Package 'lfmm'

June 29, 2020

Type Package

Title Latent Factor Mixed Models

Version 1.0

Date 2020-06-22

Author Kevin Caye <kevin.caye@gmail.com>
Basile Jumentier <basile.jumentier@gmail.com>
Olivier François <olivier.francois@univ-grenoble-alpes.fr>

Maintainer Basile Jumentier <basile.jumentier@gmail.com>

Description Fast and accurate inference of
gene-environment associations (GEA) in genome-wide studies
(Caye et al., 2019, <doi:10.1093/molbev/msz008>).
We developed a least-squares estimation approach for confounder and effect sizes
estimation that provides a unique framework for several categories of genomic data,
not restricted to genotypes.
The speed of the new algorithm is several times faster than the existing GEA approaches,
then our previous version of the 'LFMM' program present in the 'LEA' package
(Frichot and Francois, 2015, <doi:10.1111/2041-210X.12382>).

License GPL-3

LazyData TRUE

Encoding UTF-8

Depends R (>= 3.2.3)

Suggests testthat

Imports foreach, rmarkdown, knitr, MASS, RSpectra, stats, ggplot2,
readr, methods, purrr, Rcpp

LinkingTo RcppEigen, Rcpp

VignetteBuilder knitr

RoxygenNote 6.1.1

URL

BugReports https://github.com/bcm-uga/lfmm/issues

NeedsCompilation yes

Repository CRAN
Date/Publication 2020-06-29 12:24:21 UTC

R topics documented:

`effect_size` *Direct effect sizes estimated from latent factor models*

Description

This function returns 'direct' effect sizes for the regression of X (of dimension 1) on the matrix Y, as usually computed in genome-wide association studies.

Usage

```
effect_size(Y, X, lfmm.object)
```

Arguments

`Y`	a response variable matrix with n rows and p columns. Each column is a response variable (numeric).
`X`	an explanatory variable with n rows and d = 1 column (numeric).
`lfmm.object`	an object of class `lfmm` returned by the lfmm_lasso or lfmm_ridge function.

Details

The response variable matrix Y and the explanatory variable are centered.

Value

a vector of length p containing all effect sizes for the regression of X on the matrix Y

Author(s)

Kevin Caye, Basile Jumentier, Olivier Francois

Examples

```
library(lfmm)

## Simulation of 1000 genotypes for 100 individuals (y)
u <- matrix(rnorm(300, sd = 1), nrow = 100, ncol = 2)
v <- matrix(rnorm(3000, sd = 2), nrow = 2, ncol = 1000)
y <- matrix(rbinom(100000, size = 2,
                   prob = 1/(1 + exp(-0.3*(u%*%v
                   + rnorm(100000, sd = 2)))),
                   nrow = 100,
                   ncol = 1000)

#PCA of genotypes, 3 main axes of variation (K = 2)
plot(prcomp(y))

## Simulation of 1000 phenotypes (x)
## Only the last 10 genotypes have significant effect sizes (b)
b <- matrix(c(rep(0, 990), rep(6000, 10)))
x <- y%*%b + rnorm(100, sd = 100)

## Compute effect sizes using lfmm_ridge
## Note that centering is important (scale = F).
mod.lfmm <- lfmm_ridge(Y = y,
                   X = x,
                   K = 2)

## Compute direct effect sizes using lfmm_ridge estimates
b.estimates <- effect_size(y, x, mod.lfmm)

## plot the last 30 effect sizes (true values are 0 and 6000)
plot(b.estimates[971:1000])
abline(0, 0)
abline(6000, 0, col = 2)

## Prediction of phenotypes
candidates <- 991:1000 #set of causal loci
x.pred <- scale(y[,candidates], scale = FALSE) %*% matrix(b.estimates[candidates])

## Check predictions
plot(x - mean(x), x.pred,
     pch = 19, col = "grey",
     xlab = "Observed phenotypes (centered)",
     ylab = "Predicted from PRS")
     abline(0,1)
     abline(lm(x.pred ~ scale(x, scale = FALSE)), col = 2)
```

