Package 'taxodist'

September 24, 2026

Title Taxonomic Hierarchy Distances and Lineage Analysis

Version 0.8.0

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. Functions are provided for auditable name resolution, online or user-supplied lineage analysis, clade membership, pairwise and matrix distance calculation, hierarchical clustering, principal coordinates analysis, portable 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), purrr (>= 0.3.0), cli (>= 3.0.0), jsonlite (>= 1.8.0), utils, stats, graphics

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

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 2026-09-24 08:50:02 UTC

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_bundle . . . 16
print.taxodist_focal . . . 16
print.taxodist_path . . . 17
print.taxodist_resolution . . . 17
print.taxodist_result . . . 18
read_taxodist_bundle . . . 18
save_cache . . . 19
shared_clades . . . 19
summary.taxodist_ord . . . 20
taxobase . . . 21
taxo_bundle . . . 22
taxo_cluster . . . 22
taxo_distance . . . 23
taxo_from_lineages . . . 25
taxo_heatmap . . . 26
taxo_ordinate . . . 26
taxo_path . . . 27
taxo_resolve . . . 28
taxo_search . . . 29
write_taxodist_bundle . . . 30

Index **31**

`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
- `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
- `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()`.

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 *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 indicates it was not.

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)
```

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 (ultrametric 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_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")
```

`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)
```

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`.

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. An empty input returns an empty `"dist"` object.

See Also

`taxo_distance()`, `closest_relative()`, `taxo_resolve()`, `taxo_bundle()`

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)
```

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

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_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 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 get_taxonomicon_id().
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). 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.

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 *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
```

---

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)
```

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 *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_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)
```

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.

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))
```

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
)
```

Arguments

`taxa`	A character vector accepted by `taxo_resolve()` or an existing `taxodist_resolution` object.
`ambiguity, verbose, progress`	Passed to `taxo_resolve()` when `taxa` is a character vector.

Value

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

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).

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_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):

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

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

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

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.

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()`, `is_member()`, `taxo_path()`

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_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")
```

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.

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)
```

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). If fewer than two taxa are supplied, plotting is skipped.

Examples

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

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().

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 fromstats::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
)
```

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.

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.

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)
```

---

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_distance()

Examples

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

---

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, 21

cache_info, 4
check_coverage, 5
check_coverage(), *3*, *29*
clear_cache, 5
clear_cache(), *3*, *4*, *13*, *19*
closest_relative, 6
closest_relative(), *3*, *7*, *9*
compare_lineages, 6
compare_lineages(), *3*, *28*

distance_matrix, 7, *26*
distance_matrix(), *3*, *9*, *23–27*, *29*

filter_clade, 8
filter_clade(), *3*
focal_distances, 8
focal_distances(), *16*

get_lineage, 9
get_lineage(), *3*, *10*, *11*, *24*, *30*
get_lineage_by_id, 10
get_lineage_by_id(), *9*
get_taxonomicon_id, 11
get_taxonomicon_id(), *9*, *10*
graphics::text(), *15*

heatmap, *26*

is_member, 12
is_member(), *3*, *24*

lineage_depth, 12
lineage_depth(), *3*
load_cache, 13
load_cache(), *4*, *19*

mrca, 14
mrca(), *3*, *24*, *28*

plot.taxodist_cluster, 14
plot.taxodist_ord, 15
print.data.frame(), *17*
print.taxodist_bundle, 16
print.taxodist_focal, 16
print.taxodist_path, 17
print.taxodist_resolution, 17
print.taxodist_result, 18

read_taxodist_bundle, 18

save_cache, 19
save_cache(), *4*, *13*
shared_clades, 19
shared_clades(), *3*, *28*
stats::cmdscale(), *27*
stats::hclust(), *23*
stats::plot.hclust(), *15*
summary.taxodist_ord, 20

taxo_bundle, 22
taxo_bundle(), *3*, *7*, *25*, *30*
taxo_cluster, 22
taxo_cluster(), *14*, *15*, *27*
taxo_distance, 23
taxo_distance(), *3*, *7*, *9–11*, *18*, *30*
taxo_from_lineages, 25
taxo_from_lineages(), *3*
taxo_heatmap, 26
taxo_ordinate, *20*, 26
taxo_ordinate(), *15*, *23*
taxo_path, 27
taxo_path(), *17*, *24*
taxo_resolve, 28
taxo_resolve(), *3*, *7*, *17*, *22*, *25*
taxo_search, 29
taxo_search(), *29*
taxobase, 21
taxodist *(*taxodist-package*)*, 3
taxodist-package, 3

write_taxodist_bundle, 30
write_taxodist_bundle(), *18*, *22*