Package 'ExpoRiskR'

June 5, 2026

Type Package

Title Exposure-Aware Multi-Omics Risk Modeling

Version 1.1.0

Description ExpoRiskR provides tools for exposure-aware multi-omics risk modeling in translational and environmental health studies. The package aligns sample identifiers across exposure and multi-omics blocks, performs lightweight preprocessing, and fits exposure-adjusted association models to build interpretable microbe–metabolite networks. It also computes simple exposure perturbation summaries and generates publication-ready visualizations. Workflows support both matrix-based inputs and SummarizedExperiment objects.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

URL https://github.com/ppchaudhary/ExpoRiskR

BugReports https://github.com/ppchaudhary/ExpoRiskR/issues

Imports stats, ggplot2, igraph, SummarizedExperiment, S4Vectors, utils

Suggests BiocStyle, BiocCheck, knitr, rmarkdown, withr, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

biocViews Software, Network, SystemsBiology, Metabolomics, Microbiome, Regression

git_url https://git.bioconductor.org/packages/ExpoRiskR

git_branch devel

git_last_commit 1e05ac3

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-06-04

Author Prem Prashant Chaudhary [aut, cre] (ORCID: <https://orcid.org/0000-0002-3467-8608>)

Maintainer Prem Prashant Chaudhary <chaudharyp2@nih.gov>

Contents

`align_omics` *Align exposures and multi-omics blocks by sample ID*

Description

Ensures that microbiome, metabolome, exposures, and metadata all refer to the same set of samples in the same order. Sample IDs are taken from rownames of matrices/ data.frames, or from a column in `meta` if `id_col` is provided.

Usage

```
align_omics(
  microbiome,
  metabolome,
  exposures,
  meta,
  id_col = NULL,
  strict = TRUE
)
```

Arguments

`microbiome`	Matrix/data.frame of samples x microbes.
`metabolome`	Matrix/data.frame of samples x metabolites.
`exposures`	Matrix/data.frame of samples x exposures.
`meta`	data.frame of sample-level metadata (must include outcome later).
`id_col`	Optional column name in `meta` containing sample IDs. If NULL, rownames(meta) are used (if present).
`strict`	If TRUE, errors if any block has samples not found in others. If FALSE, intersects common samples and drops others.

Value

A list with aligned `microbiome`, `metabolome`, `exposures`, `meta`, and `sample_id`.

Examples

```
set.seed(4)
d <- generate_dummy_exporisk(n = 20, p_micro = 6, p_metab = 8, p_expo = 3)
aligned <- align_omics(d$microbiome, d$metabolome, d$exposures, d$meta,
                       id_col = "sample_id", strict = TRUE)
names(aligned)
```

`align_omics_se` *Align two SummarizedExperiment objects and extract exposures from colData*

Description

Convenience wrapper to (i) align microbiome, metabolome, and exposures by sample ID and (ii) return two `SummarizedExperiment` objects (microbiome + metabolome) that share the same `colData` (meta + exposures). This is useful for Bioconductor-style workflows.

Inputs `microbiome`, `metabolome`, `exposures` are expected to be sample-by-feature matrices (or coercible to matrices). Sample IDs are taken from rownames when present; otherwise from `meta[[id_col]]`.

Usage

```
align_omics_se(
  microbiome,
  metabolome,
  exposures,
  meta,
  id_col = "sample_id",
  strict = TRUE
)
```

Arguments

`microbiome`	Matrix/data.frame (samples x microbes).
`metabolome`	Matrix/data.frame (samples x metabolites).
`exposures`	Matrix/data.frame (samples x exposures).
`meta`	Data.frame with sample metadata including `id_col`.
`id_col`	Column name in `meta` holding sample IDs (default "sample_id").
`strict`	If TRUE, require that all blocks contain the same sample IDs; otherwise subset to the intersection (default TRUE).

Value

A list with:

- `se_microbiome`: SummarizedExperiment for microbiome (features x samples)
- `se_metabolome`: SummarizedExperiment for metabolome (features x samples)
- `exposures`: aligned numeric matrix (samples x exposures)
- `meta`: aligned meta data.frame
- `sample_ids`: character vector of aligned sample IDs

Examples

```
set.seed(7)
d <- generate_dummy_exporisk(n = 12, p_micro = 5, p_metab = 6, p_expo = 3)
out <- align_omics_se(
  d$microbiome, d$metabolome, d$exposures, d$meta,
  id_col = "sample_id", strict = TRUE
)
out$se_microbiome
out$se_metabolome
```

`build_exposure_network` *Build an exposure-adjusted microbe-metabolite association network*

Description

For each (microbe, metabolite) pair, fits a linear model:

$$metabolite\ microbe + exposures + covariates$$

and uses the microbe coefficient as the edge weight.

