Package 'stmgp'

July 18, 2021

Type Package

Title Rapid and Accurate Genetic Prediction Modeling for Genome-Wide Association or Whole-Genome Sequencing Study Data

Version 1.0.4

Date 2021-07-18

Author Masao Ueki

Maintainer Masao Ueki <uekimrsd@nifty.com>

Description Rapidly build accurate genetic prediction models for genome-wide association or whole-genome sequencing study data by smooth-threshold multivariate genetic prediction (STMGP) method. Variable selection is performed using marginal association test p-values with an optimal p-value cutoff selected by Cp-type criterion. Quantitative and binary traits are modeled respectively via linear and logistic regression models. A function that works through PLINK software (Purcell et al. 2007 <DOI:10.1086/519795>, Chang et al. 2015 <DOI:10.1186/s13742-015-0047-8>) <https://www.cog-genomics.org/plink2> is provided. Covariates can be included in regression model.

SystemRequirements PLINK must be installed

License GPL (>= 2)

Depends MASS

NeedsCompilation no

Repository CRAN

Date/Publication 2021-07-18 08:20:01 UTC

R topics documented:

stmgp-package *Rapid and Accurate Genetic Prediction Modeling for Genome-Wide Association or Whole-Genome Sequencing Study Data*

Description

Rapidly build accurate genetic prediction models for genome-wide association or whole-genome sequencing study data by smooth-threshold multivariate genetic prediction (STMGP) method. Variable selection is performed using marginal association test p-values with an optimal p-value cutoff selected by Cp-type criterion. Quantitative and binary traits are modeled respectively via linear and logistic regression models. A function that works through PLINK software (Purcell et al. 2007 <DOI:10.1086/519795>, Chang et al. 2015 <DOI:10.1186/s13742-015-0047-8>) <https://www.cog-genomics.org/plink2> is provided. Covariates can be included in regression model.

Details

The DESCRIPTION file:

Index of help topics:

stmgeplink	Smooth-threshold multivariate genetic prediction (incorporating gene-environment interactions) for genome-wide association or whole-genome sequencing data in PLINK format
stmgp	Smooth-threshold multivariate genetic prediction
stmgp-package	Rapid and Accurate Genetic Prediction Modeling for Genome-Wide Association or Whole-Genome Sequencing Study Data
stmgplink	Smooth-threshold multivariate genetic prediction for genome-wide association or whole-genome sequencing data in PLINK format

Author(s)

Maintainer: Masao Ueki <uekimrsd@nifty.com>

References

Ueki M, Tamiya G, and for Alzheimer's Disease Neuroimaging Initiative. (2016) Smooth-thresholdmultivariate genetic prediction with unbiased model selection. Genet Epidemiol 40:233-43. <https://doi.org/10.1002/gepi.21958>

Ueki M. (2009) A note on automatic variable selection using smooth-threshold estimating equations. Biometrika 96:1005-11. <https://doi.org/10.1093/biomet/asp060>

stmgeplink *Smooth-threshold multivariate genetic prediction (incorporating gene-environment interactions) for genome-wide association or whole-genome sequencing data in PLINK format*

Description

Build prediction model from training data and predict test data phenotype through smooth-threshold multivariate genetic prediction (STMGP) method incorporating gene-environment (GxE) interactions, in which GxE interaction effects are linearly added to the STMGP model with marginal effects. Data must be in PLINK binary format and marginal test p-values (i.e. test for each variant) are computed by PLINK software, which enables rapid computation even for data having very large number of variants. An optimal p-value cutoff is selected by Cp-type criterion. Both quantitative and binary phenotypes are acceptable, in which data must be in PLINK fam file format or in a separate file (PLINK format, i.e. FID and IID are needed). Environment variables need be in covariate file by specifying the column names.

Usage

```
stmgeplink(trainbed, Z ,Enames ,Zte = NULL, testbed = NULL, gamma = 1, taun = NULL,
   lambda = 1, plink = "plink --noweb", maf = 0.01, hwe = 1e-4, geno = 0.1,
   fout = "stp", trainfout = "train", testfout = "test", ll = 50, maxal = NULL,
  alc = NULL, tdir = NULL, Znames = NULL, trainphenofile = NULL, testphenofile = NULL,
   phenoname = NULL, Assc = FALSE, AsscGE = FALSE, centerE = TRUE)
```

Arguments

trainbed	A training data file name in PLINK binary format or a vector of three file names of .bed, .bim, .fam; Binary phenotype must be coded as 1 or 2 in PLINK fam file. Missing values in .fam file are -9 as usual.
Z	A covariate file name for training data including environment variables (PLINK format, i.e. FID and IID are needed) or data matrix, missing values are "-9".
Enames	Vector of (column) names of environment variables in the covariate file specified in Z.
Zte	A covariate file name for test data including environment variables (PLINK format, i.e. FID and IID are needed) or data matrix, missing values are "-9"; NULL (default) means unspecified.
testbed	A test data file name in PLINK binary format or a vector of three file names of .bed, .bim, .fam; NULL (default) means unspecified. Missing values in .fam are -9 as usual.
gamma	gamma parameter; gamma=1 is default as suggested in Ueki and Tamiya (2016).
taun	tau parameter divided by (sample size) (allowed to be a vector object; optimal parameter is chosen by Cp); NULL (default) specifies tau=n/log(n)^0.5 as suggested in Ueki and Tamiya (2016).
lambda	lambda parameter (default=1).

