Package: MiCARA (via r-universe)

November 17, 2026

Title Microbiome Confounder-Adjusted Robust Analysis

Version 0.99.3

Description Provides a comprehensive framework for multi-cohort microbiome differential abundance and residual correlation analysis. It integrates confounder-adjusted linear models with centered log-ratio transformed taxonomic and functional pathway profiles. The package features robust preprocessing, automated covariate handling, multiple-testing correction, and quality control diagnostics for multi-omics data. Interactive visualization tools are included to facilitate downstream interpretation of cross-domain interaction networks.

License MIT + file LICENSE

URL https://github.com/BushnaqSafa/MiCARA

BugReports https://github.com/BushnaqSafa/MiCARA/issues

Depends R (>= 4.5.0)

Imports ANCOMBC, Hmisc, htmlwidgets, mice, networkD3, stats, utils

Suggests BiocStyle, knitr, MultiAssayExperiment, rmarkdown, SummarizedExperiment, testthat (>= 3.0.0), webshot2

VignetteBuilder knitr

biocViews Microbiome, DifferentialExpression, Software, Metagenomics, NetworkInference, Pathways, Network, Preprocessing, QualityControl

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/pak/sysreqs cmake libglpk-dev make libgsl0-dev libicu-dev libuv1-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-11-08 18:45:55 UTC

RemoteUrl https://github.com/bioc/MiCARA

RemoteRef HEAD

RemoteSha cd5e8dc1d0242c3261449de5cadc9a947e3492a3

Contents

compute_global_fdr 2
compute_residualCLR_correlations 4
example_metadata 7
example_pathways 8
example_taxa 8
impute_metadata 9
micara-print 11
plot_interaction_sankey 12
run_diff_abundance 14
validate_inputs 17

Index **20**

`compute_global_fdr` *Apply global multiple-testing correction to taxon-pathway associations*

Description

Applies a single multiple-testing correction across all taxon-pathway associations tested across all analysed disease groups in a `micara_interactions` object.

Usage

```
compute_global_fdr(
  interactions,
  method = "BH",
  q_cutoff = 0.05,
  r_cutoff = 0.2,
  verbose = TRUE
)
```

Arguments

`interactions`	An object of class `micara_interactions`, typically returned by `compute_residualCLR_correlations`.
`method`	Multiple-testing correction method passed to `p.adjust`. Default is `"BH"`.
`q_cutoff`	Global adjusted p-value threshold for retaining an association. Default is `0.05`.
`r_cutoff`	Minimum absolute Spearman correlation required for retaining an association. Default is `0.2`. The package convention is strictly `|rho| > r_cutoff`.
`verbose`	Logical; if `TRUE`, prints a summary of the global correction and the number of retained associations per disease.

Details

The function operates on the `$all_links` component produced by `compute_residualCLR_correlations`. The existing disease-specific `$links` results are not modified. Instead, globally supported associations are stored in a new `$global_links` component for each disease, and the globally adjusted p-values are added to `$all_links` as `q_global`.

This correction is optional and is most appropriate when inference is intended across multiple disease groups, disease stages, cancer types, cohorts, or other collections of simultaneous taxon-pathway tests.

The global correction is performed once across the complete set of available `$all_links` tables from all analysed diseases. It is therefore different from the disease-specific BH correction already performed by `compute_residualCLR_correlations`.

Global correction is performed only on the p-values from tested taxon-pathway pairs. The correlation threshold is applied after correction and does not enter the multiplicity adjustment.

Value

An object of class `micara_interactions`. For each disease, the returned object contains:

- `all_links`: all tested taxon-pathway pairs with an added `q_global` column;
- `links`: the original disease-specific BH-significant links, unchanged;
- `global_links`: associations passing the global adjusted p-value and correlation thresholds;
- `n_global_links`: number of globally significant associations.

The returned object also contains a `"global_fdr"` attribute with correction settings, the total number of tested pairs, and the total number of globally significant associations.