This is an MVP, interpretable approach suitable for Bioconductor submission.

Usage

```
build_exposure_network(
  X,
  Y,
  E,
  covar = NULL,
  fdr = 0.1,
  max_pairs = 5000,
  seed = NULL
)
```

Arguments

`X`	Numeric matrix (samples x microbes).
`Y`	Numeric matrix (samples x metabolites).
`E`	Numeric matrix (samples x exposures).
`covar`	Optional data.frame of sample-level covariates (rows = samples).
`fdr`	FDR threshold for keeping edges (BH adjusted p-value).
`max_pairs`	Max number of (microbe, metabolite) pairs to test (for speed). If NULL, tests all pairs (may be slow).
`seed`	Optional random seed used only when `max_pairs` is not NULL and sampling is required. If `NULL`, the current RNG state is used.

Value

A list with:

- edges: data.frame of significant edges (microbe, metabolite, weight, p_value, fdr)
- graph: igraph object (bipartite)
- meta: list of settings and counts

Examples

```
set.seed(1)
d <- generate_dummy_exporisk(n = 30, p_micro = 10, p_metab = 12, p_expo = 4)
al <- align_omics(d$microbiome, d$metabolome, d$exposures, d$meta,
                  id_col = "sample_id", strict = TRUE)
pr <- prep_omics(al$microbiome, al$metabolome, al$exposures)
net <- build_exposure_network(pr$X, pr$Y, pr$E, fdr = 0.5, max_pairs = 120, seed = 1)
utils::head(net$edges)
```

---

exposure_perturbation_score

Score exposures by network perturbation (leave-one-exposure-out)

Description

Builds a reference network using all exposures, then for each exposure j builds a network leaving out exposure j, and computes a perturbation score based on differences in edge weights for a subset of tested pairs.

This is an MVP perturbation metric designed to be interpretable and fast enough for simulated/demo datasets.

Usage

```
exposure_perturbation_score(
  X,
  Y,
  E,
  covar = NULL,
  fdr = 0.2,
  max_pairs = 3000,
  seed = 1
)
```

Arguments

X	Microbiome matrix (samples x microbes).
Y	Metabolome matrix (samples x metabolites).
E	Exposures matrix (samples x exposures).
covar	Optional covariates data.frame.
fdr	FDR threshold passed to build_exposure_network().
max_pairs	Number of pairs to test per network build (speed control).
seed	Random seed.

Value

A data.frame with exposure, perturbation_score, n_edges_ref, n_edges_drop.

Examples

```
set.seed(2)
d <- generate_dummy_exporisk(n = 30, p_micro = 10, p_metab = 12, p_expo = 4)
al <- align_omics(d$microbiome, d$metabolome, d$exposures, d$meta,
                  id_col = "sample_id", strict = TRUE)
pr <- prep_omics(al$microbiome, al$metabolome, al$exposures)
scores <- exposure_perturbation_score(pr$X, pr$Y, pr$E, fdr = 0.5, max_pairs = 120, seed = 1)
scores
```

`generate_dummy_exporisk` *Generate simulated exposure + multi-omics data with a binary outcome*

Description

Creates a reproducible toy dataset for demonstrating ExpoRiskR workflows: exposures (E), microbiome-like positive features (X), metabolome-like positive features (Y), and a binary disease outcome.

If `seed` is provided, reproducibility is ensured locally without modifying the global RNG state.

Usage

```
generate_dummy_exporisk(
  n = 120,
  p_micro = 50,
  p_metab = 80,
  p_expo = 10,
  n_signal = 6,
  seed = NULL
)
```

Arguments

`n`	Number of samples.
`p_micro`	Number of microbiome features.
`p_metab`	Number of metabolomics features.
`p_expo`	Number of exposure variables.
`n_signal`	Number of truly associated features per block.
`seed`	Optional random seed for reproducible simulation.

Value

A list with matrices: microbiome, metabolome, exposures; and meta data.frame.

Examples

```
d <- generate_dummy_exporisk(n = 20, p_micro = 6, p_metab = 8, p_expo = 3, seed = 1)
str(d)
```

`plot_exposure_network` *Plot exposure-adjusted multi-omics network (bipartite)*

Description

Plots a bipartite igraph network returned by `build_exposure_network()`. Uses base igraph plotting (no extra dependencies).

