

Package: perturbMatch (via r-universe)

October 13, 2026

Type Package

Title Match Transcriptomics Signatures to Single-Gene Perturbation References

Version 0.99.1

Description Matches a query transcriptomics signature against a reference database of single-gene perturbation signatures, ranking the perturbations whose expression response most resembles the query and so nominating the regulators most likely to drive the observed change. Query signatures derived from a differential expression result are scored using cosine, GSEA and signed-Jaccard similarity metrics. Reference databases are supplied as HDF5-backed SummarizedExperiment objects for memory-efficient handling of large perturbation collections, and helper functions rank and visualise the best-matching regulators.

Depends R (>= 4.5.0)

Imports methods, AnnotationDbi, org.Hs.eg.db, SummarizedExperiment, BiocFileCache, BiocParallel, ExperimentHub, dplyr, ggplot2, HDF5Array, fgsea, DelayedMatrixStats, DelayedArray, MatrixGenerics, S4Vectors, cowplot, ggrepel, rlang

License GPL-3

URL <https://github.com/soulj/perturbMatch>

BugReports <https://github.com/soulj/perturbMatch/issues>

biocViews GeneRegulation, Transcriptomics, Transcription, GeneSetEnrichment, GeneTarget, GeneExpression, DifferentialExpression

LazyData FALSE

VignetteBuilder knitr

RoxygenNote 7.3.2

Encoding UTF-8

Suggests testthat, curl, rmarkdown, BiocStyle, knitr, GO.db, airway, DESeq2, muscData, scuttle, SingleCellExperiment, singscore, scater, scales, tibble

Collate 'perturbMatch-package.R' 'perturbMatch_class.R' 'perturbMatch_get_reference_database.R' 'perturbMatch_prepareQuery.R' 'perturbMatch_calcEnrichment.R' 'method_cosine.R' 'method_gsea.R' 'method_jaccard.R' 'perturbMatch_plot_similarity.R' 'perturbMatch_regulator_go.R' 'perturbMatch_volcanoPlot.R'

Config/pak/sysreqs libicu-dev libpng-dev libssl-dev zlib1g-dev

Repository <https://biocstaging.r-universe.dev>

Date/Publication 2026-09-02 15:40:29 UTC

RemoteUrl <https://github.com/BiocStaging/perturbMatch>

RemoteRef HEAD

RemoteSha 48a25c74e5947a044ecdcb0ded10be4173bf0d9

Contents

background	2
calcEnrichment	3
downRegulated	5
enrichment	6
getReferenceDatabase	7
PerturbMatch-class	9
plotRegulatorGSEA	9
plotSimilarity	11
plotVolcano	12
prepareQuery	13
regulatorGSEA	15
regulatorTable	17
show,PerturbMatch-method	18
topRegulators	19
upRegulated	19
volcanoPlot	20
Index	22

background	<i>Background genes from a PerturbMatch object</i>
------------	--

Description

Returns the background gene set recorded in `metadata()` when the object was built by `calcEnrichment`. This is the universe of genes against which enrichment was scored.

Usage

```
background(object)
```

Arguments

object A PerturbMatch object.

Value

A character vector of gene identifiers, or NULL if unset.

Examples

```
pm <- methods::new(
  "PerturbMatch",
  SummarizedExperiment::SummarizedExperiment(
    assays = list(
      enrichment = matrix(0.5, dimnames = list("SIG1", "score"))
    ),
    metadata = list(background = c("GENE1", "GENE2", "GENE3")))
background(pm)
```

calcEnrichment	<i>Calculate enrichment</i>
----------------	-----------------------------

Description

Calculates the enrichment of a regulator expression signature in the query expression signature. This measure is used to prioritise which upstream regulators are driving the observed gene expression changes.

Usage

```
calcEnrichment(
  object,
  expressionMatrix = NULL,
  upRegulated = NULL,
  downRegulated = NULL,
  background = NULL,
  method = c("SignedJaccard", "GSEA", "cosine", "cosineExtreme"),
  refMetric = NULL,
  threshold = NULL,
  topN = 250L,
  topSig = 250L,
  BPPARAM = BiocParallel::SerialParam(progressbar = TRUE),
  chunkSize = NULL
)
```

