Package 'Rclade'

November 2, 2026

Title Automated Deep-Time Phylogenetic Tree Collapsing and Visualization

Version 1.1.6

Author Zeng Zichao [aut, cre]

Maintainer Zeng Zichao <zengzichao@sjtu.edu.cn>

Description Provides a single-function pipeline for automated collapsing and visualization of large phylogenetic trees with geological timescales. Automatically parses taxonomic labels from multiple formats (GTDB, Silva, NCBI, embedded, custom), identifies Most Recent Common Ancestors (MRCAs), assigns color-blind-safe palettes, executes batch collapsing with automatic nesting-aware ordering, integrates 'deeptime' geologic time scales with adaptive time breaks and unit switching, and manages smart legend layout. Supports special ancestral node identifiers (LUCA, LACA, LBCA) for highlighting key nodes in the tree of life. Provides external taxonomy file support for trees with incomplete or missing taxonomic labels. Features real-time logging with timestamps, step tracking, and multiple log levels. Includes comprehensive input validation for tree and sequence file formats. Reduces a 60-line manual workflow to a single function call while preserving full compatibility with the 'ggtree'/'deeptime' ecosystem. The visualization pipeline builds on 'ggtree' Yu et al. (2017) <doi:10.1111/2041-210X.12628>, 'deeptime' Gearty (2025) <doi:10.1080/20964471.2025.2537516>, and 'ape' Paradis and Schliep (2019) <doi:10.1093/bioinformatics/bty633>.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Imports ape (>= 5.0), ggtree (>= 4.0.0), deeptime (>= 1.0), ggplot2 (>= 3.5.0), rlang, stringr (>= 1.5), tidytree (>= 0.4), viridisLite

Suggests treeio (>= 1.0), phangorn (>= 2.0), RColorBrewer, cowplot (>= 1.1), patchwork (>= 1.1), shiny (>= 1.7), optparse (>= 1.7), yaml, vdiffr (>= 1.0), testthat (>= 3.0), knitr, rmarkdown, covr, filelock, withr

Encoding UTF-8

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

URL https://github.com/zengzichao/Rclade,
https://zengzichao.github.io/Rclade/

BugReports https://github.com/zengzichao/Rclade/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2026-11-02 17:20:09 UTC

Contents

batch_plot . . . 3
check_monophyly . . . 4
check_special_monophyly . . . 5
detect_taxonomy_format . . . 6
example_tree . . . 7
get_supported_extensions . . . 8
parse_taxonomy . . . 8
plot_timetree . . . 9
polytomy_tree . . . 16
rclade_logo . . . 16
rclade_options . . . 17
read_taxonomy_file . . . 18
read_tree_auto . . . 19
run_rclade_cli . . . 20
run_rclade_selftest . . . 21
run_rclade_shiny . . . 21
save_session_info . . . 22
save_timetree . . . 22
set_log_enabled . . . 23
set_log_file . . . 23
set_log_level . . . 24
summarize_taxonomy_quality . . . 24
summarize_taxonomy_quality_with_file . . . 25
summarize_timetree . . . 26
theme_timetree . . . 26
validate_sequence_deep . . . 27
validate_sequence_file . . . 27
validate_tree_sequence_match . . . 28

Index **29**

batch_plot *Batch plot timetrees from a directory of tree files*

Description

Batch plot timetrees from a directory of tree files

Usage

```
batch_plot(
  input_dir,
  output_dir,
  pattern = "*.tre",
  format = "pdf",
  width = 14,
  height = 10,
  overwrite = "ask",
  ignore_malformed = FALSE,
  ...
)
```

Arguments

input_dir	Input directory containing tree files
output_dir	Output directory for plots
pattern	File matching pattern (glob format, e.g., "*.tre")
format	Output format: "pdf", "png", "tiff", "svg", "eps" (default "pdf")
width	Output width in inches (default 14)
height	Output height in inches (default 10)
overwrite	Overwrite mode: "ask" (default), "force", or "no-clobber"
ignore_malformed	Logical. If TRUE, malformed tree inputs are skipped (with a warning) instead of aborting the whole batch, aligned with the CLI's --ignore_malformed flag. Default: FALSE.
...	Additional arguments passed to plot_timetree()

Value

Invisibly returns list of success/failure status

check_monophyly *Check if a taxonomic group is monophyletic*

Description

Tests whether all tips belonging to a specified taxonomic group form a monophyletic clade in the tree.

Usage

```
check_monophyly(
  tree,
  group,
  rank,
  format = "auto",
  custom_patterns = NULL,
  quiet = FALSE,
  delimiter_mode = "reverse",
  taxonomy_levels = NULL
)
```

Arguments

tree	A phylo object.
group	Character. The name of the taxonomic group to check (e.g., "P1", "Mammalia").
rank	Character. Taxonomic rank of the group. One of "domain", "phylum", "class", "order", "family", "genus", "species", or abbreviations "d", "p", "c", "o", "f", "g", "s".
format	Character. Taxonomy label format. One of "auto", "GTDB", "Silva", "NCBI", "custom_rank", "custom_regex". Default: "auto".
custom_patterns	Named list of regex patterns for custom format. Required when format = "custom_regex".
quiet	Logical. If TRUE, suppress informational messages. Default: FALSE.
delimiter_mode	Character. Embedded (Format A) parsing strategy: "reverse" (right-to-left, default), "greedy" (left-to-right), or "segment" (delimiter-to-delimiter extraction).
taxonomy_levels	Custom taxonomy level configuration (list with codes and names). Default: NULL.

