist_bundle_comparison . . . 21
print.taxodist_distinctness . . . 22
print.taxodist_focal . . . 22
print.taxodist_path . . . 23
print.taxodist_resolution . . . 23
print.taxodist_result . . . 24
read_taxodist_bundle . . . 24
save_cache . . . 25
shared_clades . . . 25
summary.taxodist_ord . . . 26
taxobase . . . 27
taxonomicon_available . . . 28
taxo_bundle . . . 28
taxo_cluster . . . 29

taxo_distance 30
taxo_distinctness 32
taxo_distinctness_test 34
taxo_from_lineages 36
taxo_heatmap 37
taxo_ordinate 38
taxo_path 39
taxo_resolve 40
taxo_search 41
taxo_tree 42
write_taxodist_bundle 43

Index **44**

`taxodist-package` *taxodist: Taxonomic Hierarchy Distances and Lineage Analysis*

Description

taxodist retrieves taxonomic lineages from The Taxonomicon (taxonomy.nl) or accepts user-supplied lineages, then computes distances from the depth of the most recent common ancestor. The resulting values describe separation within the source classification; they are not estimates of evolutionary time or phylogenetic branch length.

Core functions:

- `get_lineage()` — retrieve the full lineage of any taxon
- `taxo_resolve()` — resolve and audit a batch of names or IDs
- `taxo_from_lineages()` — create an offline resolution from local lineages
- `taxo_distance()` — compute the ultrametric distance between two taxa
- `mrca()` — find the most recent common ancestor
- `distance_matrix()` — compute all pairwise distances for a set of taxa
- `cross_distances()` — compute distances between two sets of taxa
- `taxo_tree()` — build a `phylo` tree whose nodes are named clades
- `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
- `taxo_bundle()` — preserve an analysis and its provenance
- `compare_bundles()` — report what changed between two analyses
- `save_cache()` / `load_cache()` — persist retrieved lineages
- `clear_cache()` — clear the session lineage cache

Mathematical background:

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

$$d(A, A) = 0$$

$$d(A, B) = 1/h(A, B)$$

Here, `h(A,B)` denotes the depth of the MRCA. The deeper the shared ancestor, the smaller the distance. Zero is reserved for identical hierarchy nodes; distinct ancestor-descendant pairs therefore have positive distance. The measure is an ultrametric on each connected taxonomic hierarchy. Its numerical values depend on the resolution of the classification returned by the data source.

Data source:

Online lineage data is sourced from The Taxonomicon (taxonomy.nl), based on Systema Naturae 2000 by S.J. Brands (1989 onwards). Please cite that resource when using its classifications, or record the relevant source when using `taxo_from_lineages()`.

Options:

- `taxodist.max_tries`: number of attempts made for each request to The Taxonomicon (default `3`). Network errors and server errors (5xx) are retried with exponential backoff; client errors (4xx) are not.
- `taxodist.min_interval`: minimum time in seconds between consecutive requests to The Taxonomicon (default `0.25`), so that large batches do not overload the server. Set to `0` to disable.

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 the taxa whose lineages are cached (the terminal node of each lineage).

`ids` Character vector of the Taxonomicon IDs of those lineages, in the same order as `taxa`.

`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 *Check whether a taxon is covered by The Taxonomicon*

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 that it is absent from The Taxonomicon, and NA that the answer is unknown because the server could not be reached.

See Also

taxo_resolve() for a complete audit of each name.

Examples

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

---

clear_cache *Clear the taxodist lineage cache*

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 *Find the closest relative of a taxon among a set of candidates*

Description

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

Usage

```
closest_relative(taxon, candidates, verbose = FALSE, metric = "reciprocal")
```

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.
metric	Distance metric; see taxo_distance().

Value

A data frame with columns taxon (candidate name) and distance, sorted by distance ascending. Returns NULL if the query taxon cannot be found. An empty candidate vector returns an empty data frame with the same columns.

Examples

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

compare_bundles *Compare two taxodist bundles*

Description

Reports how the resolution of each taxon and the pairwise distances changed between two bundles. A typical use is to re-run an analysis months later and check whether updates to The Taxonomicon changed any lineage, selected identifier, or distance.

Usage

```
compare_bundles(old, new, tolerance = sqrt(.Machine$double.eps))
```

Arguments

old, new	taxodist_bundle objects, or paths to JSON files written by write_taxodist_bundle().
tolerance	Numeric tolerance below which two distances are considered equal.

Value

An object of class taxodist_bundle_comparison, a list with:

taxa A data frame with one row per input label found in either bundle and columns input, change, status_old, status_new, id_old, id_new, depth_old, and depth_new. change is one of "unchanged", "added", "removed", "status_changed", "id_changed", or "lineage_changed".

distances A data frame of the pairs of inputs present in both bundles whose distance changed, with columns taxon_a, taxon_b, distance_old, and distance_new.

created_old, created_new Creation timestamps of the bundles.

