Package 'GraBLD'

September 7, 2017

Type Package

Title Gradient Boosted and LD adjusted

Version 0.1.0

Author Shihong Mao, Guillaume Pare, Wei Q. Deng

Maintainer Shihong Mao <maos@phri.ca>

Description Gradient boosted and LD adjusted (GraBLD) produces boosted and LD corrected polygenic gene scores that can be applied to polygenic traits prediction. The gradient boosted regression tree model is first used to optimize the weights of SNPs included in the polygenic scores by leveraging the large number of variants in genome-wide studies and the availability of summary association test statistics from meta-analyses. The linkage disequilibrium between SNPs is then adjusted locally in a user-defined regional. The details of the method is described in Pare et al (http://biorxiv.org/content/early/2017/02/09/107409).

License file LICENSE

Encoding UTF-8

Imports gbm, data.table

Depends R (>= 3.1.0)

LazyData true

RoxygenNote 6.0.1

R topics documented:

`annotation_data` *External Annotation.*

Description

A dataset containing external information that can be used to tune the weights in the polygenic gene score using boosted regression tree models.

Usage

```
annotation_data
```

Format

A data frame with 927 rows and 48 variables:

SNP_ID SNP rs ID

BMI_beta_adj Standardized regression coefficient for association of a SNP and BMI obtained from external consortium GIANT

CARDIOGRAM_beta_adj Standardized regression coefficient for association of a SNP and CAD obtained from external consortium CARDIOGRAM

DIAGRAM_beta_adj Standardized regression coefficient for association of a SNP and T2D obtained from external consortium DIAGRAM

HDL_beta_adj Standardized regression coefficient for association of a SNP and HDL obtained from external consortium Global Lipids Genetics Consortium Results

Source

http://www.diagram-consortium.org http://csg.sph.umich.edu/abecasis/public/lipids2013/ http://www.cardiogramplusc4d.org/ http://portals.broadinstitute.org/collaboration/giant/index.php/GIANT_consortium

---

`BMI` *Simulated Body Mass Index.*

Description

A dataset containing the simulated body mass index of 5,000 individuals.

Usage

```
BMI
```

Format

A data frame with 5000 rows and 1 variable:

The BMI values were randomly simulated for each individual independently without assuming an effect from genotypes. The values had been standardized to have mean 0 and variance 1.

BMI body mass index of 5,000 individuals

full_normal_geno *Standardizing the genotype data.*

Description

The function standardizes a genotype data such that for each SNP the mean is 0 and standard deviation is 1.

Usage

```
full_normal_geno(geno_raw, max_size = 1e+05, NAval = NA)
```

Arguments

geno_raw	the genotype matrix assuming each row is the individual and each column is the SNP.
max_size	an integer for the maximum size of SNPs to be standardized at a time, the default is 100000. This option can be ignored for data with less than 1 million SNPs.
NAval	the missing value used in the output data matrix. The default is NA, for use with PLINK, set NAval = -9.

Value

a data matrix of standardized genotypes.

Examples

```
data(geno)
norm_geno <- full_normal_geno(geno_raw = geno)
```

geno *A genotype matrix.*

Description

A dataset containing the genotypes of 927 SNPs on 5,000 individuals.

Usage

```
geno
```

Format

A data matrix with 5000 rows and 927 columns:

The genotype data matrix was generated by PLINK from .bed, .bim and .fam files based on appropriate reference alleles. In the analyses, the applied reference alleles among training data, testing data and consortium data should be identical. The individuals in the phenotype data were match with .fam file (i.e. the number of individuals and the order of individuals should be same); The number of SNPs in prediction variable matrix annotations is matched with that in .bim data;

Source

http://zzz.bwh.harvard.edu/plink/

---

genomewideLD *Genome-wide LD adjustment*

Description

The function compute the LD adjustment on the entire genotype data matrix by first dividing them into SNP blocks.

