Package: easyLSEA (via r-universe)

June 16, 2026

Title Lipid Set Enrichment Analysis with Dual KS and 'fgsea' Engines

Version 0.2.1

Description Provides biology-aware lipid set enrichment analysis (LSEA) for lipidomics data using dual engines: the Kolmogorov-Smirnov test and the fast gene set enrichment algorithm from the 'fgsea' package. Annotates lipids into biological groups at three levels (lipid class, LIPID MAPS category, functional category) and tests for coordinated directional shifts between conditions. Includes fatty acid chain analysis with trend plots weighted by lipid abundance (Spearman rank correlation, configurable smoothing), wide-format chain position output (sn-1, sn-2, sn-3, sn-4), annotation confidence filtering, and export utilities for reproducible reporting in CSV, 'Excel', and PDF formats. Vignettes are available in English and Spanish. Methods are based on Subramanian et al. (2005) <doi:10.1073/pnas.0506580102> and Korotkevich et al. (2021) <doi:10.1101/060012>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

RoxygenNote 8.0.0

Imports ggplot2, withr

Suggests fgsea, knitr, openxlsx, rmarkdown, spelling, testthat (>= 3.0.0)

URL https://github.com/DavidGO464/easyLSEA

BugReports https://github.com/DavidGO464/easyLSEA/issues

LazyData true

Config/testthat/edition 3

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author David Guardamino Ojeda [aut, cre] (ORCID:
<https://orcid.org/0000-0002-3122-2218>)

Maintainer David Guardamino Ojeda <david.guardamino@gmail.com>

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

Date/Publication 2026-06-16 19:40:02 UTC

RemoteUrl https://github.com/cran/easyLSEA

RemoteRef HEAD

RemoteSha 70cad577df5710a0bd8c23bd2e6539902e485645

Contents

`annotate_lipid` *Annotate lipid names with LIPID MAPS classification*

Description

Parses lipid names in any format used by lipidomics software (LipidSearch, MS-DIAL, LipidView) and returns a structured data frame with LIPID MAPS canonical classification, chain-level metadata, and optional shorthand notation per Liebisch et al. (2020).

Usage

```
annotate_lipid(
  molecules,
  detail = c("compact", "standard", "full"),
  shorthand = FALSE,
  sn_confirmed = FALSE,
  lyso_explicit = FALSE,
```

```
  no_match = c("warn", "remove", "ignore"),
  sphingoid_default = "d"
)
```

Arguments

`molecules`	Character vector of lipid names to parse.
`detail`	Level of detail in the output table:
	`"compact"` Class, chains, totals. Equivalent to lipidR. Default.
	`"standard"` Adds LIPID MAPS category, class ID, and structural flags.
	`"full"` All columns. Recommended for joining with LMSD.
`shorthand`	Logical. If `TRUE`, adds `shorthand_lm` column with canonical shorthand per Liebisch et al. (2020). Default `FALSE`.
`sn_confirmed`	Logical. If `TRUE`, marks chains as sn-confirmed in shorthand output. Requires MS/MS-directed analysis. Default `FALSE`.
`lyso_explicit`	Logical. If `TRUE`, lyso-lipids include the empty sn-position: `LPC(18:1/0:0)` instead of `LPC(18:1)`. Default `FALSE`.
`no_match`	How to handle unparsed names: `"warn"` (default), `"remove"`, or `"ignore"`.
`sphingoid_default`	Default sphingoid base prefix for sphingolipids without explicit prefix. `"d"` = dihydroxy (mammalian default). Use `NA` for non-mammalian data.

Value

A data frame with one row per unique lipid name. Key columns include `Class`, `lm_category`, `lm_class_id`, `annotation_level`, `is_ether`, `is_plasmalogen`, `is_istd`, `sphingoid_prefix`, `total_cl`, `total_cs`, and optionally `shorthand_lm`.

