Package 'VIGoR'

May 20, 2015

Type Package

Title Variational Bayesian Inference for Genome-Wide Regression

Version 1.0

Date 2015-05-18

Author Akio Onogi and Hiroyoshi Iwata

Maintainer Akio Onogi <onogiakio@gmail.com>

Description Conducts linear regression using variational Bayesian inference, particularly optimized for genome-wide association mapping and whole-genome prediction which use a number of DNA markers as the explanatory variables. Provides seven regression models which select the important variables (i.e., the variables related to response variables) among the given explanatory variables in different ways (i.e., model structures).

License MIT + file LICENSE

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-05-20 15:38:58

R topics documented:

Covariates *An example of Covariate objects*

Description

A (100 x 2) matrix. The first column is the intercept and filled with 1s for all samples. The second column is a randomly generated covariate consisting of two classes, 0 and 1.

Examples

```
data(sampledata)
dim(Covariates)#100 rows and 2 columns
```

Geno *An example of marker genotype objects*

Description

An example of a marker genotype object consisting of 1000 markers and 100 samples. The genotypes are coded as 0 (AA), 1 (AB), and 2 (BB). The genotypes were randomly generated as described in the details.

Details

Geno was generated by

```
N<-100
P<-1000
Geno<-matrix(sample(c(0,1,2),N*P,replace=T,
prob=c(0.49,0.42,0.09)),nc=P)
#MAF is 0.3
```

See Also

hyperpara

Examples

```
data(sampledata)
dim(Geno)#100 samples and 1000 markers
unique(Geno[1:(100*1000)])#0,1,2
```

hyperpara *Calculation of hyperparameter values*

Description

This function determines the hyperparameter values of regression methods, based on several assumptions.

Usage

```
hyperpara(Geno, Mvar, Method = c("BL", "EBL", "wBSR", "BayesB", "BayesC", "SSVS", "MIX"),
          Kappa, A = 0.9, Xtype="Geno", f = 0, BL.Phi = 1, EBL.Phi = 0.1,
          EBL.Omega = 0.1, Psi = 1, Nu = 5, Printinfo = FALSE)
```

Arguments

`Geno`	An (N x P) matrix, where N and P denote the number of individuals and markers, respectively. Marker genotypes should be coded as 0 (AA), 1 (AB), and 2 (BB). Because missing values are not allowed, missing genotypes should be imputed before analysis. Doubles between 0 and 2 are allowed. Marker genotypes are used to calculate sum (2*Q*(1-Q)*(1+f)) where Q is a vector of allele frequencies and f is the inbreeding coefficient.
`Mvar`	A scalar denoting the assumed proportion of phenotypic variance that can be explained by the markers. Mvar is < 1.0 in BL and EBL, and <= 1.0 in the other methods.
`Method`	One of the seven regression methods
`Kappa`	A scalar or vector containing the assumed proportion of markers with non-zero effects. In MIX and SSVS, Kappa is < 1; in the other methods, 0 < Kappa <= 1.
`A`	In the SSVS and MIX models. A is a scalar or vector denoting the assumed proportion of Mvar explained by the markers assigned to the normal prior distribution with the larger variance. For example, given Mvar = 0.5, A = 0.9, and Kappa = 0.01, then 0.45 (0.5 x 0.9) of the phenotypic variance is assumed to be explained by (P x 0.01) markers, and 0.05 (0.5 x (1-0.9)) of variance is assumed to be explained by (P x (1 - 0.01)) markers.
`Xtype`	Allowed Xtypes are "Geno" and "Var". Enter "Geno" when Geno contains the marker genotypes and "Var" when Geno contains variables other than marker genotypes.
`f`	A scalar representing the inbreeding coefficient. Enter 1 for inbred species.
`BL.Phi`	A scalar or vector containing Phi values of BL.
`EBL.Phi`	A scalar or vector containing Phi values of EBL.
`EBL.Omega`	A scalar or vector containing Omega values of EBL.
`Psi`	A scalar or vector containing Psi values of EBL.
`Nu`	A scalar or vector containing Nu values of wBSR, BayesB, BayesC, SSVS, and MIX.
`Printinfo`	Specifies whether to print information (TRUE) or not (FALSE).

Details

To run vigor, users must specify the following hyperparameter values.

