Package: cascade (via r-universe)

May 31, 2026

Type Package

Title Comprehensive Analysis Suite for Cell-type specificity and Accessibility-Driven QTL Effects

Version 1.0.0

Description Provides a hierarchical framework for systematically characterizing cell type specificity and regulatory mechanisms of molecular QTLs (molQTLs) from multiome analyses. Gene- and peak-level analyses classify eGenes and caPeaks by their cell type specificity patterns, with power-aware assessment using local false sign rates (LFSR) from multivariate adaptive shrinkage (mash) to identify likely shared but underpowered features. Variant-level analysis classifies fine-mapped variants by regulatory mechanism, with 25 mutually exclusive QTL patterns collapsing into eight mechanism categories that include three chromatin-to-expression cascade tiers.

License MIT + file LICENSE

URL https://github.com/mkanai/cascade

BugReports https://github.com/mkanai/cascade/issues

OS_type unix

SystemRequirements C++11

Encoding UTF-8

Depends R (>= 3.5.0)

Imports cli, data.table (>= 1.12.0), digest, GenomicRanges, IRanges, jsonlite, memoise, parallel, qs2, R.utils, Rcpp, S4Vectors, stats, utils

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

RoxygenNote 7.3.2

VignetteBuilder knitr

Config/testthat/edition 3

LinkingTo Rcpp

Config/pak/sysreqs make

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

Date/Publication 2026-05-31 18:27:49 UTC

RemoteUrl https://github.com/mkanai/cascade

RemoteRef HEAD

RemoteSha c12d2e469760731cc0993ef65bef24ddde3e3f4b

Contents

`add_variant_analysis` *Variant Heterogeneity Analysis*

Description

Functions for analyzing variant heterogeneity patterns using Cochran's Q test and CS cluster-based approaches Add Variant Analysis to Categorization Results

Add variant heterogeneity analysis using Cochran's Q test or CS clusters

Usage

```
add_variant_analysis(
  results,
  susie_results,
  data_type,
  lfsr_results,
  meta_data,
  variant_feature_specificity,
  cochran_q_threshold = 5e-08,
  use_cs_clusters = TRUE,
  cs_clusters = NULL,
  cs_cluster_variants = NULL
)
```

Arguments

`results`	Results from vectorized categorization
`susie_results`	SuSiE results
`data_type`	Either "gene" or "peak"
`lfsr_results`	Optional LFSR results
`meta_data`	Required. Pre-computed meta data with Cochran's Q values
`variant_feature_specificity`	Required. Variant-feature specificity mapping
`cochran_q_threshold`	Cochran's Q p-value threshold
`use_cs_clusters`	Logical. Whether to use CS cluster-based analysis (default TRUE)
`cs_clusters`	Optional. CS cluster mappings from load_cs_clusters()
`cs_cluster_variants`	Optional. CS cluster variant mappings from load_cs_cluster_variants()

Value

List with updated results and variant details

CAQTL_STATUS *caQTL Status Descriptions*

Description

caQTL Status Descriptions

Usage

```
CAQTL_STATUS
```

Format

An object of class `list` of length 3.

categorize_features *Categorize features (genes or peaks)*

Description

High-level dispatch for categorizing genes or peaks across cell types, powered by vectorized C++ kernels.

Usage

```
categorize_features(
  feature_data,
  lfsr_results,
  susie_results,
  meta_data,
  variant_feature_specificity,
  feature_type = "gene",
  hierarchy = DEFAULT_CELL_HIERARCHY,
  lfsr_sig_threshold = LFSR_SIG_THRESHOLD,
  lfsr_null_threshold = LFSR_NULL_THRESHOLD,
  cochran_q_threshold = 5e-08,
  use_cs_clusters = TRUE,
  cs_clusters = NULL,
  cs_cluster_variants = NULL
)
```

Arguments

feature_data	Feature data from load_feature_data(config, feature_type, chromosomes)
lfsr_results	Pre-computed LFSR results from load_lfsr_results()
susie_results	SuSiE results (optional)
meta_data	Required. Pre-computed meta data with Cochran's Q values
variant_feature_specificity	Required. Variant-feature specificity mapping from variant categorization
feature_type	Either "gene" or "peak"
hierarchy	A CellTypeHierarchy object; defaults to DEFAULT_CELL_HIERARCHY
lfsr_sig_threshold	Significance threshold used for both ACAT q-value significance and LFSR gray zone lower bound (by design, both use the same threshold). This is NOT acat_fdr_threshold (which is used only in variant data loading).
lfsr_null_threshold	LFSR null hypothesis threshold
cochran_q_threshold	P-value threshold for Cochran's Q heterogeneity test (default 5e-8)
use_cs_clusters	If TRUE, use credible-set cluster assignments when available; otherwise rely on Cochran's Q only
cs_clusters	Optional credible-set cluster assignments (loaded by load_cs_clusters())
cs_cluster_variants	Optional cluster-level variant details (loaded by load_cs_cluster_variants())

Value

List with two elements: categories (data frame with categorization results) and variant_details (data frame with variant details)

---

categorize_variants *Two-stage variant categorization*

Description

Main entry point for variant categorization. Runs per-cell-type QTL pattern detection (Stage 1), then aggregates across cell types with bulk ‘data.table‘ operations (Stage 2).