plink	PLINK command, e.g. "plink2", "./plink –noweb", or "plink1.9 –memory 100000" (default is plink --noweb) where options can be added; PLINK must be installed.
maf	Minor allele frequency (MAF) cutoff for --maf option in PLINK.
hwe	Hardy-Weinberg equilibrium (HWE) cutoff for --hwe option in PLINK.
geno	Missing call rate cutoff for --geno option in PLINK (default=0.1).
fout	An output file name (default="stp").
trainfout	An output file name for training data (default="train").
testfout	An output file name for test data (default="test").
ll	Number of candidate p-value cutoffs for search (default=50) as determined by 10^seq(log10(maxal),log10(5e-8),length=ll).
maxal	Maximum p-value cutoff for search (default=NULL); If most variants are null maxal*(number of variables to be selected) gives approximate number of filtered variants, which is useful for rapid computation even for data with large number of variants.
alc	User-specified candidate p-value cutoffs for search; ll option is effective if alc=NULL.
tdir	Location of temporary files (default=tempdir()).
Znames	Name(s) of covariate used; NULL (default) means unspecified.
trainphenofile	A phenotype file name for training data (PLINK format, i.e. FID and IID are needed) with header columns (i.e. FID, IID, phenoname1, phenoname2, ...) missing values are "-9"; NULL (default) means unspecified.
testphenofile	A phenotype file name for test data (PLINK format, i.e. FID and IID are needed) with header columns (i.e. FID, IID, phenoname1, phenoname2, ...) missing values are "-9"; NULL (default) means unspecified.
phenoname	Phenotype name in trainphenofile; NULL (default) means unspecified but users should provide if trainphenofile is provided.
Assc	Whether the marginal association result is stored or not (default=FALSE).
AsscGE	Whether the gene-environment interaction result is stored or not (default=FALSE).
centerE	Whether environment variables are centered or not by subtracting their mean (default=TRUE).

Details

See Ueki and Tamiya (2016).

Value

Muhat	Estimated phenotypes from linear model evaluated at each candidate tuning parameters (al and tau) whose size is of (sample size) x (length of al) x (length of tau).
gdf	Generalized degrees of freedom (GDF, Ye 1998) whose size is of (length of al) x (length of tau).

sig2hat	Error variance estimates (=1 for binary traits) whose size is of (length of al) x (length of tau).
df	Number of nonzero regression coefficients whose size is of (length of al) x (length of tau).
al	Candidate p-value cutoffs for search.
lopt	An optimal tuning parameter indexes for al and tau selected by Cp-type criterion, CP
BA	Estimated regression coefficient matrix whose size is of (1 + number of columns of Z + number of columns of X) x (length of al)) x (length of tau)); the first element, the second block and third block correspond to intercept, Z and X, respectively.
Loss	Loss (sum of squared residuals or -2*loglikelihood) whose size is of (length of al) x (length of tau).
sig2hato	An error variance estimate (=1 for binary traits) used in computing the variance term of Cp-type criterion.
tau	Candidate tau parameters for search.
CP	Cp-type criterion whose size is of (length of al) x (length of tau).
PE	PLINK .fam file for training data with additional column including the predicted phenotype from linear model.
PEte	PLINK .fam file for test data with additional column including the predicted phenotype from linear model estimated from training data.
nonzero	Variants with nonzero regression coefficients at the optimal parameter in PLINK file.
DataTr	Training dataset used (y, X and Z)
lapprox	lapprox values for gene-environment interaction tests (discrepancy from 1 suggests model misspecification)
ASSC	Marginal association result from PLINK if Assc is TRUE
ASSCGE	Gene-environment interaction result (approx test) from PLINK if AsscGE is TRUE
ASSCGEa	Gene-environment interaction result (all tests) from PLINK if AsscGE is TRUE

References

Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira M, Bender D, Maller J, Sklar P, de Bakker P, Daly MJ, Sham PC. (2007) PLINK: A tool set for whole-genome and population-based linkage analyses. Am J Hum Genet 81:559-75.

Chang CC, Chow CC, Tellier LCAM, Vattikuti S, Purcell SM, Lee JJ. (2015) Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience 4.

Ueki M, Fujii M, Tamiya G. (2019) Quick assessment for systematic test statistic inflation/deflation due to null model misspecifications in genome-wide environment interaction studies. PLoS ONE 14: e0219825.