Examples

```
# Construct minimal mock interactions input
mock_interactions <- list(
    IBD = list(
        status = "analysed",
        all_links = data.frame(
            taxon = c("TaxonA", "TaxonB", "TaxonC"),
            pathway = c("Path1", "Path2", "Path1"),
            p = c(0.001, 0.030, 0.450),
            rho = c(0.45, -0.28, 0.05),
            stringsAsFactors = FALSE
        ),
        links = data.frame(
            taxon = c("TaxonA", "TaxonB"),
            pathway = c("Path1", "Path2"),
            p = c(0.001, 0.030),
            rho = c(0.45, -0.28),
            stringsAsFactors = FALSE
        )
    )
)
class(mock_interactions) <- c("micara_interactions", "list")
```

```
# Run global FDR evaluation
interactions_global <- compute_global_fdr(mock_interactions)

# Inspect global links
interactions_global$IBD$sig_links_global
attr(interactions_global, "global_hits")
```

`compute_residualCLR_correlations` *Compute confounder-adjusted Residual CLR taxon-pathway correlations for Disease-Associated Features*

Description

Applies the second analytical layer of MiCARA after differential abundance analysis. For each disease with at least one robust differentially abundant taxon and pathway, the function:

Usage

```
compute_residualCLR_correlations(
  micara_obj,
  diffab_obj = NULL,
  taxa_diffab = NULL,
  pathway_diffab = NULL,
  diseases = NULL,
  covariates = NULL,
  q_cutoff = 0.05,
  r_cutoff = 0.2,
  pseudocount = NULL,
  min_samples_per_disease = 10L,
  min_residual_df = 3L,
  retain_all_pairs = TRUE,
  verbose = TRUE
)
```

Arguments

`micara_obj`	A `micara_input` or `micara_imputed` object.
`diffab_obj`	Optional combined `micara_diffab_list` object with elements `$taxa` and `$pathways`. A single `micara_diffab` object is also accepted when the complementary object is supplied through `taxa_diffab` or `pathway_diffab`.
`taxa_diffab`	Optional `micara_diffab` object produced by `run_diff_abundance` with `feature_type = "taxa"`.
`pathway_diffab`	Optional `micara_diffab` object produced by `run_diff_abundance` with `feature_type = "pathways"`.

`diseases` Character vector of diseases to analyse. The default `NULL` analyses every disease represented in both differential abundance result objects.

`covariates` Character vector of metadata columns to remove before correlation. The default `NULL` uses all confounders retained by `validate_inputs()` (`micara_obj$confounders`); as in `run_diff_abundance()`, continuous `age` is used instead of `age_category` when both are present. Non-estimable covariates are dropped within each disease and reported.

`q_cutoff` Strict BH-adjusted p-value threshold for retaining links. Default `0.05`.

`r_cutoff` Strict minimum absolute Spearman correlation for retaining links. Default `0.2`. This is an effect-size threshold for network inclusion, not a universal biological cutoff.

`pseudocount` Zero replacement used before taking logarithms. `NULL` uses half the minimum positive value separately for taxa and pathways. A single positive number applies to both matrices. A named list such as `list(taxa = 1e-06, pathways = 1e-08)` may also be supplied.

`min_samples_per_disease`
Minimum number of disease samples required before attempting residualisation. Default `10`.

`min_residual_df`
Minimum residual degrees of freedom required after building the covariate design matrix. Default `3`.

`retain_all_pairs`
Logical. If `TRUE`, each disease result also stores every tested pair and its p/q values in `$all_links`. Default `TRUE`.

`verbose` Logical; print progress and a disease-level summary.

Details

1. CLR-transforms the complete taxonomic and pathway matrices.
2. selects the disease-specific significant taxa and pathways.
3. restricts the analysis to samples from that disease.
4. residualises both feature sets against estimable covariates.
5. computes all taxon-pathway Spearman correlations on the residuals.
6. applies Benjamini-Hochberg correction within each disease.
7. retains links satisfying both `q < q_cutoff` and `abs(rho) > r_cutoff`.

