Package ‘bugwas’

May 19, 2016

Title What the Package Does (one line, title case)

Version 0.0.0.9000

Description What the package does (one paragraph)

Depends R (>= 3.2.1)

License What license is it under?

LazyData true

Imports ape,
phangorn

RoxygenNote 5.0.1.9000

R topics documented:

all_plots *Generates all plots.*

Description

This function generates all the plots

Usage

```
all_plots(biallelic = NULL, triallelic = NULL, genVars = NULL,
  treeInfo = NULL, config = NULL)
```

Arguments

biallelic	A list called 'biallelic' created from the lin_loc function
triallelic	A list called 'triallelic' created from the lin_loc function
genVars	A list called 'genVars' created from the lin_loc function
treeInfo	A list called 'treeInfo' created from the lin_loc function
config	A list called 'config' created from the lin_loc function

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
all_plots(biallelic = data$biallelic, triallelic = data$triallelic,
genVars = data$genVars, treeInfo = data$treeInfo, config = data$config)
```

bayesianWaldTestPCsBarplot *Barplot of Bayesian Wald Test on principal components*

Description

This function generates the barplot of Bayesian Wald Test on principal components.

Usage

```
bayesianWaldTestPCsBarplot(config, biallelic, treeInfo, colourPalette = NULL,
  p.genomewidepc = NULL)
```

Arguments

config	A list called 'config' created from the lin_loc function. It is a required input.
biallelic	A list called 'biallelic' created from the lin_loc function. It is a required input.
treeInfo	A list called 'treeInfo' created from the lin_loc function. It is a required input.
colourPalette	A vector of colours colour the significant principal components identified by the Bayesian Wald test (see testGenomeWidePCs). If this is NULL then colours are chosen from a default colour palette.

`p.genomewidepc`	A matrix of the significant principal component and their correlation with lineages. This is Bayesian Wald test results produced by the function testGenomeWidePCs. If this is NULL then testGenomeWidePCs is called to generate the required test results.

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
testGenomeWidePCs(config = data$config, biallelic = data$biallelic)
```

genVarPlots *Plots for GWAS on general variants*

Description

This function generates the various Manhattan plots for general variants.

Usage

```
genVarPlots(genVars, biallelic, config, colourPalette = NULL)
```

Arguments

`genVars`	A list called 'genVars' created from the lin_loc function. It is a required input.
`biallelic`	A list called 'biallelic' created from the lin_loc function. It is a required input.
`config`	A list called 'config' created from the lin_loc function. It is a required input.
`colourPalette`	A vector of colours colour the significant principal components identified by the Bayesian Wald test (see testGenomeWidePCs). If this is NULL then colours are chosen from a default colour palette.

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
genVarPlots(genVars = data$genVars, biallelic = data$biallelic, config = data$config)
```

getDataFromGemma *This function retrieves the binary data from GEMMA input files.*

Description

This function performs the genome-wide principal components.

Usage

```
getDataFromGemma(prefix = NULL, gemmaGenFile = NULL, gemmaSnpFile = NULL,
  id = NULL)
```

Arguments

`prefix` Prefix of the output files.

`gemmaGenFile` The path to the GEMMA gen input file. It is a required input.

`gemmaSnpFile` The path to the GEMMA snp input file. It is a required input.

`id` A vector of the sample ids.

Value

A list of information on the biallelic snps

Examples

```
pheno <- read.table(file = pheno, header=T, as.is = T, sep="\t")
data <- getDataFromGemma(gemmaGenFile = gemmaGenFile, gemmaSnpFile = gemmaSnpFile,
id = pheno.df$id, prefix = prefix)
```

`getSNPDataFromGEMMA` *This function retrieves the binary data from GEMMA input files and phenotype files.*

Description

This function generates the Manhattan plot(s) for a SNP GWAS.

Usage