See Also

taxo_bundle(), read_taxodist_bundle()

Examples

```
before <- taxo_bundle(taxo_from_lineages(list(
  Alpha = c("Biota", "Animalia", "Alpha"),
  Beta  = c("Biota", "Animalia", "Beta")
)), progress = FALSE)
after <- taxo_bundle(taxo_from_lineages(list(
  Alpha = c("Biota", "Animalia", "Chordata", "Alpha"),
  Beta  = c("Biota", "Animalia", "Beta")
)), progress = FALSE)
compare_bundles(before, after)
```

---

compare_lineages *Compare lineages of two taxa side by side*

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")
```

`cross_distances` *Compute taxonomic distances between two sets of taxa*

Description

Computes the rectangular matrix of distances from every taxon in `x` to every taxon in `y`, for example from the species of a community to a set of reference taxa. This avoids computing the full square matrix of the combined set.

Usage

```
cross_distances(x, y, verbose = FALSE, progress = TRUE, metric = "reciprocal")
```

Arguments

`x`, `y`	Character vectors of taxon names, or `taxodist_resolution` objects returned by `taxo_resolve()` or `taxo_from_lineages()`.
`verbose`	Logical. If `TRUE`, prints retrieval messages.
`progress`	Logical. If `TRUE`, shows progress messages.
`metric`	Distance metric; see `taxo_distance()`.

Value

A numeric matrix with one row per element of `x` and one column per element of `y`, labeled with the inputs. Taxa that could not be found have `NA` distances; taxa without a common ancestor have `Inf`.

See Also

`distance_matrix()`, `focal_distances()`

Examples

```
community <- taxo_from_lineages(list(
  Alpha = c("Biota", "Animalia", "Chordata", "Alpha"),
  Beta  = c("Biota", "Animalia", "Arthropoda", "Beta")
))
reference <- taxo_from_lineages(list(
  Gamma = c("Biota", "Animalia", "Chordata", "Gamma"),
  Delta = c("Biota", "Plantae", "Delta")
))
cross_distances(community, reference)
```

`distance_matrix` *Compute pairwise taxonomic distances for a set of taxa*

Description

Given a vector of taxon names, computes all pairwise taxonomic hierarchy 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, metric = "reciprocal")
```

Arguments

`taxa`	A character vector of taxon names, a `taxodist_resolution` object returned by `taxo_resolve()`, or a `taxodist_bundle`. Resolution objects reuse their stored lineages and retain the original inputs as matrix labels; bundles return their stored matrix.
`verbose`	Logical. If `TRUE`, prints progress for each pair. Default `FALSE`.
`progress`	Logical. If `TRUE`, shows a progress bar. Default `TRUE`.
`metric`	Distance metric; see `taxo_distance()`. Ignored for a `taxodist_bundle`, whose stored matrix is returned.

Details

Distances are computed in vectorized form: every lineage node is first replaced by an integer identifying its complete path from the root, so the MRCA depth of two taxa is simply the number of depths at which these identifiers agree. This gives exactly the same values as `taxo_distance()` while scaling to thousands of taxa.

Value

A symmetric numeric matrix of class `"dist"` containing pairwise distances. Row and column names are set to the input taxon names, and the `"method"` attribute records the metric. Taxa that could not be found are included with `NA` distances. An empty input returns an empty `"dist"` object.

See Also

taxo_distance(), cross_distances(), closest_relative(), taxo_resolve(), taxo_bundle(), taxo_tree()

Examples

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

---

filter_clade *Filter a vector of taxa to those belonging to a given clade*

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` *Compute distances from a focal taxon to a community of taxa*

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,
  metric = "reciprocal"
)
```

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`.
`metric`	Distance metric; see `taxo_distance()`.

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. An empty community returns an empty object with the same columns and class.

See Also

`closest_relative()`, `cross_distances()`, `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_taxonomicon_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 single character string giving the taxon name or numeric Taxonomicon ID. Leading, trailing, and repeated internal whitespace is ignored, and the queried name is matched case-insensitively when truncating the lineage.
`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 single numeric or character Taxonomicon ID. Obtain this with `get_taxonomicon_id()`.
`clean`	Logical. If `TRUE` (default), removes philosophical root nodes above `Biota` (i.e., Natura, actualia, Mundus, naturalia). Dagger and superscript markers are always stripped 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). Lineage resolution varies among taxonomic groups and may change when the source classification is updated. Because hierarchy distances depend on lineage depth, analyses should record the retrieval date and package version.

Cleaned and uncleaned lineages are cached separately, so the value of `clean` used in an earlier call never affects later calls.

Value

A character vector of clade names from root to tip, or `NULL` if `taxon_id` is not a numeric ID or 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. Leading, trailing, and repeated internal whitespace is ignored.
`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. Only candidates whose lineage descends from `Biota` are retained, which excludes astronomical and other non-biological homonyms. When several biological candidates remain, those whose lineage contains the queried name are preferred and a warning lists the alternatives.

Transient network failures are retried (three attempts by default; see the `taxodist.max_tries` option in taxodist-package).

Value

A character string containing the Taxonomicon numeric ID, or `NULL` if the taxon is not found or the server cannot be reached.

See Also

`get_lineage()`, `taxo_resolve()`, `check_coverage()`

Examples

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

is_member *Test whether one taxon is nested within another*

Description

Returns TRUE if taxon is a member of clade — i.e., if the complete clade name appears in the taxon's lineage. Matching is case-insensitive and ignores leading and trailing whitespace.

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 *Get the lineage depth of a taxon*

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
```

