Package 'ACMEeqtl'

March 11, 2017

Type Package

Title Estimation of Interpretable eQTL Effect Sizes Using a Log of Linear Model

Version 1.4

Date 2017-03-11

Author Andrey A Shabalin, John Palowitch

Maintainer Andrey A Shabalin <andrey.shabalin@gmail.com>

Description We use a non-linear model, termed ACME, that reflects a parsimonious biological model for allelic contributions of cis-acting eQTLs. With non-linear least-squares algorithm we estimate maximum likelihood parameters. The ACME model provides interpretable effect size estimates and p-values with well controlled Type-I error. Includes both R and (much faster) C implementations.

BugReports https://github.com/andreyshabalin/ACMEeqtl/issues

Depends R (>= 3.3.0), filematrix

Imports parallel

Suggests knitr, rmarkdown, BiocStyle, pander

URL https://github.com/andreyshabalin/ACMEeqtl

License LGPL-3

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-03-11 09:18:18

R topics documented:

`ACMEeqtl-package` *Estimation of Interpretable eQTL Effect Sizes Using a Log of Linear Model*

Description

We use a non-linear model, termed ACME, that reflects a parsimonious biological model for allelic contributions of cis-acting eQTLs. With non-linear least-squares algorithm we estimate maximum likelihood paramters. The ACME model provides interpretable effect size estimates and p-values with well controlled Type-I error. Includes both R and (faster) C implementations.

Details

Package:	ACMEeqtl
Type:	Package
License:	LGPL-3

Author(s)

Andrey A Shabalin <andrey.shabalin@gmail.com>, John Palowitch

References

The manuscript is available at: http://arxiv.org/abs/1605.08799

See Also

For package overview and code examples see the package vignette via:
`browseVignettes("ACMEeqtl")`
or
`RShowDoc("doc/ACMEeqtl.html", "html", "ACMEeqtl")`

For fast testing of all local gene-SNP pairs (local eQTL) see `multithreadACME`.

create_artificial_data *Create an Artificial eQTL Data Set*

Description

Create artificial genotype, gene expression, covariate, and gene/SNP location data sets.

Usage

```
create_artificial_data(
    nsample,
    ngene = 5000,
    nsnp = 500e+03,
    ncvrt = 10,
    minMAF = 0.2,
    saveDir = getwd(),
    returnData = FALSE,
    savefmat = FALSE,
    savetxt = FALSE,
    verbose = TRUE)
```

Arguments

nsample	Number of samples.
ngene	Number of genes.
nsnp	Number of SNPs.
ncvrt	Number of covariates.
minMAF	Minimum minor allele frequency of generated SNPs.
saveDir	Directory for the output.
returnData	Set to TRUE to return generated data in an R list.
savefmat	Save generated data in filematrix format. This format is used by multithreadACME.
savetxt	Save generated data in text format. This format is used by Matrix eQTL MatrixEQTL.
verbose	Set to TRUE to indicate progress.

Value

The function generates the following text files and/or filematrices:

cvrt	Covariates
gene	Gene expression
snps	SNPs (genotype)
gene_loc	Gene locations
snps_loc	SNP locations
etas	SNP effect on the genes

Author(s)

Andrey A Shabalin <andrey.shabalin@gmail.com>, John Palowitch

References

The manuscript is available at: http://arxiv.org/abs/1605.08799

See Also

For package overview and code examples see the package vignette via:
`browseVignettes("ACMEeqtl")`
or
`RShowDoc("doc/ACMEeqtl.html", "html", "ACMEeqtl")`

For fast testing of all local gene-SNP pairs (local eQTL) see `multithreadACME`.

Examples

```
z = create_artificial_data(
    nsample = 10,
    ngene = 50,
    nsnp = 500,
    ncvrt = 1,
    minMAF = 0.2,
    returnData = TRUE,
    savefmat = FALSE,
    savetxt = FALSE,
    verbose = TRUE)

    names(z)
```

---

effectSizeEstimation *Estimate Non-Linear Model for the Effect of Genotype on Gene Expression*

Description

Estimate non-linear model for the effect of genotype on the phenotype of interest, i.e. gene expression.

