Package 'BootstrapQTL'

May 12, 2021

Type Package

Title Bootstrap cis-QTL Method that Corrects for the Winner's Curse

Version 1.0.5

Author Qin Qin Huang [aut],
Scott Ritchie [aut, cre]

Maintainer Scott Ritchie <sritchie73@gmail.com>

BugReports https://github.com/sritchie73/bootstrapQTL/issues

Description Identifies genome-related molecular traits with significant evidence of genetic regulation and performs a bootstrap procedure to correct estimated effect sizes for over-estimation present in cis-QTL mapping studies (The ``Winner's Curse''), described in Huang QQ *et al.* 2018 <doi: 10.1093/nar/gky780>.

Depends MatrixEQTL

Imports foreach, data.table

Suggests doMC, doParallel, qvalue, testthat

License GPL-2

Encoding UTF-8

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2021-05-12 00:52:43 UTC

R topics documented:

BootstrapQTL *Bootstrap QTL analysis for accurate effect size estimation*

Description

Performs cis-QTL mapping using MatrixEQTL then performs a bootstrap analysis to obtain unbiased effect size estimates for traits with significant evidence of genetic regulation correcting for the "Winner's Curse" arising from lead-SNP selection.

Usage

```
BootstrapQTL(snps, gene, snpspos, genepos, cvrt = SlicedData$new(),
  n_bootstraps = 200, n_cores = 1, eGene_detection_file_name = NULL,
  bootstrap_file_directory = NULL, cisDist = 1e+06,
  local_correction = "bonferroni", global_correction = "fdr",
  correction_type = "shrinkage", errorCovariance = numeric(),
  useModel = modelLINEAR, eigenMT_tests_per_gene = NULL)
```

Arguments

snps	SlicedData object containing genotype information used as input into Matrix_eQTL_main.
gene	SlicedData object containing gene expression information used as input into Matrix_eQTL_main.
snpspos	data.frame object with information about SNP locations. Used in conjunction with 'genespos' and 'cisDist' to determine SNPs in *cis* of each gene. Must have three columns: 1. 'snpid' describing the name of the SNP and corresponding to rows in the 'snps' matrix. 2. 'chr' describing the chromosome for each SNP. 3. 'pos' describing the position of the SNP on the chromosome.
genepos	data.frame object with information about transcript locations. Used in conjunction with 'snpspos' and 'cisDist' to determine SNPs in *cis* of each gene. Must have four columns: 1. 'geneid' describing the name of the gene and corresponding to rows in the 'gene' matrix. 2. 'chr' describing the chromosome for each SNP. 3. 'left' describing the start position of the transcript. 4. 'right' describing the end position of the transcript. Note that Matrix_eQTL_main tests all variants within cisDist of the start or end of the gene. If you wish instead to test all variants within cisDist of the transcription start site, you should specify this location in both the 'left' and 'right' columns of the genepos data.frame. Similarly, when analysing a molecular phenotype that have a single chromosomal position then the 'left' and 'right' columns should both contain the same position.

cvrt — SlicedData object containing covariate information used as input into Matrix_eQTL_main. Argument can be ignored in the case of no covariates.

n_bootstraps — number of bootstraps to run.

n_cores — number of cores to parallise the bootstrap procedure over.

eGene_detection_file_name — character, connection or NULL. File to save local *cis* associations to in the eGene detection analysis. Corresponds to the output_file_name.cis argument in Matrix_eQTL_main. If a file with this name exists it is overwritten, if NULL output is not saved to file.

bootstrap_file_directory — character or NULL. If not NULL, files will be saved in this directory storing local *cis* associations for the bootstrap eGene detection group (detection_bootstrapnumber.txt) and local *cis* associations the bootstrap left-out eGene effect size estimation group (estimation_bootstrapnumber.txt). Estimation group files will only be saved where signficant eGenes are also significant in the bootstrap detection group (see Details). Corresponds to the output_file_name.cis argument in the respective calls to Matrix_eQTL_main. Files in this directory will be overwritten if they already exist.

cisDist — numeric. Argument to Matrix_eQTL_main controlling the maximum distance from a gene to consider tests for eQTL mapping.

local_correction — multiple testing correction method to use when correcting p-values across all SNPs at each gene (see EQTL mapping section in Details). Can be a method specified in p.adjust.methods, "qvalue" for the qvalue package, or "eigenMT" if EigenMT has been used to estimate the number effective independent tests (see eigenMT_tests_per_gene).