Usage

```
categorize_variants(
  variant_data,
  lfsr_results = NULL,
  output_dir = NULL,
  config = NULL
)
```

Arguments

variant_data	Variant data from 'load_variant_data_by_qtl_type()'.
lfsr_results	Pre-loaded LFSR results (optional).
output_dir	Optional output directory for saving intermediate results.
config	Configuration object (optional).

Value

List with 'per_celltype', 'cross_celltype', and 'variant_feature_specificity' results.

---

create_cell_hierarchy *Cell Type Hierarchy*

Description

Functions for defining and working with cell type hierarchies used in specificity categorization. Create a Cell Type Hierarchy

Defines the cell type hierarchy for specificity categorization. Each grouping level adds one specificity category. The package ships with DEFAULT_CELL_HIERARCHY for immune cells (2 grouping levels -> 6 categories), but users can define their own system for any tissue or organism.

The categorization logic uses the hierarchy as follows:

- "Cross-lineage shared": significant in 2+ top-level lineage groups
- "Likely shared but underpowered": LFSR gray zone evidence of hidden sharing
- "Lineage-specific": significant in exactly 1 lineage group
- "Subgroup-specific": significant in 1 subgroup (one category per subgroup level)
- "Single cell-type": significant in exactly 1 L1 cell type
- "No significance": nothing significant

Total categories = 4 fixed + N grouping levels (1 lineage level + subgroup levels).

Usage

```
create_cell_hierarchy(
  lineages,
  subgroups = list(),
  bulk = NULL,
  other = NULL,
  mapping_to_l1 = list(),
  column_prefix = "predicted.celltype"
)
```

Arguments

lineages	Named list with 2+ entries. Each entry is a character vector of L1 cell type names in that top-level group. Names become lineage labels.
subgroups	Ordered list of sub-grouping levels (optional). Each element is a named list of groups at that depth. Groups must be subsets of a single lineage. Ordered from broadest to narrowest. Use `attr(level, "label")` to set a custom category label for a level with multiple groups.
bulk	Character vector of bulk/mixed cell type names (e.g., "PBMC"). Features significant only in bulk are categorized as "Likely shared but underpowered". NULL if none.
other	Character vector of cell types excluded from lineage grouping (e.g., "other" for unclassified types). These types still participate in single-cell-type detection and are valid mapping targets. NULL if none.
mapping_to_l1	Named list mapping lower-level cell types to their L1 parents. Supports arbitrary QTL resolution depth (L2, L3, etc.) – all map directly to L1. Targets must be L1 lineage types or "other" types. Types not in this mapping are assumed to already be L1.
column_prefix	Prefix for cell type column names in data files. Default: "predicted.celltype". Columns will be "prefix.l1.name" for L1.

Value

A CellTypeHierarchy object (S3 class)

---

create_config *Create Configuration Object*

Description

Helper function to create a configuration object for cascade analysis

Usage

```
create_config(
  cell_types,
  chromosomes = NULL,
  file_patterns = list(),
 parameters = list(pip_threshold = 0.5, min_pip_threshold = 0.1, acat_fdr_threshold =
    0.05, lfsr_sig_threshold = LFSR_SIG_THRESHOLD, lfsr_null_threshold =
    LFSR_NULL_THRESHOLD, run_mash = FALSE, n_cores = NULL, mash_params =
    list(max_variants_per_gene = 5, alpha = 1, strong_z_threshold = 2)),
  column_mapping = list(),
  feature_type = "all"
)
```

Arguments

`cell_types`	Vector of cell type names
`chromosomes`	Vector of chromosomes to analyze
`file_patterns`	List with file pattern templates
`parameters`	Analysis parameters including: • pip_threshold: Maximum PIP threshold across cell types for additional filtering of 95 • min_pip_threshold: Minimum PIP threshold per cell type (default: 0.1) • acat_fdr_threshold: FDR threshold for ACAT significance (default: 0.05) • lfsr_sig_threshold: LFSR significance threshold (default: 0.05) • lfsr_null_threshold: LFSR null hypothesis threshold (default: 0.5) • run_mash: Whether to run mash analysis (default: FALSE) • n_cores: Number of cores for parallelization (default: NULL for auto-detect) • mash_params: Parameters for mash analysis
`column_mapping`	Column name mappings for input files
`feature_type`	Type of features to analyze ("gene", "peak", "variant", or "all")

Value

Configuration list

`DEFAULT_CELL_HIERARCHY`

Default Cell Type Hierarchy (Immune)

Description

The default hierarchy for immune cell QTL analysis, with myeloid/lymphoid lineages and T-cell subgroup. Produces 6 specificity categories.