Arguments

<code>object</code>	A SummarizedExperiment reference database with <code>metadata(object)\$ref_metric</code> set to one of "chrdir", "log2fc", or "pi".
<code>expressionMatrix</code>	Optional data frame with gene IDs in column 1 and score values in column 2. Required for "cosine" and "cosineExtreme" methods. If supplied, <code>upRegulated</code> , <code>downRegulated</code> , and <code>background</code> are not used.
<code>upRegulated</code>	Character vector of upregulated genes in the experiment. Only used when <code>expressionMatrix</code> is NULL.
<code>downRegulated</code>	Character vector of downregulated genes in the experiment. Only used when <code>expressionMatrix</code> is NULL.
<code>background</code>	Character vector of all genes expressed/measured in the experiment. Only used when <code>expressionMatrix</code> is NULL.
<code>method</code>	Enrichment method to use. One of "SignedJaccard", "GSEA", "cosine", "cosineExtreme". Defaults to "SignedJaccard".
<code>refMetric</code>	Reference metric to score against, one of <code>metadata(object)\$ref_metric</code> . When NULL (default), <code>metadata(object)\$default_metric</code> is used.
<code>threshold</code>	Optional numeric threshold applied to reference metric values to define reference gene sets ($ \text{metric} > \text{threshold}$). When NULL, <code>topN</code> is used instead. Default: NULL.
<code>topN</code>	Number of top and bottom genes per reference signature to use, ranked by the reference metric. Used only when <code>threshold</code> is NULL. Default: 250.
<code>topSig</code>	Number of top/bottom genes from the query signature to use when deriving gene sets from <code>expressionMatrix</code> . Default: 250.
<code>BPPARAM</code>	A BiocParallelParam object controlling parallelisation. Defaults to <code>BiocParallel::SerialParam(progress = TRUE)</code> .
<code>chunkSize</code>	Number of reference signatures (columns) per processing block. When NULL (default) a value giving roughly one block per worker in <code>BPPARAM</code> is chosen. Larger values use more memory, and <code>chunkSize >= ncol(object)</code> leaves a single block that cannot be parallelised.

Value

A [PerturbMatch](#) object containing the enrichment scores.

Input signature

Two input formats are supported:

- **Ranked list:** supply `expressionMatrix` (a data frame of gene IDs and score values). `upRegulated`, `downRegulated`, and `background` are ignored.
- **Gene sets:** supply `upRegulated`, `downRegulated`, and `background` as character vectors. `expressionMatrix` should be NULL.

If both are provided, `expressionMatrix` takes precedence.

Reference metric

Two metadata slots record which metrics a reference supports:

- `metadata(object)$ref_metric`: the available metrics, each naming a stored assay. limma references hold `c("pi", "log2fc")` and chrdir references hold `"chrdir"`.
- `metadata(object)$default_metric`: the metric used when `refMetric = NULL`.

Examples

```
# Load the small bundled demo reference database
se <- getReferenceDatabase(demo = TRUE)

# Score a query gene set against every reference signature. Here we just
# use a handful of reference genes to stand in for a real up/down
# signature.
result <- calcEnrichment(
  se,
  upRegulated = rownames(se)[seq_len(3)],
  downRegulated = rownames(se)[seq(4, 5)],
  background = rownames(se),
  method = "SignedJaccard",
  BPPARAM = BiocParallel::SerialParam(progressbar = FALSE)
)

cols <- c("pert_symbol", "pert_type", "score", "zscore")
topRegulators(result, n = 3)[, cols]
```

downRegulated	<i>Down-regulated genes from a PerturbMatch object</i>
---------------	--

Description

Returns the down-regulated gene set recorded in `metadata()` when the object was built by `calcEnrichment`.

Usage

```
downRegulated(object)
```

Arguments

`object` A `PerturbMatch` object.

Value

A character vector of gene identifiers, or `NULL` if unset.

Examples

```
pm <- methods::new(
  "PerturbMatch",
  SummarizedExperiment::SummarizedExperiment(
    assays = list(
      enrichment = matrix(0.5, dimnames = list("SIG1", "score"))
    ),
    metadata = list(down_regulated = c("GENE3", "GENE4"),
      background = c("GENE3", "GENE4"))))
downRegulated(pm)
```

enrichment

Enrichment scores from a PerturbMatch object

Description

Returns the matrix stored in the "enrichment" assay.

Usage

```
enrichment(object)
```

Arguments

object A PerturbMatch object.

Value

A matrix of enrichment scores (rows = features, columns = signatures).

Examples

```
mat <- matrix(runif(3), nrow = 3, dimnames = list(
  c("GENE1", "GENE2", "GENE3"), "score"
))
se <- SummarizedExperiment::SummarizedExperiment(
  assays = list(enrichment = mat)
)
S4Vectors::metadata(se) <- list(
  up_regulated = "GENE1",
  down_regulated = "GENE3",
  background = c("GENE1", "GENE2", "GENE3")
)
pm <- new("PerturbMatch", se)
enrichment(pm)
```

```
getReferenceDatabase
```

Load the perturbMatch reference database

Description

Retrieve a perturbMatch reference expression database as a [SummarizedExperiment::SummarizedExperiment-class] object.