Usage

```
effectSizeEstimationC(x, y, cvrt)
effectSizeEstimationR(x, y, cvrt)
```

Arguments

x	Genotype vector. Typically having 0/1/2 values.
y	Phenotype vector. Typically gene expression in normalized raw counts.
cvrt	Matrix of covariates.

Details

The function has two implementations, one fully coded in R and a faster version with core coded in C.

Value

Returns a vector with estimated paramters and diagnostics information, such as number of iterations till convergence.

The items of the vector include:

`beta0`	The constant parameter in the non-linear model (within the logarithm).
`beta1`	The effect size parameter in the non-linear model (within the logarithm).
`nits`	Number of iterations till convergence of the estimation algorithm.
`SSE`	Sum of squared residuals of the fitted model.
`SST`	Sum of squared residuals of the model with zero effect.
`F`	The F test for the significance of the genotype effect.
`eta`	The effect size parameter for simplified model (beta1/beta0).
`SE_eta`	Standard error of the eta estimate.

Author(s)

Andrey A Shabalin <andrey.shabalin@gmail.com>, John Palowitch

References

The manuscript is available at: http://arxiv.org/abs/1605.08799

See Also

For package overview and code examples see the package vignette via:
`browseVignettes("ACMEeqtl")`
or
`RShowDoc("doc/ACMEeqtl.html", "html", "ACMEeqtl")`

For fast testing of all local gene-SNP pairs (local eQTL) see `multithreadACME`.

Examples

```
# Model parameters
beta0 = 10000
beta = 50000

# Data dimensions
n = 1000
p = 19

# Standard deviation of covariate effects and noise
cvrtsd = 10
```

```
noisesd = 1

### Data generation
### Zero average covariates
cvrt = matrix(rnorm(n * p, sd = cvrtsd), n, p);
cvrt = t(t(cvrt) - colMeans(cvrt))

c_eff = rnorm(p, sd = cvrtsd)
error = rnorm(n, sd = noisesd)

# Generate SNPs
x = rbinom(n, size = 2, prob = 0.2)
y = log(beta0 + beta * x) + cvrt %*% c_eff + error

### Model estimation

z1 = effectSizeEstimationR(x, y, cvrt)
z2 = effectSizeEstimationC(x, y, cvrt)

### Compare the estimates

show(cbind(z1, z2))
```

`multisnpACME` *Estimation of Multi-SNP ACME Model for Full-Tissue Genome and All Local SNPs*

Description

This function estimates gene-wise multi-SNP ACME models. It requires the output of `multithreadACME` to know all the local SNPs for each gene. It then performs forward step-wise selection of the local SNPs, based on the adjusted-R-squared at each step. The arguments closely mirror those of `multithreadACME`; their values must correspond to a set of output files from that function (as well as the input files which originally produced the output). It saves the data in filematrix format, similar to the output of `multithreadACME`.

Note that each multi-SNP model will contain at least one SNP, even if that initial SNP was not significant under the single-SNP models. This initial SNP will be the one with the highest adjusted-R-squared value among the single-SNP models. However, after the initial SNP, further SNPs are added only if the combined model's adjusted-R-squared is greater than that from the previous combined model.

Usage

```
multisnpACME(
genefm = 'gene',
snpsfm = 'snps',
glocfm = 'gene_loc',
slocfm = 'snps_loc',
cvrtfm = 'cvrt',
```

```
acmefm = 'ACME',
workdir = NULL,
genecap = Inf,
verbose = TRUE)
```