Examples

```
## Not run:
wd = system.file("extdata",package="stmgp")


# quantitative traits
# training data (plink format)
trainbed = paste(wd,"train",sep="/")
# test data (plink format)
testbed = paste(wd,"test",sep="/")
# number of SNPs
#n.snp = length(readLines(paste(trainbed,".bim",sep="")))
n.snp = 80000


#> head(read.table(paste0(trainbed,".fam")))
# training sample .fam file (quantitative phenotype in the 6th column)
#       V1      V2 V3 V4 V5    V6
#1 id1_100 id2_100  0  0  1 -1.23
#2 id1_101 id2_101  0  0  1  1.48
#3 id1_102 id2_102  0  0  1 -4.27
#4 id1_103 id2_103  0  0  1 -2.61
#5 id1_104 id2_104  0  0  1 -0.27
#6 id1_105 id2_105  0  0  1 -0.50


#> head(read.table(paste0(testbed,".fam")))
# test sample .fam file
# (quantitative phenotype in the 6th column but missing (i.e. "-9") allowed)
#     V1    V2 V3 V4 V5    V6
#1 id1_0 id2_0  0  0  1 -0.59
#2 id1_1 id2_1  0  0  1  1.11
#3 id1_2 id2_2  0  0  1 -2.45
#4 id1_3 id2_3  0  0  1  0.11
#5 id1_4 id2_4  0  0  1 -1.17
#6 id1_5 id2_5  0  0  1  2.08


# using covariates files

Zf = paste(wd,"train.cov",sep="/")
Ztef = paste(wd,"test.cov",sep="/")


#> head(read.table(Zf,header=TRUE))   # covariate for training sample
#      FID     IID        COV1 COV2 qphen bphen
#1 id1_340 id2_340  1.27203944    1 -2.47     2
#2 id1_430 id2_430 -0.44144482    1 -0.71     2
#3 id1_199 id2_199 -0.18200011    1 -3.42     2
#4 id1_473 id2_473  0.03965880    0  0.32     1
#5 id1_105 id2_105  0.20418279    0 -0.50     2
```

```
#6 id1_188 id2_188 -0.04838519    0  2.98     1


#> head(read.table(Ztef,header=TRUE))   # covariate for test sample
#     FID    IID       COV1 COV2
#1 id1_80 id2_80 -0.2057512    0
#2 id1_53 id2_53 -0.8627601    1
#3 id1_59 id2_59 -0.2973529    1
#4 id1_71 id2_71  1.4728727    1
#5 id1_92 id2_92  3.5614472    0
#6 id1_25 id2_25  0.5135032    1


# model building from training data
# (incorporating COV1xG interaction as well as two covariates, COV1 and COV2)
sge1p0 = stmgeplink(trainbed=trainbed,Z=Zf,Enames="COV1",
                maxal=5000/n.snp,Znames=c("COV1","COV2"))
head(sge1p0$PE)

# model building from training data and predicting test data
# (incorporating COV1xG interaction as well as two covariates, COV1 and COV2)
sge1p = stmgeplink(trainbed=trainbed,testbed=testbed,Z=Zf,Zte=Ztef,Enames="COV1",
                maxal=5000/n.snp,Znames=c("COV1","COV2"))
head(sge1p$PEte)
head(sge1p$nonzero)


# model building from training data
# (incorporating COV1xG and COV2xG interactions as well as two covariates, COV1 and COV2)
sge12p0 = stmgeplink(trainbed=trainbed,Z=Zf,Enames=c("COV1","COV2"),
                maxal=5000/n.snp,Znames=c("COV1","COV2"))
head(sge12p0$PE)

# model building from training data and predicting test data
# (incorporating COV1xG and COV2xG interactions as well as two covariates, COV1 and COV2)
sge12p = stmgeplink(trainbed=trainbed,testbed=testbed,Z=Zf,Zte=Ztef,Enames=c("COV1","COV2"),
                maxal=5000/n.snp,Znames=c("COV1","COV2"))
head(sge12p$PEte)
head(sge12p$nonzero)




#### binary traits ####
# training data (plink format)
trainbed = paste(wd,"train",sep="/")
# test data (plink format)
testbed = paste(wd,"test",sep="/")
# number of SNPs
#n.snp = length(readLines(paste(trainbed,".bim",sep="")))
n.snp = 80000
```

```
#> head(read.table(paste0(trainbed,"b.fam")))
# training sample .fam file (binary phenotype (1 or 2) in the 6th column)
#       V1      V2 V3 V4 V5 V6
#1 id1_100 id2_100  0  0  1  2
#2 id1_101 id2_101  0  0  1  1
#3 id1_102 id2_102  0  0  1  2
#4 id1_103 id2_103  0  0  1  2
#5 id1_104 id2_104  0  0  1  2
#6 id1_105 id2_105  0  0  1  2


#> head(read.table(paste0(testbed,"b.fam")))
# test sample .fam file (binary phenotype (1 or 2) in the 6th column
# but missing (i.e. "-9") allowed)
#     V1    V2 V3 V4 V5 V6
#1 id1_0 id2_0  0  0  1  2
#2 id1_1 id2_1  0  0  1  1
#3 id1_2 id2_2  0  0  1  2
#4 id1_3 id2_3  0  0  1  1
#5 id1_4 id2_4  0  0  1  2
#6 id1_5 id2_5  0  0  1  1


# using covariates files

# model building from training data
# (incorporating COV1xG interaction as well as two covariates, COV1 and COV2)
sge1p0b = stmgeplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
       Z=paste(wd,"train.cov",sep="/"),Enames="COV1",
       maxal=5000/n.snp,Znames=c("COV1","COV2"))
head(sge1p0b$PE)

# model building from training data and predicting test data
# (incorporating COV1xG interaction as well as two covariates, COV1 and COV2)
sge1pb = stmgeplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
       testbed=paste0(testbed,c(".bed",".bim","b.fam")),
       Z=paste(wd,"train.cov",sep="/"),Zte=paste(wd,"test.cov",sep="/"),Enames="COV1",
       maxal=5000/n.snp,Znames=c("COV1","COV2"))
head(sge1pb$PEte)
head(sge1pb$nonzero)


# model building from training data
# (incorporating COV1xG and COV2xG interactions as well as two covariates, COV1 and COV2)
sge12p0b = stmgeplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
       Z=paste(wd,"train.cov",sep="/"),Enames=c("COV1","COV2"),
       maxal=5000/n.snp,Znames=c("COV1","COV2"))
head(sge12p0b$PE)

# model building from training data and predicting test data
# (incorporating COV1xG and COV2xG interactions as well as two covariates, COV1 and COV2)
sge12pb = stmgeplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
```

```
        testbed=paste0(testbed,c(".bed",".bim","b.fam")),
        Z=paste(wd,"train.cov",sep="/"),Zte=paste(wd,"test.cov",sep="/"),
        Enames=c("COV1","COV2"),maxal=5000/n.snp,Znames=c("COV1","COV2"))
head(sge12pb$PEte)
head(sge12pb$nonzero)




## End(Not run)
```