Value

A list with components:

is_monophyletic Logical. Whether the group is monophyletic.

group Character. The group name.

n_tips Integer. Number of tips belonging to the group.

mrca_node Integer or NULL. The MRCA node number, or NULL if the group has fewer than 2 tips.

outsiders Character vector. Tips in the MRCA clade that do not belong to the group (empty if monophyletic).

Case sensitivity (L-A2)

Group matching is **case-insensitive**. The query `group` and every parsed `Group` label are lower-cased with `tolower()` *at comparison time* (not during parsing) before matching, so `"Proteobacteria"` and `"proteobacteria"` match the same clade. `parse_taxonomy` itself preserves the original case of parsed labels; the lower-casing applied here is local to this comparison and keeps the two modules consistent about what a group name refers to.

Examples

```
data(example_tree)

# Check if phylum P1 is monophyletic
result <- check_monophyly(example_tree, "P1",
                          rank = "phylum", format = "GTDB")
if (result$is_monophyletic) {
  message("P1 is monophyletic!")
} else {
  message("P1 is NOT monophyletic.")
}
```

---

check_special_monophyly

Check if a special identifier corresponds to a monophyletic group

---

Description

Tests whether the MRCA of the specified domains (identified by LUCA/LACA/LBCA) contains only tips from those domains (i.e., is monophyletic with respect to the target domains).

Usage

```
check_special_monophyly(
  tree,
  identifier,
  format = "auto",
  quiet = FALSE,
  delimiter_mode = "reverse",
  taxonomy_levels = NULL
)
```

Arguments

tree	A phylo object.
identifier	Character. One of "LUCA", "LACA", "LBCA".
format	Character. Taxonomy label format. Default: "auto".
quiet	Logical. If TRUE, suppress informational messages. Default: FALSE.
delimiter_mode	Character. Embedded parsing strategy: "reverse", "greedy", or "segment". Default: "reverse".
taxonomy_levels	Custom taxonomy level configuration (list with codes and names). Default: NULL.

Value

A list with components:

is_monophyletic Logical. Whether the group is monophyletic.

identifier Character. The identifier name.

node Integer or NULL. The MRCA node number.

n_tips Integer. Number of tips in the target domains.

n_outsiders Integer. Number of outsider tips in the MRCA clade.

outsider_domains Character vector. Domains of outsider tips.

Examples

```
data(example_tree)
result <- check_special_monophyly(example_tree, "LBCA")
if (result$is_monophyletic) {
  message("LBCA is monophyletic!")
}
```

---

detect_taxonomy_format

Detect taxonomy format from tip labels

Description

Detects Format A (embedded: _d_Bacteria_p_...) or Format B (semicolon-delimited: d__Bacteria;p__...).

Usage

```
detect_taxonomy_format(labels)
```

Arguments

labels	Character vector of tip labels

Value

Format name: "embedded", "GTDB", "Silva", "NCBI", or "unknown"

Regex heuristic boundaries (L-C3)

Detection and parsing here are **heuristic regex** passes, not a strict grammar. Known boundaries the caller must respect:

- **Underscores in names**: under `delimiter_mode = "reverse"` the pattern uses `[^_]+?`, so an underscore inside a taxon name (e.g. `_g_Clostridium_sensu_stricto_s_X`) can be misread as a rank separator. Names that contain underscores should use `delimiter_mode = "segment"` (delimiter-to-delimiter extraction), which preserves embedded underscores. For Format B (`;`-delimited) underscores are safe.
- **Custom regex backtracking**: caller-supplied `custom_patterns` are inserted verbatim into the parser. Unbounded repeating groups (e.g. `(.+)+`) can cause **catastrophic backtracking** on adversarial labels. Patterns are rejected above the 200-character limit, but keep sub-patterns bounded and anchored.
- **Detection is content-based**: `detect_taxonomy_format` samples labels; a mixed or malformed corpus may return `"unknown"`, in which case the caller must fall back to an explicit `format=`.
- **GTDB requires a semicolon majority (M-B3)**: the GTDB rule additionally requires that more than half of sampled labels contain a semicolon. Accession-prefixed embedded labels with double-underscore rank separators (e.g. `GCA_xxx_d__Archaea_p__Nanoarchaeota`) satisfy the `[dpcofgsk]__` pattern but contain no semicolons; they are detected as `"embedded"` rather than misclassified as GTDB. The embedded parsers also tolerate double-underscore separators.

---

example_tree *Example phylogenetic tree with GTDB-style labels*

Description

A `phylo` object with 50 tips containing GTDB-format taxonomic labels spanning 5 coded phyla (P1, P2, P3, P4, P5) and 10 coded classes (C1, ..., C10) for demonstration and testing purposes. Branch lengths are in Ma (mega-annum). This tree is fully bifurcating with proper monophyletic groups at all taxonomic levels. All labels are artificial codes, not real taxon names.

Usage

```
data(example_tree)
```

Format

A `phylo` object with 50 tips and 49 internal nodes.

Source

Simulated data for package demonstration.

get_supported_extensions *Get supported file extensions*

Description

Get supported file extensions

Usage

```
get_supported_extensions()
```

Value

Named list of supported extensions by category.

parse_taxonomy *Unified taxonomy parsing entry point*

Description

Parses taxonomic information from tip labels using the specified format and returns a data.frame of group assignments. Intended for use as a stable library API by external workflows (e.g., Snakemake/Nextflow).

