Package 'CONCERTDR'

November 12, 2026

Type Package

Title Drug Response Data Analysis Using CMap Database

Version 0.99.6

Description Provides a complete pipeline for drug repurposing analysis using the Connectivity Map (CMap) L1000 database. Functionality includes database subsetting, reference profile extraction from GCTX files, signature matching via Kolmogorov-Smirnov, GSEA-based, cosine similarity, and other methods, result annotation against CMap compound metadata, z-score matrix extraction, and publication-quality clustermap visualization via ComplexHeatmap. Connectivity scoring methods (KS, GSEA, XCos, XSum, Zhang) are implemented internally based on the algorithms described in Lamb et al. (2006), Subramanian et al. (2005), Cheng et al. (2014), and Zhang & Gant (2008), removing the need for external GitHub-only dependencies. CamSum, a correlation-adjusted sum score with an analytic p-value, is also provided.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/DavidsonGroup/CONCERTDR

BugReports https://github.com/DavidsonGroup/CONCERTDR/issues

Depends R (>= 4.5.0)

Imports data.table, matrixStats, rhdf5, stats, utils

Suggests knitr, rmarkdown, testthat, BiocStyle, ComplexHeatmap, circlize, ggplot2, RecordLinkage

VignetteBuilder knitr

Config/testthat/edition 3

biocViews GeneExpression, Pharmacogenomics, Software, Transcriptomics, DataImport

Config/roxygen2/version 8.1.0

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

git_branch devel

git_last_commit 78e3770

git_last_commit_date 2026-11-09

Repository Bioconductor 3.24

Date/Publication 2026-11-12

Author Hengxin Pan [aut, cre] (ORCID: <https://orcid.org/0009-0004-1274-1612>),
Chin Wee Tan [aut, ths] (ORCID:
<https://orcid.org/0000-0001-9695-7218>),
Gad Abraham [aut, ths],
Nadia Davidson [aut, ths] (ORCID:
<https://orcid.org/0000-0002-8461-7467>)

Maintainer Hengxin Pan <pan.h@wehi.edu.au>

Contents

.add_pert_kind_and_mode . . . 4
.apply_filters . . . 5
.apply_score . . . 5
.as_numeric . . . 6
.barcode_fig_size . . . 6
.barcode_gp . . . 7
.barcode_panel . . . 7
.barcode_style . . . 8
.build_drug_context_summary . . . 9
.build_tech_view . . . 9
.build_wetlab_drug_view . . . 10
.build_wetlab_gene_view . . . 10
.camsum_blocks . . . 11
.camsum_cached . . . 11
.camsum_col_sd . . . 12
.camsum_file_key . . . 12
.camsum_gctx_columns . . . 13
.camsum_option_file . . . 13
.camsum_pert_types . . . 14
.camsum_read_meta_file . . . 14
.camsum_rho . . . 15
.camsum_rho_gctx . . . 15
.camsum_rho_stream . . . 16
.camsum_rows . . . 16
.camsum_signed . . . 17
.camsum_warn_local_rho . . . 17
.check_topN . . . 18
.combine_up_down . . . 18
.count_pvalues . . . 19
.dedup_drug_rows . . . 19
.dedup_geneinfo . . . 20
.derive_annotations . . . 20
.direction_ordered_genes . . . 21
.draw_barcode . . . 21
.fast_chunk_size . . . 22
.fast_draws . . . 22
.fast_gsea0 . . . 23
.fast_gsea0_es . . . 23
.fast_gseaW . . . 24
.fast_gseaW_es . . . 24
.fast_ks . . . 25

.fast_ks_es . . . 25
.fast_permutation_test . . . 26
.fast_positions . . . 26
.fast_shared_null_test . . . 27
.fast_sorted_set . . . 27
.fast_up_down_test . . . 28
.fast_xcos . . . 28
.fast_xsum . . . 29
.fast_zhang . . . 29
.fill_name_from_iname . . . 30
.filter_interactively . . . 30
.filter_siginfo . . . 31
.first_existing_file . . . 31
.gctx_ids . . . 32
.gctx_read_matrix . . . 32
.is_blank . . . 33
.list_column_values . . . 33
.load_signature . . . 34
.matrix_to_extreme_list . . . 34
.matrix_to_name_ranked_list . . . 35
.matrix_to_signed_rank_list . . . 35
.matrix_to_value_ranked_list . . . 36
.merge_compoundinfo . . . 36
.merge_siginfo . . . 37
.message_column_summary . . . 37
.message_rule . . . 38
.order_by_score . . . 38
.parse_compound_context . . . 39
.permutation_pvalues . . . 39
.permutation_test . . . 40
.permutation_test_up_down . . . 40
.pert_id_missing . . . 41
.print_filter_summary . . . 42
.print_reuse_filters . . . 42
.prompt_column_filter . . . 43
.rank_by_direction . . . 43
.read_cmap_table . . . 44
.read_gene_map . . . 44
.read_siginfo . . . 45
.read_signature . . . 45
.reference_matrix . . . 46
.reference_to_matrix . . . 46
.remaining_columns . . . 47
.resolve_cmap_file . . . 47
.run_score_method . . . 48
.save_siginfo . . . 48
.score_gsea . . . 49
.select_perturbations . . . 50
.show_value_preview . . . 50
.siginfo_ask . . . 51
.siginfo_labels . . . 51
.siginfo_pct . . . 52

.simplify_siginfo . . . 52
.tie_floor . . . 53
.use_fast . . . 53
.validate_ref_matrix . . . 54
.write_result_tables . . . 54
annotate_drug_results . . . 55
cmap_extract . . . 57
compute_camsum_rho . . . 57
create_signature_from_gene_lists . . . 58
create_summary_from_results . . . 59
extract_cmap_data_from_siginfo . . . 59
extract_compound_id . . . 61
extract_signature_zscores . . . 62
fast_parse_gctx . . . 64
fast_read_meta . . . 64
fuzzy_drug_match . . . 65
get_rid . . . 66
plot.cmap_signature_result . . . 66
plot_signature_direction_tile_barcode . . . 67
print.cmap_signature_result . . . 70
process_combination . . . 71
process_combinations_file . . . 72
process_signature_with_df . . . 73
score_camsum . . . 75
score_gsea0 . . . 76
score_gsea1 . . . 77
score_gsea2 . . . 78
score_ks . . . 79
score_xcos . . . 79
score_xsum . . . 80
score_zhang . . . 81
scoring_methods . . . 82
subset_siginfo_beta . . . 82
summary.cmap_signature_result . . . 83

Index **85**

`.add_pert_kind_and_mode` *Add perturbation kind, mode and dose unit from* `pert_type`

Description

Add perturbation kind, mode and dose unit from `pert_type`

Usage

```
.add_pert_kind_and_mode(df)
```

Arguments

`df`	Data frame with a `pert_type` column.

Value

`df` with added `pert_kind`, `mode` and `dose_unit` columns.

.apply_filters *Apply pre-defined filters (non-interactive mode)*

Description

Apply pre-defined filters (non-interactive mode)

Usage

```
.apply_filters(filtered_data, filters, original_count, verbose)
```

Arguments

`filtered_data`	Data frame to filter.
`filters`	Named list; names are columns, values the values to keep. Unknown columns are skipped.
`original_count`	Row count before any filtering.
`verbose`	Logical; print one line per applied filter.

Value

The filtered data frame.

.apply_score *Apply a scoring function across reference lists*

Description

Apply a scoring function across reference lists

Usage

```
.apply_score(refList, scoreFun, ...)
```

Arguments

`refList`	List of per-sample reference objects.
`scoreFun`	Function used to score one reference object.
`...`	Additional arguments passed to `scoreFun`.

Value

Named or unnamed numeric vector of scores, one per element of `refList`.

.as_numeric *Coerce to numeric without warnings*

Description

Coerce to numeric without warnings

Usage

```
.as_numeric(x)
```

Arguments

x	Vector.

Value

Numeric vector; NA where x is not numeric.

.barcode_fig_size *Figure size for the barcode heatmap*

Description

Figure size for the barcode heatmap

Usage

```
.barcode_fig_size(n_rows, n_cols, width = NULL, height = NULL)
```

Arguments

n_rows, n_cols	Number of perturbations and genes.
width, height	User-supplied size in inches, or NULL to compute from the data (about 0.22 in per gene and 0.28 in per perturbation).

Value

Numeric vector c(width, height) in inches.

.barcode_gp *Build a grid gpar, dropping NULL entries*

Description

Build a grid gpar, dropping NULL entries

Usage

```
.barcode_gp(...)
```

Arguments

...	Graphical parameters; NULL values are ignored.

Value

A gpar object.

.barcode_panel *Build one panel of the barcode heatmap*

Description

Every panel of every layout (single, clustered with absent genes, split up/down) is produced here; the layouts differ only in the arguments.

Usage

```
.barcode_panel(
  z,
  genes,
  shared,
  name,
  title = "(not in ref)",
  title_size = 10,
  title_col = NULL,
  muted = FALSE,
  na_col = "grey",
  cluster_rows = FALSE,
  row_order = NULL,
  show_row_names = FALSE,
  row_title = character(0),
  cluster_cols = FALSE,
  col_name_col = NULL,
  show_legend = TRUE,
  show_ann_name = TRUE,
  show_ann_legend = TRUE
)
```

Arguments

z	Numeric matrix (perturbations x genes) for this panel.
genes	Gene names of the columns of z, used to look up log2FC.
shared	List of options common to all panels: style (from .barcode_style), lfc_all (named log2FC vector), row_method, col_method, row_dend, col_dend.
name	Heatmap name (must be unique within a heatmap list).
title	Column title.
title_size, title_col	Font size and colour of the column title.
muted	Logical; draw a gray placeholder panel for absent genes (fixed title style, no legends, no clustering).
na_col	Fill colour for NA cells.
cluster_rows	FALSE, TRUE or an hclust object.
row_order	Optional row order used when rows are not clustered.
show_row_names	Logical; show perturbation labels.
row_title	Row title.
cluster_cols	Logical; cluster the gene columns.
col_name_col	Optional column-name colours.
show_legend	Logical; show the z-score legend.
show_ann_name, show_ann_legend	Logical; show the annotation name and the annotation legend.

Value

A ComplexHeatmap::Heatmap.

.barcode_style *Colour scales shared by all barcode heatmap panels*

Description

Colour scales shared by all barcode heatmap panels

Usage

```
.barcode_style(z_plot, logfc_vals)
```

Arguments

z_plot	Numeric matrix of z-scores (perturbations x genes).
logfc_vals	Numeric vector of signature log2FC values.

Value

List with col_fun (symmetric z-score scale), lfc_col_fun (BrBG-like log2FC scale, teal = down, brown = up), muted_lfc_col_fun (gray log2FC scale for absent genes) and muted_col (NA fill).

.build_drug_context_summary *Per-drug context summary*

Description

Per-drug context summary

Usage

```
.build_drug_context_summary(res2, score_col, direction = "reversal")
```

Arguments

res2	Annotated data frame.
score_col	Score column.
direction	"reversal" (lowest score first) or "mimic".

Value

Data frame (empty when there are no drug rows).

.build_tech_view *Full technical view with preferred columns first*

Description

Full technical view with preferred columns first

Usage

```
.build_tech_view(res2, compound_col, score_col, p_col, padj_col)
```

Arguments

res2	Annotated data frame.
compound_col, score_col, p_col, padj_col	Column names.

Value

Data frame.

`.build_wetlab_drug_view` *Drug-focused wetlab view (one row per drug, best score first)*

Description

Drug-focused wetlab view (one row per drug, best score first)

Usage

```
.build_wetlab_drug_view(res2, score_col, keep_dose, direction = "reversal")
```

Arguments

`res2`	Annotated data frame.
`score_col`	Score column.
`keep_dose`	Logical; keep `dose_uM`.
`direction`	`"reversal"` (lowest score first) or `"mimic"`.

Value

Data frame.

`.build_wetlab_gene_view` *Gene-focused wetlab view (best score first)*

Description

Gene-focused wetlab view (best score first)

Usage

```
.build_wetlab_gene_view(res2, score_col, direction = "reversal")
```

Arguments

`res2`	Annotated data frame.
`score_col`	Score column.
`direction`	`"reversal"` (lowest score first) or `"mimic"`.

Value

Data frame.

.camsum_blocks *Split 1..n into consecutive blocks of at most* chunk

Description

Split 1..n into consecutive blocks of at most chunk

Usage

```
.camsum_blocks(n, chunk)
```

Arguments

n	Number of items.
chunk	Block size.

Value

List of integer vectors (empty when n = 0).

.camsum_cached *Evaluate* value *once per key (by default, once per session)*

Description

Values are kept in a list inside env because keys can be longer than the 10,000-byte limit on variable names.