---

stmgp *Smooth-threshold multivariate genetic prediction*

Description

Smooth-threshold multivariate genetic prediction (STMGP) method, which is based on the smooth-threshold estimating equations (Ueki 2009). Variable selection is performed based on marginal association test p- values (i.e. test of nonzero slope parameter in univariate regression for each predictor variable) with an optimal p-value cutoff selected by a Cp-type criterion. Quantitative and binary phenotypes are modeled via linear and logistic regression, respectively.

Usage

```
stmgp(y, X, Z = NULL, tau, qb, maxal, gamma = 1, ll = 50,
  lambda = 1, alc = NULL, pSum = NULL)
```

Arguments

y	A response variable, either quantitative or binary (coded 0 or 1); Response type is specified by qb.
X	Predictor variables subjected to variable selection.
Z	Covariates; Z=NULL means unspecified.
tau	tau parameter (allowed to be a vector object); NULL (default) specifies tau=n/log(n)^0.5 as suggested in Ueki and Tamiya (2016).
qb	Type of response variable, qb="q" and "b" specify quantitative and binary traits, respectively.
maxal	Maximum p-value cutoff for search.
gamma	gamma parameter; gamma=1 is default as suggested in Ueki and Tamiya (2016).
ll	Number of candidate p-value cutoffs for search (default=50) as determined by 10^seq(log10(maxal),log10(5e-8),length=ll).
lambda	lambda parameter (default=1).

alc	User-specified candidate p-value cutoffs for search; ll option is effective if alc=NULL.
pSum	User-specified p-values matrix from other studies that are independent of the study data (optional, default=NULL), a matrix object having rows with the same size of X and columns for each study (multiple studies are capable). Missing p-values must be coded as NA. Summary p-values are combined with p-values in the study data by the Fisher's method.

Details

See Ueki and Tamiya (2016).

Value

Muhat	Estimated phenotypic values from linear model evaluated at each candidate tuning parameters (al and tau) whose size is of (sample size) x (length of al) x (length of tau).
gdf	Generalized degrees of freedom (GDF, Ye 1998) whose size is of (length of al) x (length of tau).
sig2hat	Error variance estimates (=1 for binary traits) whose size is of (length of al) x (length of tau).
df	Number of nonzero regression coefficients whose size is of (length of al) x (length of tau).
al	Candidate p-value cutoffs for search.
lopt	An optimal tuning parameter indexes for al and tau selected by Cp-type criterion, CP
BA	Estimated regression coefficient matrix whose size is of (1 + number of columns of Z + number of columns of X) x (length of al)) x (length of tau)); the first element, the second block and third block correspond to intercept, Z and X, respectively.
Loss	Loss (sum of squared residuals or -2*loglikelihood) whose size is of (length of al) x (length of tau).
sig2hato	An error variance estimate (=1 for binary traits) used in computing the variance term of Cp-type criterion.
tau	Candidate tau parameters for search.
CP	Cp-type criterion whose size is of (length of al) x (length of tau).

References

Ye J. (1988) On measuring and correcting the effects of data mining and model selection. J Am Stat Assoc 93:120-31.

Ueki M. (2009) A note on automatic variable selection using smooth-threshold estimating equations. Biometrika 96:1005-11.

