User Manual and Tutorial for GWAStest

Lance Merrick & Samuel Prather

March 30, 2020

Figure 1:

Table of contents

Introduction

A common problem associated with preforming Genome Wide Association Studies (GWAS) is the abundance of false positive and false negative associated with population structure. One way of dealing with this issue is to first calculate Principal Components (PC) and then use those PCs as to account for population structure when running the GWAS. This method has shown to not only reduce false positives but also increase the power of the test. Here we present an R based package that can preform GWAS using PCA and user imputed covariate to calculate SNPs associated with a phenotype called "GWAStest". Using the freely availably R studio software this packaged is designed to quickly and efficiently calculate a P-value for every SNP phenotype association. The results from this package are including but not limited to a Manhattan plot, QQ-plot and table with every recorded p-value.

Getting started

First if you do not already have R and R studio installed on your computer head over to https://www.r-project.org/ and install the version appropriate for you machine. Once R and R studio is installed you will need to install the GWAStest package since this is a working package in it's early stages of development it's only available through Github. To download files off Github first download and load the library of the packaged "devtools" using the code below.

```
#code to install package from github
#install.packages("devtools")
library(devtools)
```

```
## Loading required package: usethis
```

Next using the code below download and install the package GWAStest from Github. The bottom two line of code in the chunk below make sure the dependencies GWAStest relies on are also downloaded and installed.

```
#install package
#install_github("stp4freedom/HW4_GLM_GWAS", force =TRUE)
library(GWAStest)#package name
#Load dependencies
if (!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
pacman::p_load(ggplot2,knitr,gridExtra,kableExtra)
```

In order to have an effective tutorial you'll need some data to play with, this code below downloads and loads into your environment data that for this tutorial.

```
if (!file.exists("GAPIT_Tutorial_Data.zip"))
{
  download.file("http://zzlab.net/GAPIT/GAPIT_Tutorial_Data.zip", destfile = "GAPIT_Tutorial_Data.zip")
  unzip("GAPIT_Tutorial_Data.zip")
}
download.file("http://zzlab.net/GAPIT/data/CROP545_Covariates.txt", destfile = "CROPS545_Covariates.txt
download.file("http://zzlab.net/GAPIT/data/CROP545_Phenotype.txt", destfile = "CROPS545_Phenotype.txt")
# Import the GAPIT demo data genotypes
gt_scan <- data.frame(read.table("GAPIT_Tutorial_Data/mdp_numeric.txt", header = T, stringsAsFactors = 
classes <- sapply(gt_scan, class)
genotypes <- data.frame(read.table("GAPIT_Tutorial_Data/mdp_numeric.txt", header = T, row.names = 1, co

GM <- read.table("GAPIT_Tutorial_Data/mdp_SNP_information.txt", header = T, stringsAsFactors = F, sep =
CV <- read.table("CROPS545_Covariates.txt", header = T, stringsAsFactors = F, sep = "\t")
phenotypes <- read.table("CROPS545_Phenotype.txt", header = T, stringsAsFactors = F, sep = "\t")
```

Required Inputes

There are three data components required to run GWAStest, the SNP data the phenotype(s) and a map showing the chromosome and positions of each SNP. The SNP data must be numeric coded as "0", "1" and "2" where the 0 stands for homozygous for parent A, 2 stands for homozygous for parent B and 1 is for heterozygous SNP calls. Below is an example showing the first 5 rows and columns of our tutorial SNP data.

```
genotypes[1:5,1:5]
```

```
##       PZB00859.1 PZA01271.1 PZA03613.2 PZA03613.1 PZA03614.2
## 33-16          2          0          0          2          2
## 38-11          2          2          0          2          2
## 4226           2          0          0          2          2
## 4722           2          2          0          2          2
## A188           0          0          0          2          2
```

The phenotype can and integers negative or positive that correspond to each individual. Note: Below I have the head of the phenotype printed with the taxa column still included, before running GWAStest you will need to remove the taxa column leaving only the phenotypic values for the function to work.

```
phenotypes[1:5,1:2]
```

```
##    Taxa        Obs
## 1 33-16 -1.2731037
## 2 38-11  0.5515271
## 3  4226 -0.2549273
## 4  4722 -6.1229643
## 5  A188 -2.5939825
```

