Package 'sp.gwas'

July 23, 2020

Type Package

Title Variable Selection Package Using Selection Probability for Genome-Wide Association Studies

Version 1.2.0

Author Kipoong Kim

Maintainer Kipoong Kim <kkp7700@gmail.com>

Description Selection probabilities using generalized linear model with regularization for a SNP data in the hapmap format.

License GPL-2 | GPL-3

Depends R (>= 3.6.0)

Imports ggplot2, compiler, glmnet, dplyr, CMplot, readxl, data.table, writexl, bestNormalize, gridExtra, tidyr, gtools, calibrate, genetics, pbapply, HardyWeinberg

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

R topics documented:

`sp.gwas-package` *Selection probabilities using generalized linear model with regularization for a SNP data in the hapmap format.*

Description

sp.gwas

Details

The penalty function of `elastic-net` is defined as

$$\alpha||\beta||_1 + (1-\alpha)||\beta||_2/2,$$

where α is a mixing proportion of ridge and the lasso, and β is regression coefficients. This penalty is equivalent to the Lasso penalty if `alpha=1`.

Value

A list of data files(genotype, phenotype, etc.), results for selection probabilities, and manhattan plot for multiple traits.

References

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical methodology), 67(2), 301-320.

`import.hapmap` *A function to import the hapmap formatted SNP data and the corresponding phenotype data*

Description

Input: Hapmap-formatted SNP data, phenotype data

Output: Matched data files (genotype, numerical, SNP information, QC information, and phenotype) with QC and/or imputation.

Usage

```
import.hapmap(
  genotype = NULL,
  phenotype = NULL,
  input.type = c("object", "path"),
  save.path,
  y.col = NULL,
  y.id.col = 2,
  family = "gaussian",
  normalization = TRUE,
  remove.missingY = TRUE,
  imputation = FALSE,
  QC = TRUE,
  callrate.range = c(0, 1),
  maf.range = c(0, 1),
  HWE.range = c(0, 1),
  heterozygosity.range = c(0, 1)
)
```

Arguments

genotype	Either R object or file path can be considered. A genotype data is not a data.frame but a matrix with dimension p by (n+11). It is formatted by hapmap which has (rs, allele, chr, pos) in the first four(1-4) columns, (strand, assembly, center, protLSID, assayLSID, panel, Qcode) in the following seven(5-11) columns. If NULL, user can choose a path in interactive use.
phenotype	Either R object or file path can be considered. A phenotype data is an n by p matrix. Since the first some columns can display attributes of the phenotypes, you should enter the arguments, y.col and y.id.col, which represent the columns of phenotypes to be analyzed and the column of sample ID. If NULL, user can choose a path in interactive use.
input.type	Default is "object". If input.type is "object", obejects of genotype/phenotype will be entered, and if "path", paths of genotype/phenotype will be enterd. If you want to use an object, you have to make sure that the class of each column of genotype data is equal to "character".
save.path	A save.path which has all output files. If there exists save.path, sp.gwas will check if there is an output file. Note that if there is an output RData file in "save.path", sp.gwas will just load the output files(.RData) in there, thereby not providing the results for new "genotype" and "phenotype".
y.col	The columns of phenotypes. At most 4 phenotypes can be considered, because the plot of them will be fine. Default is 2.
y.id.col	The column of sample ID in the phenotype data file. Default is 1.
family	A family of response variable(phenotype). It is "gaussian" for continuous response variable, "binomial" for binary, "poisson" for count, etc. Now you can use only the same family for the multi phenotypes. For more details, see the function(stats::glm). Default is "gaussian".
normalization	If TRUE. phenotypes are converted to be normal-shape using box-cox transformation when all phenotypes are positive.
remove.missingY	If TRUE, the samples with missing values in phenotype data are removed. Accordingly, the corresponding genotype samples are also filtered out. Default is TRUE.
imputation	TRUE or FALSE for whether imputation will be conducted.
QC	TRUE or FALSE for whether QC for SNPs will be conducted.
maf.range	A numeric vector indicating the range of minor allele frequency (MAF) to be used. Default is c(0, 1).
HWE.range	A numeric vector indicating the range of pvalue by Hardy-Weinberg Equillibrium to be used. Default is c(0, 1).
heterozygosity.range	A numeric vector indicating the range of heterozygosity values to be used, because, in some cases, heterozygosity higher than expected indicates the low quality variants or sample contamination. Default is c(0, 1).