- BL: Phi, Omega
- EBL: Phi, Omega, Psi, Theta
- wBSR: Nu, S2, Kappa
- BayesB: Nu, S2, Kappa
- BayesC: Nu, S2, Kappa
- SSVS: c, Nu, S2, Kappa
- MIX: c, Nu, S2, Kappa

This function calculates the Omega of BL; Theta of EBL; S2 of wBSR, BayesB, and BayesC; and c and S2 of SSVS and MIX. Mvar, Kappa, and the other hyperparameters required by each method are specified by the user. The SSVS and MIX models also require A. The definitions of Mvar, Kappa, and A are intuitively understandable and relatively easy to specify (see Arguments). We recommend the default values for the other hyperparameters. When the arguments of hyperpara are vectors, all value-combinations are returned as a matrix. The hyperparameters are explained in the details of vigor and in the pdf manual of VIGoR (Onogi 2015).

Value

This function returns a vector when yielding a single hyperparameter set, and a matrix when yielding multiple hyperparameter sets. The rows and columns of the matrix correspond to the sets (value combinations) and the hyperparameters, respectively. See examples below.

References

Onogi & Iwata, VIGoR: variational Bayesian inference for genome-wide regression, in prep.
Onogi A, 2015, Documents for VIGoR (May 2015).https://github.com/Onogi/VIGoR

See Also

vigor

Examples

```
#data
data(sampledata)
dim(Geno)#100 samples and 1000 markers
unique(Geno[1:(100*1000)])#coded as 0, 1, 2

#A single Kappa value is assumed for BL. A vector is returned.
hyperpara(Geno,0.5,"BL",0.01,Printinfo=TRUE)

#Phi is set to 1 as default. To change Phi, use BL.Phi.
hyperpara(Geno,0.5,"BL",0.01,BL.Phi=5)
```

```
#Calculate multiple hyperparameter value sets of BayesC assuming that Kappa is 0.01, 0.1, and 1.
#A matrix is returned.
hyperpara(Geno,0.5,"BayesC",c(0.01,0.1,1))

#The output vector can be used as the argument of vigor
Result<-vigor(Pheno$Height,Geno,"wBSR",hyperpara(Geno,0.5,"wBSR",0.01))

#Calculate multiple hyperparameter sets of SSVS
#assuming that Mvar is 0.5, Kappa is 0.01 and 0.1, and A is 0.9 and 0.99.
hyperpara(Geno,0.5,"SSVS",c(0.01,0.1),c(0.9,0.99))
#2 x 2 sets are created.

#Calculate hyperparameter values of BayesB
#assuming that Mvar is 0.5, and Kappa is 0.01. Inbred lines are analyzed.
hyperpara(Geno,0.5,"BayesB",0.01,f=1)

#Calculate hyperparameter values of EBL
#assuming that Mvar is 0.5, and Kappa is 0.01. Consider marker genotypes as general variables.
hyperpara(Geno,0.5,"EBL",0.01,Xtype="Var")
```

Pheno *An example of Pheno objects*

Description

An object consisting of 100 samples and three traits, "Height"", "Weight"", and "Length".

Details

Five QTLs are dispersed across the genomes. The QTLs are not shared among traits. The QTL effects were drawn from a normal distribution with mean = 0 and SD = 4. The heritabilities of the traits are 0.8 (Height), 0.5 (Weight), and 0.7 (Length). The "Weight"" record of the fifth sample is missing (NA).

Examples

```
data(sampledata)
dim(Pheno)#100 samples and 3 traits
any(is.na(Pheno))#TRUE. Pheno includes missing records
which(is.na(Pheno$Weight))#5. The fifth sample lacks the Weight record.
```

vigor *Variational Bayesian inference for genome-wide regression*

Description

This function performs Bayesian genome-wide regression using variational Bayesian algorithms. The available regression methods are Bayesian lasso (BL), extended Bayesian lasso (EBL), weighted Bayesian shrinkage regression (wBSR), BayesB, BayesC, stochastic search variable selection (SSVS), and Bayesian mixture model (MIX).

Usage

```
vigor(Pheno, Geno, Method = c(”BL”, ”EBL”, "wBSR", "BayesB", "BayesC", "SSVS", ”MIX”),
      Hyperparameters, Function = ”fitting”, Nfold = 10, CVFoldTuning = 5,
      Partition=NULL, Covariates = ”Intercept”, Threshold = 2+log10(ncol(Geno)),
      Maxiterations=1000, RandomIni=FALSE, Printinfo=TRUE)
```

Arguments

`Pheno` An N-length Vector of response variables (e.g., phenotypic values), where N is the number of individuals. Missing data (coded as NA) are allowed.