Usage

```
.camsum_cached(key, value, env = .camsum_cache)
```

Arguments

key	Character cache key.
value	Expression, evaluated only on a cache miss; must not be NULL.
env	Environment used as the cache.

Value

The cached value.

.camsum_col_sd *Per-profile standard deviation over all genes, computed in column blocks*

Description

Per-profile standard deviation over all genes, computed in column blocks

Usage

```
.camsum_col_sd(refMatrix, chunk = 2000L)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
chunk	Number of columns processed at a time.

Value

Numeric vector, one standard deviation per column.

.camsum_file_key *Cache key for a file: path plus modification time*

Description

Cache key for a file: path plus modification time

Usage

```
.camsum_file_key(path)
```

Arguments

path	File path.

Value

Character string.

.camsum_gctx_columns *Read whole columns of the GCTX data matrix*

Description

Read whole columns of the GCTX data matrix

Usage

```
.camsum_gctx_columns(gctx_file, cols)
```

Arguments

gctx_file	GCTX file path.
cols	Integer column indices.

Value

Numeric matrix, all genes x cols.

.camsum_option_file *Path from a CONCERTDR file option, or NULL when unset or missing*

Description

Path from a CONCERTDR file option, or NULL when unset or missing

Usage

```
.camsum_option_file(name)
```

Arguments

name	Option name.

Value

Character path or NULL.

.camsum_pert_types *Look up the perturbation type of reference profiles*

Description

Uses `metadata` (the `"metadata"` attribute written by `extract_cmap_data_from_siginfo`) when it covers the profiles, otherwise the siginfo file in option `CONCERTDR.siginfo_file`.

Usage

```
.camsum_pert_types(sig_ids, metadata = NULL)
```

Arguments

`sig_ids`	Character vector of profile identifiers.
`metadata`	Optional data frame with `sample_id` and `pert_type` columns.

Value

Character vector of perturbation types, `NA` where unknown.

.camsum_read_meta_file *Read columns of a tab-separated CMap metadata file once per session*

Description

Read columns of a tab-separated CMap metadata file once per session

Usage

```
.camsum_read_meta_file(path, select)
```

Arguments

`path`	File path.
`select`	Columns to read.

Value

data.frame.

.camsum_rho *Streaming rho_bar on an in-memory matrix*

Description

Streaming rho_bar on an in-memory matrix

Usage

```
.camsum_rho(refMatrix, U, D, s_i, chunk = 2000L)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
U, D	Integer row indices of up- and down-regulated query genes.
s_i	Per-profile standard deviations.
chunk	Number of columns processed at a time.

Value

Numeric scalar with attributes k, kU, kD, n_profiles.

.camsum_rho_gctx *rho_bar over every profile of one perturbation type in the GCTX file*

Description

Profiles are all siginfo rows of pert_type; every one of them must be in the GCTX file. Each profile's s_i is taken over the GCTX rows whose gene symbols are the row names of the reference, so the gene space matches the reference. The result is cached for the session.

Usage

```
.camsum_rho_gctx(pert_type, genes, U, D, chunk = 2000L)
```

Arguments

pert_type	Perturbation type.
genes	Row names of the reference matrix.
U, D	Integer row indices (into genes) of the query genes.
chunk	Number of profiles read at a time.

Value

Numeric scalar with attributes k, kU, kD, n_profiles; NULL when the CMap file options are not set, a reference gene is not in the GCTX file, or the GCTX file lacks part of the library.

.camsum_rho_stream *Two-pass streaming estimate of rho_bar*

Description

read_block(b) returns block b of .camsum_signed() output. Uses $1'R1 = \mathrm{Var}_n(w'(y_n - \bar{y}))$ with $w_g = 1/\mathrm{sd}_g$, so the k x k matrix is never formed.

Usage

```
.camsum_rho_stream(read_block, n_blocks, k)
```

Arguments

read_block	Function of a block number returning a k x m matrix.
n_blocks	Number of blocks.
k	Number of query genes.

Value

List with rho_bar and n (profiles used).

.camsum_rows *Match query genes to reference rows for CamSum*

Description

Genes listed in both directions are dropped from both.

Usage

```
.camsum_rows(genes, queryUp, queryDown)
```

Arguments

genes	Character vector of reference row names.
queryUp, queryDown	Character vectors of gene symbols.

Value

List with integer row indices U and D.

.camsum_signed *Scale query rows by s_i, negate down rows, drop unusable profiles*

Description

Scale query rows by s_i, negate down rows, drop unusable profiles

Usage

```
.camsum_signed(Z, s, kU)
```

Arguments

Z	k x m block of query rows (up rows first, then down rows).
s	Standard deviations of the m profiles.
kU	Number of up rows.

Value

k x m' matrix of $y = \pm z/s_i$.

.camsum_warn_local_rho *Warn that rho_bar came from the supplied reference, not the full library*

Description

Warn that rho_bar came from the supplied reference, not the full library

Usage

```
.camsum_warn_local_rho(local, n_rho)
```

Arguments

local	Perturbation types whose rho_bar was estimated on the supplied reference.
n_rho	Number of profiles behind each rho_bar, named by type.

Value

NULL, invisibly.

.check_topN *Stop when* topN *exceeds half of the reference genes*

Description

Stop when topN exceeds half of the reference genes

Usage

```
.check_topN(topN, refMatrix)
```

Arguments

topN	Number of top/bottom genes per reference profile.
refMatrix	Numeric matrix with genes as rows.

Value

NULL, invisibly.

.combine_up_down *Combine up and down enrichment scores, zeroing them when they agree in sign*

Description

Combine up and down enrichment scores, zeroing them when they agree in sign

Usage

```
.combine_up_down(scoreUp, scoreDown)
```

Arguments

scoreUp, scoreDown	Enrichment scores of the up and down gene sets.

Value

Numeric.

.count_pvalues *Turn permutation exceedance counts into a result data frame*

Description

Turn permutation exceedance counts into a result data frame

Usage

```
.count_pvalues(score, count, permuteNum, pAdjMethod, sample_names)
```

Arguments

score	Observed scores.
count	Number of permutations with |permuted| >= |observed|, ties up to rounding included (see .tie_floor).
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.
sample_names	Row names of the result.

Value

Data frame with Score, pValue, pAdjValue columns.

.dedup_drug_rows *Collapse drug rows to one per drug*

Description

Keeps the complete row with the best score in the requested direction.

Usage

```
.dedup_drug_rows(
  df,
  score_col,
  group_keys = c("pert_id", "display_name"),
  direction = "reversal"
)
```

Arguments

df	Data frame.
score_col	Name of the score column.
group_keys	Columns defining a drug (those present in df).
direction	"reversal" (lowest score first) or "mimic".

Value

Data frame with one row per group.

.dedup_geneinfo *Select one feature per gene symbol*

Description

Prefer landmark, then best inferred, then inferred, then unrecognised feature spaces. Ties retain the first input row. Retained rows stay in input order; a warning identifies every discarded gene ID and its replacement.

Usage

```
.dedup_geneinfo(geneinfo_df)
```

Arguments

geneinfo_df	Gene information with gene_id and gene_symbol columns, and optionally feature_space.

Value

Gene information with unique gene symbols.

.derive_annotations *Derive names, context, perturbation kind, effect direction and MOA status*

Description

Derive names, context, perturbation kind, effect direction and MOA status

Usage

```
.derive_annotations(res2, compound_col, score_col)
```

Arguments

res2	Merged data frame.
compound_col	Signature id column.
score_col	Score column.

Value

res2 with the derived columns added.

.direction_ordered_genes *Order signature genes down-regulated first, then up-regulated*

Description

Order signature genes down-regulated first, then up-regulated

Usage

```
.direction_ordered_genes(sig, gene_col, log2fc_col, max_genes = NULL)
```

Arguments

sig	Cleaned signature data frame (see .read_signature).
gene_col, log2fc_col	Names of the gene and log2FC columns in sig.
max_genes	Optional cap applied separately to each direction; NULL keeps all genes.

Value

Character vector: most negative to zero, then most positive first.

.draw_barcode *Draw a barcode heatmap (list) with the standard legend placement*

Description

Draw a barcode heatmap (list) with the standard legend placement

Usage

```
.draw_barcode(ht, extra = list())
```

Arguments

ht	A Heatmap or HeatmapList.
extra	Named list of further arguments for ComplexHeatmap::draw.

Value

Called for its side effect of drawing.

.fast_chunk_size *Number of profiles scored at a time by the vectorised engine*

Description

Number of profiles scored at a time by the vectorised engine

Usage

```
.fast_chunk_size()
```

Value

Integer; option `CONCERTDR.chunk_size` (default: 2000).

.fast_draws *Draw the permuted queries in the order the row-by-row code draws them*

Description

Draw the permuted queries in the order the row-by-row code draws them

Usage

```
.fast_draws(permuteNum, draw_one)
```

Arguments

`permuteNum`	Number of permutations.
`draw_one`	Function of no arguments drawing one permuted query.

Value

List of `permuteNum` draws.

.fast_gsea0 *Vectorised GSEA weight-0 score*

Description

Vectorised GSEA weight-0 score

Usage

```
.fast_gsea0(refMatrix, queryUp, queryDown, permuteNum, pAdjMethod)
```

Arguments

refMatrix	Numeric matrix, genes x profiles.
queryUp, queryDown	Character vectors of gene symbols.
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.

.fast_gsea0_es *Unweighted GSEA enrichment of sorted positions*

Description

The running sum ends at 0, so its maximum sits at a hit (or is 0) and its minimum just before a hit (or is 0). The comparison is done on integers (the sums scaled by $k(G-k)$), so a maximum that equals the absolute minimum is resolved the same way for every profile.

Usage

```
.fast_gsea0_es(SP, k, G)
```

Arguments

SP	k x profiles matrix of sorted positions.
k	Size of the gene set.
G	Number of genes.

Value

Numeric vector, one value per profile.

.fast_gseaW *Vectorised weighted GSEA score*

Description

Vectorised weighted GSEA score

Usage

```
.fast_gseaW(refMatrix, queryUp, queryDown, permuteNum, pAdjMethod, weight)
```

Arguments

refMatrix	Numeric matrix, genes x profiles.
queryUp, queryDown	Character vectors of gene symbols.
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.
weight	Exponent applied to the absolute reference values.

.fast_gseaW_es *Weighted GSEA enrichment of sorted positions*

Description

Weighted GSEA enrichment of sorted positions

Usage

```
.fast_gseaW_es(SP, W, k, G)
```

Arguments

SP	k x profiles matrix of sorted positions.
W	k x profiles matrix of the weights, sorted like SP.
k	Size of the gene set.
G	Number of genes.

Value

Numeric vector, one value per profile.

.fast_ks *Vectorised KS score*

Description

Vectorised KS score

Usage

```
.fast_ks(refMatrix, queryUp, queryDown, permuteNum, pAdjMethod)
```

Arguments

refMatrix	Numeric matrix, genes x profiles.
queryUp, queryDown	Character vectors of gene symbols.
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.

.fast_ks_es *KS enrichment of sorted positions*

Description

KS enrichment of sorted positions

Usage

```
.fast_ks_es(SP, k, G)
```

Arguments

SP	k x profiles matrix of sorted positions.
k	Size of the gene set.
G	Number of genes.

Value

Numeric vector, one value per profile.

.fast_permutation_test *Score every profile of a reference matrix, block by block*

Description

Score every profile of a reference matrix, block by block

Usage

```
.fast_permutation_test(
  refMatrix,
  make_ctx,
  score,
  draws,
  permuteNum,
  pAdjMethod
)
```

Arguments

refMatrix	Numeric matrix, genes x profiles.
make_ctx	Function of a block matrix returning the per-block context.
score	Function (ctx, draw) returning one score per profile of the block; draw is NULL for the observed query and one element of draws for a permutation.
draws	List of permuted queries.
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.

Value

Data frame with Score, pValue, pAdjValue per profile.

.fast_positions *Positions (1 = highest value) of all genes in every profile of a block*

Description

Positions (1 = highest value) of all genes in every profile of a block

Usage

```
.fast_positions(M)
```

Arguments

M	Numeric matrix, genes x profiles.

Value

Integer matrix of the same shape.

.fast_shared_null_test *KS or unweighted GSEA with one null shared by all profiles*

Description

The score depends on the gene set only through its positions in a profile, and a random gene set has the same distribution of positions in every profile, so the permutation null does not depend on the profile.

Usage

```
.fast_shared_null_test(
  refMatrix,
  queryUp,
  queryDown,
  es,
  permuteNum,
  pAdjMethod
)
```

Arguments

refMatrix	Numeric matrix, genes x profiles.
queryUp, queryDown	Character vectors of gene symbols.
es	Enrichment function of sorted positions (.fast_ks_es or .fast_gsea0_es).
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.