Details

Hardy-Weinberg Equillibrium test was derived from "genetics" package. In imputation process, we first calculate the empirical allele frequencies. If we use a beta distribution as a prior in order to estimate the posterior distribution of allele frequency, then the posterior distribution of allele frequecy is also beta distribution. Accordingly, we impute the missing values with samples from the posterior distribution.

Value

A folder containing a genomic data set in which the samples of genotype and phenotype data are matched, and that quality control steps can be conducted for genotype data

Author(s)

Kipoong Kim <kkp7700@gmail.com>

Examples

```
genotype <- sp.gwas::genotype # load("genotype.rda")
phenotype <- sp.gwas::phenotype # load("phenotype.rda")

# object
import.hapmap(genotype = genotype,
              phenotype = phenotype,
              input.type = c("object", "path")[1],
              imputation = FALSE,
         QC = TRUE,  # if TRUE, the following QC steps (callrate, maf, HWE, heterozygosity) are conducted.
              callrate.range = c(0.95, 1),
              maf.range = c(1e-3, 1),
              HWE.range = c(0, 1),
              heterozygosity.range = c(0, 1),
         remove.missingY = TRUE,  # if TRUE, the samples with any missing phenotypes are filtered out in all dat
              save.path = "./EXAMPLE_obj",
              y.id.col = 1,
              y.col = 2:4,
         normalization = FALSE, #if family is not "gaussian", i.e. not continuous variable, normalization shoul
              family="gaussian")



# path

write.table( x = sp.gwas::genotype, file = "./genotype.csv", row.names = FALSE, col.names = FALSE, sep=",")
write.table( x = sp.gwas::phenotype, file = "./phenotype.csv", row.names = FALSE, sep="," )

import.hapmap(genotype = "./genotype.csv",
              phenotype = "./phenotype.csv",
              input.type = c("object", "path")[2],
         QC = TRUE,  # if TRUE, the following QC steps (callrate, maf, HWE, heterozygosity) are conducted.
              callrate.range = c(0.95, 1),
              maf.range = c(1e-3, 1),
              HWE.range = c(0, 1),
              heterozygosity.range = c(0, 0.5),
         remove.missingY = TRUE,  # if TRUE, the samples with any missing phenotypes are filtered out in all dat
              save.path = "./EXAMPLE_path",
              y.id.col = 1,
              y.col = 2:4,
         normalization = FALSE, #if family is not "gaussian", i.e. not continuous variable, normalization shoul
              family="gaussian")
```

qqman_manhattan *Creates a manhattan plot*

Description

Creates a manhattan plot from PLINK assoc output (or any data frame with chromosome, position, and p-value).

Usage

```
qqman_manhattan(
  x,
  chr = "CHR",
  bp = "BP",
  p = "P",
  snp = "SNP",
  col = c("gray10", "gray60"),
  col.highlight = "green",
  chrlabs = NULL,
  suggestiveline = -log10(1e-05),
  genomewideline = -log10(5e-08),
  highlight = NULL,
  logp = TRUE,
  annotatePval = NULL,
  annotateTop = TRUE,
  ...
)
```

Arguments

x	A data.frame with columns "BP," "CHR," "P," and optionally, "SNP."
chr	A string denoting the column name for the chromosome. Defaults to PLINK's "CHR." Said column must be numeric. If you have X, Y, or MT chromosomes, be sure to renumber these 23, 24, 25, etc.
bp	A string denoting the column name for the chromosomal position. Defaults to PLINK's "BP." Said column must be numeric.
p	A string denoting the column name for the p-value. Defaults to PLINK's "P." Said column must be numeric.
snp	A string denoting the column name for the SNP name (rs number). Defaults to PLINK's "SNP." Said column should be a character.
col	A character vector indicating which colors to alternate.
col.highlight	A character vector of colors corresponding to the list of "highlight".
chrlabs	A character vector equal to the number of chromosomes specifying the chromosome labels (e.g., `c(1:22,"X","Y","MT")`).
suggestiveline	Where to draw a "suggestive" line. Default -log10(1e-5). Set to FALSE to disable.
genomewideline	Where to draw a "genome-wide sigificant" line. Default -log10(5e-8). Set to FALSE to disable.

highlight	A list of character vector of SNPs in your dataset to highlight. These SNPs should all be in your dataset.
logp	If TRUE, the -log10 of the p-value is plotted. It isn't very useful to plot raw p-values, but plotting the raw value could be useful for other genome-wide plots, for example, peak heights, bayes factors, test statistics, other "scores," etc.
annotatePval	If set, SNPs below this p-value will be annotated on the plot.
annotateTop	If TRUE, only annotates the top hit on each chromosome that is below the annotatePval threshold.
...	Arguments passed on to other plot/points functions

Value

A manhattan plot.