`lineage_ranks` *Retrieve the ranks of the nodes in a lineage*

Description

Returns each node of the lineage of a taxon together with its rank in The Taxonomicon classification (for example "Kingdom", "Clade", "Superfamily", or "Genus"). The nodes are exactly those returned by `get_lineage()` for the same arguments.

Usage

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

Arguments

taxon	A single character string giving the taxon name or numeric Taxonomicon ID. Leading, trailing, and repeated internal whitespace is ignored, and the queried name is matched case-insensitively when truncating the lineage.
clean	Logical. If TRUE (default), removes philosophical root nodes and cleans formatting markers.
verbose	Logical. If TRUE, prints progress messages. Default FALSE.

Details

Ranks are read from the label that precedes each name in the classification, without the [crown], [plesion], and [stem] qualifiers or extinction markers. Rows without a label are species, subspecies, or subgenera, whose rank is inferred from the form of the name. Many nodes are unranked clades ("Clade"). Ranks are used by the "ranked" distance method; see `taxo_distance()`.

Value

A data frame with columns node, rank, and depth, one row per lineage node from root to tip, or NULL if the taxon cannot be found. rank is NA for nodes without a stated rank (such as the philosophical root nodes above Biota).

See Also

`get_lineage()`, `taxo_distance()`

Examples

```
lineage_ranks("Passer")
```

`load_cache` *Load a previously saved taxodist cache from disk*

Description

Restores lineage data saved with `save_cache()` into the current session cache, avoiding network requests for taxa already retrieved in a previous session. The file structure is validated before the current cache is modified.

Usage

```
load_cache(file = NULL)
```

Arguments

`file`	Path to an `.rds` file created by `save_cache()`. If `NULL` (default), the persistent per-user cache written by `save_cache()` is used.

Value

Invisibly returns `NULL`.

See Also

`save_cache()`, `clear_cache()`

Examples

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

mrca *Compute the most recent common ancestor of two taxa*

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 *Plot method for taxodist_cluster objects*

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$hclust)) x$hclust$method else "None"),
  ...
)
```

Arguments

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

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 taxo_ordinate().
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 graphics::text().

Value

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

Examples

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

`print.taxodist_bundle` *Print a taxodist bundle*

Description

Print a taxodist bundle

Usage

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

Arguments

`x`	A `taxodist_bundle` object.
`...`	Additional arguments (currently ignored).

Value

Invisibly returns `x`.

`print.taxodist_bundle_comparison` *Print a taxodist bundle comparison*

Description

Print a taxodist bundle comparison

Usage

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

Arguments

x	A taxodist_bundle_comparison returned by compare_bundles().
...	Additional arguments (currently ignored).

Value

Invisibly returns x.

---

print.taxodist_distinctness *Print taxonomic distinctness results*

Description

Print taxonomic distinctness results

Usage

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

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

Arguments

x	A taxodist_distinctness object from taxo_distinctness() or a taxodist_distinctness_test object from taxo_distinctness_test().
...	Additional arguments passed to print.data.frame().

Value

Invisibly returns x.

---

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_resolution *Print an auditable taxon-resolution table*

Description

Print an auditable taxon-resolution table

Usage

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

Arguments

x	A taxodist_resolution object returned by taxo_resolve().
...	Additional arguments passed to print.data.frame().

Value

Invisibly returns `x`.

---

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).

---

read_taxodist_bundle *Read a portable taxodist JSON bundle*

Description

Read a portable taxodist JSON bundle

Usage

```
read_taxodist_bundle(file)
```

Arguments

`file`	Path to a JSON file created by `write_taxodist_bundle()`.

Value

A reconstructed `taxodist_bundle` object.

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 = NULL)
```

Arguments

`file`	Path to the `.rds` file to write. If `NULL` (default), the cache is written to a persistent per-user location given by `tools::R_user_dir()`, which is created if necessary.

Value

Invisibly returns the path of the written file.

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.

taxobase *Reference Taxonomic Dataset (taxobase)*

Description

A pre-computed reference dataset for 52 candidate genera distributed across major taxonomic groups. The data were retrieved from The Taxonomicon using the package functions and are intended for examples, documentation, and offline inspection of package output structures.