Examples

```
## Not run:


set.seed(22200)

wd = system.file("extdata",package="stmgp")

D = read.table(unzip(paste(wd,"snps.raw.zip",sep="/"),exdir=tempdir()),header=TRUE)

X = D[,-(1:6)]
X = (X==1) + 2*(X==2)
p = ncol(X)
n = nrow(X)
ll = 30
p0 = 50; b0 = log(rep(1.2,p0))
iA0 = sample(1:p,p0)
Z = as.matrix(cbind(rnorm(n),runif(n)))  # covariates
eta = crossprod(t(X[,iA0]),b0) - 4 + crossprod(t(Z),c(0.5,0.5))


# quantitative trait
mu = eta
sig = 1.4
y = mu + rnorm(n)*sig
STq = stmgp(y,X,Z,tau=n*c(1),qb="q",maxal=0.1,gamma=1,ll=ll)
boptq = STq$BA[,STq$lopt[1],STq$lopt[2]]  # regression coefficient in selected model
nonzeroXq = which( boptq[(1+ncol(Z))+(1:p)]!=0 )  # nonzero regression coefficient
# check consistency
cor( STq$Muhat[,STq$lopt[1],STq$lopt[2]], crossprod(t(cbind(1,Z,X)),boptq) )
cor( STq$Muhat[,STq$lopt[1],STq$lopt[2]], eta)  # correlation with true function
# proportion of correctly identified true nonzero regression coefficients
length(intersect(which(boptq[-(1:(ncol(Z)+1))]!=0),iA0))/length(iA0)


# binary trait
mu = 1/(1+exp(-eta))
Y = rbinom(n,size=1,prob=mu)
STb = stmgp(Y,X,Z,tau=n*c(1),qb="b",maxal=0.1,gamma=1,ll=ll)
boptb = STb$BA[,STb$lopt[1],STb$lopt[2]]  # regression coefficient in selected model
nonzeroXb = which( boptb[(1+ncol(Z))+(1:p)]!=0 )  # nonzero regression coefficient
# check consistency
cor( STb$Muhat[,STb$lopt[1],STb$lopt[2]], crossprod(t(cbind(1,Z,X)),boptb) )
Prob = 1/(1+exp(-STb$Muhat[,STb$lopt[1],STb$lopt[2]]))  # Pr(Y=1) (logistic regression)
cor( STb$Muhat[,STb$lopt[1],STb$lopt[2]], eta)  # correlation with true function
# proportion of correctly identified true nonzero regression coefficients
length(intersect(which(boptb[-(1:(ncol(Z)+1))]!=0),iA0))/length(iA0)



# simulated summary p-values
```

```
pSum = cbind(runif(ncol(X)),runif(ncol(X)));
pSum[iA0,1] = pchisq(rnorm(length(iA0),5,1)^2,df=1,low=F); # study 1 summary p-values
pSum[iA0,2] = pchisq(rnorm(length(iA0),6,1)^2,df=1,low=F); # study 2 summary p-values
pSum[sample(1:length(pSum),20)] = NA
head(pSum)


# quantitative trait using summary p-values
STqs = stmgp(y,X,Z,tau=n*c(1),qb="q",maxal=0.1,gamma=1,ll=ll,pSum=pSum)
boptqs = STqs$BA[,STqs$lopt[1],STqs$lopt[2]]  # regression coefficient in selected model
nonzeroXqs = which( boptqs[(1+ncol(Z))+(1:p)]!=0 )  # nonzero regression coefficient
# check consistency
cor( STqs$Muhat[,STqs$lopt[1],STqs$lopt[2]], crossprod(t(cbind(1,Z,X)),boptqs) )
cor( STqs$Muhat[,STqs$lopt[1],STqs$lopt[2]], eta)  # correlation with true function
# proportion of correctly identified true nonzero regression coefficients
length(intersect(which(boptqs[-(1:(ncol(Z)+1))]!=0),iA0))/length(iA0)



# binary trait using summary p-values
STbs = stmgp(Y,X,Z,tau=n*c(1),qb="b",maxal=0.1,gamma=1,ll=ll,pSum=pSum)
boptbs = STbs$BA[,STbs$lopt[1],STbs$lopt[2]]  # regression coefficient in selected model
nonzeroXbs = which( boptbs[(1+ncol(Z))+(1:p)]!=0 )  # nonzero regression coefficient
# check consistency
cor( STbs$Muhat[,STbs$lopt[1],STbs$lopt[2]], crossprod(t(cbind(1,Z,X)),boptbs) )
Prob = 1/(1+exp(-STbs$Muhat[,STbs$lopt[1],STbs$lopt[2]]))  # Pr(Y=1) (logistic regression)
cor( STbs$Muhat[,STbs$lopt[1],STbs$lopt[2]], eta)  # correlation with true function
# proportion of correctly identified true nonzero regression coefficients
length(intersect(which(boptbs[-(1:(ncol(Z)+1))]!=0),iA0))/length(iA0)




## End(Not run)
```

`stmgplink` *Smooth-threshold multivariate genetic prediction for genome-wide association or whole-genome sequencing data in PLINK format*

Description

Build prediction model from training data and predict test data phenotype through smooth-threshold multivariate genetic prediction (STMGP) method. Data must be in PLINK binary format and marginal test p-values (i.e. test for each variant) are computed by PLINK software, which enables rapid computation even for data having very large number of variants. An optimal p-value cutoff is selected by Cp-type criterion. Both quantitative and binary phenotypes are acceptable, in which data must be in PLINK fam file format or in a separate file (PLINK format, i.e. FID and IID are needed).

Usage

```
stmgplink(trainbed, testbed = NULL, gamma = 1, taun = NULL,
  lambda = 1, Z = NULL, Zte = NULL, plink = "plink --noweb",
  maf = 0.01, hwe = 1e-04, geno = 0.1, fout = "stp",
  trainfout = "train", testfout = "test", ll = 50, maxal = NULL, alc = NULL,
  tdir = NULL, Znames=NULL, trainphenofile=NULL,
  testphenofile = NULL, phenoname=NULL, pSum = NULL)
```

