

G4G

*GLM for GWAS package
For R studio*



USER MANUAL

Developed by: Cristina Ocana & Shelby Tisinai

March 2021

Table of contents

Description of functions	3
Principal Components Analysis	3
Description	3
Usage	3
Arguments	3
Apply a Function over a List or Vector	4
Description	4
Usage	4
Arguments	4
Sample Ranks	5
Description	5
Usage	5
Arguments	5
Generic X-Y Plotting	5
Description	5
Usage	5
Arguments	5
User guidance	6
FAQs	9

1. Description of functions

`G4G <- function(pheno, geno, covar, PCA.m = 3, map = NULL)`

The functions described below have been included in the background of G4G to produce a GWAS result. Most of these functions do not need to be altered in order to produce GWAS but can be customized according to the each function's arguments. The G4G function returns Manhattan and QQ plots by using the 'plot' function, but alternative functions, such as 'qq_plot' can be used.

a) *Prcomp* (default number of PCs included in G4G package is 3)

Principal Components Analysis

Description

Performs a principal components analysis on the given data matrix and returns the results as an object of class `prcomp`.

Usage

```
prcomp(x, ...)  
  
## S3 method for class 'formula'  
prcomp(formula, data = NULL, subset, na.action, ...)  
  
## Default S3 method:  
prcomp(x, retx = TRUE, center = TRUE, scale. = FALSE,  
       tol = NULL, rank. = NULL, ...)  
  
## S3 method for class 'prcomp'  
predict(object, newdata, ...)
```

Arguments

<code>formula</code>	a formula with no response variable, referring only to numeric variables.
<code>data</code>	an optional data frame (or similar: see model.frame) containing the variables in the formula <code>formula</code> . By default the variables are taken from <code>environment(formula)</code> .

subset	an optional vector used to select rows (observations) of the data matrix <code>x</code> .
<code>x</code>	a numeric or complex matrix (or data frame) which provides the data for the principal components analysis.
rank.	optionally, a number specifying the maximal rank, i.e., maximal number of principal components to be used. Can be set as alternative or in addition to <code>tol</code> , useful notably when the desired rank is considerably smaller than the dimensions of the matrix.

b) Sapply

Apply a Function over a List or Vector

Description

`lapply` returns a list of the same length as `x`, each element of which is the result of applying `FUN` to the corresponding element of `x`.

`sapply` is a user-friendly version and wrapper of `lapply` by default returning a vector, matrix or, if `simplify = "array"`, an array if appropriate, by applying `simplify2array().sapply(x, f, simplify = FALSE, USE.NAMES = FALSE)` is the same as `lapply(x, f)`.

Usage

```
sapply(X, FUN, ..., simplify = TRUE, USE.NAMES = TRUE)
```

Arguments

<code>x</code>	a vector (atomic or list) or an expression object. Other objects (including classed objects) will be coerced by <code>base::as.list</code> .
<code>FUN</code>	the function to be applied to each element of <code>x</code> : see ‘Details’. In the case of functions like <code>+</code> , <code>%*%</code> , the function name must be backquoted or quoted.
<code>...</code>	optional arguments to <code>FUN</code> .
<code>USE.NAMES</code>	logical; if <code>TRUE</code> and if <code>x</code> is character, use <code>x</code> as names for the result unless it had names already. Since this argument follows <code>...</code> its name cannot be abbreviated.
<code>expr</code>	the expression (a language object , usually a call) to evaluate repeatedly.
<code>x</code>	a list, typically returned from <code>lapply()</code> .

c) Rank

Sample Ranks

Description

Returns the sample ranks of the values in a vector. Ties (i.e., equal values) and missing values can be handled in several ways.