Value

Data frame with Score, pValue, pAdjValue per profile.

.fast_sorted_set *Positions of a gene set sorted within each profile*

Description

Positions of a gene set sorted within each profile

Usage

```
.fast_sorted_set(pos, S, values = NULL)
```

Arguments

pos	Position matrix from .fast_positions.
S	Integer row indices of the gene set.
values	Optional matrix (genes x profiles) whose rows S are reordered the same way.

Value

List with SP (k x profiles sorted positions) and, when values is given, W (the values in the same order).

`.fast_up_down_test` *Vectorised test for methods scoring an up and a down gene set*

Description

Vectorised test for methods scoring an up and a down gene set

Usage

```
.fast_up_down_test(
  refMatrix,
  queryUp,
  queryDown,
  make_ctx,
  score,
  permuteNum,
  pAdjMethod
)
```

Arguments

refMatrix	Numeric matrix, genes x profiles.
queryUp, queryDown	Character vectors of gene symbols.
make_ctx, score	See .fast_permutation_test; score is called as score(ctx, list(U =, D =)) where U and D are row indices, or with NULL for the observed query.
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.

Value

Data frame with Score, pValue, pAdjValue per profile.

`.fast_xcos` *Vectorised XCos score*

Description

Vectorised XCos score

Usage

```
.fast_xcos(refMatrix, query, topN, permuteNum, pAdjMethod)
```

Arguments

refMatrix	Numeric matrix, genes x profiles.
query	Named numeric vector.
topN	Number of top/bottom genes kept per profile.
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.

.fast_xsum *Vectorised XSum score*

Description

Vectorised XSum score

Usage

```
.fast_xsum(refMatrix, queryUp, queryDown, topN, permuteNum, pAdjMethod)
```

Arguments

refMatrix	Numeric matrix, genes x profiles.
queryUp, queryDown	Character vectors of gene symbols.
topN	Number of top/bottom genes kept per profile.
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.

.fast_zhang *Vectorised Zhang score*

Description

Vectorised Zhang score

Usage

```
.fast_zhang(refMatrix, queryUp, queryDown, permuteNum, pAdjMethod)
```

Arguments

refMatrix	Numeric matrix, genes x profiles.
queryUp, queryDown	Character vectors of gene symbols.
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.

.fill_name_from_iname *Fill blank entries of a name column from* pert_iname

Description

Fill blank entries of a name column from pert_iname

Usage

```
.fill_name_from_iname(df, col)
```

Arguments

df	Data frame.
col	Name column; created as NA when absent.

Value

df.

.filter_interactively *Run the interactive filtering session over all key columns*

Description

Run the interactive filtering session over all key columns

Usage

```
.filter_interactively(
  filtered_data,
  key_columns,
  original_count,
  verbose,
  show_preview
)
```

Arguments

filtered_data	Data frame to filter.
key_columns	Columns to filter on, in order.
original_count	Row count before any filtering.
verbose, show_preview	Logical flags.

Value

The filtered data frame.

.filter_siginfo *Apply the quality filter and signature limit to a siginfo table*

Description

Apply the quality filter and signature limit to a siginfo table

Usage

```
.filter_siginfo(sig_info, filter_quality, max_signatures, verbose)
```

Arguments

sig_info	Siginfo data frame.
filter_quality	Whether to keep only is_hiq == 1 rows.
max_signatures	Maximum number of signatures, or NULL.
verbose	Whether to print progress messages.

Value

The filtered data frame. Errors when sig_id is missing or no signature is left.

.first_existing_file *First existing file among candidates*

Description

First existing file among candidates

Usage

```
.first_existing_file(...)
```

Arguments

...	Candidate paths; NULL, NA and empty strings are skipped.

Value

A path, or NULL when none of the candidates exists.

.gctx_ids *Row and column ids of a GCTX file, read once per session*

Description

Cached by file path and modification time, so repeated block reads of a large file do not re-read and re-trim hundreds of thousands of ids.

Usage

```
.gctx_ids(fname)
```

Arguments

fname	GCTX file path.

Value

List with `row` and `col` character vectors.

.gctx_read_matrix *Read part of the GCTX data matrix*

Description

Reads sorted, unique indices and explicitly restores the requested order (including repeated indices), independently on both axes.

Usage

```
.gctx_read_matrix(fname, index)
```

Arguments

fname	GCTX file path.
index	List of row and column indices; `NULL` selects all.

Value

Numeric matrix without dimnames.

.is_blank *Whether strings are missing or empty*

Description

Whether strings are missing or empty

Usage

```
.is_blank(x)
```

Arguments

x	Vector coercible to character.

Value

Logical vector.

.list_column_values *Print the available values of a column with their impact*

Description

Print the available values of a column with their impact

Usage

```
.list_column_values(
  filtered_data,
  col,
  unique_vals,
  remaining_cols,
  original_count,
  show_preview
)
```

Arguments

filtered_data	Current data frame.
col	Column being filtered.
unique_vals	Sorted unique values of col.
remaining_cols	Columns after col.
original_count	Row count before any filtering.
show_preview	Logical; append downstream option counts.

Value

Named list (one element per value) with data, count, prop_current and prop_original.

.load_signature *Read the signature from a data frame or a file*

Description

Read the signature from a data frame or a file

Usage

```
.load_signature(signature_file, read_method)
```

Arguments

signature_file	Path to a signature file, or a data frame with Gene and log2FC columns.
read_method	One of "auto", "fread" or "read.table".

Value

List with the signature data and a label describing its origin.

.matrix_to_extreme_list *Convert expression matrix to top-N / bottom-N named-value lists (for XCos/XSum)*

Description

Convert expression matrix to top-N / bottom-N named-value lists (for XCos/XSum)

Usage

```
.matrix_to_extreme_list(refMatrix, topN)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
topN	Number of most positive and most negative genes to retain.

Value

List of named numeric vectors containing the top and bottom genes per sample.

.matrix_to_name_ranked_list

Convert expression matrix to sorted gene-name lists (for KS)

Description

Convert expression matrix to sorted gene-name lists (for KS)

Usage

```
.matrix_to_name_ranked_list(refMatrix)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.

Value

List of character vectors ordered decreasingly by expression for each sample.

.matrix_to_signed_rank_list

Convert expression matrix to signed-rank lists (for Zhang)

Description

Convert expression matrix to signed-rank lists (for Zhang)

Usage

```
.matrix_to_signed_rank_list(refMatrix)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.

Value

List of named numeric vectors containing signed ranks per sample.

.matrix_to_value_ranked_list *Convert expression matrix to sorted named-value lists (for GSEA)*

Description

Convert expression matrix to sorted named-value lists (for GSEA)

Usage

```
.matrix_to_value_ranked_list(refMatrix)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.

Value

List of named numeric vectors ordered decreasingly by expression for each sample.

.merge_compoundinfo *Merge compound annotations (target, MOA) by* pert_id

Description

Multiple annotation rows are collapsed per perturbation: distinct nonblank values in each column are joined with "; " in input order. Target and MOA columns are independent lists, not positional target-MOA pairs.

Usage

```
.merge_compoundinfo(res2, comp_info_file)
```

Arguments

res2	Data frame with a pert_id column.
comp_info_file	Compound info path or data frame.

Value

Merged data frame.

.merge_siginfo *Merge signature results with siginfo annotations*

Description

Merge signature results with siginfo annotations

Usage

```
.merge_siginfo(res, sig_info_file, compound_col)
```

Arguments

res	Results data frame.
sig_info_file	Siginfo path or data frame.
compound_col	Signature id column in res.

Value

Merged data frame with a guaranteed pert_id column.

.message_column_summary *Report the distinct values of one siginfo column*

Description

Report the distinct values of one siginfo column

Usage

```
.message_column_summary(sig_info, col, label, max_shown = Inf)
```

Arguments

sig_info	Siginfo data frame.
col	Column name; nothing is reported when it is absent.
label	Label used in the message.
max_shown	Maximum number of values listed.

Value

NULL, invisibly.

.message_rule *Print a horizontal rule of* n *copies of* char *as a message*

Description

Print a horizontal rule of n copies of char as a message

Usage

```
.message_rule(char = "=", n = 80L)
```

Arguments

char	Single character.
n	Width.

Value

NULL, invisibly.

.order_by_score *Order rows by ascending numeric score*

Description

Order rows by ascending numeric score

Usage

```
.order_by_score(df, score_col, direction = "reversal")
```

Arguments

df	Data frame.
score_col	Score column.
direction	"reversal" (lowest score first) or "mimic".

Value

Data frame.

.parse_compound_context *Parse library, cell line, dose and time from CMap signature ids*

Description

Ids look like `LIB_CELL_6H:BRD-XXXX:dose:time`.

Usage

```
.parse_compound_context(df, col = "compound")
```

Arguments

`df`	Data frame.
`col`	Column holding the signature ids.

Value

`df` with added `library`, `cell_line`, `broad_id`, `dose_uM` and `time_h` columns.

.permutation_pvalues *Compute p-values and adjusted p-values via permutation*

Description

Compute p-values and adjusted p-values via permutation

Usage

```
.permutation_pvalues(
  score,
  permuteScoreMat,
  pAdjMethod = "BH",
  sample_names = NULL
)
```

Arguments

`score`	Numeric vector of observed scores (one per sample).
`permuteScoreMat`	Matrix of permuted scores (nSamples x nPerms).
`pAdjMethod`	Adjustment method passed to `stats::p.adjust`.
`sample_names`	Optional sample names used as row names of the result.

Value

Data frame with Score, pValue, pAdjValue columns.

.permutation_test *Score a query against every reference profile and against its permutations*

Description

Score a query against every reference profile and against its permutations

Usage

```
.permutation_test(
  refMatrix,
  refList,
  scoreFun,
  queryArgs,
  draw,
  permuteNum,
  pAdjMethod
)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
refList	Per-sample reference objects built from refMatrix.
scoreFun	Function scoring one reference object; called as scoreFun(ref, ...) with the arguments in queryArgs or returned by draw.
queryArgs	Named list of query arguments for the observed score.
draw	Function of no arguments returning the named list of query arguments for one permutation.
permuteNum	Number of permutations.
pAdjMethod	P-value adjustment method.

Value

Data frame with Score, pValue, pAdjValue per sample.

.permutation_test_up_down *Permutation test for methods scoring an up and a down gene set*

Description

Permuted queries draw random gene sets of the same sizes from the reference genes.

Usage

```
.permutation_test_up_down(
  refMatrix,
  refList,
  scoreFun,
  queryUp,
  queryDown,
  permuteNum,
  pAdjMethod
)
```

Arguments

`refMatrix`	Numeric matrix with genes as rows and samples as columns.
`refList`	Per-sample reference objects built from `refMatrix`.
`scoreFun`	Function (`ref`, `queryUp`, `queryDown`) scoring one reference object.
`queryUp`, `queryDown`	Character vectors of gene symbols.
`permuteNum`	Number of permutations.
`pAdjMethod`	P-value adjustment method.

Value

Data frame with Score, pValue, pAdjValue per sample.

`.pert_id_missing` *Whether the* `pert_id` *column is absent or entirely blank*

Description

Whether the `pert_id` column is absent or entirely blank

Usage

```
.pert_id_missing(df)
```

Arguments

`df`	Data frame.

Value

Logical scalar.

.print_filter_summary *Print the final filtering summary*

Description

Print the final filtering summary

Usage

```
.print_filter_summary(filtered_data, key_columns, original_count)
```

Arguments

filtered_data	Filtered data frame.
key_columns	Columns to summarise.
original_count	Row count before any filtering.

Value

NULL, invisibly.

.print_reuse_filters *Print the filters that reproduce the interactive selection*

Description

Print the filters that reproduce the interactive selection

Usage

```
.print_reuse_filters(filtered_data, sig_info, key_columns)
```

Arguments

filtered_data	Filtered data frame.
sig_info	Unfiltered data frame.
key_columns	Filter columns.

Value

NULL, invisibly.

.prompt_column_filter *Interactively filter the data on one column*

Description

Interactively filter the data on one column

Usage

```
.prompt_column_filter(
  filtered_data,
  col,
  col_idx,
  key_columns,
  original_count,
  verbose,
  show_preview
)
```

Arguments

`filtered_data`	Current data frame.
`col`	Column to filter.
`col_idx`, `key_columns`	Position of `col` within `key_columns`.
`original_count`	Row count before any filtering.
`verbose`, `show_preview`	Logical flags.

Value

The (possibly reduced) data frame.

.rank_by_direction *Order rows by score in the requested connectivity direction*

Description

Rows are only ordered, never dropped by the sign of the score; rows with a non-finite score are removed because they cannot be ranked.

Usage