Usage

```
getReferenceDatabase(
  demo = FALSE,
  type = c("chrdir", "limma"),
  batchCorrection = FALSE,
  source = c("auto", "experimenthub", "zenodo"),
  cacheDir = tools::R_user_dir("perturbMatch", "cache"),
  ask = interactive()
)
```

Arguments

demo	Logical scalar indicating whether to load the small bundled demonstration dataset instead of a full reference. Defaults to FALSE. The demo ignores type and batchCorrection. See Details for what it is and is not good for.
type	Character scalar specifying which reference database to load. One of "chrdir" (characteristic direction) or "limma" (log2 fold-change and adjusted p-values). Defaults to "chrdir".
batchCorrection	Logical scalar selecting the batch-corrected variant of a reference. Available for both type = "chrdir" and type = "limma": FALSE (default) loads the uncorrected signatures and TRUE loads signatures computed with study/batch effects removed (SVA).
source	Character scalar selecting where the full reference is retrieved from. "experimenthub" takes it from Bioconductor's ExperimentHub and errors if the records are not registered there. "zenodo" downloads it from the Zenodo deposit instead. "auto" (the default) uses ExperimentHub, falling back to Zenodo only when the records are absent from the hub, and announces which of the two it used. The records are registered for Bioconductor 3.24 and later, so an older release reaches the data through the Zenodo fallback. Ignored when demo = TRUE.
cacheDir	Directory holding the BiocFileCache cache of downloaded reference files. Defaults to a per-user cache directory (tools::R_user_dir("perturbMatch", "cache")). Only used by the Zenodo download path, as ExperimentHub manages its own cache.
ask	Logical scalar controlling whether BiocFileCache asks before creating cacheDir. Defaults to interactive(), so a session with a user present is prompted and a scripted one creates the directory. Declining the prompt, or passing TRUE outside

an interactive session, puts the cache under `tempdir()` and so re-downloads the reference in every session.

Details

Each reference is hosted as three files: an HDF5 file of the assay matrices (0.7 GB for `chrdir_sva` up to 3.1 GB for `limma`), a serialised `colData` of the signature annotation, and one gene annotation shared by all four references. They are fetched once, cached, and reassembled into a `SummarizedExperiment` whose assays are read from disk on demand.

The resources are registered on ExperimentHub, and documented by the **`perturbMatchData`** package, which does not need to be installed to use them. The signatures are derived from public NCBI GEO RNA-seq series and released under CC-BY-4.0, [doi:10.5281/zenodo.21792179](https://doi.org/10.5281/zenodo.21792179).

The bundled demo mirrors the structure of the full databases so the package can be used offline. It is a 21-signature subset tailored to the vignette examples. The rankings it produces are illustrative only. Any real analysis needs a full reference (`demo = FALSE`, the default).

Value

A [`SummarizedExperiment::SummarizedExperiment-class`] object with assay(s) stored as HDF5-backed delayed arrays. The object carries `metadata(object)$ref_metric`, which tells [calcEnrichment](#) how to compute the scoring matrix from the assay(s):

"chrdir" Single assay named "chrdir", used directly.

"log2fc" Assay named "log2fc", used directly.

"pi" Combined $\pi = \log_2 FC \times -\log_{10}(p_{adj})$ score stored alongside the "log2fc" and "padj" assays.

`ref_metric` is fixed when the reference database is built, travels with the data, and does not need to be set or modified by the user. Alongside it are `method`, `batch_correction`, `default_metric` and the `build_version` and `source_doi` of the deposit the reference came from. `metadata(object)$source` records how it was retrieved ("experimenthub", "zenodo" or "bundled"), and `metadata(object)$demo` whether it is the bundled demo database. All are carried through to the [PerturbMatch](#) object returned by [calcEnrichment](#).