`Geno` An N x P matrix of marker genotypes, where P is the number of markers. Both integers and doubles can be used. Because missing values are not allowed, missing genotypes should be imputed before analysis.

`Method` String representing the selected regression method. See details below.

`Hyperparameters`
A vector or matrix of hyperparameter values. When multiple combinations of hyperparameter values (hyperparameter sets) are used, the columns of the matrix correspond to the hyperparameters, and the rows correspond to the combinations (sets). See details below.

`Function` One of the strings "fitting", "tuning", and "cv". See details below.

`Nfold` An integer value. When n > 1, n-fold cross-validation (CV) is performed on randomly partitioned individuals. When the integer is -1, leave-one-out CV is conducted. When the integer is -9, the partitioning for the CV is defined by the argument Partition. Used when Function = "cv".

`CVFoldTuning` An integer specifying the fold number of the CV in hyperparameter tuning. Used when Function = "cv" or "tuning" and the number of hyperparameter sets (number of rows of Hyperparameters) > 1.

`Partition` A matrix defining the partitions of CV. See details and examples below. Used when Function = "cv" or "tuning" and Nfold = -9.

`Covariates` If Covariates = "Intercept", the intercept is automatically added to the regression model. An N x F matrix where F is the number of covariates also can be input as the covariates. Both integers and doubles are permitted. Missing values are not allowed. See details below.

`Threshold` Specifies the convergence threshold. Calculation terminates if the convergence metric is < 1e-Threshold. See the pdf manual of VIGoR (Onogi 2015) for the metric.

`Maxiterations` Maximum number of iterations.

`RandomIni` If TRUE, the initial values of the marker effects are randomly determined. Otherwise, they are set to 0.

`Printinfo` If TRUE, print the run information to the console.

Details

For details of vigor, the user is referred to the pdf manual (Onogi 2015).

Regression methods

Vigor assumes the following linear model for individual i;

Pheno[i] = sum(Covariates[i,]*Alpha) + sum(Gamma*Geno[i,]*Beta) + Ei

where Pheno[i] is the response variable, Alpha contains the covariate coefficients, and Gamma contains the variables that indicate whether the corresponding markers are included in the model (1) or not (0). Beta contains the marker effects, and Ei is the residual. Gamma is set to 1 except in wBSR, which infers Gamma. Ei is assumed to follow a Normal (0, 1/Tau02) distribution, where Tau02 is assumed to follow 1/Tau02.
The methods assume different prior distributions of the marker effect p (Beta[p]).

- BL
  Beta[p] ~ normal (0, sqrt(1/Tau2[p]/Tau02))
  Tau2[p] ~ inverse gamma (1, Lambda2/2)
  Lambda2 ~ gamma (Phi, Omega)
- EBL
  Beta[p] ~ normal (0, sqrt(1/Tau2[p]/Tau02))
  Tau2[p] ~ inverse gamma (1, Delta2*Eta2[p]/2)
  Delta2 ~ gamma (Phi, Omega)
  Eta2[p] ~ gamma (Psi, Theta)
- wBSR
  Beta[p] ~ normal (0, sqrt(Sigma2[p]))
  Sigma2[p] ~ scaled-inverse-chi square (Nu, S2)
  Gamma[p] ~ Bernoulli (Kappa)
- BayesB
  Beta[p] ~ normal (0, sqrt(Sigma2[p])) if Rho[p]=1, and 0 if Rho[p]=0
  Sigma2[p] ~ scaled-inverse-chi square (Nu, S2)
  Rho[p] ~ Bernoulli (Kappa)
- BayesC
  Beta[p] ~ normal (0, sqrt(Sigma2)) if Rho[p]=1, and 0 if Rho[p]=0
  Sigma2 ~ scaled-inverse-chi square (Nu, S2)
  Rho[p] ~ Bernoulli (Kappa)

- SSVS
  Beta[p] ~ normal (0, sqrt(Sigma2)) if Rho[p]=1, and normal(0, sqrt(c*Sigma2)) if Rho[p]=0
  Sigma2 ~ scaled-inverse-chi square (Nu, S2)
  Rho[p] ~ Bernoulli (Kappa)
- MIX
  Beta[p] ~ normal (0, sqrt(Sigma2[1])) if Rho[p]=1, and normal(0, sqrt(Sigma2[2])) if Rho[p]=0
  Sigma2[1] ~ scaled-inverse-chi square (Nu, S2)
  Sigma2[2] ~ scaled-inverse-chi square (Nu, c*S2)
  Rho[p] ~ Bernoulli (Kappa)

Hyperparameters
To run vigor, the following hyperparameter values must be declared as arguments for Hyperparameters.