Usage

```
DEFAULT_CELL_HIERARCHY
```

Format

An object of class `CellTypeHierarchy` of length 12.

DEFAULT_COLUMN_MAPPING *Default Column Mappings*

Description

Maps internal column names to input column names for each file type. Entries with NULL values are optional and skipped during rename. Users can override individual entries via create_config(column_mapping = list(...)).

Usage

```
DEFAULT_COLUMN_MAPPING
```

Format

An object of class `list` of length 9.

EQTL_STATUS *eQTL Status Descriptions*

Description

eQTL Status Descriptions

Usage

```
EQTL_STATUS
```

Format

An object of class `list` of length 4.

extract_cs_cluster_susie_details *Extract Per-Cell-Type SuSiE Results for CS Cluster Variants*

Description

Extracts detailed per-cell-type SuSiE results (beta, se, pip) for all variants in CS clusters using vectorized operations

Usage

```
extract_cs_cluster_susie_details(
  features,
  cs_clusters,
  cs_cluster_variants,
  susie_results,
  feature_type = "gene"
)
```

Arguments

features	Vector of feature IDs to extract
cs_clusters	Data table with CS to cluster mappings
cs_cluster_variants	Data table with cluster to variant mappings
susie_results	List of SuSiE results by cell type
feature_type	Either "gene" or "peak"

Value

Data table with per-cell-type SuSiE results for CS cluster variants

filter_l1_celltypes *Filter to only L1 cell types*

Description

Filter to only L1 cell types

Usage

```
filter_l1_celltypes(celltypes, hierarchy)
```

Arguments

celltypes	Character vector of cell type column names
hierarchy	A CellTypeHierarchy object (required)

Value

Character vector of L1-only cell types

---

get_pattern_interpretation *Get QTL Pattern Interpretation*

Description

Get QTL Pattern Interpretation

Usage

```
get_pattern_interpretation(pattern_num)
```

Arguments

pattern_num	Numeric pattern number (1-25)

Value

Character string with the pattern interpretation

---

is_l2_celltype *Check if a cell type is a mapped (lower-level) type*

Description

A cell type is considered "L2" (or lower) if it appears as a key in the hierarchy's mapping_to_l1, meaning it maps to an L1 parent.

Usage

```
is_l2_celltype(celltype, hierarchy)
```

Arguments

celltype	Character scalar or vector of cell type column names
hierarchy	A CellTypeHierarchy object (required)

Value

Logical vector

LFSR_NULL_THRESHOLD *LFSR Null Hypothesis Threshold*

Description

LFSR Null Hypothesis Threshold

Usage

```
LFSR_NULL_THRESHOLD
```

Format

An object of class `numeric` of length 1.

LFSR_SIG_THRESHOLD *LFSR Significance Threshold*

Description

LFSR Significance Threshold

Usage

```
LFSR_SIG_THRESHOLD
```

Format

An object of class `numeric` of length 1.

load_cs_cluster_variants *Load CS cluster variant data*

Description

Loads the file mapping clusters to their constituent variants

Usage

```
load_cs_cluster_variants(
  cs_cluster_variant_file,
  cache = NULL,
  column_mapping = NULL
)
```

Arguments

cs_cluster_variant_file	Path to the CS cluster variant file
cache	Cache object for memoization (optional)
column_mapping	Named list mapping internal names to input column names. If NULL, uses DEFAULT_COLUMN_MAPPING$cs_cluster_variants.

Value

Data table with cluster to variant mappings

---

load_cs_clusters *Load CS cluster mapping data*

Description

Loads the CS cluster file that maps credible sets to clusters across cell types

Usage

```
load_cs_clusters(cs_cluster_file, cache = NULL, column_mapping = NULL)
```

Arguments

cs_cluster_file	Path to the CS cluster file
cache	Cache object for memoization (optional)
column_mapping	Named list mapping internal names to input column names. If NULL, uses DEFAULT_COLUMN_MAPPING$cs_clusters.