Examples

```
# Load the bundled demonstration database (no download required)
se <- getReferenceDatabase(demo = TRUE)
se
S4Vectors::metadata(se)$ref_metric

# A full reference is a multi-gigabyte one-time download, so it is fetched
# only in an interactive session
if (interactive()) {
  seChrdir <- getReferenceDatabase(type = "chrdir")

  # chrdir reference, with batch-effect correction
  seChrdirBc <- getReferenceDatabase(
    type = "chrdir",
    batchCorrection = TRUE
```

```

    )

    # limma reference, without batch-effect correction
    seLimma <- getReferenceDatabase(type = "limma", batchCorrection = FALSE)
  }

```

PerturbMatch-class	<i>PerturbMatch enrichment results</i>
--------------------	--

Description

An S4 class extending [SummarizedExperiment](#) that holds the enrichment scores produced by [calcEnrichment](#). Scores are available through [enrichment\(\)](#), or as a tidy table through [regulatorTable\(\)](#). The usual SummarizedExperiment accessors also work.

Details of analysis are stored in [metadata\(\)](#) when the object is built. The query gene sets have accessors of their own ([upRegulated](#), [downRegulated](#), [background](#)). The scoring parameters are read straight from [metadata\(\)](#) and are printed by [show\(\)](#). The reference gene sets come from either threshold or topN.

The class contains an "enrichment" assay of numeric scores, gene sets held as character vectors, and a score_column entry. Objects failing any of these are rejected by [validObject](#).

Arguments

object	A PerturbMatch object returned by calcEnrichment .
--------	--

Value

- enrichment: matrix of enrichment scores.
- regulatorTable: tidy data.frame of scores plus row metadata.

plotRegulatorGSEA	<i>Dot plot of regulator GO enrichment</i>
-------------------	--

Description

Draws a dot plot of the GO terms returned by [regulatorGSEA](#). Terms are placed by normalised enrichment score, so those enriched among mimicking regulators (positive NES) sit to the right of zero and those enriched among reversing regulators (negative NES) to the left. Point size shows the GO set size and colour the adjusted p-value.

Usage

```
plotRegulatorGSEA(
  x,
  n = 15L,
  pvalueCutoff = 0.1,
  wrapWidth = 40,
  fontSize = 12
)
```

Arguments

<code>x</code>	A <code>data.frame</code> returned by regulatorGSEA .
<code>n</code>	Maximum number of terms to show. Taken by largest $ NES $ so both directions are represented. Default 15.
<code>pvalueCutoff</code>	Only terms with <code>padj</code> $\leq$ <code>pvalueCutoff</code> are shown. Default 0.1.
<code>wrapWidth</code>	Wrap long GO term labels to this many characters per line. NULL disables wrapping. Default 40.
<code>fontSize</code>	Base font size passed to theme_cowplot . Default 12.

Value

A `ggplot` object.

See Also

[regulatorGSEA](#)

Examples

```
se <- getReferenceDatabase(demo = TRUE)
res <- calcEnrichment(
  se,
  upRegulated = rownames(se)[seq_len(5)],
  downRegulated = rownames(se)[seq(6, 10)],
  background = rownames(se),
  BPPARAM = BiocParallel::SerialParam(progressbar = FALSE)
)
go <- regulatorGSEA(res, minSize = 1)
if (any(go$padj <= 0.5)) plotRegulatorGSEA(go, pvalueCutoff = 0.5)
```

plotSimilarity	<i>Plot signature similarity scores</i>
----------------	---

Description

Draws the top positive and top negative signatures as a lollipop plot.

Usage

```
plotSimilarity(object, topN = 10, inferActivity = FALSE)
```

Arguments

object	A SummarizedExperiment containing scores in its colData.
topN	Integer. Number of top positive and top negative hits to show.
inferActivity	Logical. If TRUE, flip the sign of the score for KD/KO perturbations so it reflects predicted gene activity rather than transcriptional similarity. If FALSE (default), the score keeps its similarity meaning and the perturbation direction is shown by point shape: up-pointing triangles for gain-of-function (OE), down-pointing triangles for loss-of-function (KD/KO).

Value

A ggplot object.

Examples

```
# A PerturbMatch object as returned by calcEnrichment(): one row per
# reference signature, scores in the 'enrichment' assay and the signature
# annotation in rowData().
enr <- cbind(
  score = c(0.9, -0.7, -0.4, 0.5, -0.2),
  zscore = c(1.6, -1.1, -0.6, 0.8, -0.3)
)
rownames(enr) <- paste0("SIG", seq_len(5))
se <- SummarizedExperiment::SummarizedExperiment(
  assays = list(enrichment = enr),
  rowData = S4Vectors::DataFrame(
    pert_symbol = c("GENE_A", "GENE_B", "GENE_C", "GENE_D", "GENE_E"),
    pert_type = c("OE", "KD", "KO", "OE", "KD")
  )
)
pm <- methods::new("PerturbMatch", se)
plotSimilarity(pm, topN = 3)
plotSimilarity(pm, topN = 3, inferActivity = TRUE)
```

plotVolcano

*Volcano Plot from a SummarizedExperiment Object***Description**

Extracts effect size and p-value assays for a given signature from a SummarizedExperiment and delegates to [volcanoPlot](#).