The final piece of information needed for GWAStest of work is a genetic map of the SNPs that had the SNP name the chromosome it's on and the position on that chromosome. This genetic map is needed in order to produce the Manhattan plot of the results.

```
GM[1:5,1:3]
```

```
##          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
```

Optional Inputes

The optional imputes for GWAStest are the covariates which if used can help increases the power of the GWAS. The covariates mush be numerical integers much like the phenotype values. Below is an example from the tutorial data.

```
CV[1:5,1:3]
```

```
##    Taxa  FactorA  FactorB
## 1 33-16 2.531331 5.501464
## 2 38-11 2.633860 4.655691
## 3  4226 1.890695 6.136883
## 4  4722 1.856035 7.841858
## 5  A188 2.552629 5.409450
```

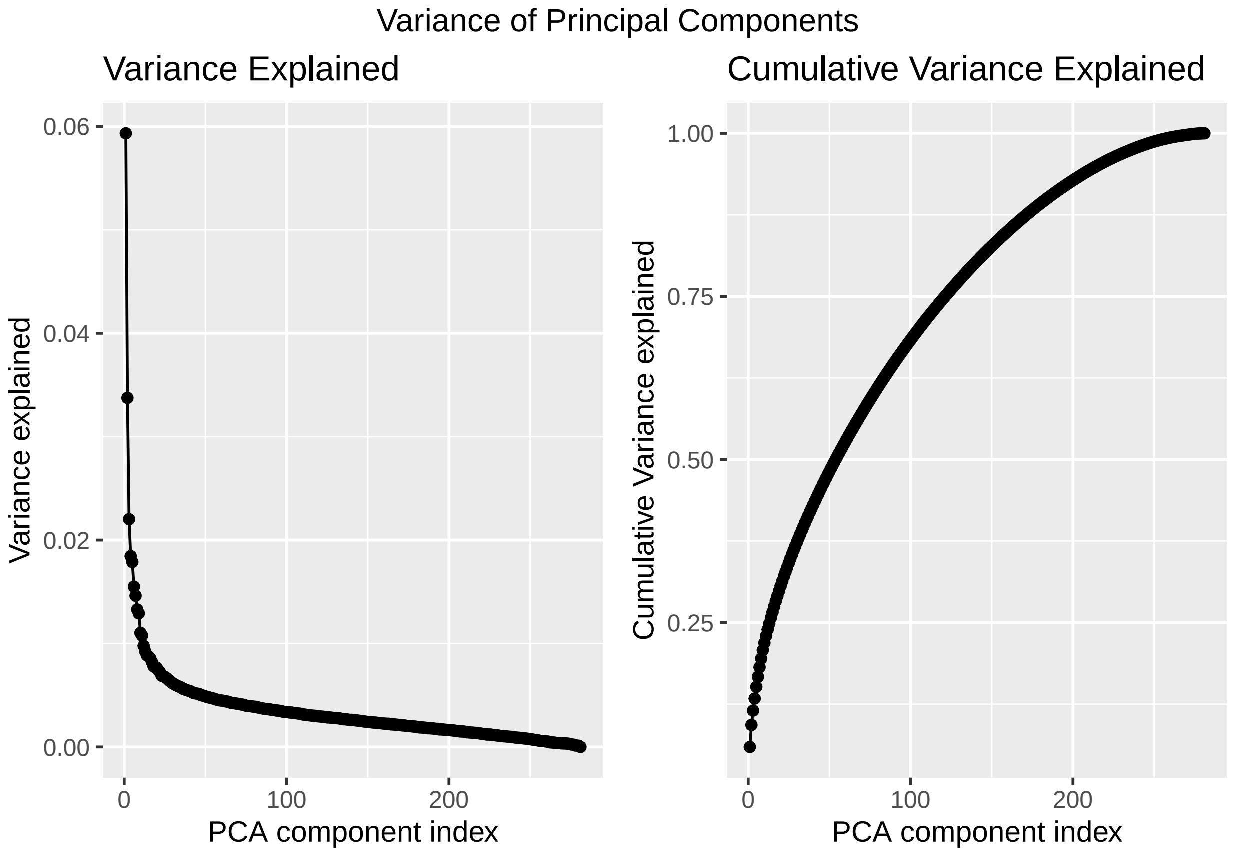
Other imputes include number of PCAs to use, with the default being set to three. The significance threshold (Cutoff) which can be set to the exact -log(10) of the p-value you want or the default of 0.05/number of SNPs. There are also some options you can use to suppress the plots automatically generated. Tp best show these options I will run a few examples below.