```
getSNPDataFromGEMMA(gemmaGenFile = NULL, gemmaSnpFile = NULL,
  pheno = NULL, prefix = NULL)
```

Arguments

`gemmaGenFile` The path to the GEMMA gen input file. It is a required input.

`gemmaSnpFile` The path to the GEMMA snp input file. It is a required input.

`pheno` The path of the phenotype file with contains sample ids in a column and phenotypes in another. The columns containing the ids and phenotypes have the repective headings id and phenotype.

`prefix` Prefix of the output files.

Value

A list of information on the biallelic snps and phenotypes.

Examples

```
data <- getSNPDataFromGEMMA(gemmaGenFile = gemmaGenFile, gemmaSnpFile = gemmaSnpFile,
pheno = pheno, prefix = prefix)
```

`linLocGEMMA` *Lineage and locus tests for bacterial GWAS using GEMMA input files.*

Description

This function tests for locus effects using GEMMA and lineage effects using a bayesian wald test for haploid data

Usage

```
linLocGEMMA(gemmaGenFile = NULL, gemmaSnpFile = NULL, pheno = NULL,
  phylo = NULL, prefix = NULL, gem.path = NULL, pcs = NULL,
  lmm.bi = NULL, lmm.tri.tetra = NULL, logreg.bi = NULL,
  logreg.tri.tetra = NULL, var.matrix = NULL, logreg.var = NULL,
  lmm.var = NULL, cutOffCor = 0, run.lmm = TRUE, maf = 0,
  relmatrix = NULL, lognull = NULL, lambda = NULL, output.dir = getwd(),
  creatingAllPlots = TRUE, allBranchAndPCCor = FALSE,
  runTriTetrallelic = TRUE)
```

Arguments

`gemmaGenFile` The path to the GEMMA gen input file. It is a required input.

`gemmaSnpFile` The path to the GEMMA snp input file. It is a required input.

`pheno` A file name specified by either a variable of mode character, or a double-quoted string, containing a column of sample names with header 'ID' and a column of the binary phenotype (coded by 0s and 1s) with column header 'pheno'. Required argument.

`phylo` A file name specified by either a variable of mode character, or a double-quoted string, containing a phylogeny of the samples, with the same names matching with arguments gen and pheno. Required argument.

`prefix` Output file prefix. Required argument.

`gem.path` A file path specified by either a variable of mode character, or a double-quoted string. gem.path is the file path to the software GEMMA (version >= ?). Required argument.

`pcs` A file name specified by either a variable of mode character, or a double-quoted string, containing the principle components of the data. Column names should be 'PC1' to 'PCn' and row names should be the sample names.

`lmm.bi` A file name specified by either a variable of mode character, or a double-quoted string, containing GEMMA results (ending '.assoc.txt') for the biallelic SNPs in argument 'gen'.

`lmm.tri.tetra` A file name specified by either a variable of mode character, or a double-quoted string, containing GEMMA results for the tri and tetra allelic SNPs in argument 'gen'. This must contain column headers 'ps' for SNP positions/IDs, 'pvals' for p-values and 'negLog10' for -log10(p).

`logreg.bi` A file name specified by either a variable of mode character, or a double-quoted string, containing logistic regression -log10(p) for the biallelic SNPs with column names 'ps' for SNP positions/IDs and 'negLog10' for -log10(p).

`logreg.tri.tetra`
: A file name specified by either a variable of mode character, or a double-quoted string, containing logistic regression -log10(p) for the tri and tetra allelic SNPs with column names 'ps' for SNP positions/IDs and 'negLog10' for -log10(p).

`var.matrix`
: A vector of file names specified by double-quoted strings. The files should contain presence absence matrices, with rows being variants (of 0s and 1s), and columns being samples. Column headers must contain 'ps' for variant positions/IDs with the others being the sample names.

`logreg.var`
: A vector of file names specified by double-quoted strings, of files containing logistic regression -log10(p-value) results for the presence absence matrices. Column names must contain 'ps' for variant positions/IDs and 'negLog10' for the -log10(p).

`lmm.var`
: A vector of file names specified by double-quoted strings, of files containing GEMMA results for the presence absence matrices.

`cutOffCor`
: Correlation cut-off for assigning and colouring variants by Principal Components (Default = 0, variants are coloured by the PC they are most correlated with).

`run.lmm`
: Whether to run GEMMA (Default = TRUE).

`maf`
: Minor allele frequency for GEMMA (Default = 0, all varaints are tested).

`relmatrix`
: A file name specified by either a variable of mode character, or a double-quoted string of a file containing the GEMMA relatedness matrix of the samples created from biallelic SNPs. The individual ordering must be in the same order as the column names in argument 'gen'.

`lognull`
: The log likelihood under the null from GEMMA.

`lambda`
: Lambda from GEMMA.

`output.dir`
: Output file directory.

`creatingAllPlots`
: Whether to create all bugwas plots. Default = TRUE.

`allBranchAndPCCor`
: Whether or not to retreive correlation matrix between branches and PCs. Default = FALSE.

Examples