`example.data` *Genetic and phenotypic data for Arabidopsis thaliana*

Description

A dataset containing SNP frequency and simulated phenotypic data for 170 plant accessions. The variables are as follows:

Usage

```
data(example.data)
```

Format

A list with 4 arguments: genotype, phenotype, causal.set, chrpos

Details

- genotype: binary (0 or 1) SNP frequency for 170 individuals (26943 SNPs).
- phenotype: simulated phenotypic data for 170 individuals.
- causal.set: set of indices for causal SNPs.
- chrpos: genetic map including chromosome position of each SNP.

Reference: Atwell et al (2010). Genome-wide association study of 107 phenotypes in Arabidopsis thaliana inbred lines. Nature 465, 627–631.

---

glm_test *GLM tests with latent factor mixed models*

Description

This function returns significance values for the association between each column of the response matrix, Y, and the explanatory variables, X, including correction for unobserved confounders (latent factors). The test is based on an LFMM fitted with a ridge or lasso penalty and a generalized linear model.

Usage

```
glm_test(Y, X, lfmm.obj, calibrate = "gif", family = binomial(link =
  "logit"))
```

Arguments

Y	a response variable matrix with n rows and p columns. Each column is a response variable (numeric).
X	an explanatory variable matrix with n rows and d columns. Each column corresponds to an explanatory variable (numeric).
lfmm.obj	an object of class lfmm returned by the lfmm_lasso or lfmm_ridge function
calibrate	a character string, "gif". If the "gif" option is set (default), significance values are calibrated by using the genomic control method. Genomic control uses a robust estimate of the variance of z-scores called "genomic inflation factor".

`family` a description of the error distribution and link function to be used in the model. For glm this can be a character string naming a family function, a family function or the result of a call to a family function.

Details

The response variable matrix Y and the explanatory variable are centered.

Value

a list with the following attributes:

- B the effect size matrix with dimensions p x d.
- score a p x d matrix which contains z-scores for each explanatory variable (columns of X),
- pvalue a p x d matrix which contains p-values for each explanatory variable,
- calibrated.pvalue a p x d matrix which contains calibrated p-values for each explanatory variable,
- gif a numeric value for the genomic inflation factor.

Author(s)

Kevin Caye, Basile Jumentier, Olivier Francois

See Also

lfmm_test

Examples

```
library(lfmm)


## An EWAS example with Y = methylation data
## and X = "exposure"
## Simulate the data

dat <- lfmm_sampler(n = 100,
                    p = 500,
                    K = 3,
                    outlier.prop = 0.01,
                    cs = 0.1,
                    sigma = 0.2,
                    B.sd = 5,
                    B.mean = 0,
                    U.sd = 1.0,
                    V.sd = 1.0)

Y <- pnorm(dat$Y)
X <- dat$X
```

```
## Fit an LFMM with 2 latent factors
mod.lfmm <- lfmm_ridge(Y = Y,
                       X = X,
                       K = 3)

## Perform association testing using the fitted model:
pv <- glm_test(Y = pnorm(Y),
               X = X,
               lfmm.obj = mod.lfmm,
               family = binomial(link = "probit"),
               calibrate = "gif")

## Manhattan plot with true associations shown
causal <- dat$outlier
pvalues <- pv$calibrated.pvalue
plot(-log10(pvalues),
     pch = 19,
     cex = .3,
     xlab = "Probe",
     col = "grey")

points(causal,
      -log10(pvalues)[causal],
       col = "blue")
```

`lfmm` *R package : Fast and Accurate statistical methods for adjusting confounding factors in association studies.*

Description

Implements statistical methods for adjusting confounding factors in association studies.