Usage

```
genomewideLD(geno_data, size = 300)
```

Arguments

geno_data	the standardized genotype matrix assuming each row is the individual and each column is the SNP.
size	an integer for the number of SNPs that should be included in a block. Usually, for genome-wide datasets, there are about 2 million SNPs and +/- 300 SNPs is roughly equivalent to 1Mb physical distance, thus 300 is set as the default size.

Value

a numeric vector of LD adjustments for all SNPs in the genotype matrix.

References

Pare, Guillaume, Shihong Mao, and Wei Q. Deng. A method to estimate the contribution of regional genetic associations to complex traits from summary association statistics. *Scientific reports* **6** (2016): 27644.

---

get_data_num *Normalize the regression coefficients and annotation data*

Description

The function process the unviariate beta regression as well as the annotation data matrix and combine the normalized data used for estimating the optimal boosted regression trees model.

Usage

```
get_data_num(betas, annotations, pos = 2, pos_sign = 3, abs_effect = 2:5,
  normalize = FALSE)
```

Arguments

betas	a matrix of regression coefficients from association analysis in the target population. The first column is the chromosome for each SNP, and the column with the regression coefficient should be specified by setting pos. The default value for pos is 2. The SNP IDs or other information could be present as additional columns. Users need to prepare univariate association beta file without headers. The betas were generated from the model: `coef(summary(lm(pheno_data ~ geno[,j])))[2,1]` Both genotype data and phenotype data over individuals need to be standardized to have mean = 0 and variance = 1.
annotations	a matrix of annotation variables used to update the beta values through gradient boosted regression tree models. Usually, this can be taken from the summary-level test statistics of matching traits from genome-wide consortia available online. The first column of the matrix must be the SNP IDs and the remaining columns could be additional annotation information. The SNP IDs must be in the same order as those in beta.
pos	an integer indicating which columns of the data matrix annotations is the corresponding consortium value and additionally which columns should also be included.
pos_sign	an integer indicating which column of the data matrix annotations should be used to update the sign of the univariate regression coefficient. Usually, it is set to be the consortium univariate regression coefficient of the same trait.
abs_effect	a vector of integers indicating which columns of the data matrix annotations should be used as absolute effect by taking the absolute sign. For example, when only the strength of the effect rather than the direction of the effect is informative for improving the polygenic score weights.
normalize	a logic indicating whether the univariate beta regression coefficients in beta should be normalized with respect to the consortium values in annotations.

Value

a data matrix that can be directly used to estimate the optimal boosted regression trees model.

Examples

```
data(annotation_data)
get_data_num(betas = univariate_beta, annotations = annotation_data)
```

---

get_formulas *Regression Tree Formula*

Description

The function generates the formula to be used in regression trees model.

Usage

```
get_formulas(name, var_names)
```

Arguments

name	a character to indicate the regression coefficients from which complex trait. It should match the trait name used in the other analyses.
var_names	the names of the variables to be included.

Value

a formula specifying the response variable (univariate regression coefficients from target population) and the predictors are used to tune the weights in boosted regression trees model.

Examples

```
data(annotation_data)
var_names <- colnames(annotation_data)
formulas = get_formulas(name = 'BMI', var_names = var_names[2:5])
print(formulas)
```

---

get_NAs *Remove missing values*

Description

The function records and removes the missing values from the input data matrix.

Usage

```
get_NAs(data, NA_value = "NaN")
```

Arguments

data	a numeric data matrix
NA_value	the missing value code in the data, the default is NaN. Could also be NA or −9 commonly used in PLINK data.

Value

a data matrix of genotypes.

GraB *Prediction using the Optimal Gradient Boosted Regression Trees using* gbm.

Description

The function returns the prediction of the polygenic gene score weights based on the optimal gradient boosted regression trees model.

