Package 'path2path'

November 7, 2026

Type Package

Title Directional Pathway-by-Pathway Interaction Maps from Genome-Scale Perturb-Seq

Version 0.99.4

Description Builds a directional pathway-by-pathway interaction map from genome-scale Perturb-seq differential-expression statistics by two stacked runs of preranked gene set enrichment analysis: once per perturbation over genes, then once per readout pathway over perturbations. Entry (j,k) of the resulting matrix measures whether perturbations targeting pathway j's members produce a coordinated transcriptional response in pathway k. The package includes target exclusion, effective-size filters, overlap masking, Stage 2 p-values with false-discovery-rate control, and a whole-pipeline permutation calibration with generalized-Pareto tail estimation.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.4.0)

Imports fgsea, msigdbr, data.table, BiocParallel, withr, methods, stats, utils, grDevices, graphics

Suggests ggplot2, testthat (>= 3.0.0), knitr, rmarkdown, BiocStyle, SummarizedExperiment, R.utils

VignetteBuilder knitr

biocViews Software, GeneSetEnrichment, Pathways, CRISPR, SingleCell, DifferentialExpression, Transcriptomics, NetworkInference, GeneExpression, GeneTarget, KEGG, WholeGenome

URL https://github.com/jli-stat/path2path

BugReports https://github.com/jli-stat/path2path/issues

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Jun Li [aut, cre] (<https://orcid.org/0000-0003-4353-5761>)

Maintainer Jun Li <jun.li@nd.edu>

git_url https://git.bioconductor.org/packages/path2path

git_branch devel

git_last_commit 7d979ae

git_last_commit_date 2026-10-09

Repository Bioconductor 3.24

Date/Publication 2026-11-06

Contents

path2path-package . . . 2
as_path2path_matrix . . . 3
as_path2path_se . . . 4
bh_populated . . . 4
build_null_pool . . . 5
calibrate_pvalues . . . 6
dataset_tag . . . 7
demo_pathways . . . 8
demo_z . . . 8
jaccard_mask . . . 9
load_pathways . . . 9
order_by_activity . . . 10
path2path . . . 11
pathway_jaccard . . . 12
permute_input . . . 13
plot_calibration . . . 13
plot_entry . . . 14
plot_map . . . 15
read_input_matrix . . . 16
select_active . . . 17
stage1_nes . . . 18
stage2_matrix . . . 19
write_result . . . 21

Index **22**

`path2path-package` *path2path: directional pathway-by-pathway maps from Perturb-seq*

Description

The package converts a perturbation-by-gene matrix of signed differential-expression statistics into a directional pathway-by-pathway map using two successive preranked gene set enrichment analyses. It also provides pathway-overlap safeguards, two routes for significance assessment, plotting functions, and conversion to and from ‘SummarizedExperiment‘ objects.

Details

See ‘vignette("path2path")‘ for an introduction and worked example.

Author(s)

Jun Li

References

Subramanian A, et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences 102:15545-15550. doi:10.1073/pnas.0506580102.

Korotkevich G, Sukhov V, Sergushichev A (2019). Fast gene set enrichment analysis. bioRxiv. doi:10.1101/060012.

See Also

Useful links:

- https://github.com/jli-stat/path2path
- Report bugs at https://github.com/jli-stat/path2path/issues

---

`as_path2path_matrix` *Convert a SummarizedExperiment assay to a path2path input matrix.*

Description

A SummarizedExperiment stores features in rows and samples or perturbations in columns. path2path uses the transpose: perturbations in rows and measured genes in columns. This helper performs that conversion while retaining the assay dimnames.

Usage

```
as_path2path_matrix(x, assay = 1L)
```

Arguments

`x`	a 'SummarizedExperiment' with genes in rows and perturbations in columns.
`assay`	character(1) or integer(1): assay name or index passed to 'SummarizedExperiment::assay()'.

Value

a numeric perturbation-by-gene matrix.

Examples

```
if (requireNamespace("SummarizedExperiment", quietly = TRUE)) {
    data(demo_z)
    se <- SummarizedExperiment::SummarizedExperiment(
        assays = list(z = t(demo_z))
    )
    z <- as_path2path_matrix(se, "z")
    stopifnot(identical(z, demo_z))
}
```

as_path2path_se *Convert a path2path map to a SummarizedExperiment.*

Description

The returned object stores the map and its p-value matrices as assays, allowing the pathway-by-pathway result to be used with standard Bioconductor infrastructure. The Stage 1 matrix remains available in the original result object.