Usage

```
parse_taxonomy(
  labels,
  rank,
  format = "auto",
  custom_patterns = NULL,
  taxonomy_levels = NULL,
  delimiter_mode = "reverse"
)
```

Arguments

labels	Character vector of tip labels
rank	Taxonomic rank (abbreviation or full name)
format	Format: "auto", "embedded", "GTDB", "Silva", "NCBI", "custom_rank", "custom_regex"
custom_patterns	Custom regex patterns (required when format = "custom_regex")
taxonomy_levels	Custom taxonomy level configuration (list with codes and names)
delimiter_mode	Character. Embedded parsing strategy: "reverse", "greedy", "segment".

Value

data.frame with columns: label, Group

`plot_timetree` *Plot a phylogenetic tree with geological timescale and taxonomic collapsing*

Description

The main entry point of Rclade. Takes a phylogenetic tree object or file path and produces a publication-ready visualization with automatic taxonomic collapsing, geological timescale integration, and smart legend layout.

Usage

```
plot_timetree(
  tree,
  tree_index = NULL,
  multi_tree_mode = "error",
  rank = "none",
  triangle_mode = "mixed",
  space_mode = "proportional",
  layout = "rectangular",
  angle = 360,
  color_palette = "viridis",
  color_mapping = NULL,
  line_width = 1,
  show_tip_labels = FALSE,
  tip_label_size = 2,
  add_timescale = TRUE,
  timescale_levels = c("eras", "eons"),
  unit = NULL,
  taxonomy_format = "auto",
  custom_patterns = NULL,
  taxonomy_file = NULL,
  taxonomy_file_sep = "auto",
  taxonomy_file_header = FALSE,
  taxonomy_file_priority = TRUE,
  taxonomy_source_priority = NULL,
  taxonomy_table_sep = ";",
  taxonomy_delimiter_mode = "reverse",
  legend_position = "bottom",
  legend_nrow = NULL,
  legend_ncol = NULL,
  legend_title = NULL,
  clade = NULL,
```

```
  strict = FALSE,
  groups = NULL,
  groups_label_only = FALSE,
  show_clade_label = FALSE,
  show_clade_count = TRUE,
  clade_label_offset = 50,
  clade_label_fontsize = 3,
  show_support = FALSE,
  support_threshold = 0.95,
  show_hpd = FALSE,
  hpd_color = "firebrick",
  geo_events = FALSE,
  timescale_version = "ICS 2023/02",
  main_title = NULL,
  sub_title = NULL,
  highlight = NULL,
  highlight_alpha = 0.2,
  theme_fun = theme_timetree,
  output = NULL,
  overwrite = "ask",
  width = 14,
  height = 10,
  taxonomy_levels = NULL,
  low_memory = FALSE,
  ignore_malformed = FALSE,
  ignore_branch_length = FALSE,
  color_rank = NULL,
  timescale_mode = "radial",
  timescale_position = "right",
  tree_start_position = "right",
  opts = NULL
)
```

Arguments

tree	A phylo or treedata object, or a file path string pointing to a Newick (.nwk, .tre, .treefile) or Nexus (.nexus, .nex) file.
tree_index	Integer. Index of tree to use from multiPhylo objects (e.g., BEAST posterior). Only used when tree is a file path. Default: NULL (uses multi_tree_mode).
multi_tree_mode	Character. How to handle multiple trees in a file. Options: "error" (default), "ask" (interactive prompt; falls back to "error" in non-interactive sessions), "first", "last", "random", "all" (returns a list of plots in batch mode), "split" (same code path as "all" but signals per-tree numbered output for pipeline dispatching). When "error", stops with informative message asking user to specify.
rank	Taxonomic rank to collapse at. One of: "none", "kingdom", "domain", "phylum", "class", "order", "family", "genus", "species", "subspecies", or abbre-

viations "k", "d", "p", "c", "o", "f", "g", "s", "ss". Default: "none". Mutually exclusive with groups.

triangle_mode
: Collapse triangle visualization mode. "max" shows the full clade range, "min" shows the minimal range, "mixed" uses adaptive selection, "none" disables triangles. Default: "mixed".

space_mode
: Space allocation strategy for collapsed clades. "equal" assigns equal vertical space to each clade, "proportional" allocates space proportional to tip count. Default: "proportional".

layout
: Tree layout. "rectangular" for standard rectangular layout, "circular" for fan/circular layout. Default: "rectangular".

angle
: Fan angle in degrees for circular layout (0-360). Only used when layout = "circular". Default: 360.

color_palette
: Color palette specification. Can be:

  - A palette name: "viridis" (default), "rainbow", or any RColorBrewer palette name (e.g., "Set1", "Paired")
  - A character vector of hex color codes (recycled if shorter than needed)

color_mapping
: Named character vector of specific color assignments. Takes highest priority over color_palette. Names should match group names. Example: c("Proteobacteria" = "#E41A1C", "Firmicutes" = "#377EB8").

line_width
: Branch line width (passed to ggtree). Default: 1.

show_tip_labels
: Logical. Whether to display tip labels. Default: FALSE (recommended for trees with >100 tips).

tip_label_size
: Numeric. Tip label font size. Default: 2.

add_timescale
: Logical. Whether to add a geological timescale to the x-axis. Requires rectangular layout and valid edge lengths. Default: TRUE.

timescale_levels
: Character vector of timescale levels to display. Options: "eras", "eons", "periods". Default: c("eras", "eons").