CLR is calculated on the full feature set before disease-specific feature selection. Canonical taxon and pathway identifiers are preserved; labels should be cleaned only in downstream plotting functions.

#' For analyses requiring correction across the pooled interaction search space, the returned object can optionally be passed to `compute_global_fdr`.

Value

An object of class `micara_interactions`. It is a named list with one element per analysed disease. Each disease element contains:

status `"analysed"` or `"skipped"`.

reason Reason for skipping, otherwise `NA`.

links Associations passing both thresholds.

all_links All tested associations when `retain_all_pairs = TRUE`.

node_directions Differential-abundance directions for tested taxa and pathways.

covariates_requested, covariates_used, covariates_dropped Covariate provenance for that disease.

n_samples, residual_df, n_taxa_tested, n_pathways_tested, n_pairs_tested Analysis scope and model diagnostics.

The object carries `"summary"` and `"settings"` attributes. Access them with `attr(x, "summary")` and `attr(x, "settings")`.

See Also

compute_global_fdr

Examples

```
set.seed(123)
sample_ids <- paste0("Sample_", 1:20)

taxa_mat <- as.data.frame(matrix(
    rpois(100, lambda = 50),
    nrow = 5, ncol = 20,
    dimnames = list(paste0("Taxon_", 1:5), sample_ids)
))

path_mat <- as.data.frame(matrix(
    rpois(100, lambda = 100),
    nrow = 5, ncol = 20,
    dimnames = list(paste0("Pathway_", 1:5), sample_ids)
))

# Inject strong signal so features pass significance testing
taxa_mat["Taxon_1", 11:20] <- taxa_mat["Taxon_1", 11:20] + 300
path_mat["Pathway_1", 11:20] <- path_mat["Pathway_1", 11:20] + 500

meta_df <- data.frame(
    sample_id = sample_ids,
    disease = rep(c("Control", "Disease"), each = 10),
    age = rnorm(20, mean = 40, sd = 5),
    row.names = sample_ids
)

validated_obj <- validate_inputs(
```

```
    taxa = taxa_mat,
    pathways = path_mat,
    metadata = meta_df,
    disease_col = "disease",
    disease_min_samples = 5,
    verbose = FALSE
)

diff_res <- run_diff_abundance(
    micara_obj = validated_obj,
    feature_type = c("taxa", "pathways"),
    rand_effect = NULL,
    n_cores = 1,
    verbose = FALSE
)

corr_res <- compute_residualCLR_correlations(
    micara_obj = validated_obj,
    diffab_obj = diff_res,
    covariates = "age",
    q_cutoff = 1.0,
    min_samples_per_disease = 5L,
    verbose = FALSE
)
```

`example_metadata` *Synthetic Example Sample Metadata*

Description

A simulated metadata data frame containing sample covariates and disease status.

Usage

```
data(example_metadata)
```

Format

A data frame with 30 rows and 6 columns:

sample_id Unique sample identifier

disease Disease group assignment ("healthy" or "Case")

age_category Age cohort ("Adult" or "Senior")

BMI Body Mass Index

gender Sex of the participant

study_name Cohort study origin

Source

Simulated with `inst/scripts/generate_example_data.R` (seed 123).

`example_pathways` *Synthetic Example Functional Pathways Table*

Description

A simulated table of functional pathway counts across samples.

Usage

```
data(example_pathways)
```

Format

A data frame of integer counts with 10 rows (pathways, `Pathway_1` to `Pathway_10`) and 30 columns (samples, `Sample_1` to `Sample_30`).

Source

Simulated with `inst/scripts/generate_example_data.R` (seed 123).

`example_taxa` *Synthetic Example Taxa Abundance Table*

Description

A simulated table of bacterial taxon counts across samples.

Usage

```
data(example_taxa)
```

Format

A data frame of integer counts with 20 rows (taxa, `Taxon_1` to `Taxon_20`) and 30 columns (samples, `Sample_1` to `Sample_30`).