Usage

```
GraB(betas, annotations, pos = 2, pos_sign = 3, abs_effect = 2:5,
  trait_name = NULL, which.var = 2:5, steps = 1, validation = 5,
  verbose = FALSE, interval = 200, sig = 1e-05, interact_depth = 5,
  shrink = 0.001, bag_frac = 0.5, max_tree = 2000, WRITE = FALSE)
```

Arguments

betas	a matrix of regression coefficients from association analysis in the target population. The first column is the chromosome for each SNP, and the column with the regression coefficient should be specified by setting pos. The default value for pos is 2. The SNP IDs or other information could be present as additional columns. Users need to prepare univariate association beta file without headers. The betas were generated from the model: `coef(summary(lm(pheno_data ~ geno[,j])))[2,1]` Both genotype data and phenotype data over individuals need to be standardized to have mean = 0 and variance = 1.
annotations	a matrix of annotation variables used to update the betas values through gradient boosted regression tree models. Usually, this can be taken from the summary-level test statistics of matching traits from genome-wide consortia available online. The first column of the matrix must be the SNP IDs and the remaining columns could be additional annotation information. The SNP IDs must be in the same order as those in betas.
pos	an integer indicating which columns of the data matrix annotations is the corresponding consortium value and additionally which columns should also be included.
pos_sign	an integer indicating which column of the data matrix annotations should be used to update the sign of the univariate regression coefficient. Usually, it is set to be the consortium univariate regression coefficient of the same trait.
abs_effect	a vector of integers indicating which columns of the data matrix annotations should be used as absolute effect by taking the absolute sign. For example, when only the strength of the effect rather than the direction of the effect is informative for improving the polygenic score weights.
trait_name	a character for the name of the quantitative trait, assuming the file is named as trait_name_univariate_beta.txt.
which.var	a vector of integers indicating which columns of annotations should be included in the formula to update the weights. Should be have at least length of two and can not take value 1 (the column for SNP ID). If not provided, it will be set to abs_effect.

steps	an integer indicating the current cross-fold
validation	an integer indicating the total number of cross folds. The default and recommended number of cross-fold is 5.
verbose	a logic indicating whether the adjusted prediction r-squared for each tested model with different number of trees should be returned.
interval	an integer indicating the number of iterated cycles to calculate the best trees using gbm. The default value is 200.
sig	the significance level for including a predictor to build the regression trees model.
interact_depth	an integer for the maximum depth of variable interactions used in gbm. See ?gbm. The default here is 5.
shrink	a shrinkage parameter or the learning rate of the tree models in gbm. See ?gbm. The default is 0.001.
bag_frac	a numeric between 0 and 1, controls the fraction of the training set observations randomly selected to propose the next tree. See ?gbm. The default value is 0.5.
max_tree	an integer indicating the total number of trees to fit in gbm. See ?gbm. The default used here is 2000.
WRITE	a logic indicating whether the results of the GraBLD weights should be written to a file with file name trait_name_gbm_beta.txt.

Details

For large datasets, it is recommended to run from the command line with

```
validation=5
for (( i = 1; i <= $validation; i++))
do
Rscript calculate_gbm.R geno_data
   trait_name annotations_file pos ${i}
   $validation interaction_depth
   shrinkage_parameter bag_fraction
   maximum_tree &
done
```

where the R script calculate_gbm.R might look something like this, while additional options can be added to the argument list:

```
#!/bin/sh
rm(list = ls())
library('GraBLD')
args = (commandArgs(TRUE))
geno_data = args[1]
 trait_name = args[2]
 annotations_file = args[3]
 pos = eval(parse(text=args[4]))
 steps = eval(parse(text=args[5]))
 validation = eval(parse(text=args[6]))
 p1 = eval(parse(text=args[7]))
```

```
p2 = eval(parse(text=args[8]))
p3 = eval(parse(text=args[9]))
p4 = eval(parse(text=args[10]))
betas = load_beta(trait_name)
annotation = load_database(annotations_file, pos = 2:3)
geno <- load_geno(geno_data)
GraB(betas = betas, annotations = annotation,
  trait_name = trait_name, steps = steps, validation = validation,
  interval = 200, sig = 1e-05, interact_depth = p1, shrink = p2,
  bag_frac = p3, max_tree = p4, WRITE = TRUE)
```

Value

a numeric vector of updated weights with length matching the number of SNPs.