Usage

```
data(taxobase)
```

Format

A structured list containing the following pre-computed components:

taxa Character vector of 52 candidate genera.

found_taxa Character vector of candidates found in The Taxonomicon.

coverage Named logical vector indicating database presence of each taxon.

matrix A pre-computed taxonomic `dist` object for `found_taxa`.

pairwise A pre-computed distance result for *Tyrannosaurus* and *Homo*.

lineage_homo Taxonomic lineage returned for *Homo*.

lineage_tyrannosaurus Taxonomic lineage returned for *Tyrannosaurus*.

closest Result of a closest-relative query for *Tyrannosaurus*.

filter Taxa in `found_taxa` assigned to Theropoda.

search Reference search result for the term `"Bacteria"`.

statistical_taxa Character vector used by the statistical applications vignette.

statistical_matrix Pre-computed `dist` object for `statistical_taxa`.

metadata Data source, generation date, package version, and distance definition.

Source

http://taxonomicon.taxonomy.nl

Examples

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

taxonomicon_available *Check whether The Taxonomicon can be reached*

Description

Sends a single quick request to The Taxonomicon and reports whether it answered. Useful to skip code that needs the server, such as examples, when there is no internet connection or the server is down; lineage retrieval itself already handles failed requests.

Usage

```
taxonomicon_available(timeout = 3)
```

Arguments

timeout	Maximum time to wait for an answer, in seconds.

Value

TRUE if the server answered with a successful response, FALSE otherwise, including when there is no internet connection.

Examples

```
taxonomicon_available()
```

taxo_bundle *Create a portable taxodist analysis bundle*

Description

Combines an auditable taxon-resolution table, its distance matrix, the distance definition, and provenance metadata in one self-contained object. The bundle can be written to a language-independent JSON file with write_taxodist_bundle().

Usage

```
taxo_bundle(
  taxa,
  ambiguity = c("warn", "first", "error"),
  verbose = FALSE,
  progress = TRUE,
  within = NULL,
  metric = "reciprocal"
)
```

Arguments

taxa	A character vector accepted by taxo_resolve() or an existing taxodist_resolution object.
ambiguity, verbose, progress, within	Passed to taxo_resolve() when taxa is a character vector.
metric	Distance metric used for the stored matrix; see taxo_distance(). The bundle records it, so the matrix can be verified from the stored lineages (and, for "ranked", the stored ranks).

Value

An object of class taxodist_bundle containing schema_version, created_at, source, software, metric, resolution, and matrix.

See Also

write_taxodist_bundle(), read_taxodist_bundle(), compare_bundles()

Examples

```
bundle <- taxo_bundle(c("Tyrannosaurus", "Triceratops", "Homo"))
distance_matrix(bundle)
```

---

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, or metric).

Value

An object of class "taxodist_cluster" — a list with:

hclust The stats::hclust() result.

dist The underlying distance matrix.

If fewer than two taxa are supplied, hclust is NULL and the distance object is preserved.

See Also

taxo_tree() for a tree whose internal nodes are the named clades of the classification, 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 taxonomic hierarchy 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). With the default metric,

Usage

```
taxo_distance(taxon_a, taxon_b, verbose = FALSE, metric = "reciprocal")
```

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.
metric	Distance metric: one of "reciprocal" (default), "path", "wu_palmer", or "ranked". See the Metrics section.

Details

$$d(A, A) = 0$$
$$d(A, B) = 1/h(A, B)$$

Here, `h(A,B)` is the depth of the MRCA of `A` and `B`. The deeper the shared ancestor, the smaller (closer to zero) the distance. Distinct nodes always have positive distance, including when one taxon is an ancestor of the other. This makes the measure an ultrametric on each connected taxonomic hierarchy. Use `is_member()` when the question is whether one taxon belongs to a clade rather than how far apart the two hierarchy nodes are.

Value

A named list of class `"taxodist_result"` with the following elements:

`distance` Numeric. The distance between the two taxa. Returns 0 only when their complete lineages are identical.

`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.

`metric` Character. The metric used.

Returns `NULL` if either taxon cannot be found.

Metrics

Let h be the MRCA depth and d_A, d_B the lineage depths (numbers of nodes, root included). Every metric is zero for identical lineages.

`"reciprocal"` **(default)** $1/h$. An ultrametric that weights shallow divergences heavily and compresses deep ones.

`"path"` $d_A + d_B - 2h$, the number of edges between the two nodes in the classification tree. A tree metric, not an ultrametric when depths differ.

`"wu_palmer"` $1 - 2h/(d_A + d_B)$, the complement of the Wu-Palmer similarity. It lies in $[0, 1)$ and normalizes for the different resolution of different parts of the classification. It is not guaranteed to satisfy the triangle inequality.