```
.rank_by_direction(df, score_col = "Score", direction = c("reversal", "mimic"))
```

Arguments

`df`	Data frame containing scores.
`score_col`	Score column name.
`direction`	`"reversal"` (lowest score first) or `"mimic"` (highest score first).

Value

Data frame in best-score-first order.

.read_cmap_table *Read a CMap table from a path or pass a data frame through*

Description

Column names are trimmed of surrounding whitespace.

Usage

```
.read_cmap_table(x, what, select = NULL, sep = "auto")
```

Arguments

x	File path or data frame.
what	Name of the calling function's argument, used in error messages (for example "geneinfo_file").
select	Optional character vector of columns to keep. Columns that are absent from the table are dropped; errors if none are present.
sep	Field separator passed to data.table::fread (default: auto-detect).

Value

Data frame.

.read_gene_map *Read a geneinfo file into a symbol -> gene_id lookup*

Description

Read a geneinfo file into a symbol -> gene_id lookup

Usage

```
.read_gene_map(geneinfo_file)
```

Arguments

geneinfo_file	Path to a geneinfo file with gene_symbol and gene_id columns.

Value

Named character vector of gene ids named by upper-cased symbol.

.read_siginfo *Read a siginfo file*

Description

Read a siginfo file

Usage

```
.read_siginfo(siginfo_file, verbose)
```

Arguments

siginfo_file	Path to the siginfo table.
verbose	Logical; print progress messages.

Value

Data frame.

.read_signature *Read and clean a signature (gene and log2FC columns)*

Description

Read and clean a signature (gene and log2FC columns)

Usage

```
.read_signature(signature_file)
```

Arguments

signature_file	Signature file path or data frame. Falls back to the first two columns when Gene/log2FC are absent.

Value

List with the cleaned data frame sig (upper-cased genes, numeric log2FC, incomplete rows dropped) and the column names gene_col and log2fc_col.

.reference_matrix *Convert a reference data frame to a numeric matrix*

Description

Convert a reference data frame to a numeric matrix

Usage

```
.reference_matrix(reference_df)
```

Arguments

reference_df	Data frame with a gene_symbol column and one column per profile.

Value

Numeric matrix with gene symbols as row names.

.reference_to_matrix *Coerce a reference data frame or matrix to a numeric gene x sample matrix*

Description

Coerce a reference data frame or matrix to a numeric gene x sample matrix

Usage

```
.reference_to_matrix(reference_df)
```

Arguments

reference_df	Data frame (optionally with a gene_symbol column) or matrix with genes as row names and perturbation ids as column names.

Value

Numeric matrix with upper-cased gene row names.

.remaining_columns *Columns that follow position* col_idx *in* key_columns

Description

Columns that follow position col_idx in key_columns

Usage

```
.remaining_columns(key_columns, col_idx)
```

Arguments

key_columns	Character vector of filter columns.
col_idx	Current position.

Value

Character vector (possibly empty).

.resolve_cmap_file *Resolve a CMap file from explicit argument, option, env var or data_dir*

Description

Candidates are tried in order: explicit, option CONCERTDR.<key>_file, env var CONCERTDR_<KEY>_FILE, data_dir/default, then default in the working directory. The last two are skipped when default is NULL.

Usage

```
.resolve_cmap_file(explicit, key, default = NULL, data_dir = NULL)
```

Arguments

explicit	User-supplied path (may be NULL).
key	Short file key (for example "gctx"), used to build the option and environment variable names.
default	Default file name, or NULL for none.
data_dir	Optional directory searched for default.

Value

Path of the first candidate that exists, or NULL.

.run_score_method *Run one scoring method*

Description

Run one scoring method

Usage

```
.run_score_method(
  method,
  ref,
  query,
  common_up,
  common_down,
  permutations,
  topN,
  camsum,
  vectorized = TRUE
)
```

Arguments

method	One of .score_methods.
ref	Reference matrix (genes x profiles).
query	Named numeric vector of signature log2FC values (XCos).
common_up, common_down	Signature genes present in ref.
permutations	Number of permutations.
topN	Number of top/bottom genes for XCos and XSum.
camsum	List with CamSum's alternative, rho_bar and pert_type arguments.
vectorized	Whether to use the vectorised engine.

Value

Data frame with Score, pValue, pAdjValue per profile.

.save_siginfo *Write the filtered siginfo table*

Description

Write the filtered siginfo table

Usage

```
.save_siginfo(filtered_data, output_file, verbose)
```

Arguments

filtered_data	Data frame to save.
output_file	Destination path; missing directories are created.
verbose	Logical; print progress messages.

Value

NULL, invisibly.

.score_gsea *Weighted GSEA connectivity score*

Description

Genes are ranked by decreasing value; each gene in the query contributes $|v|^{weight}$ to the running enrichment sum (weight 0 gives the classical unweighted KS-like statistic).

Usage

```
.score_gsea(
  refMatrix,
  queryUp,
  queryDown,
  permuteNum,
  pAdjMethod,
  weight,
  vectorized = TRUE
)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
queryUp	Character vector of up-regulated gene symbols.
queryDown	Character vector of down-regulated gene symbols.
permuteNum	Number of permutations (default: 10000).
pAdjMethod	P-value adjustment method (default: "BH").
weight	Exponent applied to the absolute reference values.
vectorized	Whether to use the vectorised engine (default: TRUE), which scores blocks of samples at once. It is skipped automatically for reference matrices with missing values or duplicated gene names. Scores are the same as the row-by-row code; p-values of KS and GSEA weight 0 use one permutation null shared by all samples (same distribution, different random numbers), and the other methods reproduce the row-by-row p-values for the same seed. Use FALSE for the row-by-row code.

Value

Data frame with Score, pValue, pAdjValue per sample.

.select_perturbations *Select the best-scoring perturbations from a results table*

Description

Select the best-scoring perturbations from a results table

Usage

```
.select_perturbations(
  results_df,
  pert_id_col,
  score_col,
  selected_drug,
  selected_drug_col,
  max_perts,
  verbose,
  direction
)
```

Arguments

`results_df`	Data frame with perturbation id and score columns.
`pert_id_col, score_col`	Names of the id and score columns.
`selected_drug`	Optional drug identifier used to filter rows.
`selected_drug_col`	Column matched against `selected_drug`; when `NULL` it is auto-detected.
`max_perts`	Maximum number of rows to keep.
`verbose`	Logical; print progress messages.
`direction`	Order in which perturbations are picked by score: `"reversal"` (default) lowest first, `"mimic"` highest first. No perturbation is excluded by the sign of its score. Precomputed matrices are plotted as supplied; choose the direction when extracting them.

Value

Data frame sorted by the selected score direction, at most `max_perts` rows.

.show_value_preview *Print the detailed downstream preview for one candidate value*

Description

Print the detailed downstream preview for one candidate value

Usage

```
.show_value_preview(preview_val, impact, remaining_cols)
```

Arguments

preview_val	The value being previewed.
impact	Its entry of the list from .list_column_values().
remaining_cols	Columns after the current one.

Value

NULL, invisibly.

.siginfo_ask *Read one line of user input*

Description

Thin wrapper around readline so that the prompt loop can be mocked.

Usage

```
.siginfo_ask(prompt)
```

Arguments

prompt	Prompt string.

Value

Character scalar.

.siginfo_labels *Build readable perturbation labels from a siginfo file*

Description

Labels look like name | dose | time | cell, where name is the first non-empty of cmap_name, pert_iname, pert_id. Ids that are not found (or when no siginfo file is available) keep their raw id.

Usage

```
.siginfo_labels(sig_ids, siginfo_file)
```

Arguments

sig_ids	Character vector of sig_ids.
siginfo_file	Path to a siginfo file, or NULL.

Value

Character vector the same length as sig_ids.

.siginfo_pct *Percentage rounded to one decimal*

Description

Percentage rounded to one decimal

Usage

```
.siginfo_pct(n, total)
```

Arguments

n, total	Numerator and denominator.

Value

Numeric scalar.

.simplify_siginfo *Reduce a siginfo table to the metadata columns used downstream*

Description

Reduce a siginfo table to the metadata columns used downstream

Usage

```
.simplify_siginfo(sig_info)
```

Arguments

sig_info	Siginfo data frame.

Value

Data frame with sample_id and, when present in sig_info, time, dose, cell, pert_name, pert_type and is_hiq.

.tie_floor *Smallest absolute permuted score that counts as reaching the observed one*

Description

Permuted scores within a relative tolerance of `sqrt(.Machine$double.eps)` of the observed score count as ties, so that permutation p-values do not depend on platform-specific rounding of sums (for example long double accumulation on x86_64).

Usage

```
.tie_floor(x)
```

Arguments

`x`	Observed scores.

Value

Numeric vector, `|x|` minus the tolerance.

.use_fast *Whether the vectorised engine can score this reference matrix*

Description

Matrices with missing values or duplicated gene names take the row-by-row path, which defines the behaviour for those cases.

Usage

```
.use_fast(refMatrix, vectorized)
```

Arguments

`refMatrix`	Numeric matrix with genes as rows and samples as columns.
`vectorized`	Whether the user asked for the vectorised engine.

Value

Logical scalar.

.validate_ref_matrix *Validate and coerce reference matrix*

Description

Validate and coerce reference matrix

Usage

```
.validate_ref_matrix(refMatrix)
```

Arguments

refMatrix	Numeric matrix or data frame with genes as rows and samples as columns.

Value

Numeric matrix with preserved row and column names.

.write_result_tables *Write the four result tables as TSV files*

Description

Write the four result tables as TSV files

Usage

```
.write_result_tables(tables, output_dir, verbose)
```

Arguments

tables	Named list with wetlab_drug_view, wetlab_gene_view, tech_view_all and drug_context_summary.
output_dir	Output directory (created when missing).
verbose	Logical; list written files.

Value

Named character vector of file paths.

annotate_drug_results *Annotate CONCERTDR Results with Drug Information*

Description

Integrate signature matching results with annotations from `siginfo_beta.txt` and `compoundinfo_beta.txt`. It produces three analysis-ready tables and one drug context summary table.

Usage

```
annotate_drug_results(
  results_df,
  sig_info_file,
  comp_info_file,
  output_file = NULL,
  score_col = "Score",
  padj_col = "pAdjValue",
  p_col = "pValue",
  compound_col = "compound",
  keep_dose_in_drug_view = FALSE,
  output_dir = NULL,
  write_outputs = FALSE,
  verbose = TRUE,
  direction = c("reversal", "mimic")
)
```

Arguments

`results_df`	Data frame containing signature matching results with a 'compound' column
`sig_info_file`	Path to siginfo_beta.txt file or data frame with signature information
`comp_info_file`	Path to compound information file or data frame
`output_file`	Deprecated single-file output path; when provided, `tech_view_all` is also written as CSV for compatibility.
`score_col`	Score column in `results_df` (default: `"Score"`)
`padj_col`	Adjusted p-value column name (default: `"pAdjValue"`)
`p_col`	Raw p-value column name (default: `"pValue"`)
`compound_col`	Compound/sig-id column in `results_df` (default: `"compound"`)
`keep_dose_in_drug_view`	Logical; whether to keep `dose_uM` in `wetlab_drug_view` (default: `FALSE`)
`output_dir`	Optional directory to write four TSV outputs. If NULL, files are not written unless `write_outputs = TRUE`.
`write_outputs`	Logical; write four TSV outputs (default: `FALSE`)
`verbose`	Logical; whether to print progress messages (default: TRUE)
`direction`	`"reversal"` (default) puts the lowest scores first in the wetlab views and picks each drug's lowest-scoring context as its best; `"mimic"` uses the highest. It only orders and picks; no row is dropped by the sign of its score.

Details

When compoundinfo contains multiple rows for one `pert_id`, all distinct nonblank target and MOA values are retained, separated by `"; "`. This does not duplicate signature results or inflate context counts. Targets and MOAs are independent lists, not positional pairs.