unit
: Time unit of input tree edge lengths: "Ga" (giga-annum) or "Ma" (mega-annum). **Required whenever** add_timescale = TRUE: the pipeline aborts if unit is NULL, because Rclade does not infer branch-length units and does not verify that the input tree is a time-calibrated chronogram. If "Ga", edge lengths are converted to Ma (multiplied by 1000). NULL (default) leaves the tree's native units untouched and is only valid with add_timescale = FALSE. Rclade displays the user-supplied time calibration only; it never estimates divergence times. Default: NULL.

taxonomy_format
: Taxonomy label format. "auto" enables automatic detection via prefix-matching heuristic. Manual options: "GTDB", "Silva", "NCBI", "custom_rank", "custom_regex". Default: "auto".

custom_patterns
: Named list of regex patterns for custom format parsing. Required when taxonomy_format = "custom_regex". Example: list(domain = "Domain:([^|]+)", phylum = "Phylum:([^|]+)").

`taxonomy_file` Character. Path to an external taxonomy file. The file should have two columns: (1) tip labels and (2) taxonomy strings in GTDB format (e.g., `d__Bacteria;p__Proteobacteria;c__Gar`. Supports tab-delimited or comma-delimited files. When provided, this file supplements or overrides label-based taxonomy parsing. Default: `NULL`.

`taxonomy_file_sep` Character. Column separator for taxonomy file. `"auto"` auto-detects tab or comma. Default: `"auto"`.

`taxonomy_file_header` Logical. Whether taxonomy file has a header row. Default: `FALSE`.

`taxonomy_file_priority` Logical. If `TRUE` (default), file taxonomy takes priority over label-based parsing. If `FALSE`, file is used only for labels that cannot be parsed from the tree. Default: `TRUE`. Superseded by `taxonomy_source_priority` if provided.

`taxonomy_source_priority` Character. Which taxonomy source takes priority when both embedded (label-based) and table (file-based) taxonomy are available: `"embedded"` or `"table"`. Default: `NULL` (use `taxonomy_file_priority`).

`taxonomy_table_sep` Character. Separator between taxonomy ranks in the second column of an external taxonomy file. Default: `";"`.

`taxonomy_delimiter_mode` Character. Embedded (Format A) parsing strategy: `"reverse"` (right-to-left, default), `"greedy"` (left-to-right), or `"segment"` (delimiter-to-delimiter extraction).

`legend_position` Legend placement. Can be a cardinal direction (`"bottom"`, `"top"`, `"right"`, `"left"`, `"none"`) or a length-2 numeric vector `c(x, y)` for inside placement (ggplot2 >= 3.5). Default: `"bottom"` (horizontal layout).

`legend_nrow` Integer. Number of rows in legend grid. Auto-computed if NULL.

`legend_ncol` Integer. Number of columns in legend grid. Auto-computed if NULL.

`legend_title` Character. Custom legend title. If `NULL`, the rank name (e.g., "Phylum", "Class") is used automatically. Default: `NULL`.

`clade` Character. Specific clade name to collapse (e.g., "Cyanobacteriota"). When provided, only this clade is checked for monophyly and collapsed if valid. Mutually exclusive with `rank` and `groups`. Default: `NULL`.

`strict` Logical. If `TRUE`, non-monophyletic clades cause termination with error. If `FALSE` (default), non-monophyletic clades trigger a warning and are skipped. The permissive default (`FALSE`) is intentional: in interactive exploration of large trees with potentially inaccurate taxonomy, halting on the first non-monophyletic group would prevent the user from seeing any result. Use `TRUE` in production pipelines where taxonomic integrity is critical. Default: `FALSE`.

`groups` Named list of custom tip groups for collapsing. Each element is a character vector of tip labels belonging to that group. Groups must be monophyletic; non-monophyletic groups will raise an error. When `groups` is provided, `rank` is ignored (set it to `"none"`). Example: `list("Group_A" = c("tip1", "tip2"), "Group_B" = c("tip3"))`.

groups_label_only
: Logical. If TRUE, clades defined by groups are labelled but NOT collapsed: the member branches stay fully expanded and only the group name is drawn next to them. Default: FALSE.

show_clade_label
: Logical. Whether to add clade labels next to collapsed triangles showing group name and species count. Default: FALSE.

show_clade_count
: Logical. Whether to show species count in clade labels (e.g., "Proteobacteria (n=42)"). Only used when show_clade_label = TRUE. Default: TRUE.

clade_label_offset
: Numeric. Horizontal offset for clade labels from the right edge of the collapsed triangle in Ma units (0-5000). Default: 50.

clade_label_fontsize
: Numeric. Font size for clade labels (1-20). Default: 3.

show_support
: Logical. Whether to display node support values. Requires treedata input with posterior/bootstrap annotations. Default: FALSE.

support_threshold
: Numeric. Minimum support value to display (0-1). Default: 0.95.

show_hpd
: Logical. Whether to display HPD (Highest Posterior Density) intervals. Requires node data with HPD annotations. Default: FALSE.

hpd_color
: Color for HPD bars. Default: "firebrick".

geo_events
: Data frame of geological events to annotate, with columns name, age_min, age_max (Ma), color. If TRUE, default events (GOE=2400-2000 Ma, NOE=800-550 Ma) are shown. Default: FALSE.

timescale_version
: Geological timescale version string. Currently only "ICS 2023/02" is supported. Default: "ICS 2023/02".

main_title
: Character. Main title for the plot. Default: NULL (no title).