References

Caye, K., B. Jumentier, J. Lepeule, and O. François, 2019 LFMM 2: fast and accurate inference of gene-environment associations in genome-widestudies. Mol. Biol. Evol. 36: 852–860.https://doi.org/10.1093/molbev/msz00

B. Jumentier, Caye, K., J. Lepeule, and O. François, 2019 Sparse latent factor regression models for genome-wide and epigenome-wide association studies (in prep)

`lfmm_lasso` *LFMM least-squares estimates with lasso penalty (Sparse LFMM)*

Description

This function computes regularized least squares estimates for latent factor mixed models using a lasso penalty.

Usage

```
lfmm_lasso(Y, X, K, nozero.prop = 0.01, mu.num = 100,
  mu.min.ratio = 0.01, mu = NULL, it.max = 100,
  relative.err.epsilon = 1e-06)
```

Arguments

`Y`	a response variable matrix with n rows and p columns. Each column is a response variable (e.g., SNP genotype, gene expression level, beta-normalized methylation profile, etc). Response variables must be encoded as numeric.
`X`	an explanatory variable matrix with n rows and d columns. Each column corresponds to a distinct explanatory variable (eg. phenotype, exposure, outcome). Explanatory variables must be encoded as numeric.
`K`	an integer for the number of latent factors in the regression model.
`nozero.prop`	a numeric value for the expected proportion of non-zero effect sizes.
`mu.num`	a numeric value for the number of 'mu' values (advance parameter).
`mu.min.ratio`	(advance parameter) A fraction of `mu.max`, the data derived entry value (i.e. the smallest value for which all coefficients are zero).
`mu`	(advance parameter) Smallest value of `mu`. Null value by default.
`it.max`	an integer value for the number of iterations of the algorithm.
`relative.err.epsilon`	a numeric value for a relative convergence error. Determine whether the algorithm converges or not.

Details

The algorithm minimizes the following penalized least-squares criterion

The response variable matrix Y and the explanatory variable are centered.

Value

an object of class `lfmm` with the following attributes:

- U the latent variable score matrix with dimensions n x K,
- V the latent variable axes matrix with dimensions p x K,
- B the effect size matrix with dimensions p x d.

Author(s)

Kevin Caye, Basile Jumentier, Olivier Francois

References

B. Jumentier, Caye, K., J. Lepeule, and O. François, 2019 Sparse latent factor regression models for genome-wide and epigenome-wide association studies (in prep)

Examples

```
library(lfmm)

## An EWAS example with Y = methylation data
## and X = exposure

## Simulate the data

dat <- lfmm_sampler(n = 100,
                    p = 1000,
                    K = 3,
                    outlier.prop = 0.02,
                    cs = 0.1,
                    sigma = 0.2,
                    B.sd = 5,
                    B.mean = 0,
                    U.sd = 1.0,
                    V.sd = 1.0)

Y <- scale(dat$Y)
X <- scale(dat$X)

## Fit an LFMM with 2 latent factors
mod.lfmm <- lfmm_lasso(Y = Y,
                       X = X,
                       K = 3,
                       nozero.prop = 0.02)

## Manhattan plot of sparse effect sizes
effect <- mod.lfmm$B
causal <- dat$outlier

plot(effect,
     pch = 19,
     cex = .3,
     xlab = "Probe",
     col = "grey")

points(causal,
       effect[causal],
       col = "blue")
```

`lfmm_ridge` *LFMM least-squares estimates with ridge penalty*

Description

This function computes regularized least squares estimates for latent factor mixed models using a ridge penalty.