Usage

```
rank(x, na.last = TRUE,  
     ties.method = c("average", "first", "last", "random", "max", "min"))
```

Arguments

<code>x</code>	a numeric, complex, character or logical vector.
<code>na.last</code>	for controlling the treatment of NAs . If <code>TRUE</code> , missing values in the data are put last; if <code>FALSE</code> , they are put first; if <code>NA</code> , they are removed; if <code>"keep"</code> they are kept with rank <code>NA</code> .
<code>ties.method</code>	a character string specifying how ties are treated,

d) Plot (used for Manhattan plot and QQ plot)

Generic X-Y Plotting

Description

Generic function for plotting of `R` objects.

For simple scatter plots, [plot.default](#) will be used. However, there are `plot` methods for many `R` objects, including [functions](#), [data.frames](#), [density](#) objects, etc. Use `methods(plot)` and the documentation for these. Most of these methods are implemented using traditional graphics (the **graphics** package), but this is not mandatory.

For more details about graphical parameter arguments used by traditional graphics, see [par](#).

Usage

```
plot(x, y, ...)
```

Arguments

<code>x</code>	the coordinates of points in the plot. Alternatively, a single plotting structure, function or <i>any R object with a plot method</i> can be provided.
----------------	--

`y` the y coordinates of points in the plot, *optional* if `x` is an appropriate structure.

- Arguments to be passed to methods, such as [graphical parameters](#) (see [par](#)). Many
- methods will accept the following arguments:

`type`

what type of plot should be drawn. Possible types are

- "p" for **p**oints,
- "l" for **l**ines,
- "b" for **b**oth,
- "c" for the lines part alone of "b",
- "o" for both 'overplotted',
- "h" for 'histogram' like (or 'high-density') vertical lines,
- "s" for stair **s**teps,
- "S" for other **s**teps, see 'Details' below,
- "n" for no plotting.

All other `types` give a warning or an error; using, e.g., `type = "punkte"` being equivalent to `type = "p"` for S compatibility. Note that some methods, e.g. [plot.factor](#), do not accept this.

`main`

an overall title for the plot: see [title](#).

`sub`

a sub title for the plot: see [title](#).

`xlab`

a title for the x axis: see [title](#).

`ylab`

a title for the y axis: see [title](#).

`asp`

the y/x aspect ratio, see [plot.window](#).

2. User guidance

1) Getting started

In order to use the G4G package, you need access to R studio and R, which can be downloaded from <https://www.r-project.org/>. Once installed, code for the G4G package can be found at GitHub in the following link:

<https://github.com/shibstish/G4G>. Below is the necessary code to install the package in R.

```
install.packages("devtools")
library(devtools)
install_github("shibstish/G4G")
```

2) Required and optional inputs

Table 1: Summary of required (R) and optional (O) input data for G4G package

Name	Type of data	Format	Dimensions
pheno (R)	Phenotypes - measurements	Quantitative	n by 1
geno (R)	SNP data	0,1,2	n by m
map (R)	Genetic map (SNP name, chromosome number and position in genome)	SNP names can be characters but chr and position need to be numeric	m by 3
covar (O)	Covariates	Numeric	n by t

n= number of individuals, m= number of markers, t= number of covariates

3) Example

The G4G package includes demo data to allow the user to familiarize with its functions. Below are examples of input data for the G4G package: phenotype, genetic map, genotype and covariates (from left to right).

	Taxa	Obs
1	33-16	-1.273103730
2	38-11	0.551527097
3	4226	-0.254927311
4	4722	-6.122964349
5	A188	-2.593982541