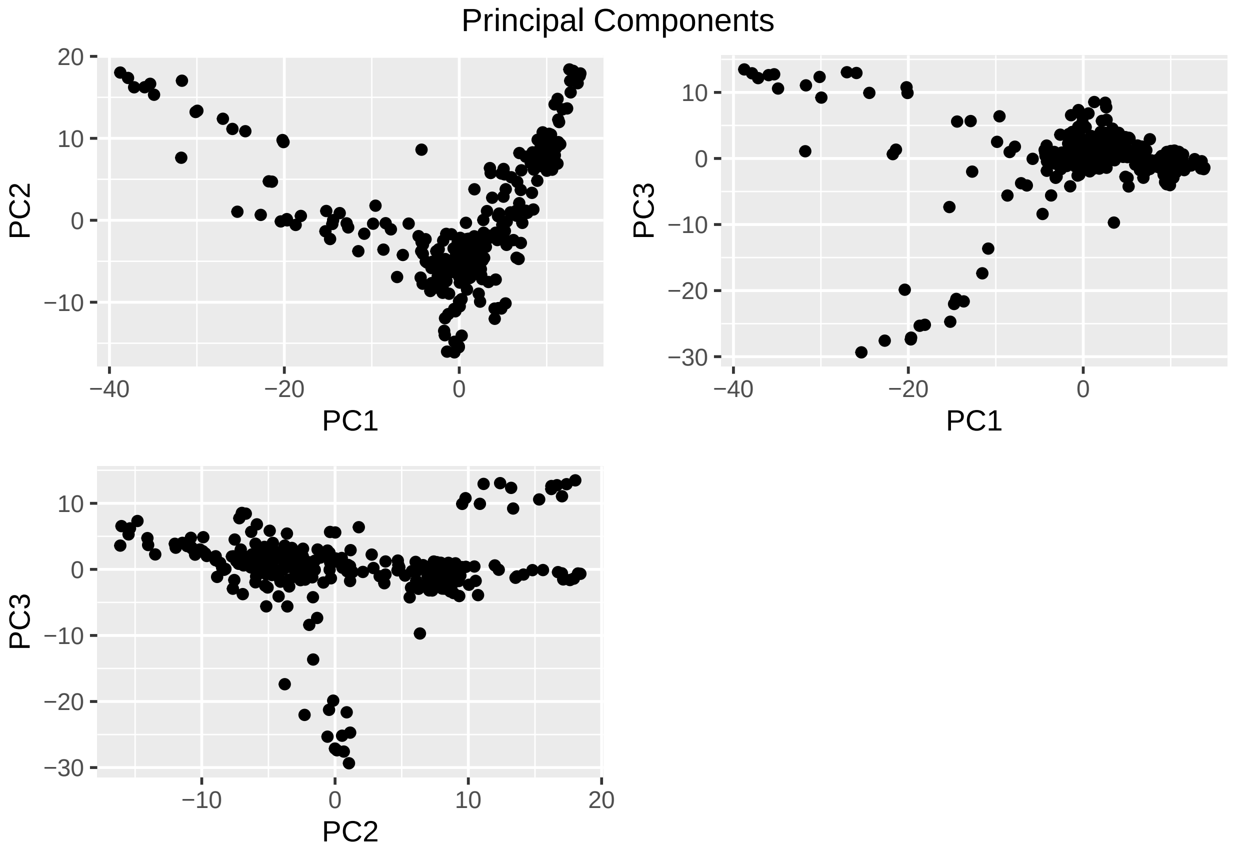
Outputs and examples

Outputs from GWAStest are table with information on the PCA, Graphs showing variance explained by the PC, scatter plots of the PC used, a Manhatten plot and a QQ-plot. A csv file with all SNPs and associated p-values and a table with the power and type-1 error are also optional outputs.