Usage

```
as_path2path_se(result)
```

Arguments

result	a list, a standard ‘path2path()‘ result containing ‘M‘.

Value

a ‘SummarizedExperiment‘ with pathways as rows and readout pathways as columns.

Examples

```
if (requireNamespace("SummarizedExperiment", quietly = TRUE)) {
    data(demo_z)
    data(demo_pathways)
    res <- path2path(
        demo_z, demo_pathways,
        stage1 = list(nperm = 100, min_size = 5),
        stage2 = list(min_pu = 5),
        BPPARAM = BiocParallel::SerialParam(), verbose = FALSE
    )
    se <- as_path2path_se(res)
    SummarizedExperiment::assayNames(se)
}
```

bh_populated *Benjamini-Hochberg over the populated (non-NA) cells of a matrix.*

Description

Exported utility: encodes the package's multiple-testing family rule (one BH family over all populated entries), for use on any p-value matrix produced by the package (e.g. negative-control stage2_matrix() output).

Usage

```
bh_populated(P)
```

Arguments

P	a numeric matrix of p-values (NA = unpopulated/masked entries).

Value

matrix of the same shape with BH-adjusted values on populated entries.

Examples

```
P <- matrix(c(0.001, 0.04, NA, 0.5), 2, 2)
bh_populated(P)
```

`build_null_pool` *Whole-pipeline permutation null calibration.*

Description

Runs the same path2path() pipeline on B permuted matrices and pools |M'| entries into a sorted null pool. Each null M' is saved to out_dir for reuse (per-pathway regrouping, tail fits).

Usage

```
build_null_pool(
  z,
  pathways,
  B = 10L,
  out_dir = NULL,
  config = NULL,
  seed_base = NULL,
  verbose = FALSE,
  ...
)
```

Arguments

z, pathways	the inputs of the observed run: 'z' a numeric matrix (perturbations x genes, rownames = perturbed gene symbols) and 'pathways' a named list of character vectors of gene symbols.
B	integer(1): number of outer permutations.
out_dir	'NULL' or character(1): optional directory persisting each null map (M_null_bNNN.rds), per-map BH discovery counts, and each null map's sorted nominal p-values; completed maps found there are reloaded instead of recomputed (resume). Use a separate directory for each input dataset, pathway collection, and pipeline configuration.
config	'NULL' or a list, a 'path2path()' result's '$config' component, so null runs repeat the observed configuration exactly.
seed_base	'NULL' or integer(1): if non-NULL, uses 'seed_base + b' for the input permutation in run b. For exact reproducibility of the GSEA calculations, also use seeded 'BPPARAM' objects in 'config' or '...'.
verbose	logical(1): report per-permutation progress messages (default FALSE).
...	additional path2path() arguments; these override entries of 'config' with the same name.

Value

a list with two components: ‘null_M‘, a length-‘B‘ list of pathway x pathway numeric matrices (one null map per permutation), and ‘pool_abs‘, a sorted numeric vector pooling the absolute values of all non-missing null-map entries.

Examples

```
data(demo_z)
data(demo_pathways)
cfg <- list(stage1 = list(nperm = 100, min_size = 5),
    stage2 = list(min_pu = 5),
    BPPARAM = BiocParallel::SerialParam())
pool <- build_null_pool(demo_z, demo_pathways, B = 2, seed_base = 1,
    stage1 = cfg$stage1, stage2 = cfg$stage2,
    BPPARAM = cfg$BPPARAM)
length(pool$pool_abs)
```

---

calibrate_pvalues *One-call calibrated p-values for an existing map.*

Description

Runs (or reuses) the outer-permutation calibration and returns the map's entries with pooled-empirical and GPD-calibrated p-values attached. Expensive: B full pipeline runs. ‘pool‘ may be supplied from a previous build_null_pool() to skip recomputation.

