Package 'iBMQ'

November 12, 2026

Type Package

Title integrated Bayesian Modeling of eQTL data

Version 1.52.0

Date 2011-10-28

Author Marie-Pier Scott-Boyer and Greg Imholte

Maintainer Greg Imholte <gimholte@uw.edu>

Description integrated Bayesian Modeling of eQTL data

License Artistic-2.0

Depends R(>= 2.15.0),Biobase (>= 2.16.0), ggplot2 (>= 0.9.2)

biocViews Microarray, Preprocessing, GeneExpression, SNP

URL http://www.rglab.org

SystemRequirements GSL and OpenMP

git_url https://git.bioconductor.org/packages/iBMQ

git_branch RELEASE_3_23

git_last_commit 384d3ce

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

iBMQ-package 2
calculateThreshold 2
eqtlClassifier 3
eqtlFinder 4
eqtlMcmc 5
gene 6
genepos 6
genotype.liver 7
hotspotFinder 7

map.liver 8
phenotype.liver 9
PPA.liver 9
probe.liver 10
snp 10
snppos 11

Index **12**

`iBMQ-package` *iBMQ : An Integrated Hierarchical Bayesian Model for Multivariate eQTL Mapping*

Description

This method is designed to detect expression QTLs (eQTLs) by incorporating genotypic and gene expression data into a single model while 1) specifically coping with the high dimensionality of eQTL data (large number of genes), 2) borrowing strength from all gene expression data for the mapping procedures, and 3) controlling the number of false positives to a desirable level.

`calculateThreshold` *Calculate PPA significance threshold leading to a desired false discovery rate*

Description

In the context of multiple testing and discoveries, a popular approach is to use a common threshold leading to a desired false discovery rate (FDR). In the Bayesian paradigm, derivation of the PPA threshold is trivial and can be calculated using a direct posterior probability calculation as described in Newton et al. (2004).

Usage

```
calculateThreshold(prob, threshold)
```

Arguments

`prob`	matrix or data frame that contains Posterior Probability of Association (output of eqtlMcmc function).
`threshold`	The desired false discovery rate.

Value

`cutoff`	The significance threshold value

References

Newton, MA., Noueiry, A., Sarkar, D. and Ahlquist, P. (2004): "Detecting differential gene expression with a semiparametric hierarchical mixture method."Biometrics, 5(2), 155-176

Examples

```
data(PPA.liver)
cutoff.liver <- calculateThreshold(PPA.liver, 0.2)
```

---

eqtlClassifier *Classifying the eQTLs*

Description

It is customary to distinguish two kinds of eQTLs: 1) cis-eQTLs (where the eQTL is on the same locus as the expressed gene); and 2) trans-eQTLs (where the eQTL is on a locus other than that of the expressed gene). The eqtlClassifier allows us to classify the eQTLs as either cis-eQTL or trans-eQTL according to their position in the genome.

Usage

```
eqtlClassifier(peak, posSNP, posGENE, max)
```

Arguments

peak	A data.frame of significant eQTLs (the output of the findEqtl function)
posSNP	A data frame specifying the genomic locations of genomic markers (i.e. SNPs).
posGENE	A data frame specifying the genomic locations of the genes (or probes).
max	A cutoff value (in base pair) corresponding to the threshold where a eQTL is considered to be cis-eQTL. A numerical value.

Value

The output of the eqtlClassifier is a data frame where the first column contains the names of each gene, the second column contains the names of markers and the third column contains the PPA value for each significant eQTL. The fourth column contains the number of the chromosome to which the gene belongs, the fifth column contains the start position of the gene and the sixth column contains the end position of the gene. The seventh column contains the number of the chromosome to which the marker belongs, the eighth column contains position of the marker and the ninth column contains a descriptor of the type of eQTL (either cis or trans). Please note that in order to ascertain that an eQTL is either cis or trans, the positions of the markers and the gene need to the given to the function. If one of the values is missing the type of eQTL will be "NA".

Examples

```
data(PPA.liver)
cutoff.liver <- calculateThreshold(PPA.liver, 0.2)
eqtl.liver <- eqtlFinder(PPA.liver, cutoff.liver)
data(map.liver)
data(probe.liver)
eqtl.type.liver <- eqtlClassifier(eqtl.liver, map.liver, probe.liver,5000000)
```

`eqtlFinder` *eqtlFinder*

Description

We can calculate how many eQTLs have PPA above the cutoff with the eqtlFinder function.

Usage

```
eqtlFinder(prob, threshold)
```

Arguments

`prob`	matrix or data frame that contains the Posterior Probability of Association values (output of eqtlMcmc function)
`threshold`	Threshold to be used to determine which QTLs are significant. This value can be the output of the calculateThreshold function.It must be a numerical value between 0 and 1.