Examples

```
#\dontrun{
#qqman_manhattan(gwasResults)
#}
```

---

qqman_qq *Creates a Q-Q plot*

Description

Creates a quantile-quantile plot from p-values from a GWAS study.

Usage

```
qqman_qq(pvector, ...)
```

Arguments

pvector	A numeric vector of p-values.
...	Other arguments passed to `plot()`

Value

A Q-Q plot.

Examples

```
## Not run:
qqman_qq(gwasResults$P)

## End(Not run)
```

sp.gwas *Selection probabilities using generalized linear model with regularization for a SNP data in the hapmap format.*

Description

For analysis of high-dimensional genomic data, penalized regression can be a solution to accomodate correlations between predictors. Moreover, selection probabilities do not depend on tuning parameter selection so that it produces a stablity selection. Thresholds are also generated to control the false positives(errors). Thresholds vary with the expected number of false positives to be controlled by the user. Input data is the hapmap formatted SNP data and phenotype data corresponding to SNP data. Output includes files of three types: (1) Matched data files (genotype, numerical, snp info, and phenotype), (2) Results file (selection probabilities and thresholds), (3) Circular manhattan plot with (blue dotted) significant line corresponding to the largest value among user-defined false discoveries.

Usage

```
sp.gwas(
  genotype = NULL,
  phenotype = NULL,
  input.type = c("object", "path"),
  imputation = FALSE,
  QC = TRUE,
  callrate.range = c(0, 1),
  maf.range = c(0, 1),
  HWE.range = c(0, 1),
  heterozygosity.range = c(0, 1),
  remove.missingY = TRUE,
  save.path = "./sp.folder",
  y.col = 2,
  y.id.col = 1,
  normalization = TRUE,
  method = "lasso",
  family = "gaussian",
  false.discovery = c(1, 5, 10),
  permutation = FALSE,
  nperm = 100,
  plot.ylim = NULL,
  lambda.min.quantile = 0.5,
  n.lambda = 10,
  K = 100,
  psub = 0.5,
  manhattan.type = "c",
  plot.name = "",
  plot.type = "jpg",
  plot.dpi = 300
)
```

Arguments

genotype	Either R object or file path can be considered. A genotype data is not a data.frame but a matrix with dimension p by (n+11). It is formatted by hapmap which has (rs, allele, chr, pos) in the first four(1-4) columns, (strand, assembly, center, protLSID, assayLSID, panel, Qcode) in the following seven(5-11) columns. If NULL, user can choose a path in interactive use.
phenotype	Either R object or file path can be considered. A phenotype data is an n by p matrix. Since the first some columns can display attributes of the phenotypes, you should enter the arguments, y.col and y.id.col, which represent the columns of phenotypes to be analyzed and the column of sample ID. If NULL, user can choose a path in interactive use.
input.type	Default is "object". If input.type is "object", obejects of genotype/phenotype will be entered, and if "path", paths of genotype/phenotype will be enterd. If you want to use an object, you have to make sure that the class of each column of genotype data is equal to "character".
imputation	TRUE or FALSE for whether imputation will be conducted.
QC	TRUE or FALSE for whether QC for SNPs will be conducted.
maf.range	A numeric vector indicating the range of minor allele frequency (MAF) to be used. Default is c(0, 1).
HWE.range	A numeric vector indicating the range of pvalue by Hardy-Weinberg Equillibrium to be used. Default is c(0, 1).
heterozygosity.range	A numeric vector indicating the range of heterozygosity values to be used, because, in some cases, heterozygosity higher than expected indicates the low quality variants or sample contamination. Default is c(0, 1).
remove.missingY	If TRUE, the samples with missing values in phenotype data are removed. Accordingly, the corresponding genotype samples are also filtered out. Default is TRUE.
save.path	A save.path which has all output files. If there exists save.path, sp.gwas will check if there is an output file. Note that if there is an output RData file in "save.path", sp.gwas will just load the output files(.RData) in there, thereby not providing the results for new "genotype" and "phenotype".
y.col	The columns of phenotypes. At most 4 phenotypes can be considered, because the plot of them will be fine. Default is 2.
y.id.col	The column of sample ID in the phenotype data file. Default is 1.
normalization	If TRUE. phenotypes are converted to be normal-shape using box-cox transformation when all phenotypes are positive.
method	A method of penalized regression. It includes "lasso" for the lasso and "enet" for the elastic-net.
family	A family of response variable(phenotype). It is "gaussian" for continuous response variable, "binomial" for binary, "poisson" for count, etc. Now you can use only the same family for the multi phenotypes. For more details, see the function(stats::glm). Default is "gaussian".
false.discovery	The expected number of false discovery to be controlled. The larger it is, the higher threshold becomes. Default is c(1, 5, 10).