sub_title
: Character. Subtitle for the plot. Default: NULL (no subtitle).

highlight
: Character vector of group names to highlight with colored backgrounds. Only monophyletic groups are highlighted; non-monophyletic groups trigger a warning. Supports special identifiers: "LUCA" (Last Universal Common Ancestor), "LACA" (Last Archaeal Common Ancestor), "LBCA" (Last Bacterial Common Ancestor). Default: NULL (no highlighting).

highlight_alpha
: Numeric. Transparency of highlight color (0-1). Default: 0.2.

theme_fun
: Theme function or NULL. Default: theme_timetree. Set to NULL to use default ggplot2 theme.

output
: Optional output file path. If provided, the plot is saved immediately (backward compatibility). Default: NULL.

overwrite
: Character. Overwrite mode when output file exists: "ask" (default; prompts interactively, and in non-interactive sessions downgrades to skip-saving with a warning), "force" (overwrite), or "no-clobber" (skip saving). Passed to save_timetree().

`width` Output width in inches. Default: `14`.

`height` Output height in inches. Default: `10`.

`taxonomy_levels` Optional list for custom taxonomy rank codes and names, e.g. `list(codes = c("k", "ss"), names = c("kingdom", "subspecies"))`. `codes` are the abbreviation letters used in labels (e.g., `"k"`, `"ss"`); `names` are human-readable rank names used as column names in the result data frame. Used to extend or override default rank handling. Default: `NULL`.

`low_memory` Logical. If `TRUE`, enable best-effort low-memory mode by triggering garbage collection between major pipeline steps. Default: `FALSE`.

`ignore_malformed` Logical. If `TRUE`, malformed inputs are skipped with a warning instead of terminating. In batch mode, failed trees are returned as `NULL`. Default: `FALSE`.

`ignore_branch_length` Logical. If `TRUE`, the tree is drawn as a cladogram with all tips aligned at the same x position, ignoring the original branch lengths (equivalent to ggtree's `branch.length = "none"`). This automatically disables the geological timescale. Default: `FALSE`.

`color_rank` Character. Taxonomic rank for coloring, independent of `rank` (which controls collapsing). When set (e.g., `"phylum"`), branches and tips are colored by this rank, while collapsing uses `rank`. `NULL` (default) uses `rank` for both coloring and collapsing.

`timescale_mode` Character. Timescale display mode for circular layout: `"radial"` (default) draws full-circle geological background bands; `"linear"` draws a rectangular geological timescale axis (via `deeptime::guide_geo()`) at the position specified by `timescale_position`, without background bands. Ignored for rectangular layout.

`timescale_position` Character. Clock position of the timescale axis when `timescale_mode = "linear"`. One of `"right"` (3 o'clock, default), `"top"` (12 o'clock), `"bottom"` (6 o'clock), `"left"` (9 o'clock). Should match `tree_start_position`. Ignored when `timescale_mode = "radial"` or for rectangular layout.

`tree_start_position` Character. Clock position where the tree starts expanding (the gap between the last and first tip). One of `"right"` (3 o'clock, default), `"top"` (12 o'clock), `"bottom"` (6 o'clock), `"left"` (9 o'clock). Should match `timescale_position` when `timescale_mode = "linear"`; a warning is issued if they differ. Ignored for rectangular layout.

`opts` A `rclade_options` object (see `rclade_options`) or a plain named list of parameter defaults. When supplied, values in `opts` serve as defaults for parameters the caller did not explicitly set; explicit arguments always take precedence. This is the recommended way to reuse a consistent parameter set across multiple calls.

Details

This function orchestrates the complete Rclade pipeline:

1. Input validation and reading (format detection, unit conversion)
2. Taxonomy parsing (GTDB/Silva/NCBI/custom, clade/groups/rank modes)
3. MRCA computation and monophyly check (with nesting conflict detection)
4. Color generation (color-blind-safe palette)
5. Tree rendering and batch clade collapsing (ggtree + depth-first collapse)
6. Timescale integration and annotations (deeptime, support, HPD, labels, highlight)
7. Plot finalization (legend, theme, title, save, metadata)

Value

A `ggplot` object with an `rclade_info` attribute containing metadata (tip count, group count, taxonomy format, etc.). Can be further customized with + layers.

Label length guard

When `tree` is a file path, Newick labels longer than 500 characters are truncated to 400 characters plus a `_RCLADE_TRUNC` suffix (with a warning) before parsing, because ape's Newick parser aborts the whole R process on labels longer than ~512 characters on Linux. Truncated labels may no longer match external taxonomy files or sequence IDs; shorten labels upstream if exact matching is required. See `read_tree_auto()`.

Parameter grouping

For complex configurations, consider organizing parameters by category:

- **Tree input:** `tree`
- **Collapsing:** `rank`, `triangle_mode`, `space_mode`
- **Layout:** `layout`, `angle`, `line_width`
- **Colors:** `color_palette`, `color_mapping`
- **Labels:** `show_tip_labels`, `tip_label_size`, `show_clade_label`, `clade_label_offset`
- **Timescale:** `add_timescale`, `timescale_levels`, `unit`
- **Taxonomy:** `taxonomy_format`, `custom_patterns`
- **Annotations:** `show_support`, `support_threshold`, `show_hpd`
- **Legend:** `legend_position`, `legend_nrow`, `legend_ncol`
- **Output:** `output`, `width`, `height`