References

Liebisch G et al. Update on LIPID MAPS classification, nomenclature, and shorthand notation for MS-derived lipid structures. *J Lipid Res.* 2020;61(12):1539-1555. doi:10.1194/jlr.S120001025

Conroy MJ et al. LIPID MAPS: update to databases and tools for the lipidomics community. *Nucleic Acids Res.* 2024;52(D1):D1677-D1682. doi:10.1093/nar/gkad896

Examples

```
lipids <- c("PC 16:0/18:1", "PC O-18:1/20:4", "Cer d18:1/16:0",
            "TG(16:0/18:1/18:1)", "Lyso PE 18:1(d7)",
            "plasmenylPE (16:0/18:1)", "Sa1P d 18:0",
            "WE 16:0/18:1", "CoA 16:0",
            "15-HETE", "PGE2", "LTB4", "Resolvin D1", "12(13)-EpOME")

annotate_lipid(lipids)
annotate_lipid(lipids, detail = "standard")
annotate_lipid(lipids, detail = "full", shorthand = TRUE)
```

annotate_lipids *Annotate lipid names with class and category information*

Description

Assigns lipid class (e.g. PC, TG, Cer), full class name, LIPID MAPS structural category, and functional category to each lipid in `data`. Returns the input data.frame with annotation columns appended, ready for use in `run_lsea` and `parse_lipid_chains`.

Usage

```
annotate_lipids(
  data,
  lipid_col = "LipidName",
  shorthand_col = "Shorthand",
  method = c("internal", "lipidAnnotator"),
  verbose = TRUE
)
```

Arguments

`data`	A `data.frame` with at least one column of lipid names.
`lipid_col`	Character(1). Name of the column containing lipid identifiers. Default: `"LipidName"`.
`shorthand_col`	Character(1) or `NULL`. Optional column with shorthand notation used as the primary annotation source when present (more standardised than common names). Falls back to `lipid_col` if `NULL` or column not found. Default: `"Shorthand"`.
`method`	Character(1). Annotation method:
	`"internal"` Regex-based hierarchical classifier validated against LIPID MAPS nomenclature. No external dependencies. Covers all major lipid classes in untargeted lipidomics (GPL, SL, GL, FA, ST, acylcarnitines, oxylipins, bile acids). Default.
	`"lipidAnnotator"` Uses the **lipidAnnotator** package (available on GitHub, archived on Zenodo). Must be installed separately. Provides enhanced structural annotation.
`verbose`	Logical(1). Print annotation summary (class distribution and count of unclassified lipids). Default: `TRUE`.

Value

The input `data.frame` with five columns appended:

`LipidClass` Abbreviated class (e.g. "PC", "TG", "Cer").

`LipidClass_Full` Descriptive class name (e.g. "Ceramide", "Ether-PC").

`LipidCategory_LMAPS` LIPID MAPS structural category (e.g. "Glycerophospholipids", "Sphingolipids").

`LipidCategory_functional` Functional category, with Oxylipins and Bile Acids as standalone groups rather than nested under Fatty Acyls.

`LipidCategory` Simplified category for plotting: same as `LipidCategory_functional` except Saccharolipids are shown as "Glycolipids".

Lipids that cannot be classified receive `LipidClass = "Unknown"`.

See Also

`run_lsea`, `parse_lipid_chains`

Examples

```
df <- data.frame(
  LipidName = c("PC 36:2", "TG(54:3)", "SM d18:1/16:0",
                "Cer(d18:1/24:0)", "LPC 18:0", "CE 18:1"),
  logFC     = c(1.2, -0.8, 0.5, -1.1, 0.3, 0.9),
  stringsAsFactors = FALSE
)

annotated <- annotate_lipids(df)
annotated[, c("LipidName", "LipidClass", "LipidCategory")]
```

`default_chain_config` *Default chain analysis class configuration*

Description

Returns the default list that maps lipid classes to their parsing strategy. Pass the output of this function as the `cls_config` argument of `parse_lipid_chains()` to override individual entries.

Usage

```
default_chain_config()
```

Value

Named list with elements `sn2`, `nacyl`, `long`, `single`, and `excl`.

easyLSEA *Lipid Set Enrichment Analysis — full pipeline*

Description

One-call interface to the complete easyLSEA workflow: lipid annotation, KS and/or fgsea enrichment across three biological levels (class, LIPID MAPS category, functional category), and fatty acid chain analysis. Returns a structured `easyLSEA_result` object that can be plotted and exported.