Source

Simulated with `inst/scripts/generate_example_data.R` (seed 123).

impute_metadata *Impute Missing Metadata Covariates Using MICE (Single Imputation)*

Description

Imputes missing values in the confounders retained by validate_inputs (for example age, gender and BMI; i.e., below the missingness cutoff). Experimental design indicators such as disease, study_name, and sequencing depth (number_reads) are strictly protected and used only as auxiliary predictors to inform imputation. age_category is also treated as an auxiliary predictor when complete to guide continuous age imputation.

Usage

```
impute_metadata(
  micara_obj,
  impute_vars = NULL,
  auxiliary_predictors = NULL,
  m = 1,
  method = NULL,
  seed = 123,
  imputation_index = 1,
  verbose = TRUE
)
```

Arguments

micara_obj	An object of class micara_input, as returned by validate_inputs.
impute_vars	Character vector of confounders to impute. Defaults to the confounders retained by validate_inputs() (micara_obj$confounders) that contain missing values, excluding the design variables study_name and age_category.
auxiliary_predictors	Character vector of additional metadata columns to include as MICE predictors (never imputation targets). Defaults to c("study_name", "age_category", "disease") when present.
m	Number of imputed datasets generated by mice::mice() (default 1, single imputation). Only the dataset selected by imputation_index is used downstream; m > 1 is useful for diagnostics (all datasets are kept in mice_fit) and sensitivity analyses.
method	Imputation method passed to mice::mice(). Defaults to NULL, allowing MICE to automatically select the optimal algorithm based on column type ("pmm" for continuous, "logreg" for binary, "polyreg" for categorical).
seed	Random seed for reproducibility (default 123).
imputation_index	Which of the m imputed datasets to use for the returned metadata (default 1). The full mids fit object is retained in micara_obj$mice_fit for diagnostic checks.
verbose	Logical; print progress and summary report to console (default TRUE).

Details

Single imputation: `impute_metadata()` returns one completed dataset (by default `m = 1`), which the downstream functions then treat as observed data. Standard errors and p-values therefore do not include the uncertainty due to the missing values; this matters less when few values are missing (at most `metadata_missing_cutoff` per covariate, 30\ `validate_inputs`). As a sensitivity analysis, results can be compared across completed datasets by rerunning with a different `seed`, or with `m > 1` and a different `imputation_index`. Pooling the downstream analyses across multiple imputations with Rubin's rules is planned for a future release.

Scientific Guidance on Downstream Covariate Selection: Imputed continuous `age` is generally recommended over binned `age_category` for differential abundance modeling (e.g., ANCOM-BC2) as it avoids discretization power loss. However, `age_category` should be used directly if modeling distinct non-linear biological thresholds (e.g., pediatric vs. adult states) or if continuous `age` missingness exceeded 50\

Value

An object of class `c("micara_imputed", "micara_input")`: the input object with updated `metadata`, plus `mice_fit` (the full `mids` object, or `NULL` if skipped), `imputed_vars`, and `imputation_performed` (logical; `FALSE` if no missingness was present or imputation was skipped).

Examples

```
set.seed(123)
sample_ids <- paste0("Sample_", 1:20)

taxa_mat <- as.data.frame(matrix(
    rpois(100, lambda = 50),
    nrow = 5, ncol = 20,
    dimnames = list(paste0("Taxon_", 1:5), sample_ids)
))

path_mat <- as.data.frame(matrix(
    rpois(100, lambda = 100),
    nrow = 5, ncol = 20,
    dimnames = list(paste0("Pathway_", 1:5), sample_ids)
))

# Create metadata with missing age values (NA)
meta_df <- data.frame(
    sample_id = sample_ids,
    disease = rep(c("Control", "Disease"), each = 10),
    age = c(
        25, NA, 30, 45, NA, 50, 33, 29, 41, 38,
        52, 60, NA, 48, 55, 62, NA, 39, 44, 51
    ),
    row.names = sample_ids
)

validated_obj <- validate_inputs(
    taxa = taxa_mat,
```

```
    pathways = path_mat,
    metadata = meta_df,
    disease_col = "disease",
    disease_min_samples = 5,
    verbose = FALSE
)

# Run imputation on missing age column
imputed_obj <- impute_metadata(
    micara_obj = validated_obj,
    impute_vars = "age",
    m = 2,
    seed = 123,
    verbose = FALSE
)
```