permutation	Permutation-based threshold values can be provided for false.discovery if permutation is TRUE. Default is FALSE.
nperm	The number of permutation replicates. Note that the calculation time increases as nperm increases. Default is 100.
plot.ylim	A range of the y-axis. If NULL, automatic range in the y-axis will be provided. For plot.ylim=c(0,1), the y-axis has a range of 0 and 1.
lambda.min.quantile	A range of lambda sequence. Default is 0.5 (median). If the range is so small that it can have many tied selection probabilities which is 1. To handle with this problem, you should increase the value of "lambda.min.quantile".
n.lambda	The length of lambda sequence. The larger n.lambda, the more detailed lambda sequence will be.
K	The number of iterations in resampling when calculating the selection probabilities.
psub	The subsampling proportion. For efficiency, default is 0.5.
manhattan.type	A type of manhattan plot to be drawn includes circular('c') and rectangular('m').
plot.name	A name of plot file.
plot.type	A type of plot file which includes "jpg", "pdf", "tiff", etc.
plot.dpi	A resolution of plot. If you want to get a high-resolution image, plot.dpi should be large.

Details

The penalty function of elastic-net is defined as

$$\alpha||\beta||_1 + (1-\alpha)||\beta||_2/2,$$

where α is a mixing proportion of ridge and the lasso, and β is regression coefficients. This penalty is equivalent to the Lasso penalty if alpha=1.

An algorithm of selection probabilities with elastic-net.

0 : Let us assume that a genomic data has n samples and p variables.
1 : For all $\Lambda = (\alpha, \lambda)$, where $\alpha in [0,1]$, $\lambda > 0$.
2 : for k=1 to K do.
3 : ——- Subsample I_k with size $[n/2]$.
4 : ——- Compute $\hat{\beta}_j^{\Lambda}(I_k)$ with regularization model.
5 : end for.
6 : $SP_j^{\Lambda} = \frac{1}{K}\#\{k \leq K : \hat{\beta}_j^{\Lambda}(I_k) \neq 0\}$.
7 : $SP_j = \max_{\Lambda} SP_j^{\Lambda}, \;\; j = 1, \cdots, p$.
8 : return SP=$(SP_1, \cdots, SP_p)$.

Value

Histogram of original and transformed phenotypes	Histogram of phenotypes with p-value by Shapiro-test on the top right corner.
myDATA	A list of myX, myGD, myGM, myGT, myY, and myY.original(for "gaussian").
sp.res	A list of sp.df and threshold.

`Circular Manhattan plot`
: Manhattan plot for the first phenotype is the innermost circle. Colors for chromosome is fixed, so that if you want to change colors, you would edit the R code of sp.manhattan function.

Note that, before your code, you have to specify the setseed value to get reproducible results, because it uses the resampling approach when calculating the selection probability.

The dataframe with new mean and sum columns

Author(s)

Kipoong Kim <kkp7700@gmail.com>

References

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical methodology), 67(2), 301-320.

Meinshausen, N., & Bühlmann, P. (2010). Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4), 417-473.

Kim, K., Koo, J., & Sun, H. (2020). An empirical threshold of selection probability for analysis of high-dimensional correlated data. Journal of Statistical Computation and Simulation, 1-12.