Usage

```
easyLSEA(
  data,
  lipid_col = "LipidName",
  fc_col = "logFC",
  pval_col = "P.Value",
  case_lbl = "Case",
  ref_lbl = "Reference",
  engine = c("both", "ks", "fgsea"),
  annotator = c("internal", "lipidAnnotator"),
  run_chains = TRUE,
  min_rank = "E",
  group_cols = NULL,
  min_n = 3L,
  n_perm = 2000L,
  fgsea_nperm = 10000L,
  plots = TRUE,
  bubble_label = c("FDR", "DS", "NES", "n"),
  output = c("combined", "separate"),
  seed = 42L,
  verbose = TRUE
)
```

Arguments

`data`	A `data.frame` with at least a lipid name column and a numeric fold-change column. Additional columns (p-values, confidence ranks, abundance) are used when present.
`lipid_col`	Character(1). Name of the lipid identifier column. Default: `"LipidName"`.
`fc_col`	Character(1). Name of the log2 fold-change column. Default: `"logFC"`.
`pval_col`	Character(1) or `NULL`. Name of the raw p-value column. Used to compute the pi-value rank metric for fgsea. Default: `"P.Value"`.
`case_lbl`	Character(1). Label for the case group, used in output tables and plot titles. Default: `"Case"`.
`ref_lbl`	Character(1). Label for the reference group. Default: `"Reference"`.

engine	Character(1). Enrichment engine: "ks", "fgsea", or "both". Default: "both".
annotator	Character(1). Lipid annotation method: "internal" (built-in regex classifier) or "lipidAnnotator" (requires optional package). Default: "internal".
run_chains	Logical(1). Whether to run fatty acid chain analysis in addition to LSEA. Default: TRUE.
min_rank	Character(1). Minimum confidence rank for chain analysis. Ranks are ordered A > B > C > D > E > P. Lipids with rank lower than min_rank (and rank "P" or NA) are excluded from chain parsing. Default: "E" (include all except P and NA). Only used when run_chains = TRUE and a rank column is present.
group_cols	Character vector. Grouping columns to test in LSEA. If NULL (default), uses the three standard levels: LipidClass, LipidCategory_LMAPS, LipidCategory_functional.
min_n	Integer(1). Minimum set size to test. Default: 3L.
n_perm	Integer(1). KS permutations for DS_perm_pval. Default: 2000L.
fgsea_nperm	Integer(1). fgsea Monte Carlo permutations. Default: 10000L.
plots	Logical(1). Whether to generate **ggplot2** objects. Set to FALSE to skip plotting and reduce runtime. Default: TRUE.
bubble_label	Character vector. Which statistics to show next to each bubble in the LSEA bubble plots. Any subset of "FDR", "DS" (KS only), "NES" (fgsea only), and "n". Use fewer to shorten labels. Default: all four.
output	Character(1). Return format when both modules run: "combined" returns a single easyLSEA_result; "separate" returns a named list with elements lsea and chains. Default: "combined".
seed	Integer(1) or NULL. RNG seed for reproducibility. Passed to with_seed — does not alter the user's global RNG state. Default: 42L.
verbose	Logical(1). Print progress messages. Default: TRUE.

Value

An object of class easyLSEA_result: a named list with five slots.

$meta Named list: call, date, labels, engine, counts.

$lsea Named list: results (data.frame with KS and/or fgsea statistics), combined (merged table with Convergence column).

$chains Named list: parsed and summary from parse_lipid_chains, or NULL if run_chains = FALSE.

$plots Named list of ggplot objects, or NULL if plots = FALSE.

$input Named list: data (annotated input), group_cols.

When output = "separate", returns list(lsea = ..., chains = ...) instead.

See Also

annotate_lipids for standalone annotation, run_lsea for the enrichment engine, parse_lipid_chains for chain analysis, plot_lsea, plot_chains, export_lsea() to save results.

Examples

```
data("lipid_example", package = "easyLSEA")

result <- easyLSEA(
  data      = lipid_example,
  lipid_col = "LipidName",
  fc_col    = "logFC",
  case_lbl  = "NASH",
  ref_lbl   = "Control",
  engine    = "ks",
  plots     = FALSE
)

print(result)
head(result$lsea$results)
```

`export_lsea` *Export easyLSEA results to disk*

Description

Saves the contents of an easyLSEA result object to a timestamped output folder. Supported formats: CSV tables, a multi-sheet Excel workbook, PDF or PNG plots, and a standalone HTML report. Any combination of formats can be requested in a single call.

