Package: karioCaS (via r-universe)

June 22, 2026

Title Kraken Confidence Scores for Reliable Domain-Specific Microbiota Inference and Discovery

Version 0.99.15

Description Provides a comprehensive framework for analyzing Kraken2 metagenomic classification using multiple confidence scores. karioCaS implements a robust comparative approach to evaluate taxonomic stability across different stringency levels, facilitating the identification of reliable microbiota components and the discovery of potentially novel or unrepresented species. A key innovation is its domain-specific analysis, separately processing Bacteria, Archaea, Eukaryota, and Viruses to ensure balanced visibility of all biological domains. Includes high-quality visualization tools for retention rates, shared taxa (Upset plots), and stability heatmaps.

License GPL (>= 3)

Depends R (>= 4.5.0)

Encoding UTF-8

RoxygenNote 7.3.3

Imports ggplot2, patchwork, UpSetR, dplyr, readr, stringr, tidyr, scales, ggtext, SummarizedExperiment, TreeSummarizedExperiment, S4Vectors, forcats, tibble, methods, rlang

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

biocViews Metagenomics, Microbiome, Software

URL https://github.com/thiagoparentefiocruz/karioCaS

BugReports https://github.com/thiagoparentefiocruz/karioCaS/issues

Config/pak/sysreqs cmake make libicu-dev libjpeg-dev libpng-dev libuv1-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

Repository https://biocstaging.r-universe.dev

Date/Publication 2026-06-21 23:45:02 UTC

RemoteUrl https://github.com/BiocStaging/karioCaS

RemoteRef HEAD

RemoteSha db28778c720e87d0b66dba398e6e6b1d47e2c2e6

Contents

`.kariocas_internal_colors`
karioCaS Visualization Styles & Helpers

Description

Single Source of Truth for visual elements in karioCaS. Follows Nature/Science publication standards.

Usage

```
.kariocas_internal_colors
```

Format

An object of class `list` of length 5.

group_upset *Cross-Sample UpSet: Core vs Unique Taxa per Biological Group (Step 008)*

Description

For each biological group (inferred from sample names by stripping trailing digits, e.g. SAMPLE33, SAMPLE34 -> SAMPLE), draws an UpSet plot comparing which taxa (at tax_level) are present across the samples of the group, per Domain. This separates the **core** taxa (present in every sample) from **unique**/rare taxa (present in one or a few samples) - the expected pattern for pathogens and false positives. A membership TSV (presence matrix plus N_Samples and a Core/Shared/Unique Category) is written alongside each plot.

Usage

```
group_upset(project_dir, tax_level = "Species", CS = NULL)
```

Arguments

project_dir	Path to the project root.
tax_level	Taxonomic rank to compare (default: "Species").
CS	Confidence Score to analyse. NULL (default) uses the final mosaic; a numeric value (Kraken fraction 0-1 or percentage 0-100) compares the imported data at that single CS.

Details

By default the analysis uses the **final mosaic** from retrieve_selected_taxa(); set CS to compare at a single Confidence Score from the imported data instead.

Value

Invisibly returns a data.frame with the full membership table. UpSet PDFs and membership TSVs are saved per group to <project_dir>/008_taxa_intersections_across_samples/.

Examples

```
toy_project <- system.file("extdata", "your_project_name", package = "karioCaS")

# Core vs unique species across samples, from the final mosaic
# group_upset(project_dir = toy_project)

# Compare at a single Confidence Score instead
# group_upset(project_dir = toy_project, CS = 40)
```

`heatmaps_karioCaS` *Generate Heatmaps of Taxa Abundance with Extinction Patterns (Step 006)*

Description

Creates Relative Abundance (Confidence Score. "Elite" survivors are clustered by similarity, while lost taxa are aggregated into "Loss Groups" at the bottom. Handles domains with zero survivors at the target CS (showing only loss groups).

Usage

```
heatmaps_karioCaS(
  project_dir,
  analysis_rank = NULL,
  confidence_score = NULL,
  top_n = 20
)
```

Arguments

`project_dir`	Path to the project root.
`analysis_rank`	Taxonomic rank to analyze. Defaults to `"Genus"`.
`confidence_score`	Target CS to define survivors (e.g., 90). Defaults to the highest available CS.
`top_n`	Number of top survivors to display individually (default: 20).

Value

Invisibly returns `NULL`. PDF plots are saved to `<project_dir>/006_relative_abundance_across_CS/`.

Examples

```
toy_project <- system.file("extdata", "your_project_name", package = "karioCaS")

# Basic usage (defaults to Genus, highest CS, top 20 taxa)
# heatmaps_karioCaS(project_dir = toy_project)

# Advanced: Species level at CS 50, top 30 taxa
# heatmaps_karioCaS(
#   project_dir      = toy_project,
#   analysis_rank    = "Species",
#   confidence_score = 50,
#   top_n            = 30
# )
```

import_karioCaS — *Import Kraken MPA Reports to TreeSummarizedExperiment (Step 001)*

Description

Reads Kraken2 MPA-style reports from the `000_mpa_original` folder. Parses filenames like `SAMPLE_CSXX.mpa` (e.g., `PILO_CS09.mpa`). Generates a detailed log file for traceability and saves a `TreeSummarizedExperiment` object for downstream analysis.