Usage

```
plotVolcano(
  object,
  signature,
  effectAssay = "logFC",
  pvalueAssay = "padj",
  numberPoints = 5,
  selectedPoints = NULL,
  logFCThreshold = log2(1.5),
  fdrThreshold = 0.05,
  pointSize = 2,
  labSize = 4,
  truncateInfPvalue = TRUE
)
```

Arguments

object	A SummarizedExperiment object. Must contain assays named by effectAssay and pvalueAssay, and rownames corresponding to gene identifiers.
signature	A single character string matching one of colnames(object), selecting the column to plot.
effectAssay	Character string giving the name of the assay that holds log2 fold-change values. Default: "logFC".
pvalueAssay	Character string giving the name of the assay that holds adjusted p-values (FDR). Default: "padj".
numberPoints	Integer. Number of top up- and down-regulated genes to label automatically (ranked by p-value, then llogFCI). Set to NULL to suppress automatic labelling. Default: 5.
selectedPoints	Character vector of gene IDs to label regardless of significance. Combined with numberPoints labels when both are supplied. Default: NULL.
logFCThreshold	Numeric. Absolute log2 fold-change cut-off used to classify genes as regulated. Default: log2(1.5).
fdrThreshold	Numeric. Adjusted p-value cut-off. Default: 0.05.
pointSize	Numeric. Size of plotted points passed to geom_point . Default: 2.
labSize	Numeric. Text size for gene labels passed to geom_text_repel . Default: 4.

truncateInfPvalue

Logical. If TRUE, genes with an adjusted p-value of exactly zero (an infinite $-\log_{10}$ FDR) are drawn at the largest finite $-\log_{10}$ FDR in the signature, so they appear as a flat row along the top of the plot. If FALSE, these genes are left at their true infinite value and dropped by the plotting layer. Default: TRUE.

Value

A ggplot object.

See Also

[volcanoPlot](#)

Examples

```
set.seed(1)
n <- 30
se <- SummarizedExperiment::SummarizedExperiment(
  assays = list(
    logFC = matrix(rnorm(n * 2), nrow = n,
      dimnames = list(paste0("Gene", seq_len(n)), c("A", "B"))),
    padj = matrix(runif(n * 2), nrow = n,
      dimnames = list(paste0("Gene", seq_len(n)), c("A", "B")))
  )
)

plotVolcano(se, signature = "A")
plotVolcano(se, signature = "A", selectedPoints = c("Gene1", "Gene2"))
```

```
prepareQuery
```

Prepare a differential expression result for calcEnrichment

Description

Extracts gene identifiers and log2 fold changes from a differential expression result, maps the identifiers to the Entrez gene IDs used by the perturbMatch reference databases, and reports the fraction of the query that was successfully mapped. The returned data frame can be passed directly to [calcEnrichment](#) as expressionMatrix.

Usage

```
prepareQuery(
  object,
  orgDb = "org.Hs.eg.db",
  keytype = NULL,
  reference = NULL,
  minMappedFraction = 0.5,
```

```

    dropLowExpressed = TRUE
  )

```

Arguments

object	A differential expression result. One of: • a DESeq2 <code>DESeqResults</code> object. • a <code>data.frame</code> (e.g. limma <code>topTable()</code> output, or a plain table read from a CSV/TSV) with gene identifiers in its row names or in a column, and a fold-change column named <code>"log2FoldChange"</code>, <code>"logFC"</code> or <code>"log2FC"</code>. An adjusted p-value column named <code>"padj"</code> or <code>"adj.P.Val"</code>, if present, is carried through. Columns are matched by name, not position, and any further columns are ignored. Only <i>adjusted</i> p-values are recognised, so a <code>"pvalue"</code> or <code>"P.Value"</code> column does not drive <code>dropLowExpressed</code>. Gene identifiers are taken from the row names when those are real names, as in <code>DESeqResults</code> and <code>topTable()</code> output. Positional row names, as produced by <code>data.frame()</code> and <code>read.csv()</code>, are ignored, as are all-digit row names when a column names itself as an identifier. The identifiers are then read from a column named like an identifier (<code>"gene"</code>, <code>"gene_id"</code>, <code>"symbol"</code>, <code>"entrezid"</code>, <code>"id"</code>, ...) if present, otherwise from the first column that is neither the fold-change nor the adjusted p-value column. Which of the two was used is reported in a message.
orgDb	An AnnotationDbi <code>OrgDb</code> object, or the name of an installed <code>OrgDb</code> package, used for ID mapping. Default <code>"org.Hs.eg.db"</code> .
keytype	The type of the query gene identifiers (e.g. <code>"ENSEMBL"</code> , <code>"SYMBOL"</code>), passed to <code>mapIds</code> . When <code>NULL</code> (default) it is guessed from the identifiers. Identifiers that are already Entrez IDs are used as-is without mapping.
reference	Optional reference database (a <code>SummarizedExperiment</code>). When supplied, the fraction of the mapped query present in <code>rownames(reference)</code> is also reported.
minMappedFraction	Numeric in $[0, 1]$. A warning is emitted if the fraction of query genes mapped to Entrez falls below this value. Default <code>0.5</code> .
dropLowExpressed	Logical. When the input carries an adjusted p-value column, tools such as DESeq2 set it to <code>NA</code> for genes that were filtered out before testing, typically because they are too lowly expressed. When <code>TRUE</code> (default) these genes are dropped and the number removed is reported. Has no effect when the input has no adjusted p-value column.