Value

Data table with CS to cluster mappings indexed by feature

load_feature_data *Load Feature Data*

Description

Generic function to load feature data (genes or peaks) with ACAT results

Usage

```
load_feature_data(config, feature_type, chromosomes = NULL, num_cores = NULL)
```

Arguments

config	Configuration list with file patterns
feature_type	Type of feature ("gene" or "peak")
chromosomes	Chromosomes to analyze
num_cores	Number of cores for parallel processing (NULL to use config$parameters$n_cores)

Value

List with feature data

load_lfsr_results *Load pre-computed LFSR results from external files*

Description

Load pre-computed LFSR results from external files

Usage

```
load_lfsr_results(config)
```

Arguments

config	Configuration list containing file paths and cell types

Value

List containing LFSR data tables for eQTL and caQTL

load_meta_data *Load Meta Data with Pre-computed Cochran's Q Values*

Description

Load meta data containing pre-computed Cochran's Q heterogeneity p-values

Usage

```
load_meta_data(meta_file, cache = NULL, column_mapping = NULL)
```

Arguments

meta_file	Path to the meta data file
cache	Cache object for memoization (optional)
column_mapping	Named list mapping internal names to input column names. If NULL, uses DEFAULT_COLUMN_MAPPING$meta.

Value

Data table with variant-phenotype pairs and heterogeneity p-values

load_variant_data *Variant Data Loading Functions*

Description

Functions for loading variant data with QTL information Load Variant Data for All Chromosomes with Parallelization

Load variant data across all specified chromosomes using parallel processing

Usage

```
load_variant_data(
  config,
  chromosomes = NULL,
  pip_threshold = 0.5,
  min_pip_threshold = 0.1,
  acat_fdr_threshold = 0.05,
  peak_bed_file = NULL,
  column_mapping = NULL,
  num_cores = NULL
)
```

Arguments

config	Configuration list
chromosomes	Chromosomes to analyze
pip_threshold	Maximum PIP threshold across cell types for filtering (default: 0.5)
min_pip_threshold	Minimum PIP threshold per cell type (default: 0.1)
acat_fdr_threshold	FDR threshold for ACAT filtering
peak_bed_file	Path to peak BED file
column_mapping	Column name mapping
num_cores	Number of cores to use (NULL for auto-detect)

Value

Combined variant data from all chromosomes

---

print.CellTypeHierarchy *Print method for CellTypeHierarchy*

Description

Print method for CellTypeHierarchy

Usage

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

Arguments

x	A CellTypeHierarchy object
...	Additional arguments (ignored)

QTL_MECHANISMS *Scientific Definitions*

Description

QTL mechanism categories, pattern definitions, variant heterogeneity codes, and other scientific constants used in categorization. QTL Mechanism Categories

Eight main categories for variant QTL mechanisms

Usage

```
QTL_MECHANISMS
```

Format

An object of class `character` of length 8.

QTL_PATTERNS *QTL Pattern Details*

Description

Maps 25 QTL patterns to their interpretations and mechanism categories Each pattern has: interpretation (detailed description) and mechanism (category index)

Usage

```
QTL_PATTERNS
```

Format

An object of class `list` of length 25.

run_cascade *Run the CASCADE pipeline*

Description

Main entry point for a CASCADE analysis. Loads configured QTL inputs, runs gene/peak/variant categorization, and writes results to disk.

Usage

```
run_cascade(config, output_dir = "results")
```

Arguments

config	Configuration object (from 'create_config()') or path to a JSON config file.
output_dir	Directory to write results to. Created if it does not exist.

Value

Invisibly, a list of categorization results (gene, peak, variant).

---

save_variant_results_cross_celltype
Save Cross-Cell-Type Variant Results

Description

Save Stage 2 results to TSV file (combined L1 + L2)

Usage

```
save_variant_results_cross_celltype(
  cross_celltype_results,
  output_dir,
  suffix = ""
)
```

Arguments

cross_celltype_results	Cross-cell-type results data frame
output_dir	Output directory
suffix	Optional suffix for file names (e.g., "l2")

save_variant_results_per_celltype *Save Per-Cell-Type Variant Results*

Description

Save Stage 1 results to separate TSV files per cell type

Usage

```
save_variant_results_per_celltype(per_celltype_results, output_dir)
```

Arguments

per_celltype_results	List of per-cell-type results
output_dir	Output directory

VARIANT_HETEROGENEITY *Variant Heterogeneity Categories*

Description

Categories for variant heterogeneity in multi-cell-type features. Maps category names to letter codes (a-d)

Usage

```
VARIANT_HETEROGENEITY
```

Format

An object of class `character` of length 4.

Index