

Package ‘Kuresi’

December 18, 2025

Type Package

Title Competitive fitness of Cancer clones

Description Kuresi is an R package for analyzing competitive fitness between cancer clones in single cell and spatial transcriptomics data. It provides tools to compute and visualize competition scores based on differential gene expression patterns.

Author Patrick C.N. Martin <patrick.martin@cshs.org>

Version 0.0.1

Maintainer Patrick C.N. Martin <patrick.martin@cshs.org>

License GPL-3 + file LICENSE

Collate competition_score.r utils.r viz_utils.r sanity_checks.r
ratio_scores.r data.r degs.r

Depends R (>= 4.0.0)

Imports grDevices, stats, utils, ggplot2, patchwork, ggpubr, rjson,
Matrix, hdf5r, Seurat, RColorBrewer, dplyr, ggnewscale, pwr,
tibble, tidyr

Suggests knitr, RUnit, rmarkdown, testthat (>= 3.0.0)

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 7.3.3

NeedsCompilation no

Contents

bin_score	2
cancer_maker_list	3
check_gene_set	3
check_grouping	4
collapse_score	4
compute_competition_outcomes	5
compute_elo	5
compute_ratio_bygroup	6
compute_ratio_score	7
compute_score	8

coordinates	8
counts	9
create_alpha	9
create_palette	10
effect_size	10
function_call	11
generate_palette	11
get_outcomes	12
k_ttest	12
k_wilcox	13
outcomes_as_cost	13
outcomes_as_elo	14
outcomes_as_score_matrix	14
score_plot	15
validate_input	15
view_elo	16
win_lose_genes	16
Index	17

bin_score	<i>Bin score values into discrete categories</i>
-----------	--

Description

Converts continuous score values into discrete bins and ranks them.

Usage

bin_score(score, bins)

Arguments

- score Data frame with a "score" column containing numeric values.
- bins Integer number of bins to create.

Value

Data frame with binned and ranked score values.

cancer_maker_list	Create Cancer marker list
-------------------	---------------------------

Description

Returns predefined lists of cancer marker genes for specific cancer types.

Usage

```
cancer_maker_list(type)
```

Arguments

type	Character string specifying cancer type: "pancreas", "breast", or "ovary".
------	--

Value

Character vector of cancer marker gene names.

check_gene_set	Check and filter gene set against count matrix
----------------	--

Description

Validates that genes in the gene set are present in the count matrix and returns filtered gene list.

Usage

```
check_gene_set(counts, genes)
```

Arguments

counts	Matrix or sparse matrix with genes as row names.
genes	Character vector of gene names to check.

Value

Character vector of genes present in counts matrix, or NULL if input is NULL.

check_grouping	<i>Check and format grouping structure</i>
----------------	--

Description

Validates and formats group assignments into a standardized data frame format.

Usage

```
check_grouping(group_id, group_name)
```

Arguments

group_id	Data frame, matrix, or named vector containing group assignments.
group_name	Character string specifying column name if group_id is a data frame.

Value

Data frame with columns "cell_id" and "group_id".

collapse_score	<i>Collapse counts matrix to vector</i>
----------------	---

Description

Collapses a counts matrix to a single vector using mean or sum aggregation.

Usage

```
collapse_score(counts, method)
```

Arguments

counts	Matrix or numeric vector of counts.
method	Character string containing "mean" or "sum" to specify aggregation method.

Value

Numeric vector of aggregated values.

`compute_competition_outcomes`*Compute competition outcomes between groups*

Description

Computes differential expression outcomes between groups using statistical tests to determine competitive fitness between cancer clones.

Usage

```
compute_competition_outcomes(  
  counts,  
  groups,  
  gene_set1,  
  group_name = NULL,  
  gene_set2 = NULL,  
  method = "wilcox",  
  log_fc = 0,  
  pval = 0.05  
)
```

Arguments

<code>counts</code>	A matrix or sparse matrix containing gene expression counts.
<code>groups</code>	A data frame, matrix, or named vector containing group assignments for cells.
<code>gene_set1</code>	A character vector of gene names representing the first gene set.
<code>group_name</code>	Character string specifying the column name in <code>groups</code> if <code>groups</code> is a data frame.
<code>gene_set2</code>	Optional character vector of gene names representing the second gene set.
<code>method</code>	Character string specifying the statistical test method (default: "wilcox").
<code>log_fc</code>	Numeric threshold for log fold change (default: 0).
<code>pval</code>	Numeric threshold for adjusted p-value (default: 0.05).