- BL: Phi, Omega
- EBL: Phi, Omega, Psi, Theta
- wBSR: Nu, S2, Kappa
- BayesB: Nu, S2, Kappa
- BayesC: Nu, S2, Kappa
- SSVS: c, Nu, S2, Kappa
- MIX: c, Nu, S2, Kappa

The argument Hyperparameters is a Nh-length vector, or a (Nh x Nset) matrix, where Nh and Nset are the number of hyperparameters and the number of hyperparameter sets, respectively. For example, when the regression method is BL, the matrix

1 0.001
1 0.01
1 0.1

indicates that Phi = 1 and Omega = 0.001 in the first set, Phi = 1 and Omega = 0.01 in the second set, and Phi = 1 and Omega = 0.1 in the third set. As another example, when the regression model is BayesB, the vector

4 0.5 0.001

indicates that Nu = 4, S2 = 0.5, and Kappa = 0.001. Vectors or matrices for Hyperparameters can be created using hyperpara.

Functions
The functions of vigor are "fitting", "tuning", and "cv". "Fitting" fits the selected regression model to the data. When Hyperparameters includes multiple hyperparameter sets (i.e., is input as a matrix), only the first set is used.
"Tuning" selects the hyperparameter set with the lowest MSE in CV. This set is then used in the model fitting. When tuning the hyperparameters, CV is performed on randomly partitioned data. The number of folds is determined by CVFoldTuning.

The "cv" function conducts CV, and returned the predicted values. When Hyperparameters includes multiple hyperparameter sets, tuning is performed at each fold of the CV.

Partition matrix
The following is a possible Partition of 20 individuals evaluated in a five-fold CV:

14 11 3 2 7
5 4 20 10 9
6 8 16 15 12
18 13 17 1 19

Individuals (row numbers in Pheno/Geno/Covariates) 14, 5, 6, and 18 are removed from the training set at the first fold of the five-fold CV. Samples 11, 4, 8, and 13 are removed at the next fold. This process is repeated up to the fold number of the CV. If the number of individuals N is 19, the gap is filled with -9. For example,

8 6 3 14 18
12 4 1 15 5
17 9 13 11 10
19 16 7 2 -9

An example of random sampling validation in which individuals can be sampled more than once is shown below.

18 3 11 16 13
17 8 13 13 18
7 15 14 19 7
1 13 12 7 2

Individuals 18, 13, and 7 are repeatedly used as testing samples.

Random partitioning (i.e., Nfold = n) outputs a Partition matrix, which can be input as the Partition matrix in subsequent analysis.

Intercept
If Covariates = "Intercept", vigor automatically adds the intercept to the regression model. If Covariates is a user-specified matrix, vigor uses this matrix as the covariates, regarding the first column as the intercept. Thus, the first column of Covariates should be filled with 1s. However, when Covariates is a matrix of population-assignment probabilities when correcting for population stratification, the intercept is not necessary.

Standardization
Vigor standardizes Pheno (response variables). Although the returned values are generally scaled back to the original scale, some estimates are reported in the standardized scale. See value.

Value

When Function = "fitting" or "tuning", a list containing the following elements is returned.

`$LB` Lower bound of the marginal log likelihood of Pheno.

`$ResidualVar` Residual variances (1/Tau02) of the iterations. Reported in the standardized scale.

`$Beta` Posterior means of the marker effects (E[Beta|Pheno]).
The wBSR model returns E[Beta|Pheno]*E[Gamma|Pheno].

`$Sd.beta` Posterior uncertainty (standard deviation) of the marker effects (sqrt(Var[Beta|Pheno])).
For wBSR,
sqrt(E[Beta^2|Pheno]*Var[Gamma|Pheno]+Var[Beta|Pheno]*E[Gamma|Pheno]^2)
is returned.

`$Tau2` Posterior mean of Tau2.

`$Sigma2` Posterior mean of Sigma2.

`$Alpha` Posterior means of Alpha.

`$Sd.alpha` Posterior uncertainty of Alpha.

`$Lambda2` Posterior mean of Lambda2. Reported in the standardized scale. Returned only by the BL model.

`$Delta2` Posterior mean of Delta2. Reported in the standardized scale. Returned only by the EBL model.

