

Package: SwitchClass (via r-universe)

September 18, 2026

Title Label-Switch Classification for Baseline- and
Perturbation-Aligned Features

Version 0.99.1

Description Quantifies baseline-aligned and perturbation-aligned molecular features across biological perturbations using a label-switch classification framework. Provides a unified workflow for identifying changes in molecular-feature alignment in longitudinal or comparative omics datasets. Includes tools for classification, visualization, dimension reduction, and pathway enrichment.

License MIT + file LICENSE

URL <https://github.com/PYangLab/SwitchClass>

BugReports <https://github.com/PYangLab/SwitchClass/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

biocViews Software, Classification, FeatureExtraction, GeneExpression, Proteomics, Transcriptomics, SingleCell, Pathways

Depends R (>= 4.5.0)

Imports randomForest, uwot, ggplot2, ggrepel, PhosR, pheatmap, dplyr, RColorBrewer, ggpubr, parallel

Suggests BiocStyle, knitr, rmarkdown, limma, matrixStats, patchwork, clusterProfiler, glmnet, reactome.db, org.Hs.eg.db, msigdb, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs

cmake libglpk-dev make libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev zlib1g-dev

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

Date/Publication 2026-08-04 23:40:27 UTC

RemoteUrl <https://github.com/BiocStaging/SwitchClass>
RemoteRef HEAD
RemoteSha 106a48d8112ce92b1a01dad1dc2a9b9069d01b08

Contents

assign_quadrants	2
CRC_subset	3
enrich_reactome_quadrants	3
label_switch_classify	4
plot_delta_vs_fc	5
plot_feature_boxplots	6
plot_heatmap_quadrants	8
plot_umap	9
run_umap_samples	10
umap_triptych_by_delta	11
Index	13

assign_quadrants	<i>Assign SwitchClass quadrants</i>
------------------	-------------------------------------

Description

Q1: delta > T & FC > 0 Q2: delta < -T & FC > 0 Q3: delta < -T & FC < 0 Q4: delta > T & FC < 0
Q0: otherwise

Usage

assign_quadrants(delta, fc, delta_thresh = 0.15)

Arguments

- delta named numeric vector (features).
- fc named numeric vector (same feature names as delta).
- delta_thresh numeric threshold for |delta|.

Value

factor vector of quadrant labels with names = features.

Examples

```
delta <- c(a = 0.2, b = -0.2, c = -0.2, d = 0.2, e = 0)
fc <- c(a = 1, b = 1, c = -1, d = -1, e = 1)
assign_quadrants(delta, fc, delta_thresh = 0.15)
```

CRC_subset	<i>CRC subset data</i>
------------	------------------------

Description

A small example dataset from the colorectal cancer (CRC) plasma proteome study. It contains a pre-filtered expression matrix and sample group labels (HC = healthy control, Pre = pre-treatment, Post = post-treatment).

Usage

`data(CRC_subset)`

Format

A list with two elements:

- expr** A numeric matrix of features (rows) × samples (columns).
- group** A character vector of sample groups, length = ncol(expr).

Details

The data are a subset of the discovery cohort used in the SwitchClass analysis. Useful for testing or demonstrating functions such as `label_switch_classify()`.

Source

Internal preprocessed subset from [Li et al. 2024, Nat Commun.](#)

Examples

```
data(CRC_subset)
dim(CRC_subset$expr)
table(CRC_subset$group)
```

enrich_reactome_quadrants	<i>Reactome ORA per quadrant</i>
---------------------------	----------------------------------

Description

- Wraps a convenience pipeline:
- builds Reactome TERM2GENE via msigdb
 - runs pathwayOverrepresent on each quadrant (Q1..Q4)
 - returns top-k barplots (-log10 p) and tidy tables

Usage

```
enrich_reactome_quadrants(
  df,
  universe,
  species = "Homo sapiens",
  top_terms = 5,
  palette = c(Q1 = "#33DBB2", Q2 = "#DFB647", Q3 = "#2ECEF8", Q4 = "#76CF49")
)
```