Value

A named list with:

`wetlab_drug_view`
: Drug-focused wetlab view

`wetlab_gene_view`
: Gene-focused wetlab view

`tech_view_all`
: Full technical table with all integrated fields

`drug_context_summary`
: Per-drug context summary

`output_files`
: Named character vector of written files (if any)

Examples

```
ex_results <- data.frame(
  compound = "CVD001_HEPG2_6H:BRD-K03652504-001-01-9:10.0497",
  Score = -0.72,
  pValue = 0.002,
  pAdjValue = 0.02
)
ex_siginfo <- data.frame(
  sig_id = ex_results$compound,
  pert_type = "trt_cp",
  pert_id = "BRD-K03652504-001-01-9",
  pert_iname = "imatinib"
)
ex_compinfo <- data.frame(
  pert_id = "BRD-K03652504-001-01-9",
  pert_name = "imatinib",
  cmap_name = "imatinib"
)
views <- annotate_drug_results(
  results_df = ex_results,
  sig_info_file = ex_siginfo,
  comp_info_file = ex_compinfo,
  write_outputs = FALSE,
  verbose = FALSE
)
names(views)


if (file.exists("sig_match_xsum_results.csv") &&
    file.exists("siginfo_beta.txt") &&
    file.exists("compoundinfo_beta.txt")) {
  results <- read.csv("sig_match_xsum_results.csv")

  views <- annotate_drug_results(
    results_df = results,
    sig_info_file = "siginfo_beta.txt",
```

```
    comp_info_file = "compoundinfo_beta.txt",
    output_dir = "results",
    write_outputs = TRUE
  )

  head(views$wetlab_drug_view)
}
```

`cmap_extract` *Extract Data from CMap GCTX Files for Specified Combinations*

Description

Functions to extract expression data from CMap GCTX files based on specified combinations of time points, dosages, and cell lines.

Value

None; this is an internal documentation topic.

`compute_camsum_rho` *Mean inter-gene correlation of a query across a reference library*

Description

Computes $\bar{\rho}$, the mean off-diagonal Pearson correlation of the query genes measured across the profiles of `refMatrix`, after dividing each profile by its standard deviation over all genes and negating the down-regulated rows. The $k \times k$ correlation matrix is never formed: two passes over column blocks use the identity $1'R1 = \mathrm{Var}_n(w'(y_n - \bar{y}))$ with $w_g = 1/\mathrm{sd}_g$.

Usage

```
compute_camsum_rho(refMatrix, queryUp, queryDown, chunk = 2000L)
```

Arguments

`refMatrix`	Numeric matrix (or data frame) with genes as rows and profiles as columns, with row and column names.
`queryUp, queryDown`	Character vectors of up- and down-regulated gene symbols. Genes present in both are dropped from both.
`chunk`	Number of columns processed at a time (default: 2000).

Details

CamSum's p-values were calibrated with $\bar{\rho}$ estimated on the whole treatment library of one perturbation type (for example all `trt_oe` or all `trt_xpr` profiles). `process_signature_with_df` does this automatically from the GCTX file when the CMap file options are set; this function estimates $\bar{\rho}$ on the matrix it is given.

Value

A single numeric value, $\bar{\rho}$, with attributes `k`, `kU`, `kD` (query genes found in `refMatrix`) and `n_profiles` (profiles used). `NA` when fewer than two query genes vary across the library.

Examples

```
set.seed(1)
ref <- matrix(rnorm(200 * 50), 200, 50,
              dimnames = list(paste0("g", 1:200), paste0("s", 1:50)))
compute_camsum_rho(ref, queryUp = paste0("g", 1:10),
                   queryDown = paste0("g", 11:20))
```

---

create_signature_from_gene_lists

Create a Signature Data Frame from Gene Lists

Description

Converts separate lists of up-regulated and down-regulated genes into a signature data frame compatible with `process_signature_with_df` and other CONCERTDR functions. This is a convenience wrapper for users who only have gene lists (e.g.\ from a pathway database) rather than continuous fold-change values.

Usage

```
create_signature_from_gene_lists(
  up_genes,
  down_genes,
  up_value = 1,
  down_value = -1
)
```

Arguments

`up_genes`	Character vector of up-regulated gene symbols.
`down_genes`	Character vector of down-regulated gene symbols.
`up_value`	Numeric value to assign to up-regulated genes (default: 1).
`down_value`	Numeric value to assign to down-regulated genes (default: -1).

Value

A data frame with columns `Gene` and `log2FC`, suitable for use as the `signature_file` argument in `process_signature_with_df` and `extract_signature_zscores`.

Examples

```
sig_df <- create_signature_from_gene_lists(
  up_genes   = c("TP53", "MYC", "BRCA1"),
  down_genes = c("EGFR", "VEGFA", "KRAS")
)
head(sig_df)
```

create_summary_from_results

Create a summary from signature matching results

Description

Create a summary from signature matching results

Usage

```
create_summary_from_results(
  results_list,
  top_n = 20,
  direction = c("reversal", "mimic")
)
```

Arguments

results_list	List of results from different methods
top_n	Number of top hits to include (default: 20)
direction	"reversal" (lowest score first) or "mimic".

Value

Data frame with summary of top hits across methods

extract_cmap_data_from_siginfo

Extract CMap Data Using Siginfo File Directly

Description

Extract expression data from CMap GCTX files using all signatures present in the provided siginfo file. This function processes all signatures found in the siginfo file without requiring a configuration file.

Usage

```
extract_cmap_data_from_siginfo(
  siginfo_file = "siginfo_beta.txt",
  geneinfo_file = "geneinfo_beta.txt",
  gctx_file = "level5_beta_trt_cp_n720216x12328.gctx",
  max_signatures = NULL,
  filter_quality = TRUE,
  verbose = TRUE,
  landmark = TRUE
)
```

Arguments

siginfo_file	Path to the signature info file (can be original or pre-filtered)
geneinfo_file	Path to the gene info file
gctx_file	Path to the GCTX file
max_signatures	Integer; maximum number of signatures to process (default: NULL for all)
filter_quality	Logical; whether to keep only high-quality signatures (is_hiq == 1) (default: TRUE)
verbose	Logical; whether to print progress messages (default: TRUE)
landmark	Logical; whether to restrict to landmark genes only (default: TRUE)

Details

Duplicate gene symbols are resolved by preferring landmark, then best inferred, then inferred features. Other feature spaces rank last and ties retain the first input row. A warning lists discarded and retained gene IDs. Gene symbols are not renamed or averaged. GCTX data are read in sorted index order and restored to the requested gene and signature order explicitly.

Value

A data frame with expression data for all signatures, with annotation columns indicating sample metadata. Metadata is stored as an attribute.

Examples

```
# Build a tiny GCTX file so the example is self-contained and runnable.
gctx_file <- tempfile(fileext = ".gctx")
rhdf5::h5createFile(gctx_file)
for (group in c("0", "0/DATA", "0/DATA/0", "0/META",
                "0/META/ROW", "0/META/COL")) {
  rhdf5::h5createGroup(gctx_file, group)
}
rhdf5::h5write(matrix(c(1.2, -0.4, 0.7, -1.1), nrow = 2),
               gctx_file, "0/DATA/0/matrix")
rhdf5::h5write(c("1", "2"), gctx_file, "0/META/ROW/id")
rhdf5::h5write(c("SIG_A", "SIG_B"), gctx_file, "0/META/COL/id")

siginfo <- data.frame(
  sig_id = c("SIG_A", "SIG_B"), is_hiq = 1,
  pert_itime = "24 h", pert_idose = "1 uM", cell_iname = "A375"
)
geneinfo <- data.frame(
  gene_id = c("1", "2"), gene_symbol = c("GENE1", "GENE2"),
  feature_space = "landmark"
)
reference_df <- extract_cmap_data_from_siginfo(
  siginfo_file = siginfo,
  geneinfo_file = geneinfo,
  gctx_file = gctx_file,
  verbose = FALSE
)
reference_df
attr(reference_df, "metadata")
unlink(gctx_file)
```

`extract_compound_id` *Extract Compound ID from Signature Matching Results*

Description

Helper function to extract compound identifiers from complex compound strings in signature matching results. This function can be customized based on the specific format of compound identifiers in your data.

Usage

```
extract_compound_id(
  compound_strings,
  method = "split_colon",
  regex_pattern = NULL,
  part_index = 2
)
```

Arguments

`compound_strings`	Vector of compound identifier strings
`method`	Method for extraction: "split_colon" (default), "split_underscore", or "regex"
`regex_pattern`	Regular expression pattern for extraction (used when method="regex")
`part_index`	Which part to extract when splitting (default: 2)

Value

Vector of extracted compound identifiers

Examples

```
extract_compound_id("A:BRD-K03652504-001-01-9:10")


# Example compound strings
compounds <- c("CVD001_HEPG2_6H:BRD-K03652504-001-01-9:10.0497",
               "CVD001_HEPG2_6H:BRD-A37828317-001-03-0:10")

# Extract using colon splitting (default)
ids <- extract_compound_id(compounds)

# Extract using custom regex
ids <- extract_compound_id(compounds, method = "regex",
                           regex_pattern = "BRD-[A-Z0-9-]+")
```

`extract_signature_zscores` *Extract z-score matrix for barcode heatmap*

Description

Performs all data-loading and GCTX-extraction steps needed for `plot_signature_direction_tile_barcode` without producing any plot. The returned matrix can be inspected, filtered, exported, or passed directly to `plot_signature_direction_tile_barcode(precomputed = )` to avoid re-reading large files on repeated calls.

Usage

```
extract_signature_zscores(
  results_df,
  signature_file,
  reference_df = NULL,
  gctx_file = NULL,
  geneinfo_file = NULL,
  siginfo_file = NULL,
  data_dir = getOption("CONCERTDR.data_dir", NULL),
  selected_drug = NULL,
  selected_drug_col = NULL,
  pert_id_col = "sig_id",
  score_col = "Score",
  max_genes = 100,
  max_perts = 60,
  split_direction = FALSE,
  output_zscores = NULL,
  verbose = TRUE,
  direction = c("reversal", "mimic")
)
```

Arguments

`results_df`	Data frame containing at least perturbation id and score columns.
`signature_file`	Path to a signature file with gene and log2FC columns, or a data frame with `Gene` and `log2FC` columns.
`reference_df`	Optional reference matrix with genes as rows and perturbation ids as columns. If supplied, z-scores are taken directly from this object and no GCTX file is required.
`gctx_file`	Optional path to GCTX expression file.
`geneinfo_file`	Optional path to geneinfo file.
`siginfo_file`	Optional path to siginfo file for readable labels.
`data_dir`	Optional directory containing CMap files.
`selected_drug`	Optional drug identifier to filter `results_df`.
`selected_drug_col`	Column used with `selected_drug`.

pert_id_col	Perturbation id column (default: "sig_id").
score_col	Score column (default: "Score").
max_genes	Maximum number of signature genes to use (default: 100). When split_direction = FALSE (default) the first max_genes rows of the signature file are taken. When split_direction = TRUE the top max_genes up-regulated genes (by log2FC) and the top max_genes down-regulated genes (by |log2FC|) are selected independently, so the matrix may contain up to 2 * max_genes gene columns in total.
max_perts	Maximum number of perturbations (default: 60).
split_direction	Logical; if TRUE, apply max_genes separately to the up-regulated (log2FC > 0) and down-regulated (log2FC < 0) gene sets rather than to the combined signature. The result contains up to 2 * max_genes columns ordered down-regulated then up-regulated, matching the layout expected by plot_signature_direction_tile_barcode with split_direction = TRUE. Default: FALSE.
output_zscores	Optional TSV path to save the matrix. NULL to skip.
verbose	Logical; print progress messages.
direction	Order in which perturbations are picked by score: "reversal" (default) lowest first, "mimic" highest first. No perturbation is excluded by the sign of its score. Precomputed matrices are plotted as supplied; choose the direction when extracting them.

Value

A named list with elements:

z_plot Numeric matrix — rows = perturbations (labelled cmap_name | dose | time | cell), cols = genes (down→up order).

ordered_genes Character vector of gene symbols.

logfc_map Named numeric vector of signature log2FC values.

sig_ids Character vector of selected sig_ids.

sig_labels Character vector of human-readable labels.

sig_scores Numeric vector of score_col values in the same order as sig_ids and the rows of z_plot.

Examples

```
is.function(extract_signature_zscores)


# Requires the full CMap GCTX file (downloaded from clue.io)
sig_file <- system.file("extdata", "example_signature.txt",
                        package = "CONCERTDR")
ref_file <- system.file("extdata", "example_reference_df.csv",
                        package = "CONCERTDR")
ref_df <- read.csv(ref_file, row.names = 1, check.names = FALSE)
ref_df$gene_symbol <- rownames(ref_df)
demo_sig <- colnames(ref_df)[1]
zmat <- extract_signature_zscores(
  results_df     = data.frame(sig_id = demo_sig, Score = -0.72),
  signature_file = sig_file,
```

```
  reference_df   = ref_df,
  pert_id_col    = "sig_id"
)
```

`fast_parse_gctx` *Fast parser for GCTX matrices with optional row/column subsetting*

Description

Fast parser for GCTX matrices with optional row/column subsetting

Usage

```
fast_parse_gctx(fname, rid = NULL, cid = NULL)
```

Arguments

`fname`	Path to GCTX file
`rid`	Optional character vector of row ids (gene ids)
`cid`	Optional character vector of column ids (signature ids)

Value

Numeric matrix with selected rows/columns

`fast_read_meta` *Read first existing HDF5 dataset from candidate paths*

Description

Read first existing HDF5 dataset from candidate paths

Usage

```
fast_read_meta(fname, paths)
```

Arguments

`fname`	HDF5/GCTX file path
`paths`	Candidate dataset paths

Value

Dataset content

`fuzzy_drug_match` *Fuzzy String Matching for Drug Names*

Description

Performs fuzzy string matching to find the best matches between query drug names and a reference list of drug names. Uses Levenshtein distance for similarity calculation.