`micara-print` *Print MiCARA objects*

Description

Compact summaries of the objects returned by `validate_inputs`, `impute_metadata`, `run_diff_abundance` and `compute_residualCLR_correlations`, instead of printing the whole underlying list. The full contents remain available as list elements (for example `x$metadata`, `x$significant` or `x[["disease"]]$links`).

Usage

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

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

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

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

Arguments

`x`	A MiCARA object.
`...`	Ignored.

Value

`x`, invisibly.

Examples

```
data("example_taxa", package = "MiCARA")
data("example_pathways", package = "MiCARA")
data("example_metadata", package = "MiCARA")

micara_obj <- validate_inputs(
    example_taxa, example_pathways, example_metadata,
    verbose = FALSE
)
micara_obj
```

`plot_interaction_sankey` *Plot a taxon-pathway interaction Sankey diagram*

Description

Renders a Sankey diagram of confounder-adjusted taxon-pathway correlations for a single disease, as computed by `compute_residualCLR_correlations`. Positive and negative correlations are rendered as separate diagrams (matching the two-panel "Figure A/B" convention used throughout this package's companion analyses) to preserve clear biological interpretation.

Usage

```
plot_interaction_sankey(
  interactions,
  disease,
  direction = c("both", "positive", "negative"),
  significance = c("local", "global"),
  top_n_per_pathway = NULL,
  node_colors = c(up = "#2196F3", down = "#F44336", unchanged = "grey70", stable =
    "gray70"),
  font_size = 18,
  node_width = 35,
  save_path = NULL
)
```

Arguments

`interactions`	A `micara_interactions` object from `compute_residualCLR_correlations`.
`disease`	Name of the disease to plot (must be present in `names(interactions)`).
`direction`	One of `"both"`, `"positive"`, or `"negative"` (default `"both"`). When `"both"`, returns and/or saves separate diagrams for positive and negative links.
`significance`	Character string specifying which interaction set to plot. `"local"` uses the disease-specific significant links stored in `$links`; `"global"` uses the globally corrected links stored in `$global_links`. Defaults to `"local"`.

top_n_per_pathway
: Optional integer; if supplied, keeps only the top_n_per_pathway strongest links (by abs(rho)) per pathway node, for readability in densely-connected diagrams. Default NULL (no trimming — all links passing the cutoffs in compute_residualCLR_correlatio are shown).

node_colors
: Named character vector mapping node group to colour. Default c(up = "#2196F3", down = "#F44336", unchanged = "grey70", stable = "gray70") — "up"/"down" refer to the ANCOM-BC2 direction of that taxon/pathway (enriched/upregulated vs depleted/downregulated), consistent with node_directions$group from compute_residualCLR_cor

font_size, node_width
: Passed to networkD3::sankeyNetwork().

save_path
: Optional file path (with or without extension) to save the diagram(s). If supplied, saves interactive .html (via htmlwidgets::saveWidget()) and static .png (via webshot2::webshot()) files. When direction = "both", suffixes _positive and _negative are automatically appended to the base path. Default NULL.

Details

When significance = "global", the input object must have been processed by compute_global_fdr.

Value

A networkD3 htmlwidget object (for "positive" or "negative"), or a named list of two networkD3 htmlwidget objects ($positive and $negative) when direction = "both" (returned invisibly if save_path is supplied).