`"ranked"` Half the path length after reducing both lineages to the nodes of the standard Linnaean ranks (kingdom, phylum, class, order, family, genus, species; see the `taxodist.rank_levels` option), the root, and the terminal node. For taxa of the same rank this is the number of rank levels up to their shared rank, i.e. the step weights of Clarke and Warwick's taxonomic distinctness: 1 for congeneric species, 2 for confamilial species, and so on. Requires lineage ranks (see `lineage_ranks()`).

Lineages without a shared root have infinite distance, except under `"wu_palmer"`, where their distance is 1.

References

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

Clarke, K.R. & Warwick, R.M. (1998). A taxonomic distinctness index and its statistical properties. *Journal of Applied Ecology* 35, 523–531.

Wu, Z. & Palmer, M. (1994). Verb semantics and lexical selection. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, 133–138.

See Also

mrca(), distance_matrix(), get_lineage(), is_member(), taxo_path(), taxo_distinctness()

Examples

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

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

# The same pair under other metrics
taxo_distance("Tyrannosaurus", "Velociraptor", metric = "path")
taxo_distance("Tyrannosaurus", "Velociraptor", metric = "ranked")
```

---

taxo_distinctness *Taxonomic diversity and distinctness indices*

Description

Computes the taxonomic diversity and distinctness indices of Warwick and Clarke for one or more samples: taxonomic diversity Δ, taxonomic distinctness Δ^*, average taxonomic distinctness Δ^+, and variation in taxonomic distinctness Λ^+.

Usage

```
taxo_distinctness(
  comm,
  d = NULL,
  metric = "ranked",
  verbose = FALSE,
  progress = TRUE
)
```

Arguments

comm	The samples. One of: • a character vector of taxon names (one sample, presence only); • a named numeric vector of abundances, named by taxon (one sample); • a matrix or data frame of abundances or presences with one row per sample and one column per taxon (column names are taxon names).
d	Optional dist object whose labels include every taxon present in comm, for example from distance_matrix(). If NULL (default), distances are computed with distance_matrix() and metric.
metric	Distance metric used when d is NULL; see taxo_distance(). The default "ranked" gives the step weights of Clarke and Warwick (1 for congeneric taxa, 2 for confamilial, ...).
verbose, progress	Passed to distance_matrix() when d is NULL.

Details

With x_i the abundance of taxon i, $n = \sum x_i$, S the number of taxa present, and ω_{ij} the distance between taxa i and j, sums running over pairs $i < j$:

$$\Delta = \sum\sum \omega_{ij} x_i x_j / [n(n-1)/2]$$

$$\Delta^* = \sum\sum \omega_{ij} x_i x_j / \sum\sum x_i x_j$$

$$\Delta^+ = \sum\sum \omega_{ij} / [S(S-1)/2]$$

$$\Lambda^+ = \sum\sum (\omega_{ij} - \Delta^+)^2 / [S(S-1)/2]$$

Δ^+ and Λ^+ use presence only and, unlike species richness, their expected values do not depend on sampling effort; see taxo_distinctness_test(). With presence-only data, Δ and Δ^* are reported for completeness and equal Δ^+.

Any metric can be used. The "ranked" metric reproduces the classical indices, which use equal steps between Linnaean ranks; the other metrics use the full resolution of The Taxonomicon classification.

Value

A data frame of class "taxodist_distinctness" with one row per sample and columns:

sample Sample name (row name of comm, or "sample").

S Number of taxa present.

Delta Taxonomic diversity Δ.

Delta_star Taxonomic distinctness Δ^*.

Delta_plus Average taxonomic distinctness Δ^+.

Lambda_plus Variation in taxonomic distinctness Λ^+.

SDelta_plus Total taxonomic distinctness $S\Delta^+$.

The indices are NA for samples with fewer than two taxa, or when a distance between present taxa is missing or infinite.

References

Warwick, R.M. & Clarke, K.R. (1995). New 'biodiversity' measures reveal a decrease in taxonomic distinctness with increasing stress. *Marine Ecology Progress Series* 129, 301–305.

Clarke, K.R. & Warwick, R.M. (1998). A taxonomic distinctness index and its statistical properties. *Journal of Applied Ecology* 35, 523–531.

Clarke, K.R. & Warwick, R.M. (2001). A further biodiversity index applicable to species lists: variation in taxonomic distinctness. *Marine Ecology Progress Series* 216, 265–278.

See Also

`taxo_distinctness_test()`, `distance_matrix()`

Examples

```
lineages <- list(
  A = c("Biota", "Animalia", "Chordata", "Mammalia", "Carnivora", "A"),
  B = c("Biota", "Animalia", "Chordata", "Mammalia", "Carnivora", "B"),
  C = c("Biota", "Animalia", "Chordata", "Mammalia", "Primates", "C"),
  D = c("Biota", "Animalia", "Arthropoda", "Insecta", "Diptera", "D")
)
d <- distance_matrix(taxo_from_lineages(lineages), metric = "path")
comm <- rbind(
  site1 = c(A = 5, B = 3, C = 1, D = 0),
  site2 = c(A = 1, B = 0, C = 2, D = 4)
)
taxo_distinctness(comm, d)
```

---

taxo_distinctness_test

Test taxonomic distinctness against random samples from a species pool

---

Description

Compares the average taxonomic distinctness Δ^+ and its variation Λ^+ of a sample with their distribution in random samples of the same number of taxa drawn from a master list (species pool), as proposed by Clarke and Warwick.

Usage