Usage

```
fuzzy_drug_match(
  query_names,
  reference_names,
  method = "levenshtein",
  threshold = 80,
  top_n = 1
)
```

Arguments

`query_names`	Vector of query drug names
`reference_names`	Vector of reference drug names to match against
`method`	Similarity method: "levenshtein" (default), "jaro", or "jarowinkler"
`threshold`	Minimum similarity threshold (0-100)
`top_n`	Number of top matches to return for each query (default: 1)

Value

Data frame with query names, matched names, and similarity scores

Examples

```
if (requireNamespace("RecordLinkage", quietly = TRUE)) {
  fuzzy_drug_match(c("asprin"), c("aspirin", "ibuprofen"), threshold = 70)
}


if (requireNamespace("RecordLinkage", quietly = TRUE)) {
  query <- c("aspirin", "ibuprofen", "acetaminophen")
  reference <- c("aspirin", "ibuprofen", "acetaminophen", "naproxen", "diclofenac")
  matches <- fuzzy_drug_match(query, reference, threshold = 80)
  print(matches)
}
```

get_rid *Get Unique Gene IDs and Gene Names*

Description

Get Unique Gene IDs and Gene Names

Usage

```
get_rid(geneinfo_df, landmark = TRUE)
```

Arguments

geneinfo_df	Data frame containing gene information
landmark	Restrict to landmark genes before resolving duplicate symbols.

Value

List with rid (gene IDs) and genenames (gene symbols)

plot.cmap_signature_result *Plot method for cmap_signature_result objects*

Description

Plot method for cmap_signature_result objects

Usage

```
## S3 method for class 'cmap_signature_result'
plot(x, method = NULL, plot_type = "scores", top_n = 20, ...)
```

Arguments

x	A cmap_signature_result object
method	Which method to plot (default: first method in results)
plot_type	Type of plot: "scores", "volcano", or "heatmap" (default: "scores")
top_n	Number of compounds to include (default: 20)
...	Additional arguments passed to plotting functions

Value

A ggplot object

Examples

```
mock_res <- structure(
  list(
    results = list(
      ks = data.frame(compound = c("imatinib", "dasatinib", "doxorubicin"),
                      Score = c(-0.72, -0.45, 0.31),
                      pValue = c(0.002, 0.041, 0.280),
                      pAdjValue = c(0.020, 0.205, 0.560),
                      stringsAsFactors = FALSE)
    ),
    summary = data.frame(compound = "imatinib", method = "ks",
                         Score = -0.72, pValue = 0.002, global_rank = 1L,
                         stringsAsFactors = FALSE),
    gene_data    = data.frame(Gene = c("TP53", "MYC"),
                              log2FC = c(1.5, -2.1), stringsAsFactors = FALSE),
    settings     = list(signature_file = "ex.txt",
                        time_completed = Sys.time(), time_taken_mins = 0.1,
                        methods = "ks", permutations = 10),
    common_genes = list(up   = list(found = "TP53", count = 1L, percent = 50),
                        down = list(found = "MYC",  count = 1L, percent = 50))
  ), class = "cmap_signature_result"
)
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plot(mock_res, method = "ks", plot_type = "scores")
}
```

---

plot_signature_direction_tile_barcode

Plot 2D Barcode Heatmap from GCTX (Single Drug Context)

Description

Build a 2D barcode heatmap using z-scores extracted from a GCTX file. Genes are ordered by signature direction (down then up), and perturbations are chosen as top `max_perts` rows by best (lowest) `score_col` from the provided technical results table.

Usage

```
plot_signature_direction_tile_barcode(
  results_df = NULL,
  signature_file = NULL,
  reference_df = NULL,
  gctx_file = NULL,
  geneinfo_file = NULL,
  siginfo_file = NULL,
  data_dir = getOption("CONCERTDR.data_dir", NULL),
  selected_drug = NULL,
  selected_drug_col = NULL,
  pert_id_col = "sig_id",
  score_col = "Score",
  max_genes = 100,
```

```
  max_perts = 60,
  cluster_rows = TRUE,
  cluster_method = "complete",
  show_row_dendrogram = TRUE,
  cluster_cols = FALSE,
  cluster_method_cols = "complete",
  show_col_dendrogram = TRUE,
  split_direction = FALSE,
  gap_width = 5,
  precomputed = NULL,
  save_png = FALSE,
  output_png = "barcode_heatmap.png",
  output_zscores = NULL,
  width = NULL,
  height = NULL,
  dpi = 150,
  verbose = TRUE,
  direction = c("reversal", "mimic")
)
```

Arguments

results_df	Data frame (typically tech_view_all) containing at least perturbation id and score columns.
signature_file	Path to a signature file with gene and log2FC columns, or a data frame with Gene and log2FC columns.
reference_df	Optional reference matrix with genes as rows and perturbation ids as columns. If supplied, no GCTX file is needed and the heatmap is built directly from this matrix.
gctx_file	Optional path to GCTX expression file. If NULL, the function tries (in order): option CONCERTDR.gctx_file, env var CONCERTDR_GCTX_FILE, data_dir/level5_beta_all_n120 then a file with that name in the working directory.
geneinfo_file	Optional path to geneinfo file with gene_symbol and gene_id columns. If NULL, the function tries option CONCERTDR.geneinfo_file, env var CONCERTDR_GENEINFO_FILE, data_dir/geneinfo_beta.txt, then geneinfo_beta.txt in the working directory.
siginfo_file	Optional path to siginfo file for readable perturbation labels. If NULL, the function tries option CONCERTDR.siginfo_file, env var CONCERTDR_SIGINFO_FILE, data_dir/siginfo_beta.txt, then siginfo_beta.txt in the working directory.
data_dir	Optional directory containing CMap files. Useful to avoid repeatedly passing full file paths.
selected_drug	Optional drug identifier to filter results_df before selecting perturbations.
selected_drug_col	Optional column used with selected_drug. If NULL, auto-detect from perturbation_name, display_name, pert_id, cmap_name, pert_name.
pert_id_col	Perturbation id column in results_df (default: "sig_id"; falls back to "compound" for older results).
score_col	Score column in results_df (default: "Score").
max_genes	Maximum number of signature genes to use (default: 100).

max_perts	Maximum number of perturbations to show (default: 60).
cluster_rows	Logical; cluster perturbation rows (default: TRUE).
cluster_method	Clustering linkage method for row clustering (default: "complete").
show_row_dendrogram	Logical; whether to draw a row dendrogram when cluster_rows = TRUE (default: TRUE).
cluster_cols	Logical; cluster gene columns (default: FALSE - genes are kept in signature direction order: down then up).
cluster_method_cols	Clustering linkage method for column clustering (default: "complete").
show_col_dendrogram	Logical; whether to draw a column dendrogram when cluster_cols = TRUE (default: TRUE).
split_direction	Logical; if TRUE, split the heatmap into two side-by-side panels — up-regulated genes (log2FC > 0) on the left and down-regulated genes (log2FC < 0) on the right — with a small gap between them. Row order is determined by clustering the full matrix so both panels stay aligned. Default: FALSE (original single-panel behaviour).
gap_width	Width of the gap between the two panels in mm when split_direction = TRUE. Default: 5.
precomputed	Optional; the output of extract_signature_zscores. If provided, skips all data-loading and GCTX-extraction steps - all data-related arguments (results_df, signature_file, gctx_file, etc.) are ignored.
save_png	Logical; save PNG output (default: FALSE).
output_png	Output PNG path.
output_zscores	Optional TSV path for exported z-score matrix. Default is NULL, so no file is written unless explicitly requested.
width, height	Figure dimensions in inches for PNG. If NULL (default), dimensions are computed automatically from the number of genes and perturbations (~0.22 in/gene width, ~0.28 in/perturbation height).
dpi	PNG resolution.
verbose	Logical; print progress messages.
direction	Order in which perturbations are picked by score: "reversal" (default) lowest first, "mimic" highest first. No perturbation is excluded by the sign of its score. Precomputed matrices are plotted as supplied; choose the direction when extracting them.

Value

Invisibly returns a list containing the plotted matrix, selected perturbation ids, and file paths.

Examples

```
is.function(plot_signature_direction_tile_barcode)


if (requireNamespace("ComplexHeatmap", quietly = TRUE) &&
    requireNamespace("circlize", quietly = TRUE)) {
```

```
  sig_file <- system.file("extdata", "example_signature.txt",
                          package = "CONCERTDR")
  ref_df <- read.csv(system.file("extdata", "example_reference_df.csv",
                                 package = "CONCERTDR"),
                     row.names = 1, check.names = FALSE)
  ref_df$gene_symbol <- rownames(ref_df)
  demo_sig <- colnames(ref_df)[1]
  plot_signature_direction_tile_barcode(
    results_df    = data.frame(sig_id = demo_sig, Score = -0.72),
    signature_file = sig_file,
    reference_df  = ref_df,
    pert_id_col   = "sig_id"
  )
}
```

`print.cmap_signature_result` *Print method for cmap_signature_result objects*

Description

Print method for cmap_signature_result objects

Usage

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

Arguments

x	A cmap_signature_result object
...	Additional arguments (not used)

Value

x invisibly

Examples

```
mock_res <- structure(
  list(
    results = list(
      ks = data.frame(compound = c("imatinib", "dasatinib"),
                      Score = c(-0.72, -0.45), pValue = c(0.002, 0.041),
                      pAdjValue = c(0.020, 0.205), stringsAsFactors = FALSE)
    ),
    summary = data.frame(compound = "imatinib", method = "ks",
                         Score = -0.72, pValue = 0.002, global_rank = 1L,
                         stringsAsFactors = FALSE),
    gene_data    = data.frame(Gene = c("TP53", "MYC"),
                              log2FC = c(1.5, -2.1), stringsAsFactors = FALSE),
    settings     = list(signature_file = "ex.txt",
```

```
                        time_completed = Sys.time(), time_taken_mins = 0.1,
                        methods = "ks", permutations = 10),
    common_genes = list(up   = list(found = "TP53", count = 1L, percent = 50),
                        down = list(found = "MYC",  count = 1L, percent = 50))
  ), class = "cmap_signature_result"
)
print(mock_res)
```

`process_combination` *Process a Single Combination of Time, Dose, and Cell Line*

Description

Process a Single Combination of Time, Dose, and Cell Line

Usage

```
process_combination(
  combination,
  rid,
  genenames,
  sig_info,
  gctx_file = "level5_beta_trt_cp_n720216x12328.gctx",
  output_dir = "."
)
```

Arguments

`combination`	Data frame row containing time, dose, and cell line information
`rid`	Vector of gene IDs to extract
`genenames`	Vector of gene names corresponding to the gene IDs
`sig_info`	Data frame containing signature information
`gctx_file`	Path to the GCTX file
`output_dir`	Directory to save output files

Value

Path to the output file (invisibly)

process_combinations_file *Process Multiple Combinations of Time, Dose, and Cell Line*

Description

Process Multiple Combinations of Time, Dose, and Cell Line

Usage

```
process_combinations_file(
  combinations_file,
  task_id = NULL,
  geneinfo_file = "geneinfo_beta.txt",
  siginfo_file = "siginfo_beta.txt",
  gctx_file = "level5_beta_trt_cp_n720216x12328.gctx",
  output_dir = "."
)
```

Arguments

combinations_file	Path to a file containing combinations to process
task_id	Specific task ID to process (for SLURM array jobs)
geneinfo_file	Path to the gene info file
siginfo_file	Path to the signature info file
gctx_file	Path to the GCTX file
output_dir	Directory to save output files

Value

List of processed files (invisibly)

Examples

```
combinations <- data.frame(
  itime = "24 h",
  idose = "1 uM",
  cell = "K562",
  stringsAsFactors = FALSE
)
combinations_file <- tempfile(fileext = ".tsv")
utils::write.table(
  combinations,
  combinations_file,
  sep = "\t",
  row.names = FALSE,
  quote = FALSE
)
file.exists(combinations_file)
```

```
# With real CMap files available locally, run:
# process_combinations_file(
#   combinations_file = combinations_file,
#   geneinfo_file = "path/to/geneinfo_beta.txt",
#   siginfo_file = "path/to/siginfo_beta.txt",
#   gctx_file = "path/to/level5_beta_all_n1201944x12328.gctx",
#   output_dir = tempdir()
# )
```