Examples

```
set.seed(123)
sample_ids <- paste0("Sample_", 1:20)

taxa_mat <- as.data.frame(matrix(
    rpois(100, lambda = 50),
    nrow = 5, ncol = 20,
    dimnames = list(paste0("Taxon_", 1:5), sample_ids)
))

path_mat <- as.data.frame(matrix(
    rpois(100, lambda = 100),
    nrow = 5, ncol = 20,
    dimnames = list(paste0("Pathway_", 1:5), sample_ids)
))

# Inject strong differential signal AND wide variance across Disease samples
taxa_mat["Taxon_1", 11:20] <- seq(100, 1000, length.out = 10)
path_mat["Pathway_1", 11:20] <- taxa_mat["Taxon_1", 11:20] * 2

meta_df <- data.frame(
    sample_id = sample_ids,
```

```
    disease = rep(c("Control", "Disease"), each = 10),
    age = rnorm(20, mean = 40, sd = 5),
    row.names = sample_ids
)

validated_obj <- validate_inputs(
    taxa = taxa_mat,
    pathways = path_mat,
    metadata = meta_df,
    disease_col = "disease",
    disease_min_samples = 5,
    verbose = FALSE
)

diff_res <- run_diff_abundance(
    micara_obj = validated_obj,
    feature_type = c("taxa", "pathways"),
    rand_effect = NULL,
    n_cores = 1,
    verbose = FALSE
)

corr_res <- compute_residualCLR_correlations(
    micara_obj = validated_obj,
    diffab_obj = diff_res,
    covariates = "age",
    q_cutoff = 1.0,
    min_samples_per_disease = 5L,
    verbose = FALSE
)

plot_interaction_sankey(corr_res, disease = "Disease")
```

`run_diff_abundance` *Run Differential Abundance Analysis Using ANCOM-BC2*

Description

Performs bias-corrected differential abundance testing for microbial taxa, functional pathways, or both using ANCOM-BC2 (`ANCOMBC::ancombc2`). Automatically integrates retained host confounders from `validate_inputs` as fixed effects and supports multi-cohort random effects.

Usage

```
run_diff_abundance(
  micara_obj,
  feature_type = c("taxa", "pathways"),
  fix_effects = NULL,
  exclude_covariates = NULL,
  rand_effect = "study_name",
```

```
  reference_level = NULL,
  p_adj_method = "BH",
  prv_cut = 0.1,
  alpha = 0.05,
  pairwise = FALSE,
  struc_zero = TRUE,
  neg_lb = TRUE,
  pseudo_sens = TRUE,
  use_robust = TRUE,
  n_cores = 1,
  verbose = TRUE
)
```

Arguments

micara_obj	An object of class micara_input (or micara_imputed), as returned by validate_inputs or impute_metadata.
feature_type	Character vector specifying feature matrices to analyze: "taxa", "pathways", or c("taxa", "pathways") (default). Keywords "all" and "both" are also supported.
fix_effects	Character vector of metadata columns to include as fixed-effect covariates. If NULL (default), auto-detects retained confounders from micara_obj$confounders (excluding "study_name" and "disease").
exclude_covariates	Character vector of covariate names to explicitly exclude from fixed effects.
rand_effect	Character string naming the metadata column for random intercepts (default "study_name"). Automatically set to NULL if the column is missing or contains only one unique level.
reference_level	Character string defining the control or reference category within metadata$disease (default "healthy"). All differential log-fold changes will be calculated relative to this baseline.
p_adj_method	Method for p-value adjustment passed to ancombc2 (default "BH").
prv_cut	Minimum prevalence threshold (0 to 1) for features to be retained prior to analysis (default 0).
alpha	Significance level threshold for q-values (default 0.05).
pairwise	Logical; whether to perform pairwise comparisons between all disease levels (default FALSE).
struc_zero	Logical; whether to detect structural zeros (default TRUE).
neg_lb	Logical; whether to apply a negative lower bound for asymptotic variance estimation (default TRUE).
pseudo_sens	Logical; whether to perform pseudo-count sensitivity analysis (default TRUE). Automatically set to TRUE if use_robust = TRUE.
use_robust	Logical; whether to use robust sensitivity flags (diff_robust_disease) rather than standard differential abundance flags (diff_disease) to define significance (default TRUE).

n_cores	Integer specifying the number of CPU cores for parallel processing (default 1).
verbose	Logical; print progress and summary report to console (default TRUE).