Arguments

trainbed	A training data file name in PLINK binary format or a vector of three file names of .bed, .bim, .fam; Binary phenotype must be coded as 1 or 2 in PLINK fam file. Missing values in .fam file are -9 as usual.
testbed	A test data file name in PLINK binary format or a vector of three file names of .bed, .bim, .fam; NULL (default) means unspecified. Missing values in .fam are -9 as usual.
gamma	gamma parameter; gamma=1 is default as suggested in Ueki and Tamiya (2016).
taun	tau parameter divided by (sample size) (allowed to be a vector object; optimal parameter is chosen by Cp); NULL (default) specifies tau=n/log(n)^0.5 as suggested in Ueki and Tamiya (2016).
lambda	lambda parameter (default=1).
Z	A covariate file name for training data (PLINK format, i.e. FID and IID are needed) or data matrix, missing values are "-9"; NULL (default) means unspecified.
Zte	A covariate file name for test data (PLINK format, i.e. FID and IID are needed) or data matrix, missing values are "-9"; NULL (default) means unspecified.
plink	PLINK command, e.g. "plink2", "./plink –noweb", or "plink1.9 –memory 100000" (default is plink --noweb) where options can be added; PLINK must be installed.
maf	Minor allele frequency (MAF) cutoff for --maf option in PLINK.
hwe	Hardy-Weinberg equilibrium (HWE) cutoff for --hwe option in PLINK.
geno	Missing call rate cutoff for --geno option in PLINK (default=0.1).
fout	An output file name (default="stp").
trainfout	An output file name for training data (default="train").
testfout	An output file name for test data (default="test").
ll	Number of candidate p-value cutoffs for search (default=50) as determined by 10^seq(log10(maxal),log10(5e-8),length=ll).
maxal	Maximum p-value cutoff for search (default=NULL); If most variants are null maxal*(number of variants) gives approximate number of filtered variants, which is useful for rapid computation even for data with large number of variants.
alc	User-specified candidate p-value cutoffs for search; ll option is effective if alc=NULL.
tdir	Location of temporary files (default=tempdir()).

Znames	Name(s) of covariate used; NULL (default) means unspecified.
trainphenofile	A phenotype file name for training data (PLINK format, i.e. FID and IID are needed) with header columns (i.e. FID, IID, phenoname1, phenoname2, ...) missing values are "-9"; NULL (default) means unspecified.
testphenofile	A phenotype file name for test data (PLINK format, i.e. FID and IID are needed) with header columns (i.e. FID, IID, phenoname1, phenoname2, ...) missing values are "-9"; NULL (default) means unspecified.
phenoname	Phenotype name in trainphenofile; NULL (default) means unspecified but users should provide if trainphenofile is provided.
pSum	User-specified p-values matrix from other studies that are independent of the study data (optional, default=NULL), a matrix object with rows for each variant and columns for each study (multiple studies are capable) where rownames must be specified from SNP IDs that exist in PLINK .bim file. Missing p-values must be coded as NA. Summary p-values are combined with p-values in the study data by the Fisher's method.

Details

See Ueki and Tamiya (2016).

Value

Muhat	Estimated phenotypes from linear model evaluated at each candidate tuning parameters (al and tau) whose size is of (sample size) x (length of al) x (length of tau).
gdf	Generalized degrees of freedom (GDF, Ye 1998) whose size is of (length of al) x (length of tau).
sig2hat	Error variance estimates (=1 for binary traits) whose size is of (length of al) x (length of tau).
df	Number of nonzero regression coefficients whose size is of (length of al) x (length of tau).
al	Candidate p-value cutoffs for search.
lopt	An optimal tuning parameter indexes for al and tau selected by Cp-type criterion, CP
BA	Estimated regression coefficient matrix whose size is of (1 + number of columns of Z + number of columns of X) x (length of al)) x (length of tau)); the first element, the second block and third block correspond to intercept, Z and X, respectively.
Loss	Loss (sum of squared residuals or -2*loglikelihood) whose size is of (length of al) x (length of tau).
sig2hato	An error variance estimate (=1 for binary traits) used in computing the variance term of Cp-type criterion.
tau	Candidate tau parameters for search.
CP	Cp-type criterion whose size is of (length of al) x (length of tau).

PE	PLINK .fam file for training data with additional column including the predicted phenotype from linear model.
PEte	PLINK .fam file for test data with additional column including the predicted phenotype from linear model estimated from training data.
nonzero	Variants with nonzero regression coefficients at the optimal parameter in PLINK file.
DataTr	Training dataset used (y, X and Z)

References

Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira M, Bender D, Maller J, Sklar P, de Bakker P, Daly MJ, Sham PC. (2007) PLINK: A tool set for whole-genome and population-based linkage analyses. Am J Hum Genet 81:559-75.

Chang CC, Chow CC, Tellier LCAM, Vattikuti S, Purcell SM, Lee JJ. (2015) Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience 4.