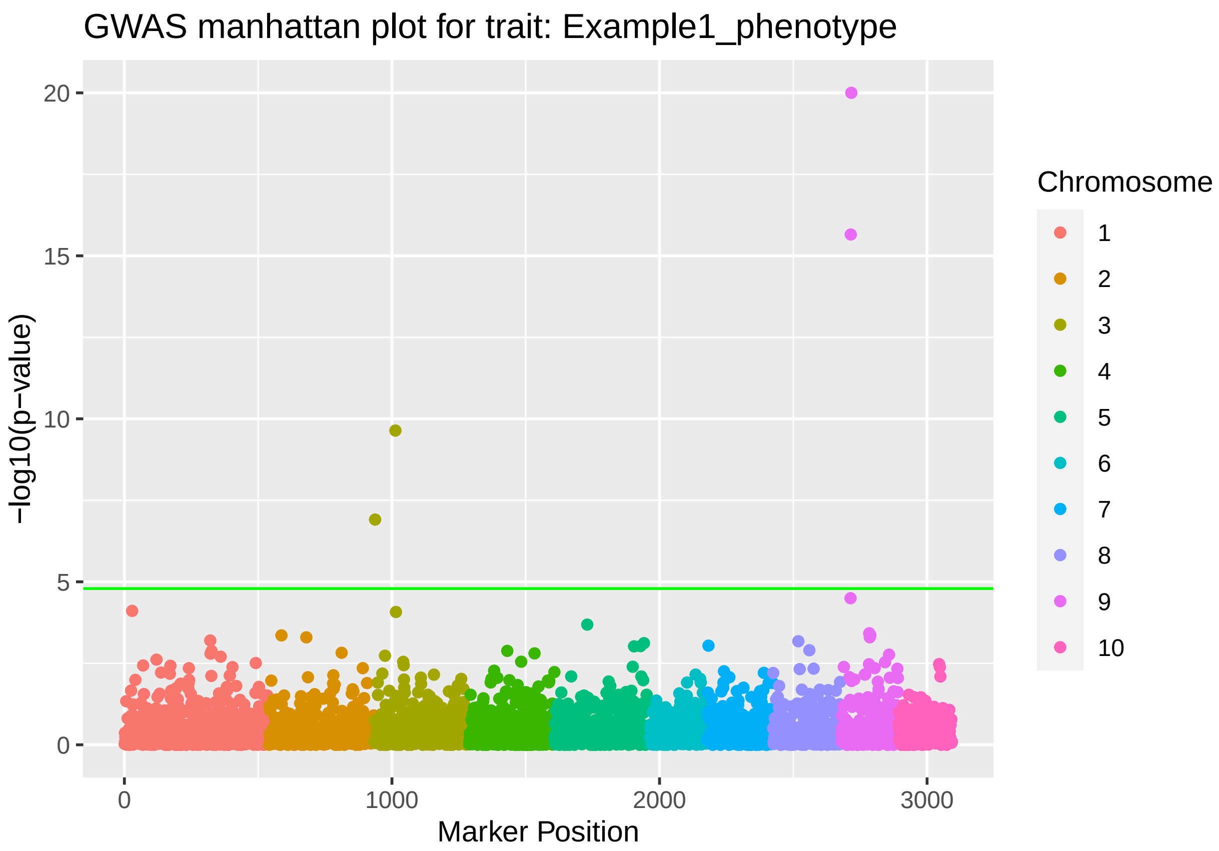
```
phenotypes_n_1=phenotypes[,2]#remember to remove the first column from the phenotype data before runnin
Example1=GWASapply(pheno=phenotypes_n_1, geno=genotypes,Cov=CV, GM=GM, PCA.M = 3, plots=TRUE, messages=
```

```
## [1] "GWASapply Starting"
## [1] "Principal Components have been calculated Successfully"
##
## \begin{tabular}{l|r|r|r|r|r|r|r|r|r|r}
## \hline
##   & PC1 & PC2 & PC3 & PC4 & PC5 & PC6 & PC7 & PC8 & PC9 & PC10\\
## \hline
## Standard deviation & 10.28 & 7.76 & 6.27 & 5.73 & 5.65 & 5.26 & 5.11 & 4.87 & 4.80 & 4.43\\
## \hline
## Proportion of Variance & 0.06 & 0.03 & 0.02 & 0.02 & 0.02 & 0.02 & 0.01 & 0.01 & 0.01 & 0.01\\
## \hline
## Cumulative Proportion & 0.06 & 0.09 & 0.12 & 0.13 & 0.15 & 0.17 & 0.18 & 0.19 & 0.21 & 0.22\\
## \hline
## \end{tabular}
```