`process_signature_with_df` *Process drug response signatures against reference data in a dataframe*

Description

Process drug response signatures against reference data in a dataframe

Usage

```
process_signature_with_df(
  signature_file,
  reference_df,
  output_dir = "results",
  permutations = 100,
  methods = c("ks", "xcos", "xsum", "gsea0", "gsea1", "gsea2", "zhang"),
  topN = 4,
  read_method = "auto",
  save_files = FALSE,
  camsum_alternative = c("two.sided", "greater", "less"),
  camsum_rho_bar = NULL,
  direction = c("reversal", "mimic"),
  vectorized = TRUE
)
```

Arguments

`signature_file`	Path to a signature gene list file with log2FC values, or a data frame with columns `Gene` and `log2FC`. When a data frame is supplied the `read_method` argument is ignored. You can create a suitable data frame from gene lists using `create_signature_from_gene_lists`.
`reference_df`	Dataframe containing reference data (combined across all conditions)
`output_dir`	Directory for output files (default: "results")
`permutations`	Number of permutations for statistical testing (default: 100)
`methods`	Vector of method names to run (default: the seven permutation methods) Options: "ks", "xcos", "xsum", "gsea0", "gsea1", "gsea2", "zhang", "camsum". "camsum" is not run by default; add it explicitly.
`topN`	Integer; number of top and bottom genes retained from each reference profile for XCos and XSum (default: 4). Does not truncate the query.

read_method	Character; method to use for reading signature file ("auto", "fread", or "read.table") (default: "auto")
save_files	Logical; whether to save results to files (default: FALSE)
camsum_alternative	Alternative hypothesis for the CamSum analytic p-value: "two.sided" (default, same convention as the permutation methods), "greater" (mimic) or "less" (reversal). Ignored by other methods.
camsum_rho_bar	Optional mean inter-gene correlation for CamSum, used for all profiles. If NULL (default), CamSum estimates it on every profile of the same perturbation type in the GCTX file (options CONCERTDR.gctx_file, CONCERTDR.siginfo_file and CONCERTDR.geneinfo_file must be set; the type is read from the "metadata" attribute of reference_df or from siginfo), and otherwise on reference_df with a warning. Ignored by other methods.
direction	Ordering of hits: "reversal" (default) ranks the lowest scores first, "mimic" the highest first. It only orders rank, the summary and plot(); no result is dropped by the sign of its score, and p-values and camsum_alternative are unaffected. Cross-method global_rank compares each hit's rank percentile among all profiles of its method, not raw score magnitudes.
vectorized	Whether the seven permutation methods use the vectorised engine (default: TRUE); see score_ks. It only changes how the scores are computed, not the formulas, and is skipped automatically when reference_df has missing values or duplicated genes. Use FALSE to reproduce results of earlier versions exactly.

Details

CamSum uses the analytic null described in compute_camsum_rho; topN and permutations do not apply to it. Its p-values were calibrated with $\bar{\rho}$ estimated on whole LINCS trt_oe / trt_xpr libraries and are anti-conservative on trt_cp. settings$camsum$rho_source records where $\bar{\rho}$ came from for each perturbation type.

Value

A structured list containing:

results	List containing results for each method
summary	Data frame of top hits across all methods
gene_data	Original signature gene data
settings	List of parameters used for the analysis. When CamSum is run, settings$camsum holds k, kU, kD, alternative, and rho_bar, VIF, rho_source and rho_n_profiles per perturbation type.
common_genes	Counts of genes found in reference data

Use print() or summary() methods to get a quick overview of results

Examples

```
sig_file <- system.file("extdata", "example_signature.txt",
                        package = "CONCERTDR")
ref_file <- system.file("extdata", "example_reference_df.csv",
                        package = "CONCERTDR")
ref_df <- read.csv(ref_file, row.names = 1, check.names = FALSE)
ref_df$gene_symbol <- rownames(ref_df)
```

```
results <- process_signature_with_df(
  signature_file = sig_file,
  reference_df = ref_df,
  output_dir = tempdir(),
  permutations = 10,
  methods = "ks",
  save_files = FALSE
)
summary(results, top_n = 3)

# You can also pass a data.frame directly:
sig_df <- read.delim(sig_file)
results2 <- process_signature_with_df(
  signature_file = sig_df,
  reference_df = ref_df,
  permutations = 10,
  methods = "ks",
  save_files = FALSE
)
```

`score_camsum` *CamSum connectivity score*

Description

Correlation-adjusted sum of reference z-scores over the query genes, with an analytic standard-normal p-value (no permutations). For profile i with mean $\bar{v}_i$ and standard deviation s_i over all genes of `refMatrix` and a query of $k = k_U + k_D$ genes found in `refMatrix`,

$$T_i = \frac{\sum_{g \in U} z_{gi} - \sum_{g \in D} z_{gi} - (k_U - k_D)\,\bar{v}_i}{s_i \sqrt{k \cdot \mathrm{VIF}}}, \qquad \mathrm{VIF} = \max\{1,\ 1 + (k-1)\bar{\rho}\}.$$

The subtracted term centres the sum at its expectation when the query genes behave like random genes of the profile: each gene then has mean $\bar{v}_i$, so the up-minus-down sum has expectation $(k_U - k_D)\bar{v}_i$. It vanishes when $k_U = k_D$ or $\bar{v}_i = 0$, and it makes the score unchanged when a constant is added to every gene of a profile. The floor at 1 matches `limma::camera` with `allow.neg.cor = FALSE`. Positive scores mean the profile mimics the query, negative scores mean it reverses it.

Usage

```
score_camsum(
  refMatrix,
  queryUp,
  queryDown,
  alternative = c("two.sided", "greater", "less"),
  rho_bar = NULL,
  pAdjMethod = "BH",
  chunk = 2000L,
  pert_type = NULL
)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
queryUp, queryDown	Character vectors of gene symbols. Genes present in both are dropped from both.
alternative	"two.sided" (default, same convention as the permutation methods), "greater" (mimic) or "less" (reversal).
rho_bar	Optional precomputed $\bar{\rho}$, used for all profiles.
pAdjMethod	P-value adjustment method (default: "BH").
chunk	Number of columns processed at a time (default: 2000).
pert_type	Optional perturbation type of each column; looked up from the siginfo file in option CONCERTDR.siginfo_file when NULL.

Details

$\bar{\rho}$ is taken, in order of preference, from rho_bar; from every profile of the same perturbation type in the GCTX file named by option CONCERTDR.gctx_file, which must hold the whole library of that type (options CONCERTDR.siginfo_file and CONCERTDR.geneinfo_file are needed too; profiles of different types get their own $\bar{\rho}$); or, with a warning, the columns of refMatrix (see compute_camsum_rho). Only the p-values depend on this choice; the ranking of profiles within one perturbation type does not.

Calibration was checked on LINCS trt_oe and trt_xpr with library-level $\bar{\rho}$; on trt_cp the score distribution is left-skewed and p-values are anti-conservative, so use the score for ranking only there.

Value

Data frame with Score, pValue, pAdjValue per sample. Attributes k, kU, kD and alternative, plus rho_bar, VIF, rho_source and rho_n_profiles as vectors named by perturbation type.

---

score_gsea0 *GSEA weight-0 connectivity score*

Description

GSEA weight-0 connectivity score

Usage

```
score_gsea0(
  refMatrix,
  queryUp,
  queryDown,
  permuteNum = 10000,
  pAdjMethod = "BH",
  vectorized = TRUE
)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
queryUp	Character vector of up-regulated gene symbols.
queryDown	Character vector of down-regulated gene symbols.
permuteNum	Number of permutations (default: 10000).
pAdjMethod	P-value adjustment method (default: "BH").
vectorized	Whether to use the vectorised engine (default: TRUE), which scores blocks of samples at once. It is skipped automatically for reference matrices with missing values or duplicated gene names. Scores are the same as the row-by-row code; p-values of KS and GSEA weight 0 use one permutation null shared by all samples (same distribution, different random numbers), and the other methods reproduce the row-by-row p-values for the same seed. Use FALSE for the row-by-row code.

Value

Data frame with Score, pValue, pAdjValue per sample.

---

score_gsea1 *GSEA weight-1 connectivity score*

Description

GSEA weight-1 connectivity score

Usage

```
score_gsea1(
  refMatrix,
  queryUp,
  queryDown,
  permuteNum = 10000,
  pAdjMethod = "BH",
  vectorized = TRUE
)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
queryUp	Character vector of up-regulated gene symbols.
queryDown	Character vector of down-regulated gene symbols.
permuteNum	Number of permutations (default: 10000).
pAdjMethod	P-value adjustment method (default: "BH").
vectorized	Whether to use the vectorised engine (default: TRUE), which scores blocks of samples at once. It is skipped automatically for reference matrices with missing values or duplicated gene names. Scores are the same as the row-by-row code; p-values of KS and GSEA weight 0 use one permutation null shared by all samples (same distribution, different random numbers), and the other methods reproduce the row-by-row p-values for the same seed. Use FALSE for the row-by-row code.

Value

Data frame with Score, pValue, pAdjValue per sample.

`score_gsea2` *GSEA weight-2 connectivity score*

Description

GSEA weight-2 connectivity score

Usage

```
score_gsea2(
  refMatrix,
  queryUp,
  queryDown,
  permuteNum = 10000,
  pAdjMethod = "BH",
  vectorized = TRUE
)
```

Arguments

`refMatrix`	Numeric matrix with genes as rows and samples as columns.
`queryUp`	Character vector of up-regulated gene symbols.
`queryDown`	Character vector of down-regulated gene symbols.
`permuteNum`	Number of permutations (default: 10000).
`pAdjMethod`	P-value adjustment method (default: "BH").
`vectorized`	Whether to use the vectorised engine (default: TRUE), which scores blocks of samples at once. It is skipped automatically for reference matrices with missing values or duplicated gene names. Scores are the same as the row-by-row code; p-values of KS and GSEA weight 0 use one permutation null shared by all samples (same distribution, different random numbers), and the other methods reproduce the row-by-row p-values for the same seed. Use `FALSE` for the row-by-row code.

Value

Data frame with Score, pValue, pAdjValue per sample.

score_ks *Kolmogorov-Smirnov connectivity score*

Description

Kolmogorov-Smirnov connectivity score

Usage

```
score_ks(
  refMatrix,
  queryUp,
  queryDown,
  permuteNum = 10000,
  pAdjMethod = "BH",
  vectorized = TRUE
)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
queryUp	Character vector of up-regulated gene symbols.
queryDown	Character vector of down-regulated gene symbols.
permuteNum	Number of permutations (default: 10000).
pAdjMethod	P-value adjustment method (default: "BH").
vectorized	Whether to use the vectorised engine (default: TRUE), which scores blocks of samples at once. It is skipped automatically for reference matrices with missing values or duplicated gene names. Scores are the same as the row-by-row code; p-values of KS and GSEA weight 0 use one permutation null shared by all samples (same distribution, different random numbers), and the other methods reproduce the row-by-row p-values for the same seed. Use FALSE for the row-by-row code.

Value

Data frame with Score, pValue, pAdjValue per sample.

score_xcos *Extreme cosine similarity score*

Description

Extreme cosine similarity score

Usage

```
score_xcos(
  refMatrix,
  query,
  topN = 500,
  permuteNum = 10000,
  pAdjMethod = "BH",
  vectorized = TRUE
)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
query	Named numeric vector (gene symbols → fold-change or rank).
topN	Number of top/bottom genes per reference profile (default: 500).
permuteNum	Number of permutations (default: 10000).
pAdjMethod	P-value adjustment method (default: "BH").

Value

Data frame with Score, pValue, pAdjValue per sample.

---

score_xsum *Extreme sum score*

Description

Extreme sum score

Usage

```
score_xsum(
  refMatrix,
  queryUp,
  queryDown,
  topN = 500,
  permuteNum = 10000,
  pAdjMethod = "BH",
  vectorized = TRUE
)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
queryUp	Character vector of up-regulated gene symbols.
queryDown	Character vector of down-regulated gene symbols.
topN	Number of top/bottom genes per reference profile (default: 500).
permuteNum	Number of permutations (default: 10000).

pAdjMethod	P-value adjustment method (default: "BH").
vectorized	Whether to use the vectorised engine (default: TRUE), which scores blocks of samples at once. It is skipped automatically for reference matrices with missing values or duplicated gene names. Scores are the same as the row-by-row code; p-values of KS and GSEA weight 0 use one permutation null shared by all samples (same distribution, different random numbers), and the other methods reproduce the row-by-row p-values for the same seed. Use FALSE for the row-by-row code.

Value

Data frame with Score, pValue, pAdjValue per sample.