Usage

```
calibrate_pvalues(
  result,
  z = NULL,
  pathways = NULL,
  B = 50L,
  pool = NULL,
  q_thresh = 0.99,
  null_dir = NULL,
  seed_base = NULL,
  verbose = FALSE
)
```

Arguments

result	a list as returned by path2path() (uses result$M, and reuses result$config so the null runs repeat the observed run's configuration exactly; there is deliberately no way to override it here, since mismatched null runs would invalidate the calibration. For full control, call build_null_pool() directly and pass its pool via ‘pool‘).
z, pathways	the inputs that produced ‘result‘: ‘z‘ a numeric matrix (perturbations x genes) and ‘pathways‘ a named list of character vectors of gene symbols. Both may be ‘NULL‘ when a precomputed ‘pool‘ is supplied.
B	integer(1): number of outer permutations (ignored if ‘pool‘ given).

pool	'NULL' or numeric vector: optional sorted |M'| pool from build_null_pool()$pool_abs.
q_thresh	numeric(1) in (0, 1): GPD threshold quantile of the pool (default 0.99); a fit-adequacy guard lowers it stepwise (0.975, 0.95, 0.90) if the fitted tail's finite endpoint falls below the largest observed |M|.
null_dir	'NULL' or character(1), optional directory: each null map is saved there as M_null_bNNN.rds (persistence only; forwarded to build_null_pool's out_dir).
seed_base	'NULL' or integer(1): forwarded to `build_null_pool()`.
verbose	logical(1): forwarded to `build_null_pool()` (per-permutation progress messages; default `FALSE`).

Value

list with K x K matrices 'pval_gpd' (the calibrated p-values: identical to 'pval_pool' up to the pool's 'q_thresh' quantile, GPD tail extrapolation beyond it when an adequate fit is available), 'padj_gpd' (single BH family over the populated entries of 'pval_gpd', the calibrated analogue of result$padj), and 'pval_pool' (pooled empirical p-values, kept for diagnostics); plus 'gpd' (fit parameters) and 'pool_abs' (the pool, reusable via 'pool =').

References

Knijnenburg TA, Wessels LFA, Reinders MJT, Shmulevich I (2009). Fewer permutations, more accurate p-values. Bioinformatics 25:i161-i168. doi:10.1093/bioinformatics/btp211.

Examples

```
data(demo_z)
data(demo_pathways)
res <- path2path(demo_z, demo_pathways,
    stage1 = list(nperm = 100, min_size = 5),
    stage2 = list(min_pu = 5),
    BPPARAM = BiocParallel::SerialParam(), verbose = FALSE)
cal <- calibrate_pvalues(res, demo_z, demo_pathways, B = 2, seed_base = 1)
summary(as.vector(cal$pval_gpd))
```

---

dataset_tag *Dataset tag from a zscore_<tag>.csv[.gz] filename.*

Description

Dataset tag from a zscore_<tag>.csv[.gz] filename.

Usage

```
dataset_tag(path)
```

Arguments

path	character(1): a file path such as `zscore_myexp.csv.gz`.

Value

the tag string (`"myexp"`).

Examples

```
dataset_tag("zscore_k562_gwps.csv.gz")
```

`demo_pathways` *Synthetic demonstration pathway collection*

Description

Six disjoint 30-gene pathways over the `demo_z` gene vocabulary. Members of PATHWAY_A, PATHWAY_C, and PATHWAY_E are among the perturbed genes; see `demo_z`.

Usage

```
data(demo_pathways)
```

Format

named list of 6 character vectors.

`demo_z` *Synthetic demonstration input matrix*

Description

A small simulated Perturb-seq differential-expression matrix (120 perturbations x 400 genes) with two planted pathway-level relations: PATHWAY_A activates PATHWAY_B and PATHWAY_C suppresses PATHWAY_D (CRISPRi orientation), plus a strong on-target knockdown on every row's own gene. Generated by `inst/scripts/make_demo_data.R` (seeded).

Usage

```
data(demo_z)
```

Format

numeric matrix, 120 x 400; rownames = perturbed gene symbols, colnames = measured gene symbols.

jaccard_mask *Mask off-diagonal entries whose pathway pair overlaps above 'tau'.*

Description

High-overlap pairs (Reactome parent/child etc.) score mechanically — j's members ride k's signal because they ARE k's members — which is gene-set structure, not causal regulation. Diagonal is never masked.