Examples

```
## Not run:
wd = system.file("extdata",package="stmgp")


# quantitative traits
# training data (plink format)
trainbed = paste(wd,"train",sep="/")
# test data (plink format)
testbed = paste(wd,"test",sep="/")
# number of SNPs
#n.snp = length(readLines(paste(trainbed,".bim",sep="")))
n.snp = 80000


#> head(read.table(paste0(trainbed,".fam")))
# training sample .fam file (quantitative phenotype in the 6th column)
#       V1      V2 V3 V4 V5    V6
#1 id1_100 id2_100  0  0  1 -1.23
#2 id1_101 id2_101  0  0  1  1.48
#3 id1_102 id2_102  0  0  1 -4.27
#4 id1_103 id2_103  0  0  1 -2.61
#5 id1_104 id2_104  0  0  1 -0.27
#6 id1_105 id2_105  0  0  1 -0.50


#> head(read.table(paste0(testbed,".fam")))
# test sample .fam file
# (quantitative phenotype in the 6th column but missing (i.e. "-9") allowed)
#     V1    V2 V3 V4 V5    V6
#1 id1_0 id2_0  0  0  1 -0.59
#2 id1_1 id2_1  0  0  1  1.11
```

```
#3 id1_2 id2_2  0  0  1 -2.45
#4 id1_3 id2_3  0  0  1  0.11
#5 id1_4 id2_4  0  0  1 -1.17
#6 id1_5 id2_5  0  0  1  2.08



pSum = read.table(paste0(wd,"/pSum.txt"));
rownames(pSum) = pSum[,2]; pSum = pSum[,7:8,drop=F]

# summary p-values format
# ("rownames(pSum)" for SNP ID should be provided; "NA" means unavailable)
#> head(pSum,15)  # summary p-values from two studies
#                   V7        V8
#rs12255619         NA        NA
#rs11252546         NA        NA
#rs7909677          NA        NA
#rs10904494         NA        NA
#rs11591988         NA        NA
#rs4508132          NA        NA
#rs10904561         NA        NA
#rs7917054          NA        NA
#rs7906287          NA        NA
#rs12775579         NA        NA
#rs4495823  0.08731436 0.2150108
#rs11253478 0.24030258 0.8726241
#rs9419557  0.49243856 0.3823173
#rs9286070  0.31277506 0.8521706
#rs9419560          NA 0.7604134



# model building from training data without covariates
sp1 = stmgplink(trainbed=trainbed,maxal=5000/n.snp)
head(sp1$PE)

# model building from training data to predict test data without covariates
sp2 = stmgplink(trainbed=trainbed,testbed=testbed,maxal=5000/n.snp)
head(sp2$PEte)
head(sp2$nonzero)

# model building from training data to predict test data without covariates
# (using 1 pSum)
sp2p = stmgplink(trainbed=trainbed,testbed=testbed,maxal=5000/n.snp,
pSum=pSum[,1,drop=F])
head(sp2p$PEte)
head(sp2p$nonzero)

# model building from training data to predict test data without covariates
# (using 2 pSum)
sp2pp = stmgplink(trainbed=trainbed,testbed=testbed,maxal=5000/n.snp,pSum=pSum)
head(sp2pp$PEte)
head(sp2pp$nonzero)
```

```
# using covariates files

Zf = paste(wd,"train.cov",sep="/")
Ztef = paste(wd,"test.cov",sep="/")


#> head(read.table(Zf,header=TRUE))   # covariate for training sample
#      FID     IID        COV1 COV2 qphen bphen
#1 id1_340 id2_340  1.27203944    1 -2.47     2
#2 id1_430 id2_430 -0.44144482    1 -0.71     2
#3 id1_199 id2_199 -0.18200011    1 -3.42     2
#4 id1_473 id2_473  0.03965880    0  0.32     1
#5 id1_105 id2_105  0.20418279    0 -0.50     2
#6 id1_188 id2_188 -0.04838519    0  2.98     1


#> head(read.table(Zfte,header=TRUE))   # covariate for test sample
#     FID    IID       COV1 COV2
#1 id1_80 id2_80 -0.2057512    0
#2 id1_53 id2_53 -0.8627601    1
#3 id1_59 id2_59 -0.2973529    1
#4 id1_71 id2_71  1.4728727    1
#5 id1_92 id2_92  3.5614472    0
#6 id1_25 id2_25  0.5135032    1


# model building from training data
sp3 = stmgplink(trainbed=trainbed,maxal=5000/n.snp,Z=Zf,Znames=c("COV1","COV2"))
head(sp3$PE)

# model building from training data and predicting test data
sp4 = stmgplink(trainbed=trainbed,testbed=testbed,maxal=5000/n.snp,Z=Zf,Zte=Ztef,
                Znames=c("COV1","COV2"))
head(sp4$PEte)
head(sp4$nonzero)

# model building from training data and predicting test data (using 1 pSum)
sp4p = stmgplink(trainbed=trainbed,testbed=testbed,maxal=5000/n.snp,
                Z=Zf,Zte=Ztef,Znames=c("COV1","COV2"),pSum=pSum[,1,drop=F])
head(sp4p$PEte)
head(sp4p$nonzero)

# model building from training data and predicting test data (using 2 pSum)
sp4pp = stmgplink(trainbed=trainbed,testbed=testbed,maxal=5000/n.snp,
                 Z=Zf,Zte=Ztef,Znames=c("COV1","COV2"),pSum=pSum)
head(sp4pp$PEte)
head(sp4pp$nonzero)



# no summary p-values
```

```
cor(sp2$PE[,6:7],use="pair")[1,2]  # training (no covariate)
cor(sp2$PEte[,6:7],use="pair")[1,2]  # test (no covariate)

cor(sp4$PE[,6:7],use="pair")[1,2]  # training (covariates)
cor(sp4$PEte[,6:7],use="pair")[1,2]  # test (covariates)