Usage

```
import_karioCaS(project_dir)
```

Arguments

`project_dir`	Path to the project root. Must contain a `000_mpa_original` subfolder with `.mpa` files.

Value

Invisibly returns a `TreeSummarizedExperiment` object.

Examples

```
toy_project <- system.file("extdata", "your_project_name", package = "karioCaS")
import_karioCaS(project_dir = toy_project)
```

reads_per_taxa — *Read Cutoff Saturation Analysis & Optimal Minimum Reads (Step 003)*

Description

Saturation analysis: progressively raises a per-taxon read-count cutoff and tracks how many taxa survive, on a log read axis. By default it draws one **group overlay** per biological group (every sample a faint line, group mean highlighted) and marks each domain's **median optimal minimum reads** - the elbow of the saturation curve, found with the same engine used for the optimal CS - as a dashed line. The per-sample optimal-reads values are written to `Reads_Audit_<rank>.tsv/.rds`, giving a quantitative threshold for excluding low-abundance background/false-positive taxa.

Usage

```
reads_per_taxa(
  project_dir,
  analysis_level = "Species",
  method = c("kneedle", "postcliff", "segmented"),
  detail_samples = NULL
)
```

Arguments

`project_dir`	Path to the project root.
`analysis_level`	Taxonomic rank to analyze (default: `"Species"`).
`method`	Elbow strategy for the optimal reads. One of `"kneedle"` (default), `"postcliff"` or `"segmented"`.
`detail_samples`	Which samples to also render as detailed per-sample saturation panels. `NULL` (default) writes only the group overlays; `"all"` renders every sample; a comma-separated string such as `"SAMPLE33, SAMPLE45"` (or a character vector) renders just those. Detailed PDFs are saved to a `per_sample/` subfolder.

Details

The optimal reads is computed *per Confidence Score*, since the saturation curve changes with CS; read it off at your chosen optimal CS (Step 002).

Value

Invisibly returns a `data.frame` with the optimal-reads audit. PDF plots and `Reads_Audit_<rank>` files are saved to `<project_dir>/003_reads_saturation/`.

Examples

```
toy_project <- system.file("extdata", "your_project_name", package = "karioCaS")

# Group saturation overlays + optimal minimum reads (default Kneedle)
# reads_per_taxa(project_dir = toy_project)
```

`retrieve_selected_taxa` *Retrieve Selected Taxa with Domain-Specific Thresholds (Step 004)*

Description

Creates a "biological mosaic" for each sample using the `karioCaS_TSE.rds` object from Step 001 and, optionally, the optimal thresholds computed earlier: the optimal Confidence Score (Stability Index audit from `taxa_retention()`, Step 002) and the optimal minimum reads (`Reads_Audit` from `reads_per_taxa()`, Step 003). Both the `CS_*` and `reads_min_*` arguments accept `"auto"`, `"secondary"`, or a manual numeric value per domain. The optimal reads is looked up at each domain's resolved CS, so the mosaic combines both data-driven thresholds automatically. The mosaic always retains **all** taxonomic ranks present in the input (as an MPA profile does); enforces a strict `> 0` read filter.

Usage

```
retrieve_selected_taxa(
  project_dir,
  tax_level = NULL,
  CS_A = "auto",
  reads_min_A = 0,
  CS_B = "auto",
  reads_min_B = 0,
  CS_E = "auto",
  reads_min_E = 0,
  CS_V = "auto",
  reads_min_V = 0
)
```

Arguments

`project_dir`	Path to the project root.
`tax_level`	Which rank's optimization audit the `"auto"` / `"secondary"` thresholds are read from, i.e. `SI_Audit_<tax_level>` and `Reads_Audit_<tax_level>` (`NULL`, the default, uses `"Species"`). This does *not* filter the output, which always contains all ranks.
`CS_A`	Character or numeric. CS for Archaea: `"auto"`, `"secondary"`, or a numeric value. Default: `"auto"`.
`reads_min_A`	Minimum reads for Archaea: `"auto"`/`"secondary"` (pulled from the Reads_Audit at the resolved CS) or a numeric value. Default: 0.
`CS_B`	Character or numeric. CS for Bacteria. Default: `"auto"`.
`reads_min_B`	Minimum reads for Bacteria. Default: 0.
`CS_E`	Character or numeric. CS for Eukaryota. Default: `"auto"`.
`reads_min_E`	Minimum reads for Eukaryota. Default: 0.
`CS_V`	Character or numeric. CS for Viruses. Default: `"auto"`.
`reads_min_V`	Minimum reads for Viruses. Default: 0.

Value

Invisibly returns `TRUE`. Mosaic files are saved to `<project_dir>/004_final_mosaic/`, with `.mpa` files under `mpa/` and `.tsv` files under `tsv/`.