```
taxo_distinctness_test(sample, d, n_iter = 999)
```

Arguments

`sample`	A character vector of the taxa present in the sample.
`d`	A `dist` object over the master list, for example from `distance_matrix()`. Every taxon in `sample` must be one of its labels.
`n_iter`	Number of random samples. Default `999`.

Details

The expected Δ^+ is the mean distance over all pairs of the master list, which does not depend on the sample size. The expected Λ^+ is the mean of the simulated values. A value of Δ^+ below the lower limit indicates that the sample is less taxonomically distinct than expected from the pool, a pattern associated with environmental degradation. Use `set.seed()` for reproducible results.

Value

An object of class `"taxodist_distinctness_test"`, a list with:

`statistics` A data frame with one row for `Delta_plus` and one for `Lambda_plus`, and columns `observed`, `expected`, `lower`, `upper` (the 2.5% and 97.5% quantiles of the random samples), `p_lower`, and `p_upper` (one-sided permutation p-values for values as low or as high as observed).

`simulated` A matrix of the simulated values, one row per random sample.

`S`, `pool_size`, `n_iter`, `metric` Test settings.

References

Warwick, R.M. & Clarke, K.R. (1995). New 'biodiversity' measures reveal a decrease in taxonomic distinctness with increasing stress. *Marine Ecology Progress Series* 129, 301–305.

Clarke, K.R. & Warwick, R.M. (1998). A taxonomic distinctness index and its statistical properties. *Journal of Applied Ecology* 35, 523–531.

Clarke, K.R. & Warwick, R.M. (2001). A further biodiversity index applicable to species lists: variation in taxonomic distinctness. *Marine Ecology Progress Series* 216, 265–278.

See Also

`taxo_distinctness()`

Examples

```
lineages <- list(
  A = c("Biota", "Animalia", "Chordata", "Mammalia", "Carnivora", "A"),
  B = c("Biota", "Animalia", "Chordata", "Mammalia", "Carnivora", "B"),
  C = c("Biota", "Animalia", "Chordata", "Mammalia", "Primates", "C"),
  D = c("Biota", "Animalia", "Arthropoda", "Insecta", "Diptera", "D"),
  E = c("Biota", "Animalia", "Arthropoda", "Insecta", "Coleoptera", "E"),
  F = c("Biota", "Plantae", "Tracheophyta", "Magnoliopsida", "Fagales", "F")
)
pool <- distance_matrix(taxo_from_lineages(lineages), metric = "path")
set.seed(1)
taxo_distinctness_test(c("A", "B", "C"), pool, n_iter = 199)
```

taxo_from_lineages *Create an auditable resolution from user-supplied lineages*

Description

Builds a taxodist_resolution object without consulting an online taxonomy service. This is useful for unpublished classifications, curated local taxonomies, frozen analyses, and fully offline workflows.

Usage

```
taxo_from_lineages(
  lineages,
  ids = NULL,
  source = "user-supplied",
  ranks = NULL
)
```

Arguments

lineages	A named list of character vectors ordered from root to the focal taxon. List names become the input labels.
ids	Optional character vector of unique identifiers, in the same order as lineages. A named vector is matched by name. If omitted, deterministic identifiers of the form custom:<input label> are created.
source	A non-empty character string describing the lineage source.
ranks	Optional named list of character vectors giving the rank of every node, aligned with lineages (use NA for unranked nodes). Ranks are required only by the "ranked" distance method. A list without names is matched by position.

Value

A taxodist_resolution object that can be passed to distance_matrix() or taxo_bundle().

See Also

taxo_resolve(), distance_matrix(), taxo_bundle()

Examples

```
lineages <- list(
  Alpha = c("Biota", "Animalia", "Alpha"),
  Beta = c("Biota", "Animalia", "Beta")
)
resolved <- taxo_from_lineages(lineages)
distance_matrix(resolved)
```

```
# With ranks, the rank-based distance can be used
ranked <- taxo_from_lineages(
  lineages,
  ranks = list(
    Alpha = c(NA, "Kingdom", "Genus"),
    Beta = c(NA, "Kingdom", "Genus")
  )
)
distance_matrix(ranked, metric = "ranked")
```

`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,
  ...,
  verbose = FALSE,
  progress = TRUE,
  metric = "reciprocal"
)
```

Arguments

`taxa`	A character vector of taxon names, a `taxodist_resolution`, a `taxodist_bundle`, or a `dist` object from `distance_matrix()`.
`...`	Additional graphical arguments passed to `stats::heatmap()` (for example `col` or `main`).
`verbose`, `progress`, `metric`	Passed to `distance_matrix()` when `taxa` is not already a `dist` object.