# 1 summary p-values
cor(sp2p$PE[,6:7],use="pair")[1,2]  # training (no covariate)
cor(sp2p$PEte[,6:7],use="pair")[1,2]  # test (no covariate)

cor(sp4p$PE[,6:7],use="pair")[1,2]  # training (covariates)
cor(sp4p$PEte[,6:7],use="pair")[1,2]  # test (covariates)


# 2 summary p-values
cor(sp2pp$PE[,6:7],use="pair")[1,2]  # training (no covariate)
cor(sp2pp$PEte[,6:7],use="pair")[1,2]  # test (no covariate)

cor(sp4pp$PE[,6:7],use="pair")[1,2]  # training (covariates)
cor(sp4pp$PEte[,6:7],use="pair")[1,2]  # test (covariates)




#### binary traits ####
# training data (plink format)
trainbed = paste(wd,"train",sep="/")
# test data (plink format)
testbed = paste(wd,"test",sep="/")
# number of SNPs
#n.snp = length(readLines(paste(trainbed,".bim",sep="")))
n.snp = 80000


#> head(read.table(paste0(trainbed,"b.fam")))
# training sample .fam file (binary phenotype (1 or 2) in the 6th column)
#       V1      V2 V3 V4 V5 V6
#1 id1_100 id2_100  0  0  1  2
#2 id1_101 id2_101  0  0  1  1
#3 id1_102 id2_102  0  0  1  2
#4 id1_103 id2_103  0  0  1  2
#5 id1_104 id2_104  0  0  1  2
#6 id1_105 id2_105  0  0  1  2


#> head(read.table(paste0(testbed,"b.fam")))
# test sample .fam file (binary phenotype (1 or 2) in the 6th column
# but missing (i.e. "-9") allowed)
#     V1    V2 V3 V4 V5 V6
#1 id1_0 id2_0  0  0  1  2
#2 id1_1 id2_1  0  0  1  1
#3 id1_2 id2_2  0  0  1  2
```

```
#4 id1_3 id2_3  0  0  1  1
#5 id1_4 id2_4  0  0  1  2
#6 id1_5 id2_5  0  0  1  1



# model building from training data without covariates
sp1b = stmgplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),maxal=5000/n.snp)
head(sp1b$PE)

# model building from training data to predict test data without covariates
sp2b = stmgplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
       testbed=paste0(testbed,c(".bed",".bim","b.fam")),maxal=5000/n.snp)
head(sp2b$PEte)
head(sp2b$nonzero)

# model building from training data to predict test data without covariates
# (using 1 pSum)
sp2bp = stmgplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
        testbed=paste0(testbed,c(".bed",".bim","b.fam")),maxal=5000/n.snp,
pSum=pSum[,1,drop=F])
head(sp2bp$PEte)
head(sp2bp$nonzero)

# model building from training data to predict test data without covariates
# (using 2 pSum)
sp2bpp = stmgplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
         testbed=paste0(testbed,c(".bed",".bim","b.fam")),maxal=5000/n.snp,pSum=pSum)
head(sp2bpp$PEte)
head(sp2bpp$nonzero)


# using covariates files

# model building from training data
sp3b = stmgplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
       maxal=5000/n.snp,Z=paste(wd,"train.cov",sep="/"),Znames=c("COV1","COV2"))
head(sp3b$PE)

# model building from training data and predicting test data
sp4b = stmgplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
       testbed=paste0(testbed,c(".bed",".bim","b.fam")),maxal=5000/n.snp,Z=Zf,Zte=Ztef,
       Znames=c("COV1","COV2"))
head(sp4b$PEte)
head(sp4b$nonzero)

# model building from training data and predicting test data (using 1 pSum)
sp4bp = stmgplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
        testbed=paste0(testbed,c(".bed",".bim","b.fam")),maxal=5000/n.snp,
        Z=Zf,Zte=Ztef,Znames=c("COV1","COV2"),pSum=pSum[,1,drop=F])
head(sp4bp$PEte)
head(sp4bp$nonzero)
```

```
# model building from training data and predicting test data (using 2 pSum)
sp4bpp = stmgplink(trainbed=paste0(trainbed,c(".bed",".bim","b.fam")),
         testbed=paste0(testbed,c(".bed",".bim","b.fam")),maxal=5000/n.snp,
         Z=Zf,Zte=Ztef,Znames=c("COV1","COV2"),pSum=pSum)
head(sp4bpp$PEte)
head(sp4bpp$nonzero)



# no summary p-values
cor(sp2b$PE[,6:7],use="pair")[1,2]  # training (no covariate)
cor(sp2b$PEte[,6:7],use="pair")[1,2]  # test (no covariate)

cor(sp4b$PE[,6:7],use="pair")[1,2]  # training (covariates)
cor(sp4b$PEte[,6:7],use="pair")[1,2]  # test (covariates)


# 1 summary p-values
cor(sp2bp$PE[,6:7],use="pair")[1,2]  # trainig (no covariate)
cor(sp2bp$PEte[,6:7],use="pair")[1,2]  # test (no covariate)

cor(sp4bp$PE[,6:7],use="pair")[1,2]  # training (covariates)
cor(sp4bp$PEte[,6:7],use="pair")[1,2]  # test (covariates)


# 2 summary p-values
cor(sp2bpp$PE[,6:7],use="pair")[1,2]  # training (no covariate)
cor(sp2bpp$PEte[,6:7],use="pair")[1,2]  # test (no covariate)

cor(sp4bpp$PE[,6:7],use="pair")[1,2]  # training (covariates)
cor(sp4bpp$PEte[,6:7],use="pair")[1,2]  # test (covariates)




## End(Not run)
```

Index