```
lin_loc()
## An example of running lin_loc with the minimum required inputs
## Assuming gemma is installed in the present working directory
gen <- system.file("extdata", "gen.txt", package = "bugwas")
pheno <- system.file("extdata", "pheno.txt", package = "bugwas")
phylo <- system.file("extdata", "tree.txt", package = "bugwas")
prefix <- "test_bugwas"
gem.path <- "./gemma"
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo, prefix = prefix, gem.path = gem.path)
```

lin_loc *Lineage and locus tests for bacterial GWAS*

Description

This function tests for locus effects using GEMMA and lineage effects using a bayesian wald test for haploid data

Usage

```
lin_loc(gen = NULL, pheno = NULL, phylo = NULL, prefix = NULL,
  gem.path = NULL, pcs = NULL, lmm.bi = NULL, lmm.tri.tetra = NULL,
  logreg.bi = NULL, logreg.tri.tetra = NULL, var.matrix = NULL,
  logreg.var = NULL, lmm.var = NULL, cutOffCor = 0, run.lmm = TRUE,
  maf = 0, relmatrix = NULL, lognull = NULL, lambda = NULL,
  output.dir = getwd(), creatingAllPlots = TRUE,
  allBranchAndPCCor = FALSE, runTriTetrallelic = TRUE)
```

Arguments

`gen` A file name specified by either a variable of mode character, or a double-quoted string, containing imputed haploid SNP data. Rows are SNPs, and columns are samples, with the first column being SNP positions. Column headers must contain 'ps' for the SNP positions with the others being the sample names. This must contain biallelic SNPs, but can also contain tri and tetra-allelic SNPs. Required argument.

`pheno` A file name specified by either a variable of mode character, or a double-quoted string, containing a column of sample names with header 'ID' and a column of the binary phenotype (coded by 0s and 1s) with column header 'pheno'. Required argument.

`phylo` A file name specified by either a variable of mode character, or a double-quoted string, containing a phylogeny of the samples, with the same names matching with arguments gen and pheno. Required argument.

`prefix` Output file prefix. Required argument.

`gem.path` A file path specified by either a variable of mode character, or a double-quoted string. gem.path is the file path to the software GEMMA (version >= ?). Required argument.

`pcs` A file name specified by either a variable of mode character, or a double-quoted string, containing the principle components of the data. Column names should be 'PC1' to 'PCn' and row names should be the sample names.

`lmm.bi` A file name specified by either a variable of mode character, or a double-quoted string, containing GEMMA results (ending '.assoc.txt') for the biallelic SNPs in argument 'gen'.

`lmm.tri.tetra` A file name specified by either a variable of mode character, or a double-quoted string, containing GEMMA results for the tri and tetra allelic SNPs in argument 'gen'. This must contain column headers 'ps' for SNP positions/IDs, 'pvals' for p-values and 'negLog10' for -log10(p).

`logreg.bi`
A file name specified by either a variable of mode character, or a double-quoted string, containing logistic regression -log10(p) for the biallelic SNPs with column names ’ps’ for SNP positions/IDs and ’negLog10’ for -log10(p).

`logreg.tri.tetra`
A file name specified by either a variable of mode character, or a double-quoted string, containing logistic regression -log10(p) for the tri and tetra allelic SNPs with column names ’ps’ for SNP positions/IDs and ’negLog10’ for -log10(p).

`var.matrix`
A vector of file names specified by double-quoted strings. The files should contain presence absence matrices, with rows being variants (of 0s and 1s), and columns being samples. Column headers must contain ’ps’ for variant positions/IDs with the others being the sample names.

`logreg.var`
A vector of file names specified by double-quoted strings, of files containing logistic regression -log10(p-value) results for the presence absence matrices. Column names must contain ’ps’ for variant positions/IDs and ’negLog10’ for the -log10(p).

`lmm.var`
A vector of file names specified by double-quoted strings, of files containing GEMMA results for the presence absence matrices.

`cutOffCor`
Correlation cut-off for assigning and colouring variants by Principal Components (Default = 0, variants are coloured by the PC they are most correlated with).

`run.lmm`
Whether to run GEMMA (Default = TRUE).

`maf`
Minor allele frequency for GEMMA (Default = 0, all varaints are tested).

`relmatrix`
A file name specified by either a variable of mode character, or a double-quoted string of a file containing the GEMMA relatedness matrix of the samples created from biallelic SNPs. The individual ordering must be in the same order as the column names in argument ’gen’.

`lognull`
The log likelihood under the null from GEMMA.

`lambda`
Lambda from GEMMA.

`output.dir`
Output file directory.

`creatingAllPlots`
Whether to create all bugwas plots. Default = TRUE.

`allBranchAndPCCor`
Whether or not to retreive correlation matrix between branches and PCs. Default = FALSE.

Examples