Value

A nested list containing differential expression outcomes between all group pairs.

`compute_elo`*Compute ELO rating updates*

Description

Updates ELO ratings based on match outcomes using the standard ELO algorithm.

Usage

```
compute_elo(local_seed_rating, local_query_rating, outcome_seed, k = 32)
```

Arguments

local_seed_rating	Numeric current ELO rating of seed group.
local_query_rating	Numeric current ELO rating of query group.
outcome_seed	Numeric outcome score for seed group (0, 0.5, or 1).
k	Numeric K-factor for rating updates (default: 32).

Details

Updates are made in-place using parent frame assignment.

compute_ratio_bygroup *Compute ratio scores by group*

Description

Computes ratio scores for each group in a grouping structure, calculating competitive fitness metrics between two gene sets.

Usage

```
compute_ratio_bygroup(
  counts,
  groups,
  group_name = NULL,
  genes_1 = NULL,
  genes_2 = NULL,
  method = c("mean_ratio", "mean_sub", "sum_ratio", "sum_sub"),
  weights_1 = NULL,
  weights_2 = NULL,
  scale = FALSE,
  center = FALSE,
  add_name = NULL,
  verbose = TRUE
)
```

Arguments

counts	A matrix or sparse matrix containing gene expression counts.
groups	A data frame, matrix, or named vector containing group assignments.
group_name	Character string specifying the column name in groups if groups is a data frame.
genes_1	Character vector of gene names for the first gene set.
genes_2	Character vector of gene names for the second gene set (optional).
method	Character vector specifying the scoring method: "mean_ratio", "mean_sub", "sum_ratio", or "sum_sub".
weights_1	Named numeric vector of weights for genes_1 (optional).
weights_2	Named numeric vector of weights for genes_2 (optional).

scale	Logical whether to scale counts (default: FALSE).
center	Logical whether to center scores (default: FALSE).
add_name	Character string to add to output column name (optional).
verbose	Logical whether to print progress messages (default: TRUE).

Value

A data frame with group assignments and computed scores.

compute_ratio_score	<i>Compute ratio score between two gene sets</i>
---------------------	--

Description

Computes competitive fitness scores based on the ratio or difference between two gene sets for individual cells or groups of cells.

Usage

```
compute_ratio_score(
  counts,
  cells = NULL,
  genes_1 = NULL,
  genes_2 = NULL,
  method = c("mean_ratio", "mean_sub", "sum_ratio", "sum_sub"),
  weights_1 = NULL,
  weights_2 = NULL,
  scale = FALSE,
  center = FALSE,
  collapse = FALSE,
  verbose = TRUE
)
```

Arguments

counts	A matrix or sparse matrix containing gene expression counts.
cells	Character vector of cell IDs to include (optional, uses all cells if NULL).
genes_1	Character vector of gene names for the first gene set.
genes_2	Character vector of gene names for the second gene set (optional, uses genes_1 if NULL).
method	Character vector specifying the scoring method: "mean_ratio", "mean_sub", "sum_ratio", or "sum_sub".
weights_1	Named numeric vector of weights for genes_1 (optional).
weights_2	Named numeric vector of weights for genes_2 (optional).
scale	Logical whether to scale counts before computing scores (default: FALSE).
center	Logical whether to center scores (default: FALSE).
collapse	Logical whether to return a single aggregated value instead of per-cell scores (default: FALSE).
verbose	Logical whether to print progress messages (default: TRUE).

Value

Numeric vector of scores (one per cell) or a single numeric value if collapse is TRUE.

compute_score	<i>Compute score from two count matrices</i>
---------------	--

Description

Computes ratio or subtraction scores between two count matrices.

Usage

```
compute_score(counts_1, counts_2, method)
```

Arguments

counts_1	Matrix or vector representing counts for first gene set.
counts_2	Matrix or vector representing counts for second gene set.
method	Character string specifying "ratio" or "sub" (subtraction) method.

Value

Numeric vector of computed scores.

coordinates	<i>Spatial coordinates</i>
-------------	----------------------------

Description

Spatial coordinates

Usage

```
data(Kuresi)
```

Format

coordinate data frame

coordinates spatial coordinates of a 2 cell circle synthetic regime produced oneiric

Source

produced by <https://github.com/WonLab-CS/oneiric>

counts	<i>Count matrix</i>
--------	---------------------

Description

Count matrix

Usage

```
data(Kuresi)
```

Format

sparse count matrix

counts gene counts of a 2 cell circle synthetic regime produced oneiric

Source

produced by <https://github.com/WonLab-CS/oneiric>

create_alpha	<i>Create alpha values for territory highlighting</i>
--------------	---

Description

Creates transparency (alpha) values for territories, with optional highlighting of specific territories.