Usage

```
jaccard_mask(M, pathways, tau = 0.5, jaccard = NULL)
```

Arguments

M	K x K numeric matrix to mask (dimnames = pathway names).
pathways	named list of character vectors of gene symbols.
tau	numeric(1) in [0, 1]: overlap threshold; entries with Jaccard > tau become NA.
jaccard	'NULL' or numeric matrix: optional precomputed pathway_jaccard() matrix.

Value

M with high-overlap off-diagonal entries set to NA.

Examples

```
data(demo_pathways)
M <- matrix(1, length(demo_pathways), length(demo_pathways),
    dimnames = list(names(demo_pathways), names(demo_pathways)))
sum(is.na(jaccard_mask(M, demo_pathways, tau = 0.1)))
```

load_pathways *Load an MSigDB collection as a named list of gene-symbol vectors.*

Description

Gene symbols are de-duplicated within each set. When 'size_filter' is TRUE the sets are restricted to 'min_size <= |S| <= max_size' measured on the *global* (genome) membership; the effective per-universe filters are applied later inside stage1_nes()/stage2_matrix(), which is where they matter statistically.

Usage

```
load_pathways(
  collection,
  min_size = 15L,
  max_size = 500L,
  species = "Homo sapiens"
)
```

Arguments

collection	character(1), one of "hallmark", "reactome", "kegg", "kegg_medicus", "kegg_legacy", "pid", "wikipathways". Note that "kegg" resolves to KEGG_MEDICUS (the current KEGG subcollection in MSigDB); use "kegg_legacy" for the classic 186-set collection.
min_size, max_size	integer(1) each: global size window applied when ‘size_filter‘ is active for the collection.
species	character(1): passed to msigdbr::msigdbr.

Value

named list of character vectors

References

Dolgalev I (2026). msigdbr: MSigDB Gene Sets for Multiple Organisms in a Tidy Data Format. R package.

Examples

```
hm <- load_pathways("hallmark")
length(hm)
```

---

order_by_activity *Pathway names ordered by number of significant entries (descending).*

Description

‘margin = "rows"‘ orders by outgoing (regulator) counts, ‘"cols"‘ by incoming (readout) counts. Returns ALL pathway names of that margin (zero-count ones last), so the result is a complete ordering usable as plot_map(order = ...).

Usage

```
order_by_activity(
  result,
  calibrated = NULL,
  padj_cutoff = 0.05,
  pval_cutoff = NULL,
  margin = c("rows", "cols"),
  include_diag = FALSE
)
```

Arguments

result	a list as returned by path2path().
calibrated	‘NULL‘ or a list as returned by calibrate_pvalues() (switches the cutoffs to Route B values).

padj_cutoff, pval_cutoff
: 'NULL' or numeric(1) each: significance cutoffs; non-NULL cutoffs conjoin, NULL disables one (at least one must be enabled).

margin
: character(1): order by outgoing ("rows") or incoming ("cols") significant counts.

include_diag
: logical(1): count the self-interaction diagonal.

Value

character vector: a complete ordering of that margin's names.

Examples

```
data(demo_z)
data(demo_pathways)
res <- path2path(demo_z, demo_pathways,
    stage1 = list(nperm = 100, min_size = 5),
    stage2 = list(min_pu = 5),
    BPPARAM = BiocParallel::SerialParam(), verbose = FALSE)
order_by_activity(res, padj_cutoff = 0.2)[1:3]
```

---

path2path *Full two-stage GSEA pipeline: z matrix -> pathway x pathway matrix.*

Description

Orchestrates stage1_nes() -> stage2_matrix() -> jaccard_mask() and global BH adjustment over populated, unmasked cells. This is the single canonical code path: production runs, calibration permutations, and sensitivity analyses must all go through this function.

Usage

```
path2path(
  z,
  pathways,
  stage1 = list(),
  stage2 = list(),
  jaccard_tau = 0.5,
  BPPARAM = BiocParallel::bpparam(),
  verbose = FALSE
)
```

Arguments

z
: numeric matrix, perturbations x genes (rownames = perturbed gene symbols, colnames = measured gene symbols).

pathways
: named list of character vectors of gene symbols.

stage1, stage2
: named lists of overrides forwarded to stage1_nes() and stage2_matrix().

jaccard_tau
: numeric(1) in [0, 1]: mask threshold; NA_real_ disables masking.