Value

Invisibly returns the underlying `dist` object. Called primarily for its side effect (plotting). If fewer than two taxa are supplied, plotting is skipped.

Examples

```
taxa <- c("Tyrannosaurus", "Velociraptor", "Homo", "Panthera", "Quercus")
taxo_heatmap(taxa, col = hcl.colors(12, "YlOrRd"), progress = FALSE)
```

`taxo_ordinate` *Ordinate taxa in taxonomic distance space*

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 `dist` object from `distance_matrix()`.
`k`	Number of dimensions. Default 2.
`...`	Additional arguments passed to `distance_matrix()` (e.g. `metric`).

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.

At least two taxa are required. When `k` is greater than the maximum possible dimension (`n - 1`), it is reduced automatically.

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 *Get the taxonomic path between two taxa*

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 or if their lineages have no common ancestor.

See Also

mrca(), shared_clades(), compare_lineages()

Examples

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

taxo_resolve *Resolve taxon names in a batch with an auditable result*

Description

Resolves taxon names or numeric Taxonomicon IDs while preserving the candidates considered, the selected identifier, the retrieved lineage, and the resolution status for every input. This is intended for reproducible workflows in which warnings emitted during a large matrix calculation would otherwise be difficult to audit afterwards.

Usage

```
taxo_resolve(
  taxa,
  ambiguity = c("warn", "first", "error"),
  verbose = FALSE,
  progress = TRUE,
  within = NULL
)
```

Arguments

taxa	A character vector of taxon names or numeric Taxonomicon IDs.
ambiguity	How to handle names with more than one valid biological candidate. "warn" (default) selects the first candidate and emits one aggregate warning, "first" selects it silently, and "error" aborts after all inputs have been inspected.
verbose	Logical. If TRUE, prints retrieval messages.
progress	Logical. If TRUE, displays a progress bar.
within	Optional character string naming a clade (for example "Animalia" or "Plantae"). When supplied, only candidates whose lineage contains this clade are considered, which resolves many homonyms across kingdoms automatically. Matching is exact and case-insensitive. Names without any candidate inside the clade are reported as "unresolved". Numeric IDs are not filtered.

Details

Ambiguous inputs retain status = "ambiguous" even when the first candidate is selected. Consequently, downstream code can distinguish an unambiguous match from a match chosen according to the requested policy. Numeric IDs bypass name search and are treated as single candidates when their lineage can be retrieved.

Names are matched after trimming and collapsing whitespace, and the stored lineage is truncated at the queried taxon exactly as in get_lineage(), so distance_matrix(taxo_resolve(x)) and distance_matrix(x) agree.

Objects returned by taxo_resolve() can be passed directly to distance_matrix(), which reuses the stored lineages and preserves the original input names as matrix labels.

Value

A data frame of class `taxodist_resolution` with one row per input and columns `input`, `resolved_name`, `id`, `status`, `n_candidates`, `lineage_depth`, `lineage`, and `candidates`. The last two are list columns. Status is one of `"resolved"`, `"ambiguous"`, `"unresolved"`, or `"retrieval_error"`.

See Also

`taxo_search()`, `distance_matrix()`, `check_coverage()`

Examples

```
resolved <- taxo_resolve(c("Tyrannosaurus", "Nereis", "50841"))
distance_matrix(resolved)

# Restrict homonyms to animals
taxo_resolve("Nereis", within = "Animalia")
```

---

taxo_search *Search The Taxonomicon for a taxon name*

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_resolve()`

Examples

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

`taxo_tree` *Build a taxonomic tree from lineages*

Description

Builds a tree whose internal nodes are the named clades of the source classification, directly from the lineages of the supplied taxa. Unlike `taxo_cluster()`, which reconstructs a dendrogram from distances, the topology here is the classification itself and internal nodes carry clade names.

Usage

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

Arguments

`taxa`	A character vector of taxon names, a `taxodist_resolution` from `taxo_resolve()` or `taxo_from_lineages()`, or a `taxodist_bundle`.
`verbose`, `progress`	Passed to `get_lineage()` retrieval when `taxa` is a character vector.

Details

Clades with a single descendant among the supplied taxa are collapsed, so every internal node is a point where the taxa diverge and is labeled with the deepest clade shared by its descendants (their MRCA).

Branch lengths are chosen so that the tree is ultrametric and its cophenetic (patristic) distances reproduce `distance_matrix()` exactly: an internal node for a clade at depth `h` is placed at height `1 / (2h)` and tips at height zero. A taxon that is an ancestor of another supplied taxon is attached to its clade node by a branch of length `1 / (2h)`, and taxa with identical lineages are joined by zero-length branches.

All lineages must share the same root node; otherwise distances between some taxa are infinite and no single tree exists.