Value

The output of the eqtlFinder is a data frame where the first column contains the names of each gene, the second column contains the names of corresponding markers and the third column contains the PPA value for each significant eQTL.

Examples

```
data(PPA.liver)
cutoff.liver <- calculateThreshold(PPA.liver, 0.2)
eqtl.liver <- eqtlFinder(PPA.liver, cutoff.liver)
```

`eqtlMcmc` *Bayesian Multiple eQTL mapping using MCMC*

Description

Compute the MCMC algorithm to produce Posterior Probability of Association values for eQTL mapping.

Usage

```
eqtlMcmc(snp, expr, n.iter, burn.in, n.sweep, mc.cores,
 write.output = TRUE, RIS = TRUE)
```

Arguments

`snp`	SnpSet class object
`expr`	ExpressionSet class object
`n.iter`	Number of samples to be saved from the Markov Chain
`burn.in`	Number of burn-in iterations for the Markov Chain
`n.sweep`	Number of iterations between samples of the Markov Chain (AKA thinning interval)
`mc.cores`	The number of cores you would like to use for parallel processing. Can be set be set via 'options(cores=4)', if not set, the code will automatically detect the number of cores.
`write.output`	Write chain iterations to file. If TRUE, output for variables will be written to files created in the working directory.
`RIS`	If TRUE, the genotype needs to be either 0 and 1. If FALSE the genotype need to be either 1,2 and 3.

Details

The value of `mc.cores` may be ignored and set to one when the iBMQ installation does not support openMP.

Value

A matrix with Posterior Probability of Association values. Rows correspond to snps from original snp data objects, columns correspond to genes from expr data objects.

References

Scott-Boyer, MP., Tayeb, G., Imholte, Labbe, A., Deschepper C., and Gottardo R. An integrated Bayesian hierarchical model for multivariate eQTL mapping (iBMQ). Statistical Applications in Genetics and Molecular Biology Vol. 11, 2012.

Examples

```
data(phenotype.liver)
data(genotype.liver)
#PPA.liver <- eqtlMcmc(genotype.liver, phenotype.liver, n.iter=100,burn.in=100,n.sweep=20,mc.cores=6, RIS=FALSE
```

---

gene *Gene expression from whole eye tissue from n = 68 BXD RIS mice.*

Description

This dataset comprises the profiles of mRNA abundance in whole eye tissue from n = 68 BXD RIS mice, as measured using Affymetrix M430 2.0 microarrays. To ease calculation and facilitate comparisons, we will use a set of G = 1000 probes

Usage

```
data(gene)
```

Format

The format is: Formal class 'ExpressionSet' [package "Biobase"]

Source

This example uses data generated by Williams and Lu, as available from the Gene Networkwebsite (genenetwork.com). This dataset comprises the profiles of mRNA abundance in whole eye tissue from n = 68 BXD RIS mice, as measured using Affymetrix M430 2.0 microarrays. To ease calculation and facilitate comparisons, we will use a set of G = 1000 probes

Examples

```
data(gene)
```

---

genepos *Gene position data frame*

Description

A data frame specifying the genomic locations of each gene/probe needs to be prepared with the following columns: gene name, chromosome number, start location (in base pairs) and the location (in base pairs).

Format

A data frame with 1000 observations with the following columns: gene name, chromosome number, start location (in base pairs) and the location (in base pairs).

Source

This example uses data generated by Williams and Lu, as available from the Gene Networkwebsite (genenetwork.com). This dataset comprises the profiles of mRNA abundance in whole eye tissue from n = 68 BXD RIS mice, as measured using Affymetrix M430 2.0 microarrays. To ease calculation and facilitate comparisons, we will use a set of G = 1000 probes

Examples

```
data(genepos)
```

`genotype.liver` *A set of 290 SNPs from 60 F2 mice.*

Description

A set of 290 single nucleotide polymorphic markers (SNPs) from 60 F2 mice.

Format

The format is: Formal class 'SnpSet' [package "Biobase"]

Source

This F2 cross data set containing genotypic and phenotypic information for 60 mice was obtained from the lab of Alan Attie at the University of Wisconsin-Madison. These data are also available at GEO (accession number GSE3330). Only the 5000 most variable expression traits out of 45,265 transcripts from the liver were used for the current example.

Examples

```
data(genotype.liver)
```

`hotspotFinder` *hotspotFinder*

Description

One main advantage of our method is its increased sensitivity for finding trans-eQTL hotspots (corresponding to situations where a single SNP is linked to the expression of several genes across the genome).