global_correction — multiple testing correction method to use when correcting p-values across all genes after performing local correction (see EQTL mapping section in Details). Must be a method specified in p.adjust.methods or "qvalue" for the qvalue package.

correction_type — character. One of "shrinkage", "out_of_sample" or "weighted". Determines which Winner's Curse correction method is used (see Details).

errorCovariance — numeric matrix argument to Matrix_eQTL_main specifying the error covariance.

useModel — integer argument to Matrix_eQTL_main specifying the type of model to fit between each SNP and gene. Should be one of modelLINEAR, modelANOVA, or modelLINEAR_CROSS.

eigenMT_tests_per_gene — data.frame containing the number of effective independent tests for each gene estimated by the EigenMT (https://github.com/joed3/eigenMT). Ignore unless 'local_correction="eigenMT"'.

Details

Although the package interface and documentation describe the use of `BootstrapQTL` for *cis*-eQTL mapping, the package can be applied to any QTL study of quantitative traits with chromosomal positions, for example *cis*-QTL mapping of epigenetic modifications. Any matrix of molecular trait data can be provided to the `'gene'` argument provided a corresponding `'genepos'` 'data.frame' detailing the chromosomal positions of each trait is provided.

Cis-eQTL mapping:: EQTL mapping is performed using the `MatrixEQTL` package. A three step hiearchical multiple testing correction procedure is used to determine significant eGenes and eSNPs. At the first step, nominal p-values from `MatrixEQTL` for all *cis*-SNPs are adjusted for each gene separately using the method specified in the `'local_correction'` argument (Bonferroni correction by default). In the second step, the best adjusted p-value is taken for each gene, and this set of locally adjusted p-values is corrected for multiple testing across all genes using the methods pecified in the `'global_correction'` argument (FDR correction by default). In the third step, an eSNP significance threshold on the locally corrected p-values is determined as the locally corrected p-value corresponding to the globally corrected p-value threshold of 0.05.

A gene is considered a significant eGene if its globally corrected p-value is < 0.05, and a SNP is considered a significant eSNP for that eGene if its locally corrected p-value < the eSNP significance threshold.

The default settings for `'local_correction'` and `'global_correction'` were found to best control eGene false discovery rate without sacrificing sensitivity (see citation).

Winner's Curse correction:: EQTL effect sizes of significant eSNPs on significant eGenes are typically overestimated when compared to replication datasets (see citation). `BootstrapEQTL` removes this overestimation by performing a bootstrap procedure after eQTL mapping.

Three Winner's Curse correction methods are available: the Shrinkage method, the Out of Sample method, and the Weighted Estimator method. All three methods work on the same basic principle of performing repeated sample bootstrapping to partition the dataset into two groups: an eQTL detection group comprising study samples select via random sampling with replacement, and an eQTL effect size estimation group comprising the remaining samples not selected via the random sampling. The default estimator, `'correction_type = "shrinkage"'`, provided the most accurate corrected effect sizes in our simulation study (see citation).

The **shrinkage method** ("shrinkage" in `'correction_type'`) corrects for the Winner's Curse by measuring the average difference between the eQTL effect size in the bootstrap detection group and the bootstrap estimation group, then subtracting this difference from the naive eQTL effect size estimate obtained from the eGene detection analysis prior to the bootstrap procedure.

The **out of sample method** ("out_of_sample" in `'correction_type'`) corrects for the Winner's Curse by taking the average eQTL effect size across bootstrap estimation groups as an unbiased effect size estimate.

The **weighted estimator method** ("weighted" in `'correction_type'`) corrects for the Winner's Curse by taking a weighted average of the nominal estimate of the eQTL effect size and the average of eQTL effect sizes across the bootstrap estimation groups: $0.368 * naive_estimate + 0.632 * mean(bootstrapestimationgroupeffectsizes)$.

In all three methods bootstrap effect sizes only contribute to the Winner's Curse correction if the corresponding eSNP is significantly associated with the eGene in the bootstrap detection group (locally corrected bootstrap P-value < eSNP significance threshold determing in the eQTL mapping step).

Note that eQTLs may not remain significant in all bootstraps, so the effective number of bootstraps used to obtain the Winner's Curse estimate will typically be lower than the number of bootstraps specified in 'n_bootstraps'. The number of bootstraps that were significant for each eQTL are reported in the 'correction_boots' column of the returned table.

