

# Package ‘bugwas’

October 21, 2015

**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

## R topics documented:

all_plots . . . . .	1
bayesianWaldTestPCsBarplot . . . . .	2
genVarPlots . . . . .	3
lin_loc . . . . .	3
logregVsLMM . . . . .	5
pcLoadingPlot . . . . .	6
plotIndividualBy2PCs . . . . .	6
snpManhattanPlot . . . . .	7
snpPCManhattanPlot . . . . .	7
testGenomeWidePCs . . . . .	8
trueAndPredPhenoOnTreePlot . . . . .	8
<b>Index</b>	<b>9</b>

---

all_plots	<i>Generates all plots.</i>
-----------	-----------------------------

---

## 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 bugwas function
triallelic	A list called 'triallelic' created from the bugwas function
genVars	A list called 'genVars' created from the bugwas function
treeInfo	A list called 'treeInfo' created from the bugwas function
config	A list called 'config' created from the bugwas function

**Examples**

```
all_plots()
```

---

```
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 bugwas function. It is a required input.
biallelic	A list called 'biallelic' created from the bugwas function. It is a required input.
treeInfo	A list called 'treeInfo' created from the bugwas 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**

```
testGenomeWidePCs(config = config, biallelic = 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 bugwas function. It is a required input.
biallelic	A list called 'biallelic' created from the bugwas function. It is a required input.
config	A list called 'config' created from the bugwas 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**

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

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)
```

**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.
phyl	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 variants 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.

### 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 bugwas function. It is a required input.

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

triallelic       A list called 'triallelic' created from the bugwas 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

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

---

pcLoadingPlot	<i>Plot PC loadings of all SNPs.</i>
---------------	--------------------------------------

---

### 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 bugwas function. It is a required input.
biallelic	A list called 'biallelic' created from the bugwas function. It is a required input.

### Examples

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

---

plotIndividualBy2PCs	<i>Plot of the sample on the first two principal components</i>
----------------------	---

---

### 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 bugwas function. It is a required input.
config	A list called 'config' created from the bugwas function. It is a required input.

### Examples

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

---

snpManhattanPlot	<i>Generates Manhattan plots for a SNP GWAS</i>
------------------	---

---

**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 bugwas function. It is a required input.
triallelic	A list called 'triallelic' created from the bugwas function. It is a required input.
config	A list called 'config' created from the bugwas 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**

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

---

snpPCManhattanPlot	<i>Plot the a Manhattan plot organised by the significance of PCs.</i>
--------------------	--

---

**Description**

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

**Usage**

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

**Arguments**

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

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

---

testGenomeWidePCs	<i>Testing the genome-wide principal components</i>
-------------------	---

---

### 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

### Examples

```
testGenomeWidePCs(config, biallelic)
```

---

trueAndPredPhenoOnTreePlot	<i>Generates all plots.</i>
----------------------------	-----------------------------

---

### 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 bugwas function. It is a required input.
biallelic	A list called 'biallelic' created from the bugwas function. It is a required input.
treeInfo	A list called 'treeInfo' created from the bugwas 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

```
trueAndPredPhenoOnTreePlot(config = config, biallelic = biallelic, treeInfo = treeInfo)
```



# Index

- \*Topic **Bayesian**
  - testGenomeWidePCs, [8](#)
- \*Topic **GEMMA**
  - lin\_loc, [3](#)
- \*Topic **GWAS**
  - lin\_loc, [3](#)
- \*Topic **Manhattan**
  - genVarPlots, [3](#)
  - snpManhattanPlot, [7](#)
  - snpPCManhattanPlot, [7](#)
- \*Topic **PCA**
  - snpPCManhattanPlot, [7](#)
- \*Topic **Reduced**
  - plotIndividualBy2PCs, [6](#)
- \*Topic **SNP**
  - logregVsLMM, [5](#)
  - snpManhattanPlot, [7](#)
  - snpPCManhattanPlot, [7](#)
- \*Topic **Scatter**
  - logregVsLMM, [5](#)
- \*Topic **Wald**
  - testGenomeWidePCs, [8](#)
- \*Topic **bacteria**
  - lin\_loc, [3](#)
- \*Topic **components**
  - testGenomeWidePCs, [8](#)
- \*Topic **lineage**
  - lin\_loc, [3](#)
- \*Topic **locus**
  - lin\_loc, [3](#)
- \*Topic **plot**
  - all\_plots, [1](#)
  - bayesianWaldTestPCsBarplot, [2](#)
  - genVarPlots, [3](#)
  - logregVsLMM, [5](#)
  - pcLoadingPlot, [6](#)
  - plotIndividualBy2PCs, [6](#)
  - snpManhattanPlot, [7](#)
  - snpPCManhattanPlot, [7](#)
  - trueAndPredPhenoOnTreePlot, [8](#)
- \*Topic **principal**
  - testGenomeWidePCs, [8](#)
- \*Topic **space**
  - plotIndividualBy2PCs, [6](#)
- \*Topic **test,**
  - testGenomeWidePCs, [8](#)
- \*Topic **wald**
  - lin\_loc, [3](#)
- all\_plots, [1](#)
- bayesianWaldTestPCsBarplot, [2](#)
- genVarPlots, [3](#)
- lin\_loc, [3](#)
- logregVsLMM, [5](#)
- pcLoadingPlot, [6](#)
- plotIndividualBy2PCs, [6](#)
- snpManhattanPlot, [7](#)
- snpPCManhattanPlot, [7](#)
- testGenomeWidePCs, [8](#)
- trueAndPredPhenoOnTreePlot, [8](#)