Usage

```
lfmm_ridge(Y, X, K, lambda = 1e-05, algorithm = "analytical",
  it.max = 100, relative.err.min = 1e-06)
```

Arguments

`Y`	a response variable matrix with n rows and p columns. Each column corresponds to a distinct response variable (e.g., SNP genotype, gene expression level, beta-normalized methylation profile, etc). Response variables must be encoded as numeric.
`X`	an explanatory variable matrix with n rows and d columns. Each column corresponds to a distinct explanatory variable (eg. phenotype, exposure, outcome). Explanatory variables must be encoded as numeric variables.
`K`	an integer for the number of latent factors in the regression model.
`lambda`	a numeric value for the regularization parameter.
`algorithm`	exact (analytical) algorithm or numerical algorithm. The exact algorithm is based on the global minimum of the loss function and computation is very fast. The numerical algorithm converges toward a local minimum of the loss function. The exact method should preferred. The numerical method is for very large n.
`it.max`	an integer value for the number of iterations for the numerical algorithm.
`relative.err.min`	a numeric value for a relative convergence error. Test whether the numerical algorithm converges or not (numerical algorithm only).

Details

The algorithm minimizes the following penalized least-squares criterion

$$L(U, V, B) = \frac{1}{2}||Y - UV^T - XB^T||_F^2 + \frac{\lambda}{2}||B||_2^2,$$

where Y is a response data matrix, X contains all explanatory variables, U denotes the score matrix, V is the loading matrix, B is the (direct) effect size matrix, and lambda is a regularization parameter.

The response variable matrix Y and the explanatory variable are centered.

Value

an object of class `lfmm` with the following attributes:

- U the latent variable score matrix with dimensions n x K,
- V the latent variable axis matrix with dimensions p x K,
- B the effect size matrix with dimensions p x d.

Author(s)

Kevin Caye, Basile Jumentier, Olivier Francois

References

Caye, K., B. Jumentier, J. Lepeule, and O. François, 2019 LFMM 2: fast and accurate inference of gene-environment associations in genome-widestudies. Mol. Biol. Evol. 36: 852–860.https://doi.org/10.1093/molbev/msz00

Examples

```
library(lfmm)

## a GWAS example with Y = SNPs and X = phenotype
data(example.data)
Y <- example.data$genotype[, 1:10000]
X <- example.data$phenotype

## Fit an LFMM with K = 6 factors
mod.lfmm <- lfmm_ridge(Y = Y,
                       X = X,
                       K = 6)

## Perform association testing using the fitted model:
pv <- lfmm_test(Y = Y,
                X = X,
                lfmm = mod.lfmm,
                calibrate = "gif")

## Manhattan plot with causal loci shown

pvalues <- pv$calibrated.pvalue
plot(-log10(pvalues), pch = 19,
     cex = .2, col = "grey", xlab = "SNP")
points(example.data$causal.set[1:5],
      -log10(pvalues)[example.data$causal.set[1:5]],
       type = "h", col = "blue")


## An EWAS example with Y = methylation data and X = exposure
Y <- skin.exposure$beta.value
X <- as.numeric(skin.exposure$exposure)
```

```
## Fit an LFMM with 2 latent factors
mod.lfmm <- lfmm_ridge(Y = Y,
                       X = X,
                       K = 2)

## Perform association testing using the fitted model:
pv <- lfmm_test(Y = Y,
                X = X,
                lfmm = mod.lfmm,
                calibrate = "gif")

## Manhattan plot with true associations shown
pvalues <- pv$calibrated.pvalue
plot(-log10(pvalues),
     pch = 19,
     cex = .3,
     xlab = "Probe",
     col = "grey")

causal.set <- seq(11, 1496, by = 80)
points(causal.set,
      -log10(pvalues)[causal.set],
       col = "blue")
```

`lfmm_sampler` *LFMM generative data sampler*

Description

Simulate data from the latent factor model.

Usage

```
lfmm_sampler(n, p, K, outlier.prop, cs, sigma = 0.2, B.sd = 1,
  B.mean = 0, U.sd = 1, V.sd = 1)
```

Arguments

`n`	number of observations.
`p`	number of response variables.
`K`	number of latent variables (factors).
`outlier.prop`	proportion of outlier.
`cs`	correlation between X and U.
`sigma`	standard deviation of residual errors.
`B.sd`	standard deviation for the effect size (B).
`B.mean`	mean of B.
`U.sd`	standard deviations for K factors.
`V.sd`	standard deviations for loadings.