```
lin_loc()
## An example of running lin_loc with the minimum required inputs
## Assuming gemma is installed in the present working directory
gen <- system.file("extdata", "gen.txt", package = "bugwas")
pheno <- system.file("extdata", "pheno.txt", package = "bugwas")
phylo <- system.file("extdata", "tree.txt", package = "bugwas")
prefix <- "test_bugwas"
gem.path <- "./gemma"
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo, prefix = prefix, gem.path = gem.path)
```

`logregVsLMM` *Plot of the P-values of logistic regression versus those of LMM*

Description

This function generates the plot of the P-values of logistic regression versus those of SNP GWAS.

Usage

```
logregVsLMM(config, biallelic, triallelic, colourPalette = NULL)
```

Arguments

`config`	A list called ’config’ created from the lin_loc function. It is a required input.
`biallelic`	A list called ’biallelic’ created from the lin_loc function. It is a required input.
`triallelic`	A list called ’triallelic’ created from the lin_loc function. It is a required input.
`colourPalette`	A vector of colours colour the significant principal components identified by the Bayesian Wald test (see testGenomeWidePCs). If this is NULL then colours are chosen from a default colour palette.

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
logregVsLMM(biallelic = data$biallelic, triallelic = data$triallelic, config = data$config)
```

`pcLoadingPlot` *Plot PC loadings of all SNPs.*

Description

This function plots the loadings of all SNPs for each significant principle component identified by the Bayesian Wald test.

Usage

```
pcLoadingPlot(config, biallelic)
```

Arguments

`config`	A list called ’config’ created from the lin_loc function. It is a required input.
`biallelic`	A list called ’biallelic’ created from the lin_loc function. It is a required input.

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
pcLoadingPlot(config = data$config, biallelic = data$biallelic)
```

plotIndividualBy2PCs *Plot of the sample on the first two principal components*

Description

This function generates a plot of the sample on the first two principal components.

Usage

```
plotIndividualBy2PCs(biallelic = NULL, config = NULL)
```

Arguments

biallelic	A list called 'biallelic' created from the lin_loc function. It is a required input.
config	A list called 'config' created from the lin_loc function. It is a required input.

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
plotIndividualBy2PCs(biallelic = data$biallelic, config = data$config)
```

snpManhattanPlot *Generates Manhattan plots for a SNP GWAS*

Description

This function generates the Manhattan plot(s) for a SNP GWAS.