Details

Multi-Feature Execution: When feature_type = c("taxa", "pathways"), the function iterates over both datasets if present in micara_obj and returns a named list of analysis objects.

Model Specification & Safeguards: Fixed effects automatically incorporate retained host confounders (e.g., age, gender, BMI) alongside disease. Variables with zero variance or missing values are automatically flagged. The random effect (rand_effect = "study_name") accounts for cohort-level batch variation and is automatically disabled if only a single cohort/study is present in the dataset.

Sensitivity Flags: By default (use_robust = TRUE), significance is defined using robust sensitivity flags (diff_robust_disease...) calculated across pseudo-count additions, providing high confidence against false positives in sparse microbiome data.

Value

If a single feature_type is analyzed, an object of class "micara_diffab". If multiple feature types are analyzed, a named list of "micara_diffab" objects with class "micara_diffab_list".

Examples

```
set.seed(123)
sample_ids <- paste0("Sample_", 1:20)

# Create count matrices required for differential abundance modeling
taxa_mat <- as.data.frame(matrix(
    rpois(100, lambda = 50),
    nrow = 5, ncol = 20,
    dimnames = list(paste0("Taxon_", 1:5), sample_ids)
))

path_mat <- as.data.frame(matrix(
    rpois(100, lambda = 100),
    nrow = 5, ncol = 20,
    dimnames = list(paste0("Pathway_", 1:5), sample_ids)
))

meta_df <- data.frame(
    sample_id = sample_ids,
    disease = rep(c("Control", "Disease"), each = 10),
    study_name = rep(c("Study1", "Study2"), each = 10),
    row.names = sample_ids
)

validated_obj <- validate_inputs(
    taxa = taxa_mat,
    pathways = path_mat,
    metadata = meta_df,
    disease_col = "disease",
```

```
    disease_min_samples = 5,
    verbose = FALSE
)

# Run differential abundance analysis (set n_cores = 1 for CRAN/Bioc compliance)
diff_res <- run_diff_abundance(
    micara_obj = validated_obj,
    feature_type = "taxa",
    rand_effect = NULL,
    n_cores = 1,
    verbose = FALSE
)
```

`validate_inputs` *Validate and preprocess MiCARA input tables*

Description

Performs quality control on taxonomic and functional pathway abundance tables and accompanying sample metadata.

Usage

```
validate_inputs(
  taxa,
  pathways = NULL,
  metadata = NULL,
  disease_col = "disease",
  confounders = NULL,
  pathways_scale = "auto",
  experiments = c(taxa = "taxa", pathways = "pathways"),
  disease_min_samples = 10,
  metadata_missing_cutoff = 0.3,
  normalise = TRUE,
  verbose = TRUE
)
```

Arguments

`taxa`	Taxonomic abundance table (features in rows, samples in columns): a data.frame or matrix, a `SummarizedExperiment` or `TreeSummarizedExperiment` (its first assay is used), or a `MultiAssayExperiment` holding both the taxa and the pathway tables (see `experiments`).
`pathways`	Functional pathway abundance table, in the same forms as `taxa`. Ignored when `taxa` is a `MultiAssayExperiment`.
`metadata`	Sample metadata. Must contain sample IDs and disease labels. Optional when `taxa` is a `(Tree)SummarizedExperiment` or a `MultiAssayExperiment`: its sample data (`colData`) is then used.

`disease_col` Name of the disease column in metadata. Default "disease".

`confounders` Optional character vector of metadata columns to treat as confounders, for example `c("age", "BMI", "antibiotics")`. The default `NULL` recognises common host and study covariates by name (case-insensitive): `study_name`, `age`, `age_category`, `gender`, `sex`, `bmi`, `body_site` and `site`. In both cases, confounders with more than `metadata_missing_cutoff` missing values are removed, and the QC report lists the metadata columns that are not used. The retained confounders are the defaults for `impute_metadata()`, `run_diff_abundance()` and `compute_residualCLR_correlations()`.