Value

An object of class `"phylo"`, compatible with the **ape** package (for example `ape::plot.phylo()`, `ape::write.tree()`, and `ape::cophenetic.phylo()`). Tip labels are the input labels and node labels are clade names. Taxa without a lineage are dropped with a warning.

See Also

taxo_cluster(), distance_matrix()

Examples

```
resolved <- taxo_from_lineages(list(
  Tyrannosaurus = c("Biota", "Dinosauria", "Theropoda", "Tyrannosaurus"),
  Velociraptor  = c("Biota", "Dinosauria", "Theropoda", "Velociraptor"),
  Triceratops   = c("Biota", "Dinosauria", "Ornithischia", "Triceratops"),
  Homo          = c("Biota", "Mammalia", "Homo")
))
tree <- taxo_tree(resolved)
tree$node.label
if (requireNamespace("ape", quietly = TRUE)) {
  plot(tree, show.node.label = TRUE)
  ape::write.tree(tree)
}
```

---

write_taxodist_bundle *Write a taxodist bundle as portable JSON*

Description

Write a taxodist bundle as portable JSON

Usage

```
write_taxodist_bundle(bundle, file, pretty = TRUE)
```

Arguments

bundle	A taxodist_bundle created by taxo_bundle().
file	Path to the JSON file to create.
pretty	Logical. If TRUE (default), indent the JSON for readability.

Value

Invisibly returns the normalized output path.

Index

∗ **datasets**
taxobase, 27

cache_info, 5
check_coverage, 5
check_coverage(), *3*, *15*, *41*
clear_cache, 6
clear_cache(), *3*, *5*, *18*, *25*
closest_relative, 7
closest_relative(), *3*, *11, 12*
compare_bundles, 7
compare_bundles(), *3*, *22*, *29*
compare_lineages, 8
compare_lineages(), *3*, *39*
cross_distances, 9
cross_distances(), *3*, *11, 12*

distance_matrix, 10
distance_matrix(), *3*, *9*, *12*, *29, 30*, *32–34*, *36–38*, *40–43*

filter_clade, 11
filter_clade(), *3*
focal_distances, 12
focal_distances(), *9*, *23*

get_lineage, 13
get_lineage(), *3*, *14, 15*, *17*, *32*, *40–42*
get_lineage_by_id, 14
get_lineage_by_id(), *13*
get_taxonomicon_id, 15
get_taxonomicon_id(), *13, 14*
graphics::text(), *20*

is_member, 16
is_member(), *3*, *31, 32*

lineage_depth, 16
lineage_depth(), *3*
lineage_ranks, 17
lineage_ranks(), *31*
load_cache, 18
load_cache(), *3*, *5*, *25*

mrca, 19
mrca(), *3*, *32*, *39*

plot.taxodist_cluster, 19
plot.taxodist_ord, 20
print.data.frame(), *22, 23*
print.taxodist_bundle, 21
print.taxodist_bundle_comparison, 21
print.taxodist_distinctness, 22
print.taxodist_distinctness_test *(*print.taxodist_distinctness*)*, 22
print.taxodist_focal, 22
print.taxodist_path, 23
print.taxodist_resolution, 23
print.taxodist_result, 24

read_taxodist_bundle, 24
read_taxodist_bundle(), *8*, *29*

save_cache, 25
save_cache(), *3*, *5*, *18*
set.seed(), *35*
shared_clades, 25
shared_clades(), *3*, *39*
stats::cmdscale(), *38*
stats::hclust(), *29, 30*
stats::heatmap(), *37*
stats::plot.hclust(), *20*
summary.taxodist_ord, 26

taxo_bundle, 28
taxo_bundle(), *3*, *8*, *11*, *36*, *43*
taxo_cluster, 29
taxo_cluster(), *19, 20*, *38*, *42, 43*
taxo_distance, 30
taxo_distance(), *3*, *7*, *9–12*, *14*, *17*, *24*, *29*, *33*

taxo_distinctness, 32
taxo_distinctness(), *22*, *32*, *35*
taxo_distinctness_test, 34
taxo_distinctness_test(), *22*, *33, 34*
taxo_from_lineages, 36
taxo_from_lineages(), *3, 4*, *9*, *42*
taxo_heatmap, 37
taxo_ordinate, *26*, 38
taxo_ordinate(), *20*, *30*
taxo_path, 39
taxo_path(), *23*, *32*
taxo_resolve, 40
taxo_resolve(), *3*, *6*, *9–11*, *15*, *23*, *29*, *36*, *41, 42*
taxo_search, 41
taxo_search(), *41*
taxo_tree, 42
taxo_tree(), *3*, *11*, *30*
taxobase, 27
taxodist *(*taxodist-package*)*, 3
taxodist-package, 3, *15*
taxonomicon_available, 28
tools::R_user_dir(), *25*

write_taxodist_bundle, 43
write_taxodist_bundle(), *8*, *24*, *28, 29*