Usage

```
create_alpha(territories, highlight, alpha)
```

Arguments

territories	Data frame with a "trial" column containing territory assignments.
highlight	Numeric or character vector describing which territories should be highlighted.
alpha	Numeric transparency factor (default alpha for non-highlighted territories is $\alpha * 0.25$).

Details

If highlight is NULL, will return the same alpha values for all territories.

Value

Numeric vector of alpha values matching territories.

create_palette	Create color palette for scores
----------------	---------------------------------

Description

Generates a color palette based on the number of unique score levels. Uses a predefined color scheme with colors randomly assigned to each territory. Note that as the territory plot returns a ggplot object, you can easily override the color scheme.

Usage

```
create_palette(score)
```

Arguments

score	Data frame with a "score" column (should be a factor).
-------	--

Value

Character vector of color values.

effect_size	Calculate effect size from p-value
-------------	------------------------------------

Description

Estimates effect size (Cohen's h) from p-value and sample sizes using power analysis methods.

Usage

```
effect_size(pval, s_counts, q_counts)
```

Arguments

pval	Numeric p-value.
s_counts	Integer number of samples in seed/reference group.
q_counts	Integer number of samples in query group.

Value

Numeric effect size value or NA if pval is NA.

function_call	<i>Generate function based on score type</i>
---------------	--

Description

Returns an appropriate aggregation function (mean, sum, or Matrix functions) based on score type and context parameters.

Usage

```
function_call(score_type, single = TRUE, by_territory = TRUE)
```

Arguments

score_type	Character string containing "mean" or "sum" to specify function type.
single	Logical whether working with single values (default: TRUE).
by_territory	Logical whether aggregating by territory (default: TRUE).

Value

A function (mean, sum, Matrix::colMeans, or Matrix::colSums).

generate_palette	<i>Create color palette to visualize seed and query scores</i>
------------------	--

Description

Generates color palettes for visualizing seed and query groups, assigning colors based on the union of groups present in both seed and query.

Usage

```
generate_palette(palette, seed, query)
```

Arguments

palette	Character vector of base colors to use for palette generation.
seed	Character or numeric vector of seed group identifiers.
query	Character or numeric vector of query group identifiers.

Value

A list with two character vectors: "seed" and "query" containing color assignments.

get_outcomes	<i>Get outcomes from differential expression results</i>
--------------	--

Description

Converts differential expression results into outcome scores based on fold change and p-value thresholds.

Usage

```
get_outcomes(deg, gene_set1, gene_set2, fold_change = 0.1, pval = 0.05)
```

Arguments

deg	Data frame containing differential expression results.
gene_set1	Character vector of gene names in the first set.
gene_set2	Character vector of gene names in the second set.
fold_change	Numeric threshold for fold change (default: 0.1).
pval	Numeric threshold for adjusted p-value (default: 0.05).

Value

Numeric vector of outcome scores (0, 0.5, or 1).

k_ttest	<i>Perform t-test for differential expression</i>
---------	---

Description

Computes differential expression between two count matrices using Student's t-tests for each gene.

Usage

```
k_ttest(s_counts, q_counts, genes)
```

Arguments

s_counts	Matrix or sparse matrix of counts for seed/reference group.
q_counts	Matrix or sparse matrix of counts for query group.
genes	Character vector of gene names (optional, uses rownames if not provided).

Value

Data frame with columns: genes, p_value, p_value_adj, fold_change, effect_size.

k_wilcox	<i>Perform Wilcoxon rank-sum test for differential expression</i>
----------	---

Description

Computes differential expression between two count matrices using Wilcoxon rank-sum tests for each gene.

Usage

```
k_wilcox(s_counts, q_counts, genes)
```

Arguments

s_counts	Matrix or sparse matrix of counts for seed/reference group.
q_counts	Matrix or sparse matrix of counts for query group.
genes	Character vector of gene names (optional, uses rownames if not provided).

Value

Data frame with columns: genes, p_value, p_value_adj, fold_change, effect_size.

outcomes_as_cost	<i>Convert outcomes to cost matrix</i>
------------------	--

Description

Converts competition outcomes into a cost matrix where values represent competition cost (inverse of score distance from 0.5).