References

Greg Ridgeway with contributions from others (2015). gbm: Generalized Boosted Regression Models. R package version 2.1.1. https://CRAN.R-project.org/package=gbm

Guillaume Pare, Shihong Mao, Wei Q Deng (2017) A machine-learning heuristic to improve gene score prediction of polygenic traits Short title: Machine-learning boosted gene scores, *bioRxiv 107409*; doi: https://doi.org/10.1101/107409; http://www.biorxiv.org/content/early/2017/02/09/107409

Examples

```
data(univariate_beta)
data(annotation_data)
GraB(betas = univariate_beta, annotations = annotation_data,
   trait_name = 'BMI', steps = 2, validation = 5)
```

GraBLD.score *Gradient Boosted and LD adjusted Prediction.*

Description

The function returns the prediction r-squared on the target population using polygenic gene score based on the GraBLD heuristic.

Usage

```
GraBLD.score(source_data = NULL, chr = NULL, geno_raw, PLINK = TRUE,
  SNPnames = NULL, max_size = 1e+05, NAval = NA, Pheno = NULL, LDadjVal,
  gbmVal, trait_name = NULL, WRITE = FALSE)
```

Arguments

source_data	the name of the file which the genotype data are to be read from. Each row of the matrix appears as one line of the file. Could be an absolute path to the file or the name of the file assuming in the present directory getwd().

chr	an integer indicating the maximum number of chromosomes to be read in, this option is used in combination with source_data, to perform analysis by each chromosome. In this case, the file name should follow: "source_data_i.raw" for all $i \leq chr$. For example, Geno_Matrix_23.raw.
geno_raw	the genotype matrix assuming each row is the individual and each column is the SNP. Can be skipped if source_data was provided.
PLINK	a logic indicating whether the supplied file is of the .raw format from PLINK, if not, the first six columns will not be removed and all columns will be read in.
SNPnames	a vector of characters for the names of SNPs used, these are only used in the output of GraBLD weights.
max_size	an integer for the maximum size of SNPs to be standardized at a time, the default is 100000. This option can be ignored for data with less than 1 million SNPs.
NAval	the missing value used in the output data matrix. The default is NA, for use with PLINK, set NAval = -9.
Pheno	a numeric vector of quantitative traits with the same length as the number of rows in the genotype matrix.
LDadjVal	a numeric vector of LD adjusted scores with length matching the number of SNPs in the genotype matrix. This can be taken from the output of LDadj().
gbmVal	a numeric vector of gradient boosted weights with length matching the number of SNPs in the genotype matrix. This can be taken from the output of GraB().
trait_name	a character indicating the name of the quantitative trait.
WRITE	a logic indicating whether the results of the GraBLD weights should be written to a file with file name trait_name_gbm_beta.txt.

Value

if Pheno is supplied, both the polygenic gene score as well as the prediction R-squared (adjusted) are returned, otherwise only the polygenic gene score is returned.

References

Guillaume Pare, Shihong Mao, Wei Q Deng (2017) A machine-learning heuristic to improve gene score prediction of polygenic traits Short title: Machine-learning boosted gene scores, *bioRxiv 107409*; doi: https://doi.org/10.1101/107409; http://www.biorxiv.org/content/early/2017/02/09/107409

Examples

```
data(geno)
data(univariate_beta)
data(annotation_data)
LD_val <- LDadj(geno_raw = geno, chr = 1, size = 200)
gbm_val <- list()
for (j in 1:5){
gbm_val[[j]] <- GraB(betas = univariate_beta, annotations = annotation_data,
trait_name = 'BMI', steps = j, validation = 5)
}
data(BMI)
gs <- GraBLD.score(geno_raw = geno, LDadjVal=LD_val, gbmVal = unlist(gbm_val),
 trait_name='BMI', Pheno = BMI[,1])
```

LDadj *LD adjustment.*

Description

The function calculates LD adjustment for each individual SNP based on the number of up/down stream SNPs for genome-wide SNPs.