Winner's Curse corrected effect sizes: The user should be aware that ability to correct for Winner's Curse can vary across significant eQTLs depending on their statistical power (*i.e. minor allele frequency, true effect size, and study sample size*). Users should be skeptical of corrected effect sizes that are larger than the nominal effect sizes estimated by MatrixEQTL, which likely reflects low power for eQTL detection rather than an underestimated effect size.

Bootstrap warning messages:: It is possible for bootstrap analyses to fail due to the reduced sample sizes of the bootstrap detection and bootstrap estimation groups. For example, the bootstrap resampling may lead to an detection or estimation groups in which all individuals are homozygous for an eSNP or have no variance in their supplied covariates (*e.g.* the estimation group may comprise individuals all of the same sex). In this case the bootstrap will fail for all eQTLs since MatrixEQTL will be unable to perform the model fitting.

Failed bootstraps are reported after the bootstrap procedure in a series of warning messages indicating the number of bootstrap failures grouped by the reason for the bootstrap failure.

Value

A data.frame (or data.table if the user has the library loaded) containing the results for each significant eQTL:

'eGene': The eQTL eGene.

'eSNP': The eQTL eSNP.

'statistic': The test statistic for the association between the eGene and eSNP.

'nominal_beta': The eQTL effect size for the eGene-eSNP pair estimated by MatrixEQTL.

'corrected_beta': The eQTL effect size after adjustment for the winners_curse.

'winners_curse': The amount of effect size overestimation determined by the bootstrap analysis (See Details).

'correction_boots': The number of bootstraps that contributed to the estimation of the winners_curse, *i.e.* the number of bootstraps in which the eSNP remained significantly associated with the eGene (see Details).

'nominal_pval': The p-value for the eGene-eSNP pair from the MatrixEQTL analysis.

'eSNP_pval': The locally corrected p-value for the eGene-eSNP pair (see Details).

'eGene_pval': The globally corrected p-value for the eGene based on its top eSNP (see Details).

Examples

```
# Locations for example data from the MatrixEQTL package
base.dir = find.package('MatrixEQTL');
SNP_file_name = paste(base.dir, "/data/SNP.txt", sep="");
snps_location_file_name = paste(base.dir, "/data/snpsloc.txt", sep="");
expression_file_name = paste(base.dir, "/data/GE.txt", sep="");
gene_location_file_name = paste(base.dir, "/data/geneloc.txt", sep="");
```

```
covariates_file_name = paste(base.dir, "/data/Covariates.txt", sep="");

# Load the SNP data
snps = SlicedData$new();
snps$fileDelimiter = "\t";      # the TAB character
snps$fileOmitCharacters = "NA"; # denote missing values;
snps$fileSkipRows = 1;          # one row of column labels
snps$fileSkipColumns = 1;       # one column of row labels
snps$fileSliceSize = 2000;      # read file in slices of 2,000 rows
snps$LoadFile(SNP_file_name);

# Load the data detailing the position of each SNP
snpspos = read.table(snps_location_file_name, header = TRUE, stringsAsFactors = FALSE);

# Load the gene expression data
gene = SlicedData$new();
gene$fileDelimiter = "\t";      # the TAB character
gene$fileOmitCharacters = "NA"; # denote missing values;
gene$fileSkipRows = 1;          # one row of column labels
gene$fileSkipColumns = 1;       # one column of row labels
gene$fileSliceSize = 2000;      # read file in slices of 2,000 rows
gene$LoadFile(expression_file_name);

# Load the data detailing the position of each gene
genepos = read.table(gene_location_file_name, header = TRUE, stringsAsFactors = FALSE);

# Load the covariates data
cvrt = SlicedData$new();
cvrt$fileDelimiter = "\t";      # the TAB character
cvrt$fileOmitCharacters = "NA"; # denote missing values;
cvrt$fileSkipRows = 1;          # one row of column labels
cvrt$fileSkipColumns = 1;       # one column of row labels
if(length(covariates_file_name)>0) {
  cvrt$LoadFile(covariates_file_name);
}

# Run the BootstrapQTL analysis
eQTLs <- BootstrapQTL(snps, gene, snpspos, genepos, cvrt, n_bootstraps=10, n_cores=2)
```

Index