References

Yu G, Smith DK, Zhu H, Guan Y, Lam TT-Y (2017). "ggtree: an R package for visualization and annotation of phylogenetic trees with their covariates and other associated data." *Methods in Ecology and Evolution*, 8(1), 28-36. doi:10.1111/2041-210X.12628

Gearty W (2025). "deeptime: an R package that facilitates highly customizable and reproducible visualizations of data over geological time intervals." *Big Earth Data*. doi:10.1080/20964471.2025.2537516

Paradis E, Schliep K (2019). "ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R." *Bioinformatics*, 35(3), 526-528. doi:10.1093/bioinformatics/bty633

Examples

```
# Quick start with the bundled example tree (no timescale)
data(example_tree)
p <- plot_timetree(example_tree, rank = "phylum",
                   taxonomy_format = "GTDB", add_timescale = FALSE)
```

`polytomy_tree` *Example phylogenetic tree with polytomies*

Description

A `phylo` object with 9 tips containing intentional polytomies (multifurcations) for testing polytomy handling. This tree has a structure of ((A,B,C),(D,E),(F,G,H,I)) with three multifurcating clades.

Usage

```
data(polytomy_tree)
```

Format

A `phylo` object with 9 tips.

Source

Simulated data for testing polytomy handling.

`rclade_logo` *Display Rclade ASCII art logo*

Description

Prints a stylized ASCII art logo for Rclade to the console.

Usage

```
rclade_logo(show_version = TRUE, show_tagline = TRUE)
```

Arguments

`show_version`	Logical. Whether to show version number. Default: TRUE
`show_tagline`	Logical. Whether to show tagline. Default: TRUE

Value

Invisible NULL

Examples

```
rclade_logo()
```

rclade_options *Construct a validated options list for* plot_timetree()

Description

rclade_options() returns a named list of commonly used rendering parameters that can be passed to plot_timetree(opts = ...) as a single object. Explicit arguments supplied directly to plot_timetree() always take precedence over values in opts.

Usage

```
rclade_options(
  rank = "none",
  layout = "rectangular",
  color_palette = "viridis",
  taxonomy_format = "auto",
  add_timescale = TRUE,
  timescale_mode = "radial",
  unit = NULL,
  legend_position = "bottom",
  line_width = 1,
  show_tip_labels = FALSE,
  width = 14,
  height = 10,
  ...
)
```

Arguments

rank	Character. Collapsing rank (e.g. "phylum", "class", "none").
layout	Character. "rectangular" or "circular".
color_palette	Character. Palette name (e.g. "viridis", "plasma", "Set1").
taxonomy_format	Character. One of "auto", "GTDB", "Silva", "NCBI", "embedded", "custom_regex".
add_timescale	Logical. Whether to add a geological timescale.
timescale_mode	Character. "radial", "linear", "none".
unit	Character or NULL. "Ma", "Ga", or NULL. Required when add_timescale = TRUE (the pipeline aborts for NULL); NULL leaves native units untouched and is only valid with add_timescale = FALSE.
legend_position	Character. Legend placement (e.g. "bottom", "right", "none").
line_width	Numeric. Branch line width.
show_tip_labels	Logical. Whether to display tip labels.

width	Numeric. Output width in inches.
height	Numeric. Output height in inches.
...	Additional named parameters to include in the options list. These are passed through without validation.

Details

This constructor is the first step toward parameter-surface convergence: it provides a single validated source of truth for parameter defaults, reducing the risk of drift between the `plot_timetree()` signature, internal forwarding lists, and documentation.

Value

A named list of class `"rclade_options"` suitable for the `opts` argument of `plot_timetree()`.

Examples

```
# Create a reusable options object
opts <- rclade_options(rank = "phylum", layout = "circular",
                       color_palette = "plasma", add_timescale = TRUE)
```

---

read_taxonomy_file *Read taxonomy information from a table file*

Description

Reads taxonomy information from a two-column table file where:

- Column 1: Tip labels (must match tree tip labels)
- Column 2: Taxonomy strings in GTDB format (e.g., `d__Archaea;p__Thermoproteota;c__Korarchaeia`)

Usage

```
read_taxonomy_file(file, sep = "auto", header = FALSE, table_sep = ";")
```

Arguments

file	Character. Path to the taxonomy table file.
sep	Character. Column separator. `"auto"` auto-detects tab or comma. Default: `"auto"`.
header	Logical. Whether the file has a header row. Default: `FALSE`.
table_sep	Character. Separator between taxonomy ranks in the second column (the taxonomy string). Default: `";"`.

Details

Supports tab-delimited or comma-delimited files. Missing ranks are indicated by empty values after the rank prefix (e.g., s__ for missing species).

Value

A data.frame with columns:

label Character. Tip labels from column 1.

domain Character. Domain (d__).

phylum Character. Phylum (p__).

class Character. Class (c__).

order Character. Order (o__).

family Character. Family (f__).

genus Character. Genus (g__).

species Character. Species (s__).

Missing ranks are NA.

---

read_tree_auto *Read tree from file with automatic format detection*

Description

Intended for use as a stable library API by external workflows (e.g., Snakemake/Nextflow).