---

score_zhang *Zhang connectivity score*

Description

Zhang connectivity score

Usage

```
score_zhang(
  refMatrix,
  queryUp,
  queryDown,
  permuteNum = 10000,
  pAdjMethod = "BH",
  vectorized = TRUE
)
```

Arguments

refMatrix	Numeric matrix with genes as rows and samples as columns.
queryUp	Character vector of up-regulated gene symbols.
queryDown	Character vector of down-regulated gene symbols.
permuteNum	Number of permutations (default: 10000).
pAdjMethod	P-value adjustment method (default: "BH").
vectorized	Whether to use the vectorised engine (default: TRUE), which scores blocks of samples at once. It is skipped automatically for reference matrices with missing values or duplicated gene names. Scores are the same as the row-by-row code; p-values of KS and GSEA weight 0 use one permutation null shared by all samples (same distribution, different random numbers), and the other methods reproduce the row-by-row p-values for the same seed. Use FALSE for the row-by-row code.

Value

Data frame with Score, pValue, pAdjValue per sample.

scoring_methods *Internal Connectivity Scoring Methods*

Description

Self-contained implementations of connectivity scoring methods for matching disease gene expression signatures to compound-induced gene expression profiles.

The seven permutation-based methods share one framework (`.permutation_test`) for computing p-values and adjusted p-values. CamSum uses an analytic null instead; see `score_camsum`.

Value

None; this is an internal documentation topic.

References

Lamb J et al. Science, 2006, 313(5795): 1929-1935 (KS method). Subramanian A et al. PNAS, 2005, 102(43): 15545-15550 (GSEA methods). Cheng J et al. Genome Medicine, 2014, 6(12): 95 (XCos, XSum methods). Zhang S D et al. BMC Bioinformatics, 2008, 9(1): 258 (Zhang method).

subset_siginfo_beta *Subset siginfo_beta file interactively with smart preview*

Description

Enhanced interactive filtering that shows what options will be available in subsequent parameters based on your current selection, helping you make informed decisions about the filtering sequence.

Usage

```
subset_siginfo_beta(
  siginfo_file,
  output_file = NULL,
  interactive = TRUE,
  filters = NULL,
  verbose = TRUE,
  show_preview = TRUE
)
```

Arguments

`siginfo_file`	Path to siginfo_beta.txt file
`output_file`	Path to save the filtered siginfo file (optional)
`interactive`	Logical; whether to run in interactive mode (default: TRUE)
`filters`	List of pre-defined filters to apply in non-interactive mode
`verbose`	Logical; whether to print progress messages (default: TRUE)
`show_preview`	Logical; whether to show preview of subsequent options (default: TRUE)

Value

A filtered data frame of the siginfo file

Examples

```
ex_sig <- data.frame(
  pert_type = c("trt_cp", "trt_cp", "trt_sh"),
  pert_itime = c("6 h", "24 h", "24 h"),
  pert_idose = c("10 uM", "10 uM", "5 uM"),
  cell_iname = c("A375", "MCF7", "A375")
)
ex_sig_file <- tempfile(fileext = ".txt")
write.table(ex_sig, ex_sig_file, sep = "\t", row.names = FALSE, quote = FALSE)
subset_siginfo_beta(
  ex_sig_file,
  interactive = FALSE,
  filters = list(pert_type = "trt_cp"),
  verbose = FALSE,
  show_preview = FALSE
)


ex_file <- system.file("extdata", "example_siginfo.txt", package = "CONCERTDR")

# Interactive mode (run only in an interactive R session)
if (interactive() && nzchar(ex_file)) {
  filtered_siginfo <- subset_siginfo_beta(ex_file)
}

# Non-interactive mode with pre-defined filters
if (nzchar(ex_file)) {
  filtered_siginfo <- subset_siginfo_beta(
    ex_file,
    interactive = FALSE,
    filters = list(
      pert_type = "trt_cp",
      pert_itime = c("6 h", "24 h"),
      pert_idose = "10 uM",
      cell_iname = c("A375", "MCF7")
    ),
    verbose = FALSE,
    show_preview = FALSE
  )
}
```

---

summary.cmap_signature_result

Summary method for cmap_signature_result objects

Description

Summary method for cmap_signature_result objects

Usage

```
## S3 method for class 'cmap_signature_result'
summary(object, top_n = 10, ...)
```

Arguments

object	A cmap_signature_result object
top_n	Number of top compounds to show (default: 10)
...	Additional arguments (not used)

Value

A data frame with the top compounds

Examples

```
mock_res <- structure(
  list(
    results = list(
      ks = data.frame(compound = c("imatinib", "dasatinib"),
                      Score = c(-0.72, -0.45), pValue = c(0.002, 0.041),
                      pAdjValue = c(0.020, 0.205), stringsAsFactors = FALSE)
    ),
    summary = data.frame(compound = c("imatinib", "dasatinib"),
                         method = c("ks", "ks"),
                         Score = c(-0.72, -0.45), pValue = c(0.002, 0.041),
                         global_rank = 1:2, stringsAsFactors = FALSE),
    gene_data    = data.frame(Gene = c("TP53", "MYC"),
                              log2FC = c(1.5, -2.1), stringsAsFactors = FALSE),
    settings     = list(signature_file = "ex.txt",
                        time_completed = Sys.time(), time_taken_mins = 0.1,
                        methods = "ks", permutations = 10),
    common_genes = list(up   = list(found = "TP53", count = 1L, percent = 50),
                        down = list(found = "MYC",  count = 1L, percent = 50))
  ), class = "cmap_signature_result"
)
summary(mock_res, top_n = 2)
```

Index

∗ **internal**
.add_pert_kind_and_mode, 4
.apply_filters, 5
.apply_score, 5
.as_numeric, 6
.barcode_fig_size, 6
.barcode_gp, 7
.barcode_panel, 7
.barcode_style, 8
.build_drug_context_summary, 9
.build_tech_view, 9
.build_wetlab_drug_view, 10
.build_wetlab_gene_view, 10
.camsum_blocks, 11
.camsum_cached, 11
.camsum_col_sd, 12
.camsum_file_key, 12
.camsum_gctx_columns, 13
.camsum_option_file, 13
.camsum_pert_types, 14
.camsum_read_meta_file, 14
.camsum_rho, 15
.camsum_rho_gctx, 15
.camsum_rho_stream, 16
.camsum_rows, 16
.camsum_signed, 17
.camsum_warn_local_rho, 17
.check_topN, 18
.combine_up_down, 18
.count_pvalues, 19
.dedup_drug_rows, 19
.dedup_geneinfo, 20
.derive_annotations, 20
.direction_ordered_genes, 21
.draw_barcode, 21
.fast_chunk_size, 22
.fast_draws, 22
.fast_gsea0, 23
.fast_gsea0_es, 23
.fast_gseaW, 24
.fast_gseaW_es, 24
.fast_ks, 25
.fast_ks_es, 25
.fast_permutation_test, 26
.fast_positions, 26
.fast_shared_null_test, 27
.fast_sorted_set, 27
.fast_up_down_test, 28
.fast_xcos, 28
.fast_xsum, 29
.fast_zhang, 29
.fill_name_from_iname, 30
.filter_interactively, 30
.filter_siginfo, 31
.first_existing_file, 31
.gctx_ids, 32
.gctx_read_matrix, 32
.is_blank, 33
.list_column_values, 33
.load_signature, 34
.matrix_to_extreme_list, 34
.matrix_to_name_ranked_list, 35
.matrix_to_signed_rank_list, 35
.matrix_to_value_ranked_list, 36
.merge_compoundinfo, 36
.merge_siginfo, 37
.message_column_summary, 37
.message_rule, 38
.order_by_score, 38
.parse_compound_context, 39
.permutation_pvalues, 39
.permutation_test, 40
.permutation_test_up_down, 40
.pert_id_missing, 41
.print_filter_summary, 42
.print_reuse_filters, 42
.prompt_column_filter, 43
.rank_by_direction, 43
.read_cmap_table, 44
.read_gene_map, 44
.read_siginfo, 45
.read_signature, 45
.reference_matrix, 46
.reference_to_matrix, 46
.remaining_columns, 47
.resolve_cmap_file, 47

.run_score_method, 48
.save_siginfo, 48
.score_gsea, 49
.select_perturbations, 50
.show_value_preview, 50
.siginfo_ask, 51
.siginfo_labels, 51
.siginfo_pct, 52
.simplify_siginfo, 52
.tie_floor, 53
.use_fast, 53
.validate_ref_matrix, 54
.write_result_tables, 54
cmap_extract, 57
create_summary_from_results, 59
fast_parse_gctx, 64
fast_read_meta, 64
get_rid, 66
process_combination, 71
score_camsum, 75
score_gsea0, 76
score_gsea1, 77
score_gsea2, 78
score_ks, 79
score_xcos, 79
score_xsum, 80
score_zhang, 81
scoring_methods, 82
.add_pert_kind_and_mode, 4
.apply_filters, 5
.apply_score, 5
.as_numeric, 6
.barcode_fig_size, 6
.barcode_gp, 7
.barcode_panel, 7
.barcode_style, 8
.build_drug_context_summary, 9
.build_tech_view, 9
.build_wetlab_drug_view, 10
.build_wetlab_gene_view, 10
.camsum_blocks, 11
.camsum_cached, 11
.camsum_col_sd, 12
.camsum_file_key, 12
.camsum_gctx_columns, 13
.camsum_option_file, 13
.camsum_pert_types, 14
.camsum_read_meta_file, 14
.camsum_rho, 15
.camsum_rho_gctx, 15
.camsum_rho_stream, 16
.camsum_rows, 16
.camsum_signed, 17
.camsum_warn_local_rho, 17
.check_topN, 18
.combine_up_down, 18
.count_pvalues, 19
.dedup_drug_rows, 19
.dedup_geneinfo, 20
.derive_annotations, 20
.direction_ordered_genes, 21
.draw_barcode, 21
.fast_chunk_size, 22
.fast_draws, 22
.fast_gsea0, 23
.fast_gsea0_es, 23
.fast_gseaW, 24
.fast_gseaW_es, 24
.fast_ks, 25
.fast_ks_es, 25
.fast_permutation_test, 26
.fast_positions, 26
.fast_shared_null_test, 27
.fast_sorted_set, 27
.fast_up_down_test, 28
.fast_xcos, 28
.fast_xsum, 29
.fast_zhang, 29
.fill_name_from_iname, 30
.filter_interactively, 30
.filter_siginfo, 31
.first_existing_file, 31
.gctx_ids, 32
.gctx_read_matrix, 32
.is_blank, 33
.list_column_values, 33
.load_signature, 34
.matrix_to_extreme_list, 34
.matrix_to_name_ranked_list, 35
.matrix_to_signed_rank_list, 35
.matrix_to_value_ranked_list, 36
.merge_compoundinfo, 36
.merge_siginfo, 37
.message_column_summary, 37
.message_rule, 38
.order_by_score, 38
.parse_compound_context, 39
.permutation_pvalues, 39
.permutation_test, 40
.permutation_test_up_down, 40
.pert_id_missing, 41
.print_filter_summary, 42
.print_reuse_filters, 42
.prompt_column_filter, 43

.rank_by_direction, 43
.read_cmap_table, 44
.read_gene_map, 44
.read_siginfo, 45
.read_signature, 45
.reference_matrix, 46
.reference_to_matrix, 46
.remaining_columns, 47
.resolve_cmap_file, 47
.run_score_method, 48
.save_siginfo, 48
.score_gsea, 49
.select_perturbations, 50
.show_value_preview, 50
.siginfo_ask, 51
.siginfo_labels, 51
.siginfo_pct, 52
.simplify_siginfo, 52
.tie_floor, 53
.use_fast, 53
.validate_ref_matrix, 54
.write_result_tables, 54

annotate_drug_results, 55

cmap_extract, 57
compute_camsum_rho, 57, *74*, *76*
create_signature_from_gene_lists, 58, *73*
create_summary_from_results, 59

extract_cmap_data_from_siginfo, *14*, 59
extract_compound_id, 61
extract_signature_zscores, *58*, 62, *69*

fast_parse_gctx, 64
fast_read_meta, 64
fuzzy_drug_match, 65

get_rid, 66

plot.cmap_signature_result, 66
plot_signature_direction_tile_barcode, *62*, *63*, 67
print.cmap_signature_result, 70
process_combination, 71
process_combinations_file, 72
process_signature_with_df, *57*, *58*, 73

score_camsum, 75
score_gsea0, 76
score_gsea1, 77
score_gsea2, 78
score_ks, 79
score_xcos, 79
score_xsum, 80
score_zhang, 81
scoring_methods, 82
subset_siginfo_beta, 82
summary.cmap_signature_result, 83