Value

A `data.frame` with a gene column (Entrez IDs) and a `log2FC` column, plus a `padj` column when one was available in object. Rows with a missing fold change or an unmapped identifier are dropped, and identifiers mapping to the same Entrez ID are collapsed to the one with the largest absolute fold change.

See Also[calcEnrichment](#)**Examples**

```
# A limma topTable-style data frame (gene IDs in the row names)
tt <- data.frame(
  logFC = c(2.5, -1.8, 0.3),
  adj.P.Val = c(0.001, 0.02, 0.4),
  row.names = c("100", "1000", "10000")
)
# IDs are already Entrez, so no OrgDb is needed
prepareQuery(tt)
```

regulatorGSEA

*GO enrichment of perturbMatch regulator hits by GSEA***Description**

Runs gene set enrichment analysis (**fgsea**) over Gene Ontology terms using the reference regulators ranked by their perturbMatch similarity score. This asks which biological processes are enriched among the regulators whose perturbation signatures most closely *mimic* the query (positive enrichment) or most strongly *reverse* it (negative enrichment).

Usage

```
regulatorGSEA(
  object,
  ontology = c("BP", "MF", "CC"),
  aggregate = c("extreme", "mean"),
  inferActivity = TRUE,
  orgDb = "org.Hs.eg.db",
  minSize = 15L,
  maxSize = 500L,
  ...
)
```

Arguments

object	A PerturbMatch object from calcEnrichment .
ontology	GO ontology to test: one of "BP" (default), "MF" or "CC".
aggregate	How to collapse the multiple signatures of a regulator into one ranking statistic. "extreme" (default) keeps the single largest-magnitude z-score. "mean" averages the activity z-scores.

<code>inferActivity</code>	Logical. When TRUE (default) the z-score sign is flipped for KD/KO perturbations before aggregating, so the ranking reflects inferred gene activity rather than raw transcriptional similarity.
<code>orgDb</code>	An AnnotationDbi OrgDb object, or the name of an installed OrgDb package, supplying the GO annotation and the symbol/Entrez mapping. Default "org.Hs.eg.db", matching the species of the packaged reference databases.
<code>minSize, maxSize</code>	Minimum and maximum GO gene-set sizes to test, passed to fgsea .
<code>...</code>	Further arguments passed to fgsea .

Details

A reference database usually holds several signatures per regulator gene (different cell lines, and knockdown versus overexpression). These are collapsed to a single ranking statistic per gene before scoring:

1. scores are put on a consistent inferred-activity scale by flipping the sign of the z-score for KD/KO perturbations (as in `plotSimilarity(inferActivity = TRUE)`), controlled by `inferActivity`.
2. the per-signature scores of each gene are aggregated with `aggregate` (the single "extreme" value with the largest magnitude, or the "mean" of the activity z-scores).

The resulting one-value-per-gene ranking is scored against GO gene sets, so the positive end of the ranking (mimicking regulators) and the negative end (reversing regulators) are tested together and separated by the sign of each term's normalised enrichment score (NES).

Gene sets are built from `orgDb` (using the ancestor-propagated GOALL mapping restricted to ontology), which also supplies the regulator symbol-to-Entrez mapping. Human-readable term names are added from **GO.db** when it is installed. Otherwise term falls back to the GO accession.

Value

A data.frame, one row per GO term ordered by adjusted p-value, with columns ID (GO accession), term, NES (positive = enriched among mimicking regulators, negative = among reversing regulators), pval, padj, size and leadingEdge (comma-separated regulator gene symbols).