Usage

```
plot_exposure_network(
  net,
  file = NULL,
  width = 10,
  height = 7,
  dpi = 300,
  layout = "layout_with_fr",
  max_label_nodes = 30
)
```

Arguments

`net`	A list returned by `build_exposure_network()` with elements `$graph` and `$edges`.
`file`	Optional output filename. If provided, saves a PNG (recommended).
`width, height`	Plot device size (in inches) when saving.
`dpi`	DPI when saving PNG.
`layout`	Layout function name passed to igraph. Default `"layout_with_fr"`.
`max_label_nodes`	Max nodes to label (largest by degree). Default 30.

Value

Invisibly returns `net$graph`.

Examples

```
d <- generate_dummy_exporisk(seed = 1, n = 12, p_micro = 5, p_metab = 6, p_expo = 3)
al <- align_omics(d$microbiome, d$metabolome, d$exposures, d$meta,
                  id_col = "sample_id", strict = TRUE)
pr <- prep_omics(al$microbiome, al$metabolome, al$exposures)
net <- build_exposure_network(pr$X, pr$Y, pr$E, fdr = 0.95, max_pairs = 120, seed = 1)
plot_exposure_network(net)
```

plot_exposure_ranking *Plot exposure perturbation ranking*

Description

Plot exposure perturbation ranking

Usage

```
plot_exposure_ranking(scores, top_n = 20)
```

Arguments

scores	A data.frame from exposure_perturbation_score().
top_n	Show only top N exposures (default 20). Use NULL for all.

Value

A ggplot object.

Examples

```
d <- generate_dummy_exporisk(n = 30, p_micro = 10, p_metab = 12, p_expo = 4)
al <- align_omics(d$microbiome, d$metabolome, d$exposures, d$meta,
                  id_col = "sample_id", strict = TRUE)
pr <- prep_omics(al$microbiome, al$metabolome, al$exposures)
scores <- exposure_perturbation_score(pr$X, pr$Y, pr$E,
                                      fdr = 0.5, max_pairs = 120, seed = 1)
plot_exposure_ranking(scores)
```

plot_feature_importance *Plot feature importance for exposures (logistic regression)*

Description

Fits a logistic regression outcome ~ exposures and ranks exposures by the absolute standardized coefficient magnitude.

Usage

```
plot_feature_importance(E, outcome, top_n = 25)
```

Arguments

E	Numeric matrix (samples x exposures).
outcome	Binary vector (0/1), length = nrow(E).
top_n	Number of top exposures to show.

Value

A ggplot object.

Examples

```
d <- generate_dummy_exporisk(seed = 1, n = 20, p_micro = 6, p_metab = 8, p_expo = 4)
outcome <- d$meta$outcome
names(outcome) <- d$meta$sample_id
p <- plot_feature_importance(E = d$exposures, outcome = outcome, top_n = 10)
print(p)
```

`plot_individual_risk_profile`
Plot individual risk profile from exposure model

Description

Fits outcome ~ exposures and shows per-exposure contribution for one sample based on standardized coefficients and standardized exposure values.

Usage

```
plot_individual_risk_profile(sample_id, E, outcome, top_n = 20)
```

Arguments

`sample_id`	Sample ID (must be in rownames(E)).
`E`	Numeric matrix (samples x exposures) with rownames.
`outcome`	Binary vector (0/1), named by sample IDs or same row order as E.
`top_n`	Number of top contributing exposures to display.

Value

A ggplot object.

Examples

```
d <- generate_dummy_exporisk(seed = 1, n = 20, p_micro = 6, p_metab = 8, p_expo = 4)
outcome <- d$meta$outcome
names(outcome) <- d$meta$sample_id
sid <- rownames(d$exposures)[1]
p <- plot_individual_risk_profile(sample_id = sid, E = d$exposures, outcome = outcome, top_n = 10)
print(p)
```

plot_network_stability *Plot network stability by bootstrap edge overlap*

Description

Builds a reference network using all samples, then repeatedly bootstraps samples with replacement, rebuilds the network, and computes Jaccard overlap between edge sets.

Usage

```
plot_network_stability(
  X,
  Y,
  E,
  n_boot = 50,
  fdr = 0.2,
  max_pairs = 2000,
  seed = NULL
)
```

Arguments

X	Numeric matrix (samples x microbes).
Y	Numeric matrix (samples x metabolites).
E	Numeric matrix (samples x exposures).
n_boot	Number of bootstrap resamples.
fdr	FDR threshold passed to build_exposure_network().
max_pairs	Maximum pairs passed to build_exposure_network().
seed	Optional seed controlling bootstrap resampling only.