BPPARAM
: a BiocParallel parameter object forwarded to both stages (a BPPARAM entry inside stage1/stage2 takes precedence for that stage).

verbose
: logical(1): emit per-stage timing messages (default FALSE).

Value

list(nes, pval_stage1, M_raw, M, pval, padj, ..., config). ‘config‘ records the stage1/stage2 overrides and jaccard_tau of this call, so that calibrate_pvalues() can rerun the identical configuration on permuted data without the user re-passing it.

Reproducibility

`BiocParallel` draws worker seeds from entropy unless the backend carries an explicit seed, so `set.seed()` alone does not make a run reproducible. For exact reproducibility pass a seeded backend, e.g. `BPPARAM = BiocParallel::MulticoreParam(RNGseed = 1)` (or `SerialParam(RNGseed = 1)`).

Examples

```
data(demo_z)
data(demo_pathways)
res <- path2path(demo_z, demo_pathways,
    stage1 = list(nperm = 100, min_size = 5),
    stage2 = list(min_pu = 5),
    BPPARAM = BiocParallel::SerialParam(), verbose = FALSE)
res$M[1:3, 1:3]
sum(res$padj < 0.05, na.rm = TRUE)
```

---

pathway_jaccard *Pairwise Jaccard similarity of pathway gene sets (dense crossproduct).*

Description

Pairwise Jaccard similarity of pathway gene sets (dense crossproduct).

Usage

```
pathway_jaccard(pathways)
```

Arguments

`pathways`	named list of character vectors of gene symbols.

Value

K x K similarity matrix (diagonal set to 0; it is never masked).

Examples

```
data(demo_pathways)
J <- pathway_jaccard(demo_pathways)
max(J)
```

permute_input *Whole-matrix permutation: one global row and one global column permutation.*

Description

Preserves the marginal value distribution and within-row gene-gene correlation (the same column permutation is applied to every row) while breaking pathway membership, because the original labels stay attached to the permuted positions.

Usage

```
permute_input(z)
```

Arguments

z	perturbations x genes numeric matrix.

Value

the permuted matrix (labels retained in place, so every label sits atop another row/column's data).

Examples

```
data(demo_z)
set.seed(1)
zp <- permute_input(demo_z)
identical(dimnames(zp), dimnames(demo_z))
```

plot_calibration *Route B diagnostics: null-pool tail versus the fitted GPD.*

Description

Left: empirical survival curve of the null pool (log10 scale) with the fitted GPD tail overlaid beyond the threshold u. Right: QQ plot of the exceedances against fitted GPD quantiles. Both panels support the judgment the user makes before quoting a pval_gpd: does the tail fit?

Usage

```
plot_calibration(calibrated)
```

Arguments

calibrated	a list as returned by `calibrate_pvalues()` carrying `pool_abs` and `gpd`.

Value

invisible(NULL); draws two base-graphics panels.

Examples

```
pool <- sort(-log(seq(0.9999, 0.0001, length.out = 5000)))
M <- matrix(c(0.5, 1, 1.5, 2), 2,
    dimnames = list(c("A", "B"), c("A", "B")))
cal <- calibrate_pvalues(list(M = M), pool = pool)
plot_calibration(cal)
```

`plot_entry` *GSEA-style enrichment plot for one map entry: why is M[j, k] strong?*

Description

Shows the Stage 2 ranking (perturbations ranked by their Stage 1 NES on readout pathway k) with tick marks at the perturbations targeting members of pathway j, via fgsea::plotEnrichment(). The title reports the entry's NES and its p-value (Route B if 'calibrated' is given).

Usage

```
plot_entry(result, pathways, j, k, calibrated = NULL)
```

Arguments

`result`	a list as returned by `path2path()`.
`pathways`	named list of character vectors: the collection used to build 'result'.
`j, k`	character(1) each, pathway names: the perturbed (row) and readout (column) side.
`calibrated`	'NULL' or a list as returned by `calibrate_pvalues()`; switches the title to Route B values.

Value

a ggplot object.