Arguments

<code>df</code>	data.frame with columns feature and quadrant (Q1..Q4). If quadrant is absent, columns delta and fc are used to assign it by sign.
<code>universe</code>	character vector of background gene symbols (e.g., rownames(discovery.cohort))
<code>species</code>	msigdb species (default "Homo sapiens")
<code>top_terms</code>	how many top terms per quadrant to plot (default 5)
<code>palette</code>	named colors for bars of Q1..Q4

Value

list with:

results named list of data.frames p1..p4 (or NULL if no hits)

barplot_df tidy data.frame used for plotting

plot ggpubr patch of 4 panels

label_switch_classify *Label-switch classification (with optional multicore)*

Description

Trains two random-forest classifiers under baseline-aligned and perturbation-aligned label schemes, then returns the directional importance score $\delta = I_{baseline} - I_{perturbation}$ using MeanDecreaseGini importances.

Usage

```
label_switch_classify(
  X,
  y_reverse,
  y_persist,
  ntimes = 25,
  seed = 1,
  ntree = 1000,
  mtry = NULL,
  parallel = FALSE,
  ncores = parallel::detectCores() - 1L
)
```

Arguments

<code>X</code>	numeric matrix with features in rows and samples in columns.
<code>y_reverse</code>	character/factor labels (length = ncol(X)) for the baseline-aligned label scheme. The argument name is retained for compatibility with earlier package versions.
<code>y_persist</code>	character/factor labels (length = ncol(X)) for the perturbation-aligned label scheme. The argument name is retained for compatibility with earlier package versions.
<code>ntimes</code>	integer; number of RF repeats to average (default 25).
<code>seed</code>	integer; base seed (default 1).
<code>ntree</code>	integer; trees per forest (default 1000).
<code>mtry</code>	integer or NULL; variables tried at each split (default NULL).
<code>parallel</code>	logical; use multicore (default FALSE).
<code>ncores</code>	integer; cores if parallel=TRUE (default detectCores()-1).

Value

list with named numeric vectors: `delta`, `importance_baseline`, `importance_perturbation`, plus compatibility aliases `I_rev` and `I_per`.

Examples

```
X <- matrix(seq_len(60) / 10, nrow = 10)
rownames(X) <- paste0("feature", seq_len(nrow(X)))
y_baseline <- rep(c("c1", "c2", "c1"), each = 2)
y_perturbation <- rep(c("c1", "c2", "c2"), each = 2)
res <- label_switch_classify(
  X, y_baseline, y_perturbation, ntimes = 1, ntree = 5
)
head(res$delta)
```

plot_delta_vs_fc

Scatter plot of delta vs fold-change with quadrant assignment

Description

Builds a tidy data.frame with quadrants and returns the ggplot scatter.

Usage

```
plot_delta_vs_fc(
  delta,
  fc,
  delta_thresh = 0.15,
  label_top = 5,
  label_size = 3,
  palette = c(Q1 = "#2E8B57", Q2 = "#C0392B", Q3 = "#C0392B", Q4 = "#2E8B57", Q0 =
```

```

    "grey70"),
  point_size = 2.2,
  alpha = 0.85,
  xlab = expression(delta),
  ylab = expression(log[2] * " FC"),
  title = "feature importance vs log2 fold-change"
)

```

Arguments

<code>delta</code>	named numeric vector of directional importance scores (features).
<code>fc</code>	named numeric vector of log2 fold-change (same names as <code>delta</code>).
<code>delta_thresh</code>	numeric; absolute delta threshold to define Q1-Q4 (default 0.15).
<code>label_top</code>	integer; number of labels per quadrant (by absolute delta). Set 0 to skip.
<code>label_size</code>	numeric; text size for labels.
<code>palette</code>	named colors for quadrants. Names should include Q1..Q4 and optional Q0.
<code>point_size</code>	numeric; scatter point size.
<code>alpha</code>	numeric from 0 to 1; point transparency.
<code>xlab, ylab</code>	axis labels.
<code>title</code>	plot title.

Value

list with elements: `data` (data.frame) and `plot` (ggplot object).