Details

`lfmm_sampler()` sample a response matrix Y and a primary variable X such that

Y = U t(V) + X t(B) + Epsilon.

U,V, B and Epsilon are simulated according to normal multivariate distributions. Moreover U and X are such that `cor(U[,i],X) = cs[i]`.

Value

A list with simulated data.

Author(s)

kevin caye, olivier francois

Examples

```
dat <- lfmm_sampler(n = 100,
                    p = 1000,
                    K = 3,
                    outlier.prop = 0.1,
                    cs = c(0.8),
                    sigma = 0.2,
                    B.sd = 1.0,
                    B.mean = 0.0,
                    U.sd = 1.0,
                    V.sd = 1.0)
```

---

lfmm_test *Statistical tests with latent factor mixed models (linear models)*

Description

This function returns significance values for the association between each column of the response matrix, Y, and the explanatory variables, X, including correction for unobserved confounders (latent factors). The test is based on an LFMM fitted with a ridge or lasso penalty (linear model).

Usage

```
lfmm_test(Y, X, lfmm, calibrate = "gif")
```

Arguments

Y	a response variable matrix with n rows and p columns. Each column is a response variable (numeric).
X	an explanatory variable matrix with n rows and d columns. Each column corresponds to an explanatory variable (numeric).

`lfmm`	an object of class `lfmm` returned by the lfmm_lasso or lfmm_ridge function
`calibrate`	a character string, "gif" or "median+MAD". If the "gif" option is set (default), significance values are calibrated by using the genomic control method. Genomic control uses a robust estimate of the variance of z-scores called "genomic inflation factor". If the "median+MAD" option is set, the pvalues are calibrated by computing the median and MAD of the zscores. If `NULL`, the pvalues are not calibrated.

Details

The response variable matrix Y and the explanatory variable are centered.

Value

a list with the following attributes:

- B the effect size matrix with dimensions p x d.
- score a p x d matrix which contains z-scores for each explanatory variable (columns of X),
- pvalue a p x d matrix which contains p-values for each explanatory variable,
- calibrated.pvalue a p x d matrix which contains calibrated p-values for each explanatory variable,
- gif a numeric value for the genomic inflation factor.

Author(s)

Kevin Caye, Basile Jumentier, Olivier Francois

See Also

glm_test

Examples

```
library(lfmm)

## a GWAS example with Y = SNPs and X = phenotype
data(example.data)
Y <- example.data$genotype[, 1:10000]
X <- example.data$phenotype

## Fit an LFMM with K = 6 factors
mod.lfmm <- lfmm_ridge(Y = Y,
                       X = X,
                       K = 6)

## Perform association testing using the fitted model:
pv <- lfmm_test(Y = Y,
                X = X,
                lfmm = mod.lfmm,
```

```
                calibrate = "gif")

## Manhattan plot with causal loci shown

pvalues <- pv$calibrated.pvalue
plot(-log10(pvalues), pch = 19,
     cex = .2, col = "grey", xlab = "SNP")
points(example.data$causal.set[1:5],
      -log10(pvalues)[example.data$causal.set[1:5]],
       type = "h", col = "blue")


## An EWAS example with Y = methylation data and X = exposure
data("skin.exposure")
Y <- scale(skin.exposure$beta.value)
X <- scale(as.numeric(skin.exposure$exposure))

## Fit an LFMM with 2 latent factors
mod.lfmm <- lfmm_ridge(Y = Y,
                       X = X,
                       K = 2)

## Perform association testing using the fitted model:
pv <- lfmm_test(Y = Y,
                X = X,
                lfmm = mod.lfmm,
                calibrate = "gif")

## Manhattan plot with true associations shown
pvalues <- pv$calibrated.pvalue
plot(-log10(pvalues),
     pch = 19,
     cex = .3,
     xlab = "Probe",
     col = "grey")

causal.set <- seq(11, 1496, by = 80)
points(causal.set,
      -log10(pvalues)[causal.set],
       col = "blue")
```

`predict_lfmm` *Predict polygenic scores from latent factor models*

Description

This function computes polygenic risk scores from the estimates of latent factor models. It uses the indirect' effect sizes for the regression of X (a single phenotype) on the matrix Y, for predicting phenotypic values for new genotype data.