Usage

```
export_lsea(
  result,
  dir,
  prefix = "easyLSEA",
  format = c("csv", "excel", "pdf"),
  overwrite = FALSE,
  plot_width = NULL,
  plot_height = NULL,
  plot_dpi = 300L,
  verbose = TRUE
)
```

Arguments

`result`	An `easyLSEA_result` object returned by easyLSEA, or a named list with elements `lsea` and/or `chains` (output of `output = "separate"`).
`dir`	Character(1). Base directory where the output folder will be created. Required: there is no default, so the function never writes to the working directory, the package directory, or the user's home filespace unless the caller explicitly provides a location. For examples, tests, or throwaway output, pass `tempdir()`.

prefix	Character(1). Prefix for the output folder name. The folder is named <prefix>_<case>_vs_<ref>_<YYYY when comparison labels are available, otherwise <prefix>_<YYYY-MM-DD_HHMM>/. Default: "easyLSEA".
format	Character vector. One or more of "csv", "excel", "pdf", "png", "html". Default: c("csv", "excel", "pdf").
overwrite	Logical(1). If TRUE, an existing output folder with the same name is overwritten. Default: FALSE.
plot_width	Numeric(1) or NULL. Plot width in inches. If NULL (default), width is auto-sized based on the number of lipid sets in each plot. Pass a number to override for all plots.
plot_height	Numeric(1) or NULL. Plot height in inches. If NULL (default), height is auto-sized per plot type. Pass a number to override for all plots.
plot_dpi	Integer(1). Resolution for PNG output. Default: 300L.
verbose	Logical(1). Print progress messages. Default: TRUE.

Details

Output folder structure:

```
<prefix>_<YYYY-MM-DD>/
  tables/
    lsea_results_ks.csv
    lsea_results_fgsea.csv
    lsea_combined.csv
    chain_results.csv
    chain_parsed.csv
    chain_wide.csv
  plots/
    lsea/
      bubble_ks.pdf
      bubble_fgsea.pdf
    chains/
      tile/
        tile_PC.pdf
        tile_TG.pdf  ...
      trend/
        trend_length_PC.pdf
        trend_unsat_PC.pdf  ...
  results.xlsx
  report.html
```

Dependencies for optional formats:
Excel export requires **openxlsx** (`install.packages("openxlsx")`). HTML export requires **rmarkdown** and **knitr**.

Value

Invisibly returns a named character vector of all file paths created. Useful for programmatic use or verification.

See Also

easyLSEA, run_lsea, parse_lipid_chains

Examples

```
data("lipid_example", package = "easyLSEA")

result <- suppressWarnings(suppressMessages(easyLSEA(
  data      = lipid_example,
  engine    = "ks",
  n_perm    = 100L,
  plots     = FALSE,
  verbose   = FALSE
)))


# Export CSV and PDF to a temporary folder
paths <- export_lsea(result, dir = tempdir(), format = c("csv", "pdf"))
paths
```

---

lipid_example *Example lipidomics dataset*

Description

A synthetic dataset of 200 lipid species simulating a case vs control lipidomics comparison, with known enrichment patterns built in: PC and PE species are enriched in the case group, TG species are depleted. Used in package examples and tests.

Usage

```
lipid_example
```

Format

A `data.frame` with 200 rows and 6 columns:

LipidName Character. Lipid identifier in shorthand notation (e.g. "PC 36:2").

LipidClass Character. Pre-assigned lipid class abbreviation.

logFC Numeric. Log2 fold change (case / control).

P.Value Numeric. Raw p-value from simulated differential analysis.

adj.P.Val Numeric. Benjamini-Hochberg adjusted p-value.

sig Integer. 1 if adj.P.Val < 0.05 and |logFC| > log2(1.25), 0 otherwise.

Source

Simulated data. See `data-raw/lipid_example.R` for the generation script. Seed: 2026.

`parse_lipid_chains` *Parse acyl chain composition from a lipidomics data.frame*

Description

Applies biology-aware chain parsing to each lipid in `data`, routing each species to the appropriate parser based on its lipid class: sn-2 (PC, PE, PE O), N-acyl (SM, Cer, HexCer, GlcCer, Hex2Cer, Hex3Cer), long-format (TG, DG, PS, PG, PA, PI, CL), single-chain (LPC, LPE, LPI, LPG, LPA, LPS, CAR, FFA, FA, CE), or excluded.