Arguments

genefm	Name of the filematrix with gene expression data. One column per gene and one row per sample.
snpsfm	Name of the filematrix with SNP data. One column per SNP and one row per sample.
glocfm	Name of the filematix with gene location information. Must contain two columns, first with gene start location and second with the gene end. The locations must be stored as numbers, the locations for different chromosomes must differ greatly. We suggest encoding (location = 1e9 * chromosome + position_on_chromosome). The rows must match the columns of the genefm filematrix.
slocfm	Name of the filematrix with SNP locations. Must have one column and rows matching columns of snpsfm filematrix. See the instructions for glocfm above.
cvrtfm	Name of the filematirx with covariates. Must not include constant (it is added automatically). One column per covariate and one row per sample.
acmefm	Name of the filematrix to in which the ACME estimates are stored. A new file matrix with the name paste0(acmefm, '_multiSNP') will be created to store the multi-SNP ACME estimates. If the filematrix exists, it will be overwritten.
workdir	Directory where the input filematrices are located.
genecap	Number of genes to estimate multi-SNP model for.
verbose	Set to TRUE to indicate progress.

Value

The function creates a filematrix named paste0(acmefm, '_multiSNP') with 4 rows and a column for a SNP when it is included in a mult-SNP model. If the SNP is included in more than one multi-SNP model, it will appear multiple times in the matrix (but with different beta estimates, corresponding to the paritular models). The rows contain gene-SNP ids, step-wise adjusted-R-squared statistics, and beta estimates:

geneid	The gene id - the column number for the gene in the genefm filematrix.
snp_id	The SNP id - the column number for the SNP in the snpsfm filematrix.
beta0	The beta0 estimate in the full model.
beta	The beta estimate for the SNP in the full model (after all chosen SNPs have been added).
forward_adjR2	The step-wise adjusted-R-squared, computed for the full model when the SNP was added.

Note

The rows of genefm, snpsfm, and cvrtfm filematrices must match. The SNPs must have increasing locations.

Author(s)

Andrey A Shabalin <andrey.shabalin@gmail.com>, John Palowitch

References

The manuscript is available at: http://arxiv.org/abs/1605.08799

See Also

For package overview and code examples see the package vignette via:
`browseVignettes("ACMEeqtl")`
or
`RShowDoc("doc/ACMEeqtl.html", "html", "ACMEeqtl")`

Examples

```
# First we generate a eQTL dataset in filematrix format
tempdirectory = tempdir();
z = create_artificial_data(
    nsample = 100,
    ngene = 500,
    nsnp = 5000,
    ncvrt = 1,
    minMAF = 0.2,
    saveDir = tempdirectory,
    returnData = FALSE,
    savefmat = TRUE,
    savetxt = FALSE,
    verbose = FALSE)

# Then we run multithreadACME to obtain single-SNP estimates.
# In this example, we use 2 CPU cores (threads)
# for testing of all gene-SNP pairs within 100,000 bp.
multithreadACME(
    genefm = "gene",
    snpsfm = "snps",
    glocfm = "gene_loc",
    slocfm = "snps_loc",
    cvrtfm = "cvrt",
    acmefm = "ACME",
    cisdist = 100e+03,
    threads = 2,
    workdir = paste0(tempdirectory,"/filematrices"),
    verbose = FALSE)

# Now the filematrix `ACME` holds estimations for all local gene-SNP pairs.

fm = fm.open(paste0(tempdirectory,"/filematrices/ACME"))
TenResults = fm[,1:10];
rownames(TenResults) = rownames(fm);
close(fm);
```

```
show(t(TenResults))

# Now we can estimate multi-SNP ACME models for each gene:
multisnpACME(
genefm = 'gene',
snpsfm = 'snps',
glocfm = 'gene_loc',
slocfm = 'snps_loc',
cvrtfm = 'cvrt',
acmefm = 'ACME',
workdir = paste0(tempdirectory, "/filematrices"),
genecap = Inf,
verbose = TRUE)
```

`multithreadACME` *Fast Parallelized Estimation of ACME Model for All Local Gene-SNP Pairs*

Description

This function estimates the ACME model (see the vignette for model details) for all gene-SNP pairs within pre-defined distance (`cisdist`). The input data must be stored in filematrices (see filematrix package) and the results are also saved in a filematrix. This allows the function to perform estimation using multiple CPU cores in parallel without having to duplicate the data across all jobs.