Usage

```
LDadj(source_data = NULL, geno_raw, size = 300, chr = NULL,
  max_size = 1e+05, write = FALSE, outname = NULL, NAval = NA)
```

Arguments

source_data	the name of the file which the genotype data are to be read from. Each row of the matrix appears as one line of the file. Could be an absolute path to the file or the name of the file assuming in the present directory getwd().
geno_raw	a matrix of the raw genotype data, contains n individuals (by row) and m SNPs (by column). Either source_data or geno_raw can be provided.
size	an integer for the number of SNPs that should be included in a block. Usually, for genome-wide datasets, there are about 2 million SNPs and +/- 300 SNPs is roughly equivalent to 1Mb physical distance, thus 300 is set as the default size.
chr	an integer for the chromosome of genotype data supplied, is not required and only used to name the output file. It is recommended to compute LD adjustments by chromosome to save memory.
max_size	an integer for the maximum size of SNPs to be standardized at a time, the default is 100000. This option can be ignored for data with less than 1 million SNPs.
write	a logic indicating whether the results should be written in a text file. If TRUE, the user should also provide the output file name by specifying outname; otherwise the default is 'LDadj_chr_size_size_SNPs.txt'.
outname	a character giving the name of the output file if write is TRUE. Otherwise, the default name 'LDadj_chr_size_size_SNPs.txt' is used.
NAval	the missing value used in the output data matrix. The default is NA, for use with PLINK, set NAval = -9.

Details

For large datasets, it is recommended to run from the command line with

```
for((i = 1; i <= chr; i++))
do
Rscript PerformLDadj.R size data_name ${i} &
done
```

where the R script PerformLDadj.R might look something like this, while additional options can be added to the argument list:

```
#!/bin/sh
rm(list = ls())
library('GraBLD')
args = (commandArgs(TRUE))
size = eval(parse(text=args[1]))
source_data = args[2]
chr = eval(parse(text=args[3]))
geno_data = load_geno(source_data = source_data, PLINK = TRUE, chr = chr)
geno_norm = full_normal_geno(geno_data)
LD_OUT <- LDadj(geno_raw = geno_norm, chr = chr, size = size, write = TRUE)
```

Value

a numeric vector of LD adjustments with length matching the number of SNPs in the genotype data provided.

References

Guillaume Pare, Shihong Mao, Wei Q Deng (2017) A machine-learning heuristic to improve gene score prediction of polygenic traits Short title: Machine-learning boosted gene scores, *bioRxiv 107409*; doi: https://doi.org/10.1101/107409; http://www.biorxiv.org/content/early/2017/02/09/107409

Pare, Guillaume, Shihong Mao, and Wei Q. Deng. A method to estimate the contribution of regional genetic associations to complex traits from summary association statistics. *Scientific reports* **6** (2016): 27644.

Examples

```
data(geno)
LDadj(geno_raw = geno, chr = 1, size = 200)
```

---

load_beta *Load univariate regression beta coefficients.*

Description

The function loads the univariate regression beta coefficients into the workspace.

Usage

```
load_beta(trait_name, file_name = NULL)
```

Arguments

trait_name	a character for the name of the quantitative trait, assuming the file is named as trait_name_univariate_beta.txt.
file_name	a character to specify the full directory of the univariate regression coefficient file instead of trait_name_univariate_beta.txt.

Details

The univariate regression coefficients are assumed to be computed univariately using: `coef(summary(lm(pheno_data ~ geno[,j])))[2,1]`. Both genotype data and phenotype data over individuals need to be standardized with mean = 0 and variance = 1. The data file should have two columns indicating chromosome number and the regression coefficients without headers.

Value

a data matrix of univariate regression coefficients.

`load_database` *Load annotations.*

Description

The function loads the annotation predictor variables into the workspace.