Examples

```
data(demo_z)
data(demo_pathways)
res <- path2path(demo_z, demo_pathways,
    stage1 = list(nperm = 100, min_size = 5),
    stage2 = list(min_pu = 5),
    BPPARAM = BiocParallel::SerialParam(), verbose = FALSE)
if (requireNamespace("ggplot2", quietly = TRUE))
    plot_entry(res, demo_pathways, names(demo_pathways)[1],
        names(demo_pathways)[2])
```

plot_map *Heatmap of a path2path map with significance dots.*

Description

Processing pipeline (in this fixed priority): (1) 'rows'/'cols' subset; (2) all-NA rows/columns dropped if 'drop_empty_*'; (3) ordering – 'order' if given (a character vector, or a list(rows=, cols=) as returned invisibly by a previous plot_map call, so a second map can be drawn under the first one's ordering), else two-directional hierarchical clustering if 'cluster = TRUE', else matrix order. Dots mark entries passing the significance cutoffs (see .sig_matrix semantics: non-NULL cutoffs conjoin; 'calibrated' switches to Route B p-values). Masked/filtered entries are gray.

Usage

```
plot_map(
  result,
  calibrated = NULL,
  padj_cutoff = 0.05,
  pval_cutoff = NULL,
  rows = NULL,
  cols = NULL,
  drop_empty_rows = TRUE,
  drop_empty_cols = TRUE,
  order = NULL,
  cluster = TRUE,
  labels = NULL,
  zlim = c(-3, 3),
  main = NULL,
  colorbar = TRUE,
  dot_cex = NULL
)
```

Arguments

result	a list as returned by path2path().
calibrated	'NULL' or a list as returned by calibrate_pvalues() (switches the cutoffs to Route B values).
padj_cutoff, pval_cutoff	'NULL' or numeric(1) each: significance cutoffs; non-NULL cutoffs conjoin, NULL disables one (at least one must be enabled).
rows, cols	'NULL' or character vectors: optional name subsets applied before anything else.
drop_empty_rows, drop_empty_cols	logical(1) each: drop all-NA rows/columns.
order	'NULL', a character vector, or list(rows=, cols=) fixing the order.
cluster	logical(1): two-directional hierarchical clustering when no 'order'.
labels	'NULL' or logical(1): force axis labels on/off (default: on up to 60 pathways).
zlim	numeric(2): color range in NES units.
main	'NULL' or character(1): plot title.

colorbar	logical(1): draw the NES colorbar.
dot_cex	'NULL' or numeric(1): significance-dot size (default scales with map size).

Value

(invisible) list(rows, cols): the plotted ordering, reusable as the 'order' argument for another map.

Examples

```
data(demo_z)
data(demo_pathways)
res <- path2path(demo_z, demo_pathways,
    stage1 = list(nperm = 100, min_size = 5),
    stage2 = list(min_pu = 5),
    BPPARAM = BiocParallel::SerialParam(), verbose = FALSE)
plot_map(res, padj_cutoff = 0.2)
```

---

read_input_matrix *Read a z-score CSV (rows = perturbed gene symbols in first column).*

Description

Read a z-score CSV (rows = perturbed gene symbols in first column).

Usage

```
read_input_matrix(path)
```

Arguments

path	character(1): CSV or CSV.GZ file; first column = perturbed gene symbols, remaining columns = measured genes. Reading gzipped files requires the **R.utils** package (listed in Suggests).

Value

numeric matrix with dimnames.

Examples

```
data(demo_z)
f <- tempfile(fileext = ".csv")
write.csv(data.frame(perturbation = rownames(demo_z), demo_z,
    check.names = FALSE), f, row.names = FALSE)
z <- read_input_matrix(f)
identical(dim(z), dim(demo_z))
```

`select_active` *Pathways with at least 'min_sig' significant entries, as regulators and as readouts.*

Description

Counts significant entries per row (outgoing, pathway as regulator) and per column (incoming, pathway as readout) separately – in practice out-degrees and in-degrees differ by an order of magnitude, so a joint count would drown the readout side. The result plugs directly into plot_map(): 'plot_map(res, rows = sel$rows, cols = sel$cols)'.