`$Eta2` Posterior means of Eta2. Reported in the standardized scale. Returned only by the EBL model.

`$Gamma` Posterior means of Gamma. Returned only by the wBSR model.

`$Rho` Posterior means of Rho. Returned only by the BayesB, BayesC, SSVS, and MIX models.

`$MSE` A data frame with (2 + Nh) columns, where Nh is the number of hyperparameters. Returned when Function = "tuning".

- Set: Hyperparameter set number. This number corresponds to the row number of Hyperparameters.
- (Hyperparameters): Hyperparameter values of the set.
- MSE: The MSE of the hyperparameter set.

When Function = "cv", a list containing the following elements is returned.

`$Prediction` A data frame with 4 columns, Test, Y, Yhat, and BV:

- Test: Tested individuals (row numbers in Pheno/Geno/Covariates).
- Y: Phenotypic values of the tested individuals (true values).
- Yhat: Predicted values. Summation of the marker and covariate effects.
- BV: Breeding values. Summation of marker effects.

`$MSE` A data frame with (3 + Nh) columns, returned when analyzing multiple hyperparameter sets.

- Fold: Fold number of CV
- ChosenSet: Chosen hyperparameter set at the fold. Set numbers correspond to the row numbers of Hyperparameters.
- (Hyperparameters): Hyperparameter values of the chosen set.
- MSE: The MSE of the chosen set.

`$Partition` A matrix representing the partition used in random partitioning. This matrix can be used as the argument Partition in subsequent analyses.

Author(s)

Akio Onogi
Hiroyoshi Iwata

References

Onogi & Iwata, VIGoR: variational Bayesian inference for genome-wide regression, in prep.
Onogi A, 2015, Documents for VIGoR (May 2015).https://github.com/Onogi/VIGoR

Examples

```
#data
data(sampledata)
dim(Geno)#100 samples and 1000 markers
dim(Pheno)#100 samples and 3 traits
dim(Covariates)#100 samples and 2 covariates (including the intercept)

#Use BL. Draw a simple Manhattan plot.
Result<-vigor(Pheno$Height,Geno,"BL",c(1,1),Covariates=Covariates)
plot(abs(Result$Beta),pch=20)
which((abs(Result$Beta)-1.96*Result$Sd.beta)>0) #Significant markers (P<0.05)

#Use BayesC without covariates.
Result<-vigor(Pheno$Height,Geno,"BayesC",matrix(c(4,1,0.01,4,1,0.001),nr=2,byrow=TRUE))
plot(abs(Result$Beta),pch=20)
which((abs(Result$Beta)-1.96*Result$Sd.beta)>0)
print(Result$Alpha)#intercept is automatically added.

#Tuning hyperparameters. Two hyperparameter sets are given as a matrix. Use BayesB.
H<-matrix(c(5,1,0.001,5,1,0.01),nc=3,byrow=TRUE)
print(H)
Result<-vigor(Pheno$Height,Geno,"BayesB",H,Function="tuning",Covariates=Covariates)
plot(abs(Result$Beta),pch=20)
print(Result$MSE)#the set with the lowest MSE was used
#When Function of vigor is "fitting", only the first set is used for regression.
#to repeat analyses under the different sets, for example,
Result<-as.list(numeric(2))
for(set in 1:2){Result[[set]]<-vigor(Pheno$Height,Geno,"wBSR",H[set,],Covariates=Covariates)}

#Perform cross-validation. Use BL. Number of hyperparameter sets is 2.
#the first set is c(1,0.01) and the second is c(1,0.1)
#6-fold CV
Result<-vigor(Pheno$Height,Geno,"BL",
matrix(c(1,0.01,1,0.1),ncol=2,byrow=TRUE),Function="cv",Nfold=6,Covariates=Covariates)
plot(Result$Prediction$Y,Result$Prediction$Yhat) #plot true and predicted values
cor(Result$Prediction$Y,Result$Prediction$Yhat) #accuracy
print(Result$MSE) #see which the set used at each fold.
print(Result$Partition) #see the partition of CV
#Perform CV using the same partition. Use BayesC.
H<-matrix(c(5,1,0.01,5,1,0.1),nc=3,byrow=TRUE)
Result2<-vigor(Pheno$Height,Geno,"BayesC",H,Function="cv",Nfold=-9,
Partition=Result$Partition,Covariates=Covariates)
```

```
cor(Result2$Prediction$Y,Result2$Prediction$Yhat) #accuracy
```

Index