Usage

```
predict_lfmm(Y, X, lfmm.object, fdr.level = 0.1, newdata = NULL)
```

Arguments

Y	a response variable matrix with n rows and p columns, typically containing genotypes. Each column is a response variable (numeric).
X	an explanatory variable with n rows and d = 1 column (numeric) representing a phenotype with zero mean across the sample.
lfmm.object	an object of class lfmm returned by the lfmm_lasso or lfmm_ridge function, computed for X and Y.
fdr.level	a numeric value for the FDR level in the lfmm test used to define candidate variables for predicting new phenotypes.
newdata	a matrix with n rows and p' columns, and similar to Y, on which predictions of X will be based. If NULL, Y is used as new data.

Details

The response variable matrix Y and the explanatory variable are centered.

Value

a list with the following attributes:

- prediction: a vector of length n containing the predicted values for X. If newdata = NULL, the fitted values are returned.
- candidates: a vector of candidate columns of Y on which the predictions are built.

Author(s)

Kevin Caye, Basile Jumentier, Olivier Francois

Examples

```
library(lfmm)

## Simulation of 1000 genotypes for 100 individuals (y)
u <- matrix(rnorm(300, sd = 1), nrow = 100, ncol = 2)
v <- matrix(rnorm(3000, sd = 2), nrow = 2, ncol = 1000)
y <- matrix(rbinom(100000, size = 2,
                   prob = 1/(1 + exp(-0.3*(u%*%v
                   + rnorm(100000, sd = 2)))),
                   nrow = 100,
                   ncol = 1000)

#PCA of genotypes, 2 main axes of variation (K = 2)
plot(prcomp(y))

## Simulation of 1000 phenotypes (x)
```

```
## Only the last 10 genotypes have significant effect sizes (b)
b <- matrix(c(rep(0, 990), rep(6000, 10)))
x <- y%*%b + rnorm(100, sd = 100)

## Compute effect sizes using lfmm_ridge
mod <- lfmm_ridge(Y = y,
                  X = x,
                  K = 2)

x.pred <- predict_lfmm(Y = y,
                       X = x,
                       fdr.level = 0.25,
                       mod)

x.pred$candidates

##Compare simulated and predicted/fitted phenotypes
plot(x - mean(x), x.pred$pred,
     pch = 19, col = "grey",
     xlab = "Observed phenotypes (centered)",
     ylab = "Predicted from PRS")
abline(0,1)
abline(lm(x.pred$pred ~ scale(x, scale = FALSE)), col = 2)
```

`skin.exposure` *Simulated (and real) methylation levels for sun exposed patient patients*

Description

A data set containing normalized beta values, and sun exposure and simulated phenotypic data for 78 tissue samples.

Usage

```
data("skin.exposure")
```

Format

A list with 6 arguments: beta.value, phenotype, causal.set, chrpos

Details

The variables are:

- beta.value: 1496 filtered normalized beta values (methyation probabilities) for 78 tissue samples.
- exposure: Sun exposure levels for 78 tissue samples.
- phenotype: Simulated binary phenotypic data for 78 tissue samples.

- age: age of patients.
- gender: sex of patients.
- tissue: category for tissue samples.

Reference: Vandiver et al (2015). Age and sun exposure-related widespread genomic blocks of hypomethylation in nonmalignant skin. Genome Biol. 16.

Index