Usage

```
select_active(
  result,
  calibrated = NULL,
  padj_cutoff = 0.05,
  pval_cutoff = NULL,
  min_sig = 1L,
  include_diag = FALSE
)
```

Arguments

`result`	a list as returned by `path2path()`.
`calibrated`	'NULL' or a list as returned by `calibrate_pvalues()` (switches the cutoffs to Route B values).
`padj_cutoff, pval_cutoff`	'NULL' or numeric(1) each: significance cutoffs; non-NULL cutoffs conjoin, NULL disables one (at least one must be enabled).
`min_sig`	integer(1): minimum number of significant entries.
`include_diag`	logical(1): count the self-interaction diagonal (default FALSE, matching the paper's off-diagonal focus; only applies to square maps).

Value

list(rows = character, cols = character)

Examples

```
data(demo_z)
data(demo_pathways)
res <- path2path(demo_z, demo_pathways,
    stage1 = list(nperm = 100, min_size = 5),
    stage2 = list(min_pu = 5),
    BPPARAM = BiocParallel::SerialParam(), verbose = FALSE)
select_active(res, padj_cutoff = 0.2)
```

stage1_nes *Stage 1: per-perturbation preranked GSEA.*

Description

For each row m of ‘z‘ (a perturbation), ranks the measured genes by the CRISPRi-negated statistic s = -z[m,] and scores every pathway with preranked GSEA. Returns the M x K NES matrix (and p-values for method = "fgsea").

Usage

```
stage1_nes(
  z,
  pathways,
  method = c("fgsea", "pooled"),
  exclude_target = TRUE,
  negate = TRUE,
  nperm = 1000L,
  min_size = 15L,
  max_size = 500L,
  b_prime = 1L,
  BPPARAM = BiocParallel::bpparam()
)
```

Arguments

z	numeric matrix, perturbations x genes; rownames = perturbed gene symbols, colnames = measured gene symbols. NAs are dropped per row.
pathways	named list of character vectors of gene symbols.
method	character(1): "fgsea" (fgseaSimple, per-perturbation permutations) or "pooled" (one pooled null per pathway across perturbations; no p-values).
exclude_target	logical(1): drop the perturbed gene from its own ranked list before scoring. This removes the tautological self-hit: the target is at the top of L_m by construction, giving every pathway that contains it a free hit. Default TRUE.
negate	logical(1): apply the CRISPRi loss-of-function sign flip (s = -z). Default TRUE; set FALSE only for gain-of-function perturbations.
nperm	integer(1), fgsea method only: permutations per ranked list for `fgsea::fgseaSimple`. The default 1000 estimates the NES accurately; downstream inference does not consume Stage 1 p-values.
min_size, max_size	integer(1) each: effective gene-set size window, applied to the intersection with each ranked list.
b_prime	integer(1), pooled method only: random null draws per perturbation pooled into the per-pathway null (1 draw x thousands of perturbations is already a large pool).
BPPARAM	a `BiocParallel` parameter object controlling parallelisation over perturbations; default `BiocParallel::bpparam()`.

Details

When ranked statistics contain ties, the function emits one summary warning for the stage rather than one warning per perturbation.

Value

list(nes = M x K matrix, pval = M x K matrix or NULL)

References

Korotkevich G, Sukhov V, Sergushichev A (2019). Fast gene set enrichment analysis. bioRxiv. doi:10.1101/060012.

Examples

```
data(demo_z)
data(demo_pathways)
s1 <- stage1_nes(demo_z, demo_pathways, nperm = 100,
    min_size = 5, BPPARAM = BiocParallel::SerialParam())
dim(s1$nes)
```

`stage2_matrix` *Stage 2: per-pathway-column GSEA in the perturbation universe.*

Description

For each column k of the Stage 1 NES matrix, ranks the perturbations by NES(., k) and asks, for every pathway j, whether the perturbations of S_j members cluster at either end. Pathways are restricted to the perturbation universe (rownames of ‘nes‘) and must retain at least ‘min_pu‘ members.