Examples

```
genotype <- sp.gwas::genotype # load("genotype.rda")
phenotype <- sp.gwas::phenotype # load("phenotype.rda")

# elastic-net model
devAskNewPage(ask = FALSE)
sp.gwas(genotype = genotype,
        phenotype = phenotype,
        input.type = c("object", "path")[1],
        QC = TRUE,
        callrate.range = c(0.95, 1),
        maf.range = c(1e-3, 1),
        HWE.range = c(0, 1),
        heterozygosity.range = c(0, 1),
        remove.missingY = TRUE,
        save.path = "./EXAMPLE_enet",
        y.id.col = 1,
        y.col = 2:4,
        normalization = FALSE,
        method="enet",
        family="gaussian",
        false.discovery = c(1,5,10),
        permutation=FALSE,
        plot.ylim = NULL,
        lambda.min.quantile = 0.5,
        n.lambda = 10,
        K = 20,
        psub = 0.5,
        manhattan.type = c("c", "r")[1],
        plot.name = "Test",
        plot.type = "jpg",
        plot.dpi = 300)
```

```
# Manhattan plot for the first phenotype (Y1)
results <- read.csv("./EXAMPLE_enet/[2]sp.results.csv")
class(results$chr) <- "numeric"
class(results$pos) <- "numeric"
thresholds <- read.csv("./EXAMPLE_enet/[2]sp.thresholds.csv")
threshold_Y1_FD1 <- subset( subset(thresholds, Method=="Theoretical"), FD==1)$Y1
threshold_Y1_FD10 <- subset( subset(thresholds, Method=="Theoretical"), FD==10)$Y1
highlight1 <- results$rs[results$Y1 > threshold_Y1_FD1]
highlight10 <- setdiff( results$rs[results$Y1 > threshold_Y1_FD10], highlight1 )

jpeg("./EXAMPLE_enet/Manhattan_from_qqman.jpeg", width=12, height=5, unit="in", res=600)
qqman_manhattan(results,
                chr="chr", bp="pos", snp="rs", p="Y1", logp=FALSE,
                suggestiveline = threshold_Y1_FD1,
                genomewideline = threshold_Y1_FD10,
                highlight = list(highlight1, highlight10),
                col.highlight = c("blue", "red"),
                ylab="Selection probabilities", ylim=c(0,1))
dev.off()



# Lasso model with permuted threshold

devAskNewPage(ask = FALSE)
sp.gwas(genotype = genotype,
        phenotype = phenotype,
        input.type = c("object", "path")[1],
        maf.range = c(1e-3, 1),
        HWE.range = c(0, 1),
        heterozygosity.range = c(0, 1),
        remove.missingY = TRUE,
        save.path = "./EXAMPLE_lasso-perm",
        y.id.col = 1,
        y.col = 2,
        normalization = FALSE,
        method="lasso",
        family="gaussian",
        false.discovery = c(1,5,10),
        permutation=TRUE,
        nperm=10,
        plot.ylim = NULL,
        lambda.min.quantile = 0.5,
        n.lambda = 10,
        K = 20,
        psub = 0.5,
        manhattan.type = c("c", "r")[1],
        plot.name = "Test_perm",
        plot.type = "jpg",
        plot.dpi = 300)

# Manhattan plot for the first phenotype (Y1) with permuted threshold
results <- read.csv("./EXAMPLE_lasso-perm/[2]sp.results.csv")
class(results$chr) <- "numeric"
class(results$pos) <- "numeric"
thresholds <- read.csv("./EXAMPLE_lasso-perm/[2]sp.thresholds.csv")
```

```
threshold_theory_FD1 <- subset( subset(thresholds, Method=="Theoretical"), FD==1)$Y1
threshold_perm_FD1 <- subset( subset(thresholds, Method=="Permuted"), FD==1)$Y1
highlight_theory <- results$rs[results$Y1 > threshold_theory_FD1]
highlight_perm <- results$rs[results$Y1 > threshold_perm_FD1]
highlight_intersect <- intersect(highlight_theory, highlight_perm)

jpeg("./EXAMPLE_lasso-perm/Manhattan_from_qqman.jpeg", width=12, height=5, unit="in", res=600)
qqman_manhattan(results,
                chr="chr", bp="pos", snp="rs", p="Y1", logp=FALSE,
                suggestiveline = threshold_theory_FD1,
                genomewideline = threshold_perm_FD1,
                highlight = list(highlight_theory, highlight_perm, highlight_intersect),
                col.highlight = c("blue", "red", "purple"),
                ylab="Selection probabilities", ylim=c(0,1))
dev.off()
```

Index