Examples

```

delta <- c(a = 0.2, b = -0.2, c = -0.2, d = 0.2, e = 0)
fc <- c(a = 1, b = 1, c = -1, d = -1, e = 1)
out <- plot_delta_vs_fc(delta, fc, label_top = 0)
head(out$data)
out$plot

```

`plot_feature_boxplots` *Boxplots for selected features across groups*

Description

Boxplots for selected features across groups

Usage

```
plot_feature_boxplots(
  X,
  group,
  features,
  feature_order = NULL,
  palette = NULL,
  ncol = 4,
  add_jitter = TRUE,
  jitter_width = 0.15,
  notch = FALSE,
  free_y = TRUE,
  point_size = 1.6,
  box_linewidth = 0.4,
  title = NULL,
  ylab = "log2 expression",
  xlab = NULL,
  save_path = NULL,
  width = 7,
  height = 4.5
)
```

Arguments

X	numeric matrix with features in rows and samples in columns.
group	factor/character of length ncol(X) giving sample groups.
features	character vector of feature (gene/protein) names to plot.
feature_order	optional character vector to set facet order (defaults to features order).
palette	optional named vector of group colors; if NULL, generated automatically.
ncol	integer; facets per row (default 4).
add_jitter	logical; add jittered points (default TRUE).
jitter_width	numeric; jitter width (default 0.15).
notch	logical; use notched boxplots (default FALSE).
free_y	logical; facet with free y scales (default TRUE).
point_size	numeric; jitter point size (default 1.6).
box_linewidth	numeric; box outline width (default 0.4).
title, ylab, xlab	plot labels.
save_path	optional file path to save (pdf/png) using ggplot2::ggsave.
width, height	numeric dimensions (inches) if saving.

Value

ggplot object.

Examples

```
X <- matrix(seq_len(40) / 10, nrow = 5)
rownames(X) <- paste0("feature", seq_len(nrow(X)))
group <- rep(c("A", "B"), each = 4)
plot_feature_boxplots(X, group, features = rownames(X)[1:2], ncol = 2)
```

```
plot_heatmap_quadrants
```

Plot heatmap for quadrant-classified features

Description

Generates a standardized heatmap of feature expression grouped by quadrant (Q1-Q4) and sample group (e.g., HC / Pre / Post). Quadrants are optionally filtered by delta cutoff thresholds.

Usage

```
plot_heatmap_quadrants(
  X,
  group,
  df_quad,
  cutoff = 0.15,
  palette_group = NULL,
  palette_quad = NULL,
  scale = TRUE,
  cap = 2,
  show_rownames = TRUE,
  show_colnames = TRUE,
  cluster_rows = FALSE,
  cluster_cols = FALSE
)
```

Arguments

X	numeric matrix with features in rows and samples in columns.
group	factor/character of length = ncol(X); sample group labels.
df_quad	data.frame with columns: feature, quadrant, and delta.
cutoff	numeric; delta threshold for selecting strong quadrant features (default 0.15). Quadrants Q1 and Q4 use $\text{delta} > \text{cutoff}$; Q2 and Q3 use $\text{delta} < -\text{cutoff}$.
palette_group	optional named color vector for sample groups.
palette_quad	optional named color vector for quadrants.
scale	logical; z-score features (default TRUE).
cap	numeric; cap scaled values at $\pm \text{cap}$ (default 2).
show_rownames, show_colnames	logical; show labels (default TRUE).
cluster_rows, cluster_cols	logical; cluster rows/columns (default FALSE).

Value

Invisibly returns the pheatmap object.