Usage

```
snpManhattanPlot(biallelic, triallelic, config, colourPalette = NULL)
```

Arguments

biallelic	A list called 'biallelic' created from the lin_loc function. It is a required input.
triallelic	A list called 'triallelic' created from the lin_loc function. It is a required input.
config	A list called 'config' created from the lin_loc function. It is a required input.
colourPalette	A vector of colours colour the significant principal components identified by the Bayesian Wald test (see testGenomeWidePCs). If this is NULL then colours are chosen from a default colour palette.

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
snpManhattanPlot(biallelic = data$biallelic, triallelic = data$triallelic,
 config = data$config)
```

snpPCManhattanPlot *Plot the a Manhattan plot organised by the signficance of PCs.*

Description

This function generates a Manhattan plot organised by the signficance of the principal components.

Usage

```
snpPCManhattanPlot(config, biallelic, triallelic, colourPalette = NULL)
```

Arguments

`config`	A list called 'config' created from the lin_loc function. It is a required input.
`biallelic`	A list called 'biallelic' created from the lin_loc function. It is a required input.
`triallelic`	A list called 'triallelic' created from the lin_loc function. It is a required input.
`colourPalette`	A vector of colours colour the significant principal components identified by the Bayesian Wald test (see testGenomeWidePCs). If this is NULL then colours are chosen from a default colour palette.
`p.genomewidepc`	A matrix of the significant principal component and their correlation with lineages. This is Bayesian Wald test results produced by the function testGenomeWidePCs. If this is NULL then testGenomeWidePCs is called to generate the required test results.

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
snpPCManhattanPlot(biallelic = data$biallelic,  triallelic = data$triallelic,
 config = data$config)
```

testGenomeWidePCs *Testing the genome-wide principal components*

Description

This function performs the genome-wide principal components.

Usage

```
testGenomeWidePCs(config = NULL, biallelic = NULL)
```

Arguments

`config`	A list called 'config' created from the bugwas function.
`biallelic`	A list called 'biallelic' created from the bugwas function.

Value

The p-value of the Bayesian Wald test for the genome-wide effect of principal components.

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
testGenomeWidePCs(config = data$config, biallelic = data$biallelic)
```

`testGenomeWidePCsAll` *Testing the genome-wide principal components for all PCs.*

Description

This function identifies the non-genome-wide principal components.

Usage

```
testGenomeWidePCsAll(config = NULL, biallelic = NULL)
```

Arguments

`config`	A list called 'config' created from the bugwas function
`biallelic`	A list called 'biallelic' created from the bugwas function

Value

The p-value of the Bayesian Wald test for the genome-wide effect of all principal components.

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
testGenomeWidePCsAll(config = data$config, biallelic = data$biallelic)
```

`trueAndPredPhenoOnTreePlot` *Generates all plots.*

Description

This function that plots the true and predicted phenotypes on a tree.

Usage

```
trueAndPredPhenoOnTreePlot(config, biallelic, treeInfo, p.genomewidepc = NULL,
  colourPalette = NULL)
```

Arguments

`config`	A list called 'config' created from the lin_loc function. It is a required input.
`biallelic`	A list called 'biallelic' created from the lin_loc function. It is a required input.
`treeInfo`	A list called 'treeInfo' created from the lin_loc function. It is a required input.
`p.genomewidepc`	A matrix of the significant principal component and their correlation with lineages. This is Bayesian Wald test results produced by the function testGenomeWidePCs. If this is NULL then testGenomeWidePCs is called to generate the required test results.
`colourPalette`	A vector of colours colour the significant principal components identified by the Bayesian Wald test (see testGenomeWidePCs). If this is NULL then colours are chosen from a default colour palette.

Examples

```
data <- lin_loc(gen = gen, pheno = pheno, phylo = phylo,
 prefix = prefix, gem.path = gem.path)
trueAndPredPhenoOnTreePlot(biallelic = data$biallelic, treeInfo = data$treeInfo,
 config = data$config)
```

Index