Examples

```
toy_project <- system.file("extdata", "your_project_name", package = "karioCaS")

# Fully data-driven mosaic: optimal CS and optimal min-reads per domain
# retrieve_selected_taxa(
#   project_dir = toy_project,
#   tax_level   = "Species",
#   CS_B        = "auto", reads_min_B = "auto",
#   CS_A        = "auto", reads_min_A = "auto",
#   CS_E        = 40,     reads_min_E = 10,
#   CS_V        = 0,      reads_min_V = 0
# )
```

---

taxa_resolution *Generate Taxa Resolution Analysis (Step 007)*

Description

Creates stacked bar plots showing resolution efficiency between a Parent Rank and a Child Rank (how much of each parent clade's reads are resolved down to the child rank versus remaining parent-exclusive). Uses `max()` for parent aggregation to prevent double-counting of children in cumulative data.

Usage

```
taxa_resolution(
  project_dir,
  parent_level = "Genus",
  child_level = "Species",
  CS = NULL,
  top_n = 10
)
```

Arguments

`project_dir`	Path to the project root.
`parent_level`	Name of the parent rank (default: `"Genus"`).
`child_level`	Name of the child rank (default: `"Species"`).
`CS`	Confidence Score to analyse. `NULL` (default) uses the final mosaic from `retrieve_selected_taxa()`. A numeric value (Kraken fraction `0-1` or percentage `0-100`) analyses the imported data at that single CS instead of every CS.
`top_n`	Number of top taxa to display per domain (default: 10).

Details

By default the analysis runs on the **final mosaic** produced by `retrieve_selected_taxa` (`004_final_mosaic/`), i.e. the data-driven high-confidence selection - one figure per sample. Alternatively, set `CS` to analyse the raw imported data at a single Confidence Score.

Value

Invisibly returns NULL. PDF plots are saved to <project_dir>/007_taxa_resolution/.

Examples

```
toy_project <- system.file("extdata", "your_project_name", package = "karioCaS")

# Default: resolution of the final mosaic (Genus vs Species, top 10)
# taxa_resolution(project_dir = toy_project)

# Analyse the imported data at a single Confidence Score
# taxa_resolution(project_dir = toy_project, CS = 40)
```

---

taxa_retention *Run Confidence Score Retention & Optimization (Step 002)*

Description

Executes taxa retention analysis based on Confidence Score (Kraken/Bracken) and, in the same step, computes the objective optimal CS (Stability Index, SI) for each domain. By default it produces a single, low-clutter **group overlay** per biological group: every sample of the group is drawn as a faint line with the group mean (±SD) highlighted and each domain's median optimal CS marked with a dashed line, faceted by Domain. It also writes the machine-readable SI audit (SI_Audit_<rank>.tsv/.rds) used by retrieve_selected_taxa. Detailed per-sample panels (all ranks, taxa vs reads) are written only on request.

Usage

```
taxa_retention(
  project_dir,
  tax_level = "Species",
  method = c("kneedle", "postcliff", "segmented", "dynamic", "manual"),
  manual_toll = 1,
  detail_samples = NULL
)
```

Arguments

project_dir	Root path of the project.
tax_level	Taxonomic rank used for the group overlay and SI (default: "Species").
method	Optimal-CS strategy. One of "kneedle" (default), "postcliff", "segmented", "dynamic", or "manual".
manual_toll	Numeric or named list. Acceptable step-wise loss percentage, used only when method = "manual" (default: 1.0).
detail_samples	Which samples to also render as detailed per-sample panels. NULL (default) writes only the group overlay; "all" renders every sample; a comma-separated string such as "SAMPLE33, SAMPLE45" (or a character vector) renders just those. Detailed PDFs are saved to a per_sample/ subfolder.

Details

Groups are inferred from sample names by stripping trailing digits (e.g. SAMPLE33, SAMPLE34 both belong to group SAMPLE).

The optimal CS is found with a multi-strategy engine: "kneedle" (default, parameter-free elbow detection), "postcliff" (a more conservative threshold past the steepest drop), "segmented" (broken-stick regression), "dynamic", or "manual".

Value

Invisibly returns a data.frame with the full SI audit trail. PDF plots and SI_Audit_<rank> files are saved to <project_dir>/002_taxa_retention/.

Examples

```
toy_project <- system.file("extdata", "your_project_name", package = "karioCaS")

# Group retention overlay + optimal CS (default Kneedle method)
# taxa_retention(project_dir = toy_project)
```

---

upset_kariocas *Generate UpSet Plots per Sample and Domain (Step 005)*

Description

"karioCaS never are upset!" Generates UpSet plots showing taxon persistence across Confidence Score levels, for a single taxonomic rank, with detailed logging. One plot per sample and domain is written to a per-sample subfolder.

Usage

```
upset_kariocas(project_dir, tax_level = "Species")
```

Arguments

project_dir	Path to the project root.
tax_level	Taxonomic rank to analyze (default: "Species").

Value

Invisibly returns NULL. PDF plots are saved to <project_dir>/005_taxa_intersections_across_CS/.

Examples

```
toy_project <- system.file("extdata", "your_project_name", package = "karioCaS")

# upset_kariocas(project_dir = toy_project)
# upset_kariocas(project_dir = toy_project, tax_level = "Genus")
```

Index