Usage

```
read_tree_auto(filepath, tree_index = NULL, multi_tree_mode = "error")
```

Arguments

filepath	Path to tree file (.tre, .nwk, .newick, .nexus, .nex, .treefile, .xml)
tree_index	Integer. Index of tree to use from multiPhylo objects (e.g., BEAST posterior). Default: NULL (will use multi_tree_mode to determine behavior).
multi_tree_mode	Character. How to handle multiple trees in a file. Options: • "error" (default): Stop with error and ask user to specify • "ask": Interactively prompt the user to choose a tree or handling mode. Falls back to "error" in non-interactive sessions. • "first": Use the first tree • "last": Use the last tree • "random": Use a randomly selected tree • "all": Return all trees (as multiPhylo) • "split": Return all trees (as multiPhylo); callers write per-tree outputs with numeric suffixes (e.g. output_1.pdf)

Details

Newick labels longer than 500 characters are automatically truncated to 400 characters plus a `_RCLADE_TRUNC` suffix (with a warning), because ape's Newick parser aborts the whole R process on labels longer than ~512 characters on Linux. Truncated labels may no longer match external taxonomy files or sequence IDs; shorten labels upstream if exact matching is required.

Value

phylo object (or multiPhylo if multi_tree_mode = "all" or "split")

`run_rclade_cli` *Run Rclade from the command line*

Description

Provides a command-line interface for Rclade. Requires the optparse package.

Usage

```
run_rclade_cli(args = commandArgs(trailingOnly = TRUE))
```

Arguments

`args`	Character vector of command-line arguments (default: commandArgs(trailingOnly = TRUE))

Value

An invisible integer exit code following standard Unix conventions: `0L` (success), `1L` (runtime error), `2L` (parameter error), `3L` (input-data error), `130L` (user interrupt / SIGINT). **The caller MUST pass this value to `q(status = ...)` or `quit(status = ...)` for the exit code to propagate to the operating system**; simply calling `run_rclade_cli()` without forwarding the return value will always exit with code 0 regardless of errors.

Config-file override trap (L-E3 — READ BEFORE USING `-config`)

A `--config` YAML file supplies **defaults** for any option, with precedence `CLI explicit argument > config file > built-in default`. Because `optparse` does not expose "was this flag passed?", the override is applied by comparing each option against its **built-in default**: **any option the user left at its default value is eligible to be overridden by the config file** — *even if the user explicitly typed a value identical to the default*. Concretely:

- `plot_timetree --rank phylum` (where "phylum" is the default) **will be overridden** by `rank: class` in the config.
- `plot_timetree --rank species` (non-default) is preserved.

This is an `optparse` limitation, not a bug, and is **intentional but surprising**. To make an option immune to config, pass a non-default value, or avoid relying on config for options you care about. Unknown config keys are warned and ignored.

run_rclade_selftest *Run Rclade self-test*

Description

Performs comprehensive self-check:

1. Required package availability and versions
2. Example tree parsing and taxonomy extraction
3. Monophyly logic validation

Usage

```
run_rclade_selftest(verbose = TRUE)
```

Arguments

verbose	Logical. If TRUE (default), progress and results are reported via message (suppressible with suppressMessages). If FALSE, the self-test runs silently and only the exit code is returned.

Value

Integer. Exit code (0 = all passed, 1 = failures).

run_rclade_shiny *Launch Rclade Shiny app*

Description

Provides an interactive web interface for Rclade. Requires the shiny package.

Usage

```
run_rclade_shiny()
```

Value

No return value, called for side effects (launches a Shiny web app).

Concurrency

The logger, step-progress, and interrupt subsystems use package-level global environments (.logger_env, .interrupt_env). Within a single R process, multiple concurrent Shiny sessions will share (and may corrupt) this state — log messages, step counters, and interrupt flags can cross between sessions. For production multi-user deployment, run one Shiny instance per R process (e.g., behind a load balancer), or assign a dedicated log file per task via --log_file. A session-scoped state refactor is on the roadmap.

save_session_info *Save sessionInfo() for reproducibility*

Description

Save sessionInfo() for reproducibility

Usage

```
save_session_info(file = "session_info.txt")
```

Arguments

file	Output file path (default "session_info.txt")

Value

Invisibly returns sessionInfo

save_timetree *Save a timetree plot to file*

Description

Save a timetree plot to file

Usage

```
save_timetree(p, file, width = 14, height = 10, dpi = 300, overwrite = "ask")
```

Arguments

p	ggplot object
file	Output file path
width	Width in inches
height	Height in inches
dpi	Resolution (for PNG/TIFF only)
overwrite	Character. Overwrite mode: "ask" (default), "force", "no-clobber".

Value

Invisibly returns the ggplot object

set_log_enabled *Enable or disable logging*

Description

Enable or disable logging

Usage

```
set_log_enabled(enabled)
```

Arguments

enabled	Logical.

Value

No return value, called for side effects (toggles the package-level logging enabled flag).

set_log_file *Set log file for dual output*

Description

Set log file for dual output

Usage

```
set_log_file(filepath)
```

Arguments

filepath	Character. Path to log file. NULL to disable.

Value

No return value, called for side effects (opens or closes a log file connection for dual console + file logging).

set_log_level *Set log level*

Description

Set log level

Usage

```
set_log_level(level)
```

Arguments

level	Character. One of "DEBUG", "INFO", "WARNING", "ERROR", "CRITICAL".

Value

No return value, called for side effects (sets the package-level log threshold).

summarize_taxonomy_quality *Report taxonomy label parsing quality*

Description

Provides a detailed report on how well taxonomic labels can be parsed, including per-rank parse rates and failed labels.