Usage

```
multithreadACME(
    genefm = "gene",
    snpsfm = "snps",
    glocfm = "gene_loc",
    slocfm = "snps_loc",
    cvrtfm = "cvrt",
    acmefm = "ACME",
    cisdist = 1e+06,
    threads = -1,
    workdir = NULL,
    verbose = TRUE)
```

Arguments

`genefm`	Name of the filematrix with gene expression data. One column per gene and one row per sample.
`snpsfm`	Name of the filematrix with SNP data. One column per SNP and one row per sample.
`glocfm`	Name of the filematix with gene location information. Must contain two columns, first with gene start location and second with the gene end. The locations must be stored as numbers, the locations for different chromosomes must differ greatly.

	We suggest encoding (location = 1e9 * chromosome + position_on_chromosome). The rows must match the columns of the genefm filematrix.
slocfm	Name of the filematrix with SNP locations. Must have one column and rows matching columns of snpsfm filematrix. See the instructions for glocfm above.
cvrtfm	Name of the filematirx with covariates. Must not include constant (it is added automatically). One column per covariate and one row per sample.
acmefm	Name of the filematrix to store the estimates. The filemarix will be created. If the filematrix exists, it will be overwritten.
cisdist	The maximum allowed distance between genes and SNPs. Gene-SNP pairs further than cisdist apart will not be tested.
threads	The number of local jobs (CPU cores) used for calculation. If negative, threads is set to the number of cores of the host machine.
workdir	Directory where the input filematrices are located.
verbose	Set to TRUE to indicate progress.

Value

The function creates a filematrix named acmefm with 10 rows and a column for each tested gene-SNP pair. The rows contain gene-SNP ids and the estimates by effectSizeEstimationC:

geneid	The gene id - the column number for the gene in the genefm filematrix.
snp_id	The SNP id - the column number for the SNP in the snpsfm filematrix.
beta0	The constant parameter in the non-linear model (within the logarithm).
beta1	The effect size parameter in the non-linear model (within the logarithm).
nits	Number of iterations till convergence of the estimation algorithm.
SSE	Sum of squared residuals of the fitted model.
SST	Sum of squared residuals of the model with zero effect.
F	The F test for the significance of the genotype effect.
eta	The effect size parameter for simplified model (beta1/beta0).
SE_eta	Standard error of the eta estimate.

Note

The rows of genefm, snpsfm, and cvrtfm filematrices must match. The SNPs must have increasing locations.

Author(s)

Andrey A Shabalin <andrey.shabalin@gmail.com>, John Palowitch

References

The manuscript is available at: http://arxiv.org/abs/1605.08799

See Also

For package overview and code examples see the package vignette via:
`browseVignettes("ACMEeqtl")`
or
`RShowDoc("doc/ACMEeqtl.html", "html", "ACMEeqtl")`

Examples

```
# First we generate a eQTL dataset in filematrix format
tempdirectory = tempdir();
z = create_artificial_data(
    nsample = 100,
    ngene = 500,
    nsnp = 5000,
    ncvrt = 1,
    minMAF = 0.2,
    saveDir = tempdirectory,
    returnData = FALSE,
    savefmat = TRUE,
    savetxt = FALSE,
    verbose = FALSE)

# In this example, we use 2 CPU cores (threads)
# for testing of all gene-SNP pairs within 100,000 bp.
multithreadACME(
    genefm = "gene",
    snpsfm = "snps",
    glocfm = "gene_loc",
    slocfm = "snps_loc",
    cvrtfm = "cvrt",
    acmefm = "ACME",
    cisdist = 100e+03,
    threads = 2,
    workdir = paste0(tempdirectory,"/filematrices"),
    verbose = FALSE)

# Now the filematrix `ACME` holds estimations for all local gene-SNP pairs.

fm = fm.open(paste0(tempdirectory,"/filematrices/ACME"))
TenResults = fm[,1:10];
rownames(TenResults) = rownames(fm);
close(fm);

show(t(TenResults))
```

Index