`pathways_scale` Scale of the pathway table. The default `"auto"` recognises integer counts, proportions, percentages and HUMAnN copies per million (samples summing to about one million). HUMAnN RPK tables (the default HUMAnN output) cannot be recognised automatically; use `pathways_scale = "rpk"`. Other values: `"counts"`, `"proportion"`, `"percentage"` and `"cpm"`. CPM and RPK values are converted to within-sample proportions and then handled like relative abundances (see the section "Abundance scales").

`experiments` Only used when `taxa` is a `MultiAssayExperiment`: the names (or positions) of its taxa and pathway experiments. Default `c(taxa = "taxa", pathways = "pathways")`.

`disease_min_samples` Minimum required samples per disease group. Default 10.

`metadata_missing_cutoff` Allowed missing value fraction for covariates. Default 0.30.

`normalise` Logical indicating whether feature normalization should be performed.

`verbose` Logical; print QC summary. Default TRUE.

Details

Rows named `UNMAPPED` or `UNINTEGRATED` and per-taxon (stratified) HUMAnN pathway rows are removed, so community-level pathway tables such as those from curatedMetagenomicData can be used directly. To import raw MetaPhlAn or HUMAnN output files, use `mia::importMetaPhlAn()` or `mia::importHUMAnN()`.

Value

An object of class `"micara_input"`.

Abundance scales

Relative abundances (proportions, percentages, or HUMAnN CPM or RPK converted to proportions) are multiplied by `metadata$number_reads` and rounded when that column is available. Pathway abundances, and MetaPhlAn relative abundances, are not in units of reads, so the result is a count-scaled abundance rather than an observed read count. In principle, this per-sample rescaling does not change the log-fold changes: ANCOM-BC2 estimates and removes a sample-specific bias term, and the CLR transformation used for the correlations is unchanged when a sample is multiplied by a constant (apart from zero replacement). The rescaling is needed because ANCOM-BC2's pseudo-count sensitivity analysis adds fixed pseudo-counts, which are small relative to the observed values only on a count-like scale.

Examples

```
set.seed(123)
sample_ids <- paste0("Sample_", 1:20)

taxa_mat <- as.data.frame(matrix(
    rpois(100, lambda = 50),
    nrow = 5, ncol = 20,
    dimnames = list(paste0("Taxon_", 1:5), sample_ids)
))

path_mat <- as.data.frame(matrix(
    rpois(100, lambda = 100),
    nrow = 5, ncol = 20,
    dimnames = list(paste0("Pathway_", 1:5), sample_ids)
))

meta_df <- data.frame(
    sample_id = sample_ids,
    disease = rep(c("Control", "Disease"), each = 10),
    age = rnorm(20, mean = 40, sd = 5),
    row.names = sample_ids
)

validated_obj <- validate_inputs(
    taxa = taxa_mat,
    pathways = path_mat,
    metadata = meta_df,
    disease_col = "disease",
    disease_min_samples = 5,
    verbose = FALSE
)
```

Index

∗ **datasets**
example_metadata, 7
example_pathways, 8
example_taxa, 8

compute_global_fdr, 2, *5, 6*, *13*
compute_residualCLR_correlations, *2, 3*, 4, *11, 12*

example_metadata, 7
example_pathways, 8
example_taxa, 8

impute_metadata, 9, *11*, *15*

micara-print, 11

p.adjust, *2*
plot_interaction_sankey, 12
print.micara_diffab *(*micara-print*)*, 11
print.micara_diffab_list *(*micara-print*)*, 11
print.micara_input *(*micara-print*)*, 11
print.micara_interactions *(*micara-print*)*, 11

run_diff_abundance, *4*, *11*, 14

validate_inputs, *9–11*, *14, 15*, 17