Package: multipoolR (via r-universe)

November 2, 2026

Title Efficient Multi-Locus Genetic Mapping in R

Version 0.10.5

Description An R package port for Multitool for efficient multi-locus genetic mapping of quantitative traits using a multi-pool sequencing approach. The package implements a Bayesian hierarchical model to estimate allele frequencies and test for associations between genetic variants and phenotypes. It provides functions for data preprocessing, model fitting, and visualization of results.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0)

Imports checkmate, data.table, ggplot2, lifecycle, matrixStats (>= 1.2.0), ggrepel, stats, AnnotationDbi, BiocGenerics, future.apply, GenomeInfoDb, GenomicFeatures, GenomicRanges, S4Vectors, TxDb.Scerevisiae.UCSC.sacCer3.sgdGene, org.Sc.sgd.db, testthat (>= 3.0.0)

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/edition 3

LazyData true

Config/pak/sysreqs cmake make libbz2-dev liblzma-dev libpng-dev libuv1-dev libxml2-dev libssl-dev xz-utils zlib1g-dev

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

Date/Publication 2025-12-26 13:37:34 UTC

RemoteUrl https://github.com/clstacy/multipoolR

RemoteRef HEAD

RemoteSha 9ea6c2f54a0d3f768c3ff80a91fb0d7b8bb84ae6

Contents

multipool 2
multipool_example_data 5
multipoolR 6

Index **8**

`multipool` *Perform Multipool Genome Scan*

Description

This is the main function to run the multipool analysis. It loads data for two pools, performs QTL mapping using a Kalman filter approach, optionally runs permutation tests to estimate significance (p-values and q-values), annotates significant QTL regions with yeast gene information, and generates a genome-wide plot of the results.

Usage

```
multipool(
  pool1,
  pool2,
  N,
  mode = c("replicates", "contrast"),
  res = 100,
  cM = 3300,
  filter = TRUE,
  nperm = 0,
  alpha = NULL,
  plot = TRUE,
  col_chr = "chr",
  col_pos = "pos",
  col_a = "a",
  col_b = "b",
  assume_chr = "chrI",
  header = TRUE,
  seed = NULL,
  parallel = FALSE,
  txdb = NULL,
  orgdb = NULL,
  q_threshold = DEFAULT_Q_THRESHOLD,
  verbose = TRUE
)
```

Arguments

pool1	A file path (character string) to the first pool's data, or a data frame containing the required columns (chr, pos, a, b).
pool2	A file path (character string) to the second pool's data, or a data frame containing the required columns (chr, pos, a, b).
N	Numeric. The effective number of individuals contributing to each sequencing pool. This accounts for the variance introduced during pooling and sequencing.
mode	Character string. Specifies the analysis mode: **"replicates"** Assumes the two pools are biological or technical replicates selected under the *same* condition. The LOD score tests for deviations from the expected 0.5 allele frequency in *both* pools simultaneously. **"contrast"** Assumes the two pools were selected under *different* conditions. The LOD score tests for significant *differences* in allele frequency between the two pools.
res	Numeric. The desired bin resolution in base pairs for analysis (default: 100). Marker data within each bin is aggregated.
cM	Numeric. An estimate of the average number of base pairs per centiMorgan for the organism (default: 3300 for yeast). Used to calculate the recombination rate between adjacent bins.
filter	Logical. If TRUE (default), apply filtering to remove markers with zero counts for either allele ('fixated' markers) and bins with extremely high coverage (potential PCR duplicates/mapping artifacts).
nperm	Integer. The number of permutations to perform for establishing significance (p-values and q-values) (default: 0, no permutations). Setting this to > 0 (e.g., 1000) is recommended for robust analysis.
alpha	Deprecated. Significance level is now typically determined by choosing a q-value threshold (e.g., q < 0.05) on the results.
plot	Logical. If TRUE (default), generate a genome-wide plot of the results using ggplot2.
txdb	A TxDb object for gene annotations (default: TxDb.Scerevisiae.UCSC.sacCer3.sgdGene). See GenomicFeatures package.
orgdb	An OrgDb object for gene annotations (default: org.Sc.sgd.db). See AnnotationDbi package.
q_threshold	Numeric. The q-value threshold for calling significance in annotation and highlighting genes on the plot when nperm > 0. (Default: 0.05).

Value

A list containing:

results A data frame with genome-wide results, including columns for chromosome (chr), bin start position (pos), observed allele frequencies (freq1_obs, freq2_obs), smoothed posterior allele frequencies (freq1_fit, freq2_fit), LOD scores (LOD), MLE allele frequencies or differences (mu_MLE_adj), and if nperm > 0, empirical p-values (p_value) and q-values (q_value).

fdr_lod_threshold The LOD score threshold corresponding to the chosen `q_threshold`, if permutations were run and significant results found. Otherwise NULL.

plot A ggplot object containing the genome-wide plot. NULL if `plot` was FALSE.

annotated_genes A data frame containing information about genes overlapping bins. If `nperm > 0`, this includes genes overlapping bins meeting the `q_threshold`. If `nperm = 0`, this includes genes overlapping *any* bin. NULL if annotation databases unavailable.

parameters A list storing the key parameters used for the analysis.