Usage

```
outcomes_as_cost(scores)
```

Arguments

scores	A nested list containing competition outcomes from compute_competition_outcomes.
--------	--

Value

A matrix with cost values between all group pairs.

outcomes_as_elo	<i>Convert outcomes to ELO ratings</i>
-----------------	--

Description

Converts competition outcomes into ELO ratings through simulated tournaments, providing a ranking system for competitive fitness.

Usage

```
outcomes_as_elo(scores, initial_elo = 1000, k = 32, n_tournaments = 100)
```

Arguments

scores	A nested list containing competition outcomes from <code>compute_competition_outcomes</code> .
initial_elo	Numeric initial ELO rating for all groups (default: 1000).
k	Numeric K-factor for ELO updates (default: 32).
n_tournaments	Integer number of tournament rounds to simulate (default: 100).

Value

A list containing "elo_seed" and "elo_query" numeric vectors with ELO ratings.

outcomes_as_score_matrix	<i>Convert outcomes to score matrix</i>
--------------------------	---

Description

Converts nested list of competition outcomes into a matrix where each entry represents the average score between groups.

Usage

```
outcomes_as_score_matrix(scores)
```

Arguments

scores	A nested list containing competition outcomes from <code>compute_competition_outcomes</code> .
--------	--

Value

A matrix with average scores between all group pairs.

score_plot	<i>Plot competition scores spatially</i>
------------	--

Description

Creates a ggplot visualization of competition scores with optional spatial image background and score binning.

Usage

```
score_plot(score, cex = 10, cex_pt = 1, alpha = 0.65, bins = NULL, img = NULL)
```

Arguments

score	Data frame with columns for x, y coordinates and score values.
cex	Numeric text size multiplier (default: 10).
cex_pt	Numeric point size (default: 1).
alpha	Numeric transparency value between 0 and 1 (default: 0.65).
bins	Integer number of bins for discrete coloring, or NULL for continuous (default: NULL).
img	Optional imager object for background image.

Value

A ggplot object.

validate_input	<i>Validate input parameters for ratio score computation</i>
----------------	--

Description

Validates and processes input parameters for compute_ratio_score function, checking counts matrix, cells, genes, weights, and method specifications.

Usage

```
validate_input(counts, cells, genes_1, genes_2, method, weights_1, weights_2)
```

Arguments

counts	Matrix, sparse matrix, or data frame containing gene expression counts.
cells	Character vector of cell IDs (optional).
genes_1	Character vector of gene names for first gene set.
genes_2	Character vector of gene names for second gene set (optional).
method	Character vector specifying scoring method.
weights_1	Named numeric vector of weights for genes_1 (optional).
weights_2	Named numeric vector of weights for genes_2 (optional).

Details

Modifies variables in the parent frame using `assign()`.

Value

Returns 0 on success, stops with error if validation fails.

<code>view_elo</code>	<i>Visualize ELO ratings</i>
-----------------------	------------------------------

Description

Creates a bar plot showing ELO ratings for seed and query groups.

Usage

```
view_elo(elo_scores)
```

Arguments

`elo_scores` List containing "elo_seed" and "elo_query" numeric vectors.

Value

A ggplot object showing ELO ratings as bar plots.

<code>win_lose_genes</code>	<i>Create Win/Lose gene list</i>
-----------------------------	----------------------------------

Description

Returns predefined lists of "win" and "lose" genes associated with competitive fitness in cancer cells.

Usage

```
win_lose_genes()
```

Value

A list containing two character vectors: "win" (pro-competitive genes) and "lose" (anti-competitive genes).

Index

- * **datasets**
 - coordinates, [8](#)
 - counts, [9](#)
- bin_score, [2](#)
- cancer_maker_list, [3](#)
- check_gene_set, [3](#)
- check_grouping, [4](#)
- collapse_score, [4](#)
- compute_competition_outcomes, [5](#)
- compute_elo, [5](#)
- compute_ratio_bygroup, [6](#)
- compute_ratio_score, [7](#)
- compute_score, [8](#)
- coordinates, [8](#)
- counts, [9](#)
- create_alpha, [9](#)
- create_palette, [10](#)
- effect_size, [10](#)
- function_call, [11](#)
- generate_palette, [11](#)
- get_outcomes, [12](#)
- k_ttest, [12](#)
- k_wilcox, [13](#)
- outcomes_as_cost, [13](#)
- outcomes_as_elo, [14](#)
- outcomes_as_score_matrix, [14](#)
- score_plot, [15](#)
- validate_input, [15](#)
- view_elo, [16](#)
- win_lose_genes, [16](#)