Usage

```
parse_lipid_chains(
  data,
  lipid_col = "LipidName",
  class_col = "LipidClass",
  shorthand_col = "Shorthand",
  rank_col = "Confidence_rank",
  min_rank = "E",
  cls_config = default_chain_config()
)
```

Arguments

`data`	A `data.frame` with at least the columns specified by `lipid_col` and `class_col`. Typically the output of `annotate_lipids()`.
`lipid_col`	Character(1). Name of the lipid identifier column. Default: `"LipidName"`.
`class_col`	Character(1). Name of the lipid class column (must contain abbreviated class names such as "PC", "TG", "SM"). Default: `"LipidClass"`.
`shorthand_col`	Character(1) or `NULL`. Name of an optional shorthand column used as fallback for sn-2 and single-chain parsing when the primary name is a common name. Default: `"Shorthand"`.
`rank_col`	Character(1) or `NULL`. Name of a confidence-rank column. Rows with rank `"P"` or `NA` are always excluded. Set to `NULL` to skip rank filtering entirely. Default: `"Confidence_rank"`.
`min_rank`	Character(1). Minimum confidence rank to include in analysis. Ranks are ordered `A > B > C > D > E > P`. Setting `min_rank = "B"` includes only ranks A and B, excluding C, D, E and P. Default: `"E"` (include all except P and NA).
`cls_config`	Named list from `default_chain_config`. Override individual elements to change class routing.

Value

A named list with two elements:

`parsed` Long-format `data.frame` with one row per chain observation. Contains all columns from `data` plus chain fields (`analysis_chain_cl`, `analysis_chain_cs`, `chain_type`, etc.).

`summary` Per-lipid parsing log `data.frame` with columns `LipidName`, `LipidClass`, `Confidence_rank`, `status`, and `chain_type`.

See Also

`default_chain_config`, `plot_chains()`

Examples

```
data("lipid_example", package = "easyLSEA")
annotated <- annotate_lipids(lipid_example)
chains <- parse_lipid_chains(annotated)
head(chains$parsed)
head(chains$summary)
```

---

plot_chains *Generate chain analysis plots*

Description

Produces tile and trend plots for each lipid class with sufficient chain observations. Returns a named list of `ggplot` objects; does not write files. Use `export_lsea()` to save.

Usage

```
plot_chains(
  chains_result,
  case_lbl = "Case",
  ref_lbl = "Reference",
  fdr_thresh = 0.05,
  min_n_tile = 4L,
  min_n_trend = 5L,
  smooth_method = c("loess", "lm"),
  smooth_span = 0.75,
  smooth_weighted = TRUE,
  smooth_se = TRUE,
  show_points = TRUE,
  tile_label = c("both", "n", "sig", "none"),
  trend_test = c("spearman", "lm", "none"),
  trend_x_step_length = 2L,
  trend_x_step_unsat = 1L
)
```

Arguments

chains_result	Named list returned by parse_lipid_chains.
case_lbl	Character(1). Label for the case group. Default: "Case".
ref_lbl	Character(1). Label for the reference group. Default: "Reference".
fdr_thresh	Numeric(1). FDR threshold to colour individual lipid points in trend plots (red = FDR sig, grey = NS) and to label significant counts in tile cells. Default: 0.05.
min_n_tile	Integer(1). Minimum chain observations per class to produce a tile plot. Default: 4L.
min_n_trend	Integer(1). Minimum chain observations per class to produce trend plots. Default: 5L.
smooth_method	Character(1). Smoothing method for trend plots. "loess" (default) fits a local polynomial; "lm" fits a global linear model. Use "lm" for small datasets or when a monotone trend is expected a priori.
smooth_span	Numeric(1). Span for loess smoothing (only used when smooth_method = "loess"). Smaller values (e.g. 0.4) produce a more flexible curve; larger values (e.g. 0.9) produce a smoother curve. Default: 0.75. A warning is issued when smooth_span < 0.5 and fewer than 10 observations are available, as this combination risks overfitting.
smooth_weighted	Logical(1). If TRUE (default), the smoothing curve is weighted by the number of chain observations per x-axis position, giving more influence to well-represented chain lengths/unsaturations. Mathematically more appropriate than unweighted loess when observation counts are unequal across positions.
smooth_se	Logical(1). Whether to display the 95\ interval ribbon around the smoothing curve. Default: TRUE.
show_points	Logical(1). Whether to display individual lipid points in trend plots, coloured by FDR significance. Default: TRUE. Set to FALSE to show only the smoothing curve (cleaner for classes with many lipids).
tile_label	Character(1). What to display inside each tile cell: "both" (default) shows total and significant lipid counts; "n" shows only the total; "sig" shows only significant; "none" shows no text.
trend_test	Character(1). Statistical test to annotate on trend plots. "spearman" (default) computes Spearman rank correlation between chain position (length or unsaturation) and logFC across individual lipids, reporting ρ and p-value. "lm" fits a weighted linear regression (weighted by n observations per position) and reports the slope β and p-value. "none" shows no statistical annotation. Note: these tests are computed on the individual lipid observations, not on the smoothed curve.
trend_x_step_length	Integer(1) or NULL. Step size for x-axis tick marks in chain length trend plots. Default: 2L (every 2 carbons), suitable for the typical range of 8–36 carbons. Use 1L for fine-grained resolution or 4L for very wide ranges. When NULL, ggplot2 chooses breaks automatically.