See Also

[plotRegulatorGSEA](#), [calcEnrichment](#)

Examples

```
se <- getReferenceDatabase(demo = TRUE)
res <- calcEnrichment(
  se,
  upRegulated = rownames(se)[seq_len(5)],
  downRegulated = rownames(se)[seq(6, 10)],
  background = rownames(se),
  BPPARAM = BiocParallel::SerialParam(progressbar = FALSE)
)
# minSize is lowered because the demo database is tiny. Keep the default
```

```
# on a full reference.
go <- regulatorGSEA(res, minSize = 1)
head(go)
```

regulatorTable

Tidy table of regulator scores from a PerturbMatch object

Description

Combines rowData with the enrichment scores into a single data.frame, one row per reference signature.

Usage

```
regulatorTable(object)
```

Arguments

object A PerturbMatch object.

Details

The name deliberately avoids result/results, which would sit one letter from [results](#) in a script that uses both.

Value

A data.frame with one row per feature holding every rowData column plus a score column.

Examples

```
mat <- matrix(runif(3), nrow = 3, dimnames = list(
  c("GENE1", "GENE2", "GENE3"), "score"
))
se <- SummarizedExperiment::SummarizedExperiment(
  assays = list(enrichment = mat)
)
S4Vectors::metadata(se) <- list(
  up_regulated = "GENE1",
  down_regulated = "GENE3",
  background = c("GENE1", "GENE2", "GENE3")
)
pm <- new("PerturbMatch", se)
regulatorTable(pm)
```

`show,PerturbMatch-method`*Show a PerturbMatch object*

Description

Prints the inherited SummarizedExperiment summary followed by the gene sets and parameters recorded when the object was built.

Usage

```
## S4 method for signature 'PerturbMatch'
show(object)
```

Arguments

`object` A PerturbMatch object.

Value

Invisibly returns NULL. Called for its printing side effect.

Examples

```
pm <- methods::new(
  "PerturbMatch",
  SummarizedExperiment::SummarizedExperiment(
    assays = list(
      enrichment = matrix(0.5, dimnames = list("SIG1", "score"))
    ),
    metadata = list(
      up_regulated = c("GENE1", "GENE2"),
      down_regulated = "GENE3",
      background = c("GENE1", "GENE2", "GENE3"),
      threshold = 0.5,
      topN = 250L,
      topSig = 100L
    )
  )
)
show(pm)
```

topRegulators	<i>Top-scoring regulator signatures</i>
---------------	---

Description

Returns the n highest-scoring rows of `regulatorTable`, ranked by absolute enrichment score.

Usage

```
topRegulators(object, n = 5, ...)
```

Arguments

<code>object</code>	A <code>PerturbMatch</code> object.
<code>n</code>	Number of signatures to return, capped at <code>nrow(object)</code> . Default 5.
<code>...</code>	Additional arguments passed to methods.

Value

A data.frame with at most n rows, ordered by descending absolute score.

Examples

```
mat <- matrix(runif(5), nrow = 5, dimnames = list(
  paste0("GENE", seq_len(5)), "score"
))
se <- SummarizedExperiment::SummarizedExperiment(
  assays = list(enrichment = mat)
)
S4Vectors::metadata(se) <- list(
  up_regulated = "GENE1",
  down_regulated = "GENE5",
  background = paste0("GENE", seq_len(5))
)
pm <- new("PerturbMatch", se)
topRegulators(pm, n = 3)
```

upRegulated	<i>Up-regulated genes from a PerturbMatch object</i>
-------------	--

Description

Returns the up-regulated gene set recorded in `metadata()` when the object was built by `calcEnrichment`.

Usage

```
upRegulated(object)
```

Arguments

object A PerturbMatch object.

Value

A character vector of gene identifiers, or NULL if unset.

Examples

```
pm <- methods::new(
  "PerturbMatch",
  SummarizedExperiment::SummarizedExperiment(
    assays = list(
      enrichment = matrix(0.5, dimnames = list("SIG1", "score"))
    ),
    metadata = list(up_regulated = c("GENE1", "GENE2"),
      background = c("GENE1", "GENE2")))
  upRegulated(pm)
```

volcanoPlot

Volcano Plot of Differential Expression Results

Description

Renders a volcano plot of log₂ fold-change against $-\log_{10}(\text{FDR})$, colouring genes as up-regulated, down-regulated, or unchanged relative to supplied thresholds. The most significant genes at each tail are labelled automatically, and arbitrary genes can be highlighted via selectedPoints.