Usage

```
hotspotFinder(peak, numgene)
```

Arguments

peak	A data frame (3 columns) corresponding to the output of the eqtlFinder function or the data frame (9 columns) corresponding to the output of the eqtlClassifier function.
numgene	The minimum of gene to detect.

Value

The output of this function is a list, where each element is a marker. For each marker there is a data frame with all the eQTLs linked to this marker.

Examples

```
data(PPA.liver)
cutoff.liver <- calculateThreshold(PPA.liver, 0.2)
eqtl.liver <- eqtlFinder(PPA.liver, cutoff.liver)
hotspot.liver <- hotspotFinder(eqtl.liver,20)
```

---

map.liver *SNP position data frame*

Description

A data frame specifying the genomic locations of each SNP with following columns: SNP name, chromosome number, SNP location (in base pair).

Format

A data frame with 290 observations with the following columns: SNP name, chromosome number, SNP location (in base pair).

Source

This F2 cross data set containing genotypic and phenotypic information for 60 mice was obtained from the lab of Alan Attie at the University of Wisconsin-Madison. These data are also available at GEO (accession number GSE3330). Only the 5000 most variable expression traits out of 45,265 transcripts from the liver were used for the current example.

Examples

```
data(map.liver)
```

phenotype.liver *Gene expression from liver tissue from n = 60 F2 mice.*

Description

Gene expression of 5000 probe from liver tissue from n = 60 F2 mice.

Format

The format is: Formal class 'ExpressionSet' [package "Biobase"]

Source

This F2 cross data set containing genotypic and phenotypic information for 60 mice was obtained from the lab of Alan Attie Lab at the University of Wisconsin-Madison. These data are also available at GEO (accession number GSE3330). Only the 5000 most variable expression traits out of 45,265 transcripts from the liver were used for the current example.

Examples

```
data(phenotype.liver)
```

PPA.liver *A matrix with Posterior Probabilities of Association*

Description

The result is a matrix with Posterior Probabilities of Association for each gene (row) and SNP (column). The PPA matrix was previously calculated with 100,000 iterations for liver tissue from n = 60 F2 mice dataset

Examples

```
data(PPA.liver)
```

probe.liver *Gene position data frame*

Description

A data frame specifying the genomic locations of each gene/probe needs to be prepared with the following columns: gene name, chromosome number, start location (in base pairs) and the location (in base pairs).

Usage

```
data(probe.liver)
```

Format

A data frame with 4427 observations with the following columns: gene name, chromosome number, start location (in base pairs) and the location (in base pairs).

Source

This F2 cross data set containing genotypic and phenotypic information for 60 mice was obtained from the lab of Alan Attie Lab at the University of Wisconsin-Madison. These data are also available at GEO (accession number GSE3330). Only the 5000 most variable expression traits out of 45,265 transcripts from the liver were used for the current example.

Examples

```
data(probe.liver)
```

snp *A set of 1700 SNP from 68 BXD RIS mice.*

Description

A set of 1700 single nucleotide polymorphic markers (SNPs) from 68 BXD RIS mice.

Usage

```
data(snp)
```

Format

The format is: Formal class 'SnpSet' [package "Biobase"]

Source

This example uses data generated by Williams and Lu, as available from the Gene Networkwebsite (genenetwork.com). This dataset comprises the profiles of mRNA abundance in wholeeye tissue from n = 68 BXD RIS mice, as measured using Affymetrix M430 2.0 microarrays . To ease calculation and facilitate comparisons, we will use a set of G = 1000 probes and 1700 single nucleotide polymorphic markers (SNPs).

Examples

```
data(snp)
```

---

snppos *SNP position data frame*

Description

A data frame specifying the genomic locations of each SNP with following columns: SNP name, chromosome number, SNP location (in base pair).

Usage

```
data(snppos)
```

Format

A data frame with 1700 observations with the following columns: SNP name, chromosome number, SNP location (in base pair).

Source

This example uses data generated by Williams and Lu, as available from the Gene Network website (genenetwork.com).

Examples

```
data(snppos)
```

Index

∗ **datasets**
gene, 6
genepos, 6
genotype.liver, 7
map.liver, 8
phenotype.liver, 9
PPA.liver, 9
probe.liver, 10
snp, 10
snppos, 11

calculateThreshold, 2

eqtlClassifier, 3
eqtlFinder, 4
eqtlMcmc, 5

gene, 6
genepos, 6
genotype.liver, 7

hotspotFinder, 7

iBMQ *(*iBMQ-package*)*, 2
iBMQ-package, 2

map.liver, 8

phenotype.liver, 9
PPA.liver, 9
probe.liver, 10

snp, 10
snppos, 11