trend_x_step_unsat
: Integer(1) or NULL. Step size for x-axis tick marks in unsaturation trend plots. Default: 1L (every double bond), suitable for the typical range of 0–8. When NULL, ggplot2 chooses breaks automatically.

Value

Named list of ggplot objects with elements tile_<CLASS>, trend_length_<CLASS>, trend_unsat_<CLASS>.

See Also

parse_lipid_chains, export_lsea()

---

plot_distribution *Distribution enrichment boxplot per lipid set*

Description

Produces a boxplot of logFC distributions for each lipid set, with jittered individual lipid points, FDR/DS/NES labels for significant sets, and red borders for significant sets. When engine = "both" (KS + fgsea), fill colour encodes convergence (KS only, fgsea only, or KS+fgsea).

Usage

```
plot_distribution(
  data,
  lsea_result,
  group_col,
  fc_col = "logFC",
  case_lbl = "Case",
  ref_lbl = "Control",
  fdr_thresh = 0.05,
  min_n = 3L,
  sig_only = FALSE,
  label_angle = 0
)
```

Arguments

data	A data.frame with at least fc_col and the grouping column (e.g. LipidClass).
lsea_result	A named list as returned by run_lsea, with elements ks, fgsea, and/or combined.
group_col	Character(1). Grouping column name (e.g. "LipidClass").
fc_col	Character(1). Column with log fold-change values. Default: "logFC".
case_lbl	Character(1). Label for the case group. Default: "Case".
ref_lbl	Character(1). Label for the reference group. Default: "Control".
fdr_thresh	Numeric(1). FDR threshold for significance. Default: 0.05.

min_n	Integer(1). Minimum number of lipids per set to include. Default: 3L.
sig_only	Logical(1). If TRUE, show only significant sets. Default: FALSE.
label_angle	Numeric(1). Angle for FDR labels. 0 = horizontal (default); 90 = vertical (useful when many groups).

Value

A ggplot object, or NULL if no groups pass min_n.

---

plot_lsea *Generate LSEA enrichment plots*

Description

Produces bubble, barplot, and running sum plots from a run_lsea() result. Returns a named list of ggplot objects.

Usage

```
plot_lsea(
  lsea_result,
  which = c("bubble_ks", "bubble_fgsea", "bubble_combined", "barplot", "running_sum"),
  fdr_thresh = 0.05,
  case_lbl = "Case",
  ref_lbl = "Reference",
  bubble_label = c("FDR", "DS", "NES", "n")
)
```

Arguments

lsea_result	Named list returned by run_lsea.
which	Character vector. Which plots to generate: "bubble_ks", "bubble_fgsea", "bubble_combined", "barplot", "running_sum". Default: all.
fdr_thresh	Numeric(1). Significance threshold for highlighting. Default: 0.05.
case_lbl	Character(1). Case label for plot annotations.
ref_lbl	Character(1). Reference label for plot annotations.
bubble_label	Character vector. Which statistics to display next to each bubble. Any subset of "FDR", "DS" (KS plots only), "NES" (fgsea plots only), and "n". Default: all four.

Value

Named list of ggplot objects.

See Also

run_lsea, export_lsea()

print.easyLSEA_result *Print method for easyLSEA_result*

Description

Print method for easyLSEA_result

Usage

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

Arguments

x	An easyLSEA_result object.
...	Ignored.