Usage

```
volcanoPlot(
  data,
  numberPoints = 5,
  selectedPoints = NULL,
  logFCThreshold = log2(1.5),
  fdrThreshold = 0.05,
  pointSize = 2,
  labSize = 4,
  truncateInfPvalue = TRUE
)
```

Arguments

data A data frame with at least three columns, in this order:

1. Gene / feature identifier (character).
2. Log₂ fold-change (numeric).
3. Adjusted p-value / FDR (numeric, in the interval [0, 1]).

	Additional columns are ignored.
numberPoints	Integer. Number of top up- and down-regulated genes to label (ranked first by p-value, then by $ \log FC $). Pass NULL to disable automatic labelling. Default: 5.
selectedPoints	Character vector of gene identifiers to label regardless of their statistics. Merged with automatic labels when numberPoints is also set. Default: NULL.
logFCThreshold	Numeric. Minimum absolute $\log_2$ fold-change required for a gene to be called regulated. Default: $\log_2(1.5)$ (≈ 0.585).
fdrThreshold	Numeric. Maximum adjusted p-value for a gene to be called regulated. Shown as a horizontal dashed line. Default: 0.05.
pointSize	Numeric. Passed to <code>geom_point</code> as size. Default: 2.
labSize	Numeric. Passed to <code>geom_text_repel</code> as size. Default: 4.
truncateInfPvalue	Logical. If TRUE, genes with $p = 0$ (an infinite $-\log_{10}$ FDR) are placed at the largest finite $-\log_{10}$ FDR, forming a flat row along the top of the plot. If FALSE, they keep their infinite value and are dropped by the plotting layer. Default: TRUE.

Details

Infinite $-\log_{10}$ p-values (arising from $p = 0$) are, by default, truncated to the largest finite value to keep the plot axis bounded. See `truncateInfPvalue` to switch this off.

Value

A `ggplot` object. Colour scale:

firebrick3 Up-regulated ($\log FC \geq \log FC_{\text{Threshold}}$ and $\text{padj} \leq \text{fdrThreshold}$).

dodgerblue3 Down-regulated ($\log FC \leq -\log FC_{\text{Threshold}}$ and $\text{padj} \leq \text{fdrThreshold}$).

black Unchanged (all other genes).

Examples

```
set.seed(42)
n <- 500
df <- data.frame(
  gene = paste0("Gene", seq_len(n)),
  logFC = rnorm(n, sd = 2),
  padj = rbeta(n, 0.5, 5)
)
volcanoPlot(df)

# Highlight specific genes
volcanoPlot(df, selectedPoints = c("Gene1", "Gene42"))

# Stricter thresholds, no automatic labels
volcanoPlot(df, numberPoints = NULL,
  logFCThreshold = log2(2), fdrThreshold = 0.01)
```

Index

background, [2](#), [9](#)
background, PerturbMatch-method
 (background), [2](#)
BiocParallelParam, [4](#)

calcEnrichment, [3](#), [8](#), [9](#), [13](#), [15](#), [16](#)
calcEnrichment, SummarizedExperiment-method
 (calcEnrichment), [3](#)

downRegulated, [5](#), [9](#)
downRegulated, PerturbMatch-method
 (downRegulated), [5](#)

enrichment, [6](#)
enrichment, PerturbMatch-method
 (enrichment), [6](#)

fgsea, [16](#)

geom_point, [12](#), [21](#)
geom_text_repel, [12](#), [21](#)
getReferenceDatabase, [7](#)
ggplot, [21](#)

mapIds, [14](#)

PerturbMatch, [4](#), [8](#), [15](#)
PerturbMatch-class, [9](#)
plotRegulatorGSEA, [9](#), [16](#)
plotSimilarity, [11](#)
plotSimilarity, PerturbMatch-method
 (plotSimilarity), [11](#)
plotVolcano, [12](#)
prepareQuery, [13](#)

regulatorGSEA, [9](#), [10](#), [15](#)
regulatorTable, [9](#), [17](#), [19](#)
regulatorTable, PerturbMatch-method
 (regulatorTable), [17](#)
results, [17](#)

show, PerturbMatch-method, [18](#)

SummarizedExperiment, [9](#)

theme_cowplot, [10](#)
topRegulators, [19](#)
topRegulators, PerturbMatch-method
 (topRegulators), [19](#)

upRegulated, [9](#), [19](#)
upRegulated, PerturbMatch-method
 (upRegulated), [19](#)

validObject, [9](#)
volcanoPlot, [12](#), [13](#), [20](#)