Usage

```
load_database(annotation_file, pos = 2)
```

Arguments

`annotation_file`	the directory to a `data.frame` of annotation variables used to update the `beta` values through gradient boosted regression tree models. The first column of the matrix must be the SNP IDs and the remaining columns could be additional annotation information. The SNP IDs must be in the same order as those in `beta`.
`pos`	an integer indicating which columns of the data matrix `annotations` is the corresponding consortium value and additionally which columns should also be included.

Details

The annotation matrix provides the necessary predictor variables used to update the weights of polygenic gene score via gradient boosted regression tree. The data.frame should have at least two columns, the first column is SNP_ID; the rest are the adjusted consortia regression coefficient or summary statistics. It is recommended to adjust the consortia regression coefficient by the minor allele frequency of the SNP:

```
SNP_SD = sqrt(2 * as.numeric(MAF[,5]) * (1 - as.numeric(MAF[,5])))
beta_adj = as.numeric(beta) * SNP_SD
```

For any one trait, at least one column of corresponding adjusted beta from the consortium is required. For instance, if we work on BMI, at least the adjusted regression coefficient for association with BMI in a consortium study should be provided. Additional annotations such as related regression coefficients of other traits, or SNP functional annotations can also be included.

Value

a data frame of predictor variables that can be used to update SNPs weights.

`load_geno` *Loading the genotype data.*

Description

The function automatically loads the genotype data matrix assuming each row is an individual and each column is the genotype of a biallelic SNP. If standard PLINK .raw file was provided, the function returns only the genotype matrix by removing the first 6 columns.

Usage

```
load_geno(source_data, PLINK = TRUE, chr = NULL)
```

Arguments

`source_data`	the name of the file which the genotype data are to be read from. Each row of the matrix appears as one line of the file. Could be an absolute path to the file or the name of the file assuming in the present directory `getwd()`.
`PLINK`	a logic indicating whether the supplied file is of the .raw format from PLINK, if not, the first six columns will not be removed and all columns will be read in.
`chr`	an integer indicating which chromosome is read in, this option is used in combination with `source_data`, to perform analysis by each chromosome. In this case, the file name should follow: `source_data_i.raw`. For example, `Geno_Matrix_1.raw` or `Geno_Matrix_23.raw` if all 23 files are available.

Value

a data matrix of genotypes.

`regionalLD` *Regional LD adjustment*

Description

The function calculates the LD adjustment of each SNP for one given size of genotype data block.

Usage

```
regionalLD(geno_data, position = "start", size = 300)
```

Arguments

`geno_data`	the standardized genotype matrix assuming each row is the individual and each column is the SNP.
`position`	a character indicating the position of the SNP in a block of SNPs, one of `start` (the start of the block), `mid` (the middle of the block), `end` (the end of the block), or `together` (combining SNPs in a few blocks), `short` (in a short block), or `all` (using all regions).
`size`	an integer for the number of SNPs that should be included in a block. Usually, for genome-wide datasets, there are about 2 million SNPs and +/- 300 SNPs is roughly equivalent to 1Mb physical distance, thus 300 is set as the default size.

Value

a numeric vector of LD adjustments with the same length as the number of SNPs in the genotype matrix.

References

Pare, Guillaume, Shihong Mao, and Wei Q. Deng. A method to estimate the contribution of regional genetic associations to complex traits from summary association statistics. *Scientific reports* **6** (2016): 27644.

Examples

```
data(geno)
norm_geno <- full_normal_geno(geno_raw = geno)
regLD <- regionalLD(geno_data = norm_geno, position = 'mid', size = 300)
print(regLD)
```

`univariate_beta` *BMI Univariate beta regression coefficient.*

Description

A dataset containing the univariate regression coefficient of 927 SNPs for association with BMI from the simulated data.

Usage

```
univariate_beta
```

Format

A data frame with 927 rows and 2 variables:

The univariate regression coefficients were computed from the simulated datasets `BMI` and `geno` using:

`coef(summary(lm(pheno_data ~ geno[,j])))[2,1].`

Both genotype data and phenotype data over individuals need to be standardized with mean = 0 and variance = 1.

V1 chromosome

V2 BMI univariate regression coefficient ...

Index