Value

Invisibly returns the input easyLSEA_result object (x). Called for its side effect of printing a formatted summary of the enrichment results to the console.

run_lsea *Lipid Set Enrichment Analysis*

Description

Runs KS-based LSEA, fgsea, or both for each grouping level in group_cols and returns a tidy data.frame with enrichment statistics.

Usage

```
run_lsea(
  data,
 group_cols = c("LipidClass", "LipidCategory_LMAPS", "LipidCategory_functional"),
  fc_col = "logFC",
  pval_col = "P.Value",
  lipid_id_col = NULL,
  case_lbl = "Case",
  ref_lbl = "Reference",
  engine = c("both", "ks", "fgsea"),
  fgsea_rank = c("pi_value", "logFC", "t_stat"),
  min_n = 3L,
  n_perm = 2000L,
  fgsea_nperm = 10000L,
```

```
  fgsea_eps = 0,
  seed = 42L,
  verbose = TRUE
)
```

Arguments

data	A data.frame with at minimum a lipid identifier column, a log2 fold-change column, and one or more grouping columns (e.g. LipidClass).
group_cols	Character vector. Names of grouping columns to test. Each column defines one level of analysis (e.g. class, LIPID MAPS category, functional category). Default: c("LipidClass", "LipidCategory_LMAPS", "LipidCategory_functional").
fc_col	Character(1). Log2 fold-change column. Default: "logFC".
pval_col	Character(1) or NULL. Raw p-value column used to compute the pi-value rank metric. If NULL, logFC is used as the fgsea rank metric. Default: "P.Value".
lipid_id_col	Character(1) or NULL. Column with unique lipid identifiers. If NULL, auto-detected from common column names.
case_lbl	Character(1). Case group label. Default: "Case".
ref_lbl	Character(1). Reference group label. Default: "Reference".
engine	Character(1). Enrichment engine: "ks", "fgsea", or "both". Default: "both".
fgsea_rank	Character(1). Rank metric for fgsea: "pi_value", "logFC", or "t_stat". Default: "pi_value".
min_n	Integer(1). Minimum set size to test. Default: 3L.
n_perm	Integer(1). KS permutations for DS_perm_pval. Default: 2000L.
fgsea_nperm	Integer(1). fgsea Monte Carlo permutations. Default: 10000L.
fgsea_eps	Numeric(1). fgsea epsilon (0 = reduce approximation error). Default: 0.
seed	Integer(1) or NULL. RNG seed passed to withr::with_seed() – does not alter the user's global RNG state. Default: 42L.
verbose	Logical(1). Print progress messages. Default: TRUE.

Value

A named list with elements:

ks data.frame of KS results (or NULL if engine = "fgsea").

fgsea data.frame of fgsea results (or NULL if engine = "ks" or fgsea is not installed).

combined data.frame merging both engines by Group and Level, including a Convergence column.

References

Korotkevich G, Sukhov V, Budin N, Shpak B, Artyomov MN, Sergushichev A (2021). Fast gene set enrichment analysis. *bioRxiv*. doi:10.1101/060012

Xiao Y, Hsiao TH, Suresh U, Chen HI, Wu X, Wolf SE, Chen Y (2014). A novel significance score for gene selection and ranking. *Bioinformatics*, 30(6), 801–807. doi:10.1093/bioinformatics/btr671

See Also

annotate_lipids(), plot_lsea, export_lsea()

Examples

```
data("lipid_example", package = "easyLSEA")
annotated <- annotate_lipids(lipid_example)

result <- run_lsea(
  data      = annotated,
  fc_col    = "logFC",
  engine    = "ks",
  case_lbl  = "NASH",
  ref_lbl   = "Control",
  n_perm    = 100L
)

head(result$ks)
```

---

summary.easyLSEA_result

Summary method for easyLSEA_result

Description

Summary method for easyLSEA_result

Usage

```
## S3 method for class 'easyLSEA_result'
summary(object, padj_cutoff = 0.05, ...)
```

Arguments

object	An easyLSEA_result object.
padj_cutoff	Numeric(1). FDR threshold for significant sets. Default: 0.05.
...	Ignored.

Value

Invisibly returns the input easyLSEA_result object (object). Called for its side effect of printing a summary table of the significant lipid sets to the console.

Index