Examples

```
plot_heatmap_quadrants(
  X = matrix(
    seq_len(24), nrow = 4,
    dimnames = list(paste0("feature", 1:4), paste0("sample", 1:6))
  ),
  group = rep(c("A", "B"), each = 3),
  df_quad = data.frame(
    feature = paste0("feature", 1:4),
    quadrant = c("Q1", "Q2", "Q3", "Q4"),
    delta = c(0.2, -0.2, -0.2, 0.2)
  ),
  show_rownames = FALSE,
  show_colnames = FALSE
)
```

plot_umap

Quick ggplot for UMAP embedding

Description

Quick ggplot for UMAP embedding

Usage

```
plot_umap(
  emb,
  groups,
  title = "",
  point_size = 3,
  shape = 21,
  color = "black",
  stroke = 1.1,
  alpha = 0.9,
  palette = NULL
)
```

Arguments

emb	matrix from <code>run_umap_samples()</code> (samples x 2).
groups	vector of sample groups (length = nrow(emb)).
title	character; plot title.
point_size	numeric; point size (default 3).

shape integer; point shape (default 21 = filled circle).
color, stroke, alpha aesthetics for outlines and transparency.
palette optional named vector of fill colors; if NULL, generated automatically.

Value

ggplot object.

Examples

```
emb <- matrix(c(0, 0, 1, 1, 0, 1), ncol = 2, byrow = TRUE)
rownames(emb) <- paste0("sample", seq_len(nrow(emb)))
plot_umap(emb, groups = c("A", "A", "B"))
```

run_umap_samples	<i>UMAP on samples using selected features</i>
------------------	--

Description

UMAP on samples using selected features

Usage

```
run_umap_samples(  
  X,  
  feats,  
  n_neighbors = 15,  
  min_dist = 0.1,  
  metric = "euclidean",  
  seed = 1  
)
```

Arguments

X numeric matrix with features in rows and samples in columns.
feats character; feature names to use (non-matching are ignored with a message).
n_neighbors integer; UMAP neighbor size (default 15).
min_dist numeric; UMAP min_dist (default 0.1).
metric character; distance metric (default "euclidean").
seed integer; RNG seed for reproducibility (default 1).

Value

matrix with samples in rows and two UMAP columns, with rownames = sample IDs.

Examples

```
X <- matrix(seq_len(80) / 10, nrow = 10)
rownames(X) <- paste0("feature", seq_len(nrow(X)))
colnames(X) <- paste0("sample", seq_len(ncol(X)))
emb <- run_umap_samples(X, rownames(X)[1:4], n_neighbors = 3)
head(emb)
```

umap_triptych_by_delta

UMAP triptych using delta-ranked features

Description

Generates UMAPs for (1) baseline-aligned (+delta), (2) perturbation-aligned (-delta), and (3) all features. Returns the three embeddings and a combined plot.

Usage

```
umap_triptych_by_delta(
  X,
  delta,
  groups,
  top_n = 100,
  n_neighbors = 15,
  min_dist = 0.1,
  metric = "euclidean",
  seed = 1,
  palette = NULL
)
```

Arguments

X	numeric matrix with features in rows and samples in columns.
delta	named numeric vector of delta scores (names = feature IDs).
groups	factor or character of sample groups (length = ncol(X)).
top_n	integer; number of features to use per side (default 100).
n_neighbors, min_dist, metric, seed	passed to run_umap_samples() .
palette	optional named color vector for groups; if NULL, generated automatically.

Value

list with emb_baseline, emb_perturbation, emb_all, and plot.

Examples

```
X <- matrix(seq_len(100) / 10, nrow = 10)
rownames(X) <- paste0("feature", seq_len(nrow(X)))
colnames(X) <- paste0("sample", seq_len(ncol(X)))
delta <- stats::setNames(seq(-1, 1, length.out = nrow(X)), rownames(X))
groups <- rep(c("A", "B"), each = 5)
triptych <- umap_triptych_by_delta(
  X, delta, groups, top_n = 2, n_neighbors = 3
)
names(triptych)
```

Index

* **datasets**

CRC_subset, [3](#)

assign_quadrants, [2](#)

CRC_subset, [3](#)

enrich_reactome_quadrants, [3](#)

label_switch_classify, [4](#)

plot_delta_vs_fc, [5](#)

plot_feature_boxplots, [6](#)

plot_heatmap_quadrants, [8](#)

plot_umap, [9](#)

run_umap_samples, [10](#)

run_umap_samples(), [9](#), [11](#)

umap_triptych_by_delta, [11](#)