Value

A ggplot object.

Examples

```
d <- generate_dummy_exporisk(seed = 1, n = 20, p_micro = 8, p_metab = 10, p_expo = 4)
al <- align_omics(d$microbiome, d$metabolome, d$exposures, d$meta,
                  id_col = "sample_id", strict = TRUE)
pr <- prep_omics(al$microbiome, al$metabolome, al$exposures)
p <- plot_network_stability(pr$X, pr$Y, pr$E, n_boot = 2, fdr = 0.95, max_pairs = 120, seed = 1)
print(p)
```

plot_risk_roc *Plot disease risk stratification ROC curves (MVP)*

Description

Plot disease risk stratification ROC curves (MVP)

Usage

```
plot_risk_roc(X, Y, E, outcome, edges, top_edges = 200)
```

Arguments

X	Microbiome matrix (samples x features)
Y	Metabolome matrix (samples x features)
E	Exposures matrix (samples x features)
outcome	Binary vector (0/1)
edges	Network edges data.frame
top_edges	Number of strongest edges for network feature

Value

A ggplot object

Examples

```
d <- generate_dummy_exporisk(seed = 1, n = 25, p_micro = 8, p_metab = 10, p_expo = 4)
al <- align_omics(d$microbiome, d$metabolome, d$exposures, d$meta,
                  id_col = "sample_id", strict = TRUE)
pr <- prep_omics(al$microbiome, al$metabolome, al$exposures)
net <- build_exposure_network(pr$X, pr$Y, pr$E, fdr = 0.95, max_pairs = 150, seed = 1)
outcome <- d$meta$outcome
names(outcome) <- d$meta$sample_id
p <- plot_risk_roc(pr$X, pr$Y, pr$E, outcome = outcome, edges = net$edges, top_edges = 30)
print(p)
```

prep_omics *Preprocess exposures and multi-omics blocks for modeling*

Description

Lightweight preprocessing for MVP and Bioconductor-friendly workflows. Converts inputs to numeric matrices, checks sample alignment, optionally imputes missing values, applies log1p transforms, and scales features.

Usage

```
prep_omics(
  microbiome,
  metabolome,
  exposures,
  log1p_micro = TRUE,
  log1p_metab = TRUE,
  z_expo = TRUE,
  scale_omics = TRUE,
  na_action = c("error", "impute")
)
```

Arguments

microbiome	Matrix/data.frame of samples x microbes.
metabolome	Matrix/data.frame of samples x metabolites.
exposures	Matrix/data.frame of samples x exposures.
log1p_micro	If TRUE (default), apply log1p to microbiome.
log1p_metab	If TRUE (default), apply log1p to metabolome.
z_expo	If TRUE (default), z-score exposures.
scale_omics	If TRUE (default), center/scale microbiome and metabolome features.
na_action	What to do with NA values: "error" (default) or "impute".

Value

A list with processed matrices: X, Y, E.

Examples

```
set.seed(1)
d <- generate_dummy_exporisk(n = 20, p_micro = 6, p_metab = 8, p_expo = 3)
al <- align_omics(d$microbiome, d$metabolome, d$exposures, d$meta,
                  id_col = "sample_id", strict = TRUE)
pr <- prep_omics(al$microbiome, al$metabolome, al$exposures)
str(pr)
```

---

prep_omics_se *Preprocess SummarizedExperiment-based omics blocks and exposures*

Description

Preprocess SummarizedExperiment-based omics blocks and exposures

Usage

```
prep_omics_se(aligned, assay_micro = NULL, assay_metab = NULL, ...)
```

Arguments

aligned	Output from align_omics_se() or align_omics().
assay_micro	Assay name for microbiome SE (default: first assay).
assay_metab	Assay name for metabolome SE (default: first assay).
...	Passed to prep_omics().

Value

A list with preprocessed matrices: X, Y, E.

Examples

```
set.seed(8)
d <- generate_dummy_exporisk(n = 12, p_micro = 5, p_metab = 6, p_expo = 3)
aligned <- align_omics_se(d$microbiome, d$metabolome, d$exposures, d$meta,
                          id_col = "sample_id", strict = TRUE)
se2 <- prep_omics_se(aligned)
se2
```

Index