Usage

```
stage2_matrix(
  nes,
  pathways,
  method = c("fgsea", "pooled"),
  score_type = c("std", "both"),
  min_pu = 10L,
  min_col_size = 2L,
  max_size = 500L,
  min_valid = 20L,
  eps = 0,
  b_prime = 1L,
  BPPARAM = BiocParallel::bpparam()
)
```

Arguments

nes	numeric matrix, M x K Stage 1 NES values (rownames = perturbed gene symbols, colnames = pathway names).
pathways	named list of character vectors: the full pathway collection (gene symbols); restricted internally.
method	character(1): "fgsea" (fgseaMultilevel, p-values) or "pooled".
score_type	character(1): "std" (default GSEA two-sided score), or "both" to run scoreType "pos" and "neg" separately and return both extremes (M_pos / M_neg), which resolves bidirectionally-regulated pairs that cancel under "std".
min_pu	integer(1): minimum members of a set in the perturbation universe (rownames of ‘nes‘) for the set to be tested at all.
min_col_size	integer(1): minimum members of a set actually present in one column’s non-NA ranking for that entry to be scored. The universe filter ‘min_pu‘ is global; within a single column, NA entries can reduce a set’s effective size below ‘min_pu‘, and ‘min_col_size‘ bounds it from below (default 2, ruling out degenerate single-member scores). Raise it (e.g. to ‘min_pu‘) for a stricter per-entry guarantee.
max_size	integer(1): maximum effective set size within a column’s ranking.
min_valid	integer(1): minimum non-NA entries a column needs to be ranked at all.
eps	numeric(1): `fgsea::fgseaMultilevel` accuracy floor (0 = exact estimation of arbitrarily small p-values).
b_prime	integer(1), pooled method only: random null draws per column pooled into the per-pathway-row null.
BPPARAM	a `BiocParallel` parameter object controlling parallelisation over readout columns; default `BiocParallel::bpparam()`.

Details

When ranked statistics contain ties, the function emits one summary warning for the stage rather than one warning per readout column.

Value

list with K x K matrices: M, pval (std) or M_pos, M_neg, pval_pos, pval_neg (both). Rows cover all pathways; pathways failing the min_pu filter are all-NA rows.

Examples

```
data(demo_z)
data(demo_pathways)
s1 <- stage1_nes(demo_z, demo_pathways, nperm = 100, min_size = 5,
    BPPARAM = BiocParallel::SerialParam())
s2 <- stage2_matrix(s1$nes, demo_pathways, min_pu = 5,
    BPPARAM = BiocParallel::SerialParam())
s2$M[1:3, 1:3]
```

write_result *Write pipeline outputs with dataset/collection namespacing.*

Description

Files: nes_matrix_<ns>.csv.gz, pval_matrix_stage1_<ns>.csv.gz, M_matrix_<ns>_raw.csv, M_matrix_<ns>.csv, pval/padj_matrix_<ns>.csv (or *_pos/_neg variants for score_type = "both").

Usage

```
write_result(result, dir, tag, collection, method = "fgsea", calibrated = NULL)
```

Arguments

result	a list as returned by path2path().
dir	character(1): output directory.
tag, collection, method	character(1) each: components of the file namespace.
calibrated	'NULL' or a list as returned by calibrate_pvalues().

Details

If 'calibrated' (a calibrate_pvalues() result) is supplied, Route B outputs are written alongside: pval_gpd_matrix_<ns>.csv, pval_pool_matrix_<ns>.csv, gpd_params_<ns>.csv, and the null pool as null_pool_<ns>.rds (for reuse via calibrate_pvalues(pool =)).

Value

(invisible) the namespace string used in filenames.

Examples

```
data(demo_z)
data(demo_pathways)
res <- path2path(demo_z, demo_pathways,
    stage1 = list(nperm = 100, min_size = 5),
    stage2 = list(min_pu = 5),
    BPPARAM = BiocParallel::SerialParam(), verbose = FALSE)
d <- tempdir()
write_result(res, d, "demo", "demo6", method = "fgsea")
list.files(d, pattern = "demo")
```

Index

∗ **datasets**
    demo_pathways, 8
    demo_z, 8
∗ **internal**
    path2path-package, 2

as_path2path_matrix, 3
as_path2path_se, 4

bh_populated, 4
build_null_pool, 5

calibrate_pvalues, 6

dataset_tag, 7
demo_pathways, 8
demo_z, 8

jaccard_mask, 9

load_pathways, 9

order_by_activity, 10

path2path, 11
path2path-package, 2
pathway_jaccard, 12
permute_input, 13
plot_calibration, 13
plot_entry, 14
plot_map, 15

read_input_matrix, 16

select_active, 17
stage1_nes, 18
stage2_matrix, 19

write_result, 21