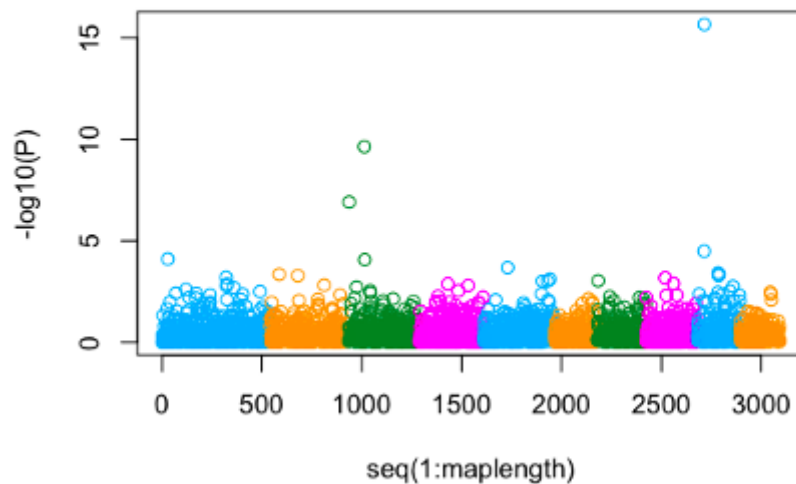
	SNP	Chromosome	Position
1	PZB00859.1	1	157104
2	PZA01271.1	1	1947984
3	PZA03613.2	1	2914066
4	PZA03613.1	1	2914171
5	PZA03614.2	1	2915078

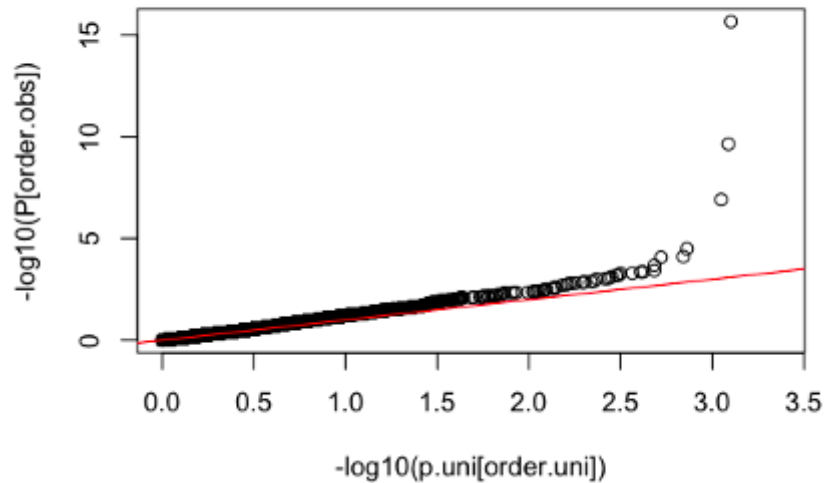
	PZB00859.1	PZA01271.1	PZA03613.2	PZA03613.1	PZA03614.2	PZA03614.1	PZA00258.3
1	2	0	0	2	2	2	2
2	2	2	0	2	2	2	0
3	2	0	0	2	2	2	0
4	2	2	0	2	2	2	1
5	0	0	0	2	2	2	0

	FactorA	FactorB
1	2.53133090	5.501464
2	2.63386050	4.655691
3	1.89069549	6.136883
4	1.85603497	7.841858
5	2.55262921	5.409450

The required inputs need to have the dimensions and format stated above and look similar to the demo dataset. The G4G package will output a list of p-values (not shown here), values for the three first principal components (default number of PCs), a manhattan plot and QQ plot.

	PC1	PC2	PC3
1	1.67807746	-4.937338203	1.002927904
2	-1.60217489	-4.732279005	-0.688618712
3	-0.89995172	-6.218609021	2.273765498
4	2.13344773	-6.287930060	5.683716125
5	0.63023723	-4.894741571	0.818900120





3. FAQs

1. What format do I require as an input to utilise G4G?

Refer to section 2b for a summary of the formats and dimensions that input data requires.

2. Are other types of SNP data formats allowed in G4G?

The first version of G4G only accepts numeric SNPs (0, 1, 2).

3. Can I turn off the plotting functions?

Unfortunately the default option does not permit to switch off plotting but we'll be working on that for future versions.

4. Are updated versions of the package installed automatically?

No, both R studio and packages need to be updated manually.