Examples

```
## Not run:
# --- Example with dummy data ---

# Create dummy data frames
make_dummy_pool <- function(n_markers = 5000, chr = "chrI") {
  pos <- sort(sample(1:200000, n_markers, replace = TRUE))
  a <- rpois(n_markers, lambda = 20)
  b <- rpois(n_markers, lambda = 20)
  # Introduce a fake QTL
  qtl_region <- pos > 80000 & pos < 120000
  a[qtl_region] <- rpois(sum(qtl_region), lambda = 35)
  b[qtl_region] <- rpois(sum(qtl_region), lambda = 10)
  data.frame(chr = chr, pos = pos, a = a, b = b)
}

pool1_df <- make_dummy_pool(chr = "chrI")
pool2_df <- make_dummy_pool(chr = "chrI") # Replicate

# Combine into two chromosomes for a more realistic example
pool1_combined <- rbind(pool1_df, make_dummy_pool(chr="chrII", n_markers=3000))
pool2_combined <- rbind(pool2_df, make_dummy_pool(chr="chrII", n_markers=3000))

# Ensure required packages are loaded (needed for default annotation DBs)
# library(TxDb.Scerevisiae.UCSC.sacCer3.sgdGene)
# library(org.Sc.sgd.db)

# Run analysis in "replicates" mode with permutations for FDR
results_rep <- multipool(
  pool1 = pool1_combined,
  pool2 = pool2_combined,
  N = 50,
  mode = "replicates",
  res = 500,
  cM = 3300,
  nperm = 1000, # Use >= 1000 for reliable FDR
  q_threshold = 0.10 # Use a 10% FDR threshold
)

# View results with p/q-values
print(head(results_rep$results))
# Print the LOD threshold corresponding to the FDR cutoff
```

```
print(paste("LOD threshold for q <", results_rep$parameters$q_threshold, ":",
            round(results_rep$fdr_lod_threshold, 3)))
# View plot (will highlight genes with q < 0.1 and show threshold line)
if (!is.null(results_rep$plot)) {
  print(results_rep$plot)
}
print(results_rep$annotated_genes) # Genes meeting q-threshold

# Run analysis without permutations (nperm = 0)
results_no_perm <- multipool(
  pool1 = pool1_combined,
  pool2 = pool2_combined,
  N = 50,
  mode = "replicates",
  res = 500,
  cM = 3300,
  nperm = 0
)

# View plot (should label genes near peaks)
if (!is.null(results_no_perm$plot)) {
  print(results_no_perm$plot)
}
print(results_no_perm$annotated_genes) # All overlapping genes


## End(Not run) # end dontrun
```

`multipool_example_data` *Example Multipool Data with Sloped QTLs*

Description

A simulated dataset containing allele counts for two pools (1 and 2) across two chromosomes (chrIII and chrV), designed to illustrate the usage of the multipoolR package.

Usage

```
data(multipool_example_data)
```

Format

A list containing two data frames, which are loaded into the environment when `data(multipool_example_data)` is called:

multipoolR_example_pool1 Data frame for Pool 1 (7000 rows)

multipoolR_example_pool2 Data frame for Pool 2 (7000 rows)

Each data frame has columns:

chr Chromosome name (chrIII or chrV)

pos Genomic position (integer)

a Allele count for reference/parent 1 (integer)

b Allele count for reference/parent 2 (integer)

Details

This dataset includes simulated QTLs with effects that ramp linearly from the baseline frequency (0.5) at the edges to a peak frequency at the center of the defined QTL region.

Pool 1 QTLs:

- chrIII: Peak frequency 0.80 between 140kb and 175kb.
- chrV: Peak frequency 0.35 between 400kb and 420kb.

Pool 2 QTLs:

- chrIII: Peak frequency 0.65 between 140kb and 175kb.
- chrV: Peak frequency 0.60 between 400kb and 420kb.

The data was generated using the `generate_multipool_data` function (see README or examples for its definition) with specific seeds for reproducibility.

Source

Simulated data generated for package examples. See README for generation code.

`multipoolR` *multipoolR: Multipool QTL Mapping in R*

Description

A comprehensive package to perform multipool QTL scans using a Kalman filter approach, permutation testing for significance (calculating q-values via FDR), yeast gene annotation for significant QTLs, and genome-wide visualization of results. Adapted from the original Python implementation by Edward and Giffords.

Main Function

`multipool`

Data Format

Input data (either files or data frames) should contain columns:

chr Chromosome identifier (e.g., "chrI", "chrII"). Must be consistent between pools.

pos Genomic position (numeric).

a Read count supporting allele 'A' (numeric).

b Read count supporting allele 'B' (numeric).

Algorithm Details

The core algorithm uses a Kalman filter and smoother to estimate the underlying allele frequency trajectory along the genome for each pool, accounting for recombination and sampling variance. LOD scores are calculated based on the likelihood ratio comparing a model with a QTL at a specific location versus a null model (no QTL), maximized over possible true allele frequencies using numerical optimization. Genome-wide permutation tests can be performed to estimate empirical p-values and q-values (False Discovery Rate).

References

Magwene, P. M., Willis, J. H., & Kelly, J. K. (2011). The statistics of bulk segregant analysis using next generation sequencing. PLoS computational biology, 7(11), e1002255.

Index

∗ **datasets**
multipool_example_data, 5

multipool, 2, *6*
multipool_example_data, 5
multipoolR, 6
multipoolR_example_pool1 *(*multipool_example_data*)*, 5
multipoolR_example_pool2 *(*multipool_example_data*)*, 5