Usage

```
summarize_taxonomy_quality(
  labels,
  format = "auto",
  custom_patterns = NULL,
  taxonomy_levels = NULL,
  delimiter_mode = "reverse"
)
```

Arguments

labels	Character vector of tip labels
format	Format: "auto", "embedded", "GTDB", "Silva", "NCBI", "custom_rank", "custom_regex"
custom_patterns	Custom regex patterns (for "custom_regex" format)

taxonomy_levels
: Custom taxonomy level configuration

delimiter_mode
: Character. Embedded parsing strategy: "reverse", "greedy", "segment".

Value

Invisibly returns a list with parsing statistics

---

summarize_taxonomy_quality_with_file

Summarize taxonomy quality with external file support

Description

Extended version of `summarize_taxonomy_quality()` that can use an external taxonomy file.

Usage

```
summarize_taxonomy_quality_with_file(
  labels,
  format = "auto",
  custom_patterns = NULL,
  taxonomy_file = NULL,
  file_sep = "auto",
  file_header = FALSE,
  file_priority = TRUE,
  table_sep = ";",
  delimiter_mode = "reverse",
  taxonomy_levels = NULL
)
```

Arguments

labels
: Character vector of tip labels.

format
: Format: "auto", "GTDB", "Silva", "NCBI", "custom_rank", "custom_regex".

custom_patterns
: Custom regex patterns (for "custom_regex" format).

taxonomy_file
: Character. Path to external taxonomy file. Default: `NULL`.

file_sep
: Character. Column separator for taxonomy file. Default: `"auto"`.

file_header
: Logical. Whether taxonomy file has header row. Default: `FALSE`.

file_priority
: Logical. If `TRUE`, file taxonomy takes priority. Default: `TRUE`.

table_sep
: Character. Separator between taxonomy ranks in the second column of the taxonomy file. Default: `";"`.

delimiter_mode
: Character. Embedded parsing strategy: "reverse", "greedy", "segment".

taxonomy_levels
: Custom taxonomy level configuration (list with codes and names). Default: `NULL`.

Value

Invisibly returns a list with parsing statistics.

---

summarize_timetree *Print a summary of a Rclade timetree plot*

Description

Print a summary of a Rclade timetree plot

Usage

```
summarize_timetree(p)
```

Arguments

p	ggplot object returned by plot_timetree() (with rclade_info attribute)

Value

Invisibly returns the info list

---

theme_timetree *Publication-ready theme for timetree plots*

Description

Based on ggtree::theme_tree2(), customized for publication quality. Includes `coord_cartesian(clip = "off")` to prevent collapsed clade triangles from being clipped at the plot panel boundary — triangle vertices (especially the MRCA node apex) often extend beyond the tip-based y-axis range.

Usage

```
theme_timetree(base_size = 12)
```

Arguments

base_size	Base font size

Value

A `list` with components `theme` (ggplot2 theme) and `coord` (`coord_cartesian(clip = "off")`). Callers should apply both via `p + result$theme + result$coord`.

validate_sequence_deep *Deep validation of sequence files*

Description

Validates FASTA/FASTQ files: format detection, duplicate IDs, alphabet detection, and alignment length consistency.

Usage

```
validate_sequence_deep(
  filepath,
  expected_alphabet = NULL,
  check_alignment = FALSE
)
```

Arguments

filepath	Character. Path to sequence file.
expected_alphabet	Character or NULL. Expected alphabet: "DNA", "RNA", "protein", or NULL for auto-detect.
check_alignment	Logical. If TRUE, check all sequences have equal length.

Value

List with validation results.

validate_sequence_file *Validate sequence file format*

Description

Validate sequence file format

Usage

```
validate_sequence_file(filepath)
```

Arguments

filepath	Character. Path to sequence file.

Value

Character. Detected format: "fasta", "fastq", "unknown".

`validate_tree_sequence_match` *Cross-validate tree tip labels against sequence IDs*

Description

Checks that all tree tips have corresponding sequences and vice versa.

Usage

```
validate_tree_sequence_match(
  tree,
  sequence_file,
  quiet = FALSE,
  mol_type = NULL,
  skip_length_check = FALSE,
  multi_tree_mode = "error"
)
```

Arguments

`tree`	phylo object or path to tree file.
`sequence_file`	Character. Path to sequence file (FASTA/FASTQ).
`quiet`	Logical. If TRUE, suppress messages.
`mol_type`	Character. Molecule type for sequence validation: "DNA", "RNA", "protein", or "auto" (NULL). Default: `NULL`.
`skip_length_check`	Logical. If TRUE, skip alignment length consistency check. Default: `FALSE`.
`multi_tree_mode`	Character. How to handle multiple trees in a file when `tree` is a path. Default: `"error"`.

Value

List with match status and differences.

Index

∗ **datasets**
    example_tree, 7
    polytomy_tree, 16

batch_plot, 3

check_monophyly, 4
check_special_monophyly, 5

detect_taxonomy_format, 6

example_tree, 7

get_supported_extensions, 8

message, *21*

parse_taxonomy, *5*, 8
plot_timetree, 9
polytomy_tree, 16

rclade_logo, 16
rclade_options, *14*, 17
read_taxonomy_file, 18
read_tree_auto, 19
read_tree_auto(), *15*
run_rclade_cli, 20
run_rclade_selftest, 21
run_rclade_shiny, 21

save_session_info, 22
save_timetree, 22
set_log_enabled, 23
set_log_file, 23
set_log_level, 24
summarize_taxonomy_quality, 24
summarize_taxonomy_quality_with_file, 25
summarize_timetree, 26
suppressMessages, *21*

theme_timetree, 26

validate_sequence_deep, 27
validate_sequence_file, 27
validate_tree_sequence_match, 28