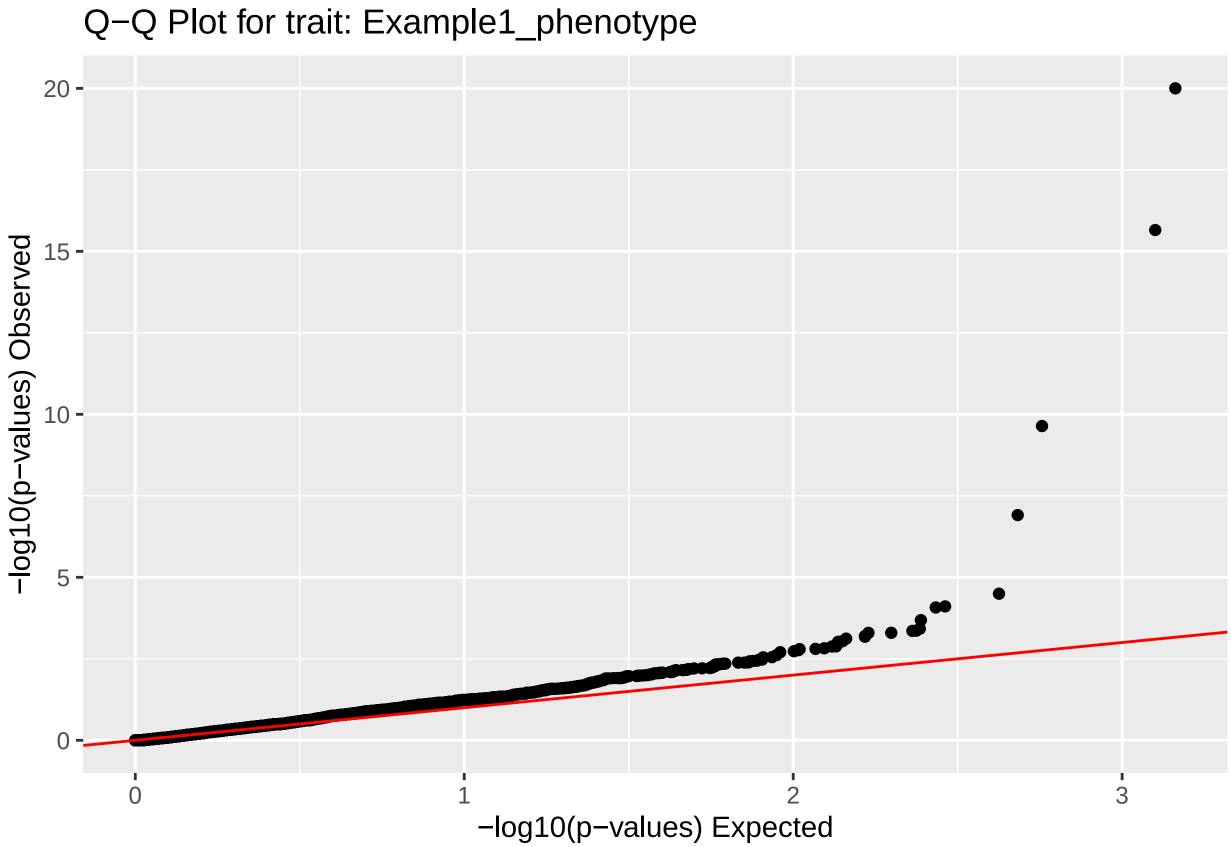




```
## [1] "Principal Components plots have been printed Successfully"
## [1] "The final cuttoff for a significant p-value is 1.61655350792111e-05"
## [1] "4 Significant SNPs were found"
```



```
## [1] "Manhattan Plot Printed without QTN"
```



```
## [1] "QQ-Plot Printed"
## [1] "GWASapply results have printed"
## [1] "GWASapply ran successfully and took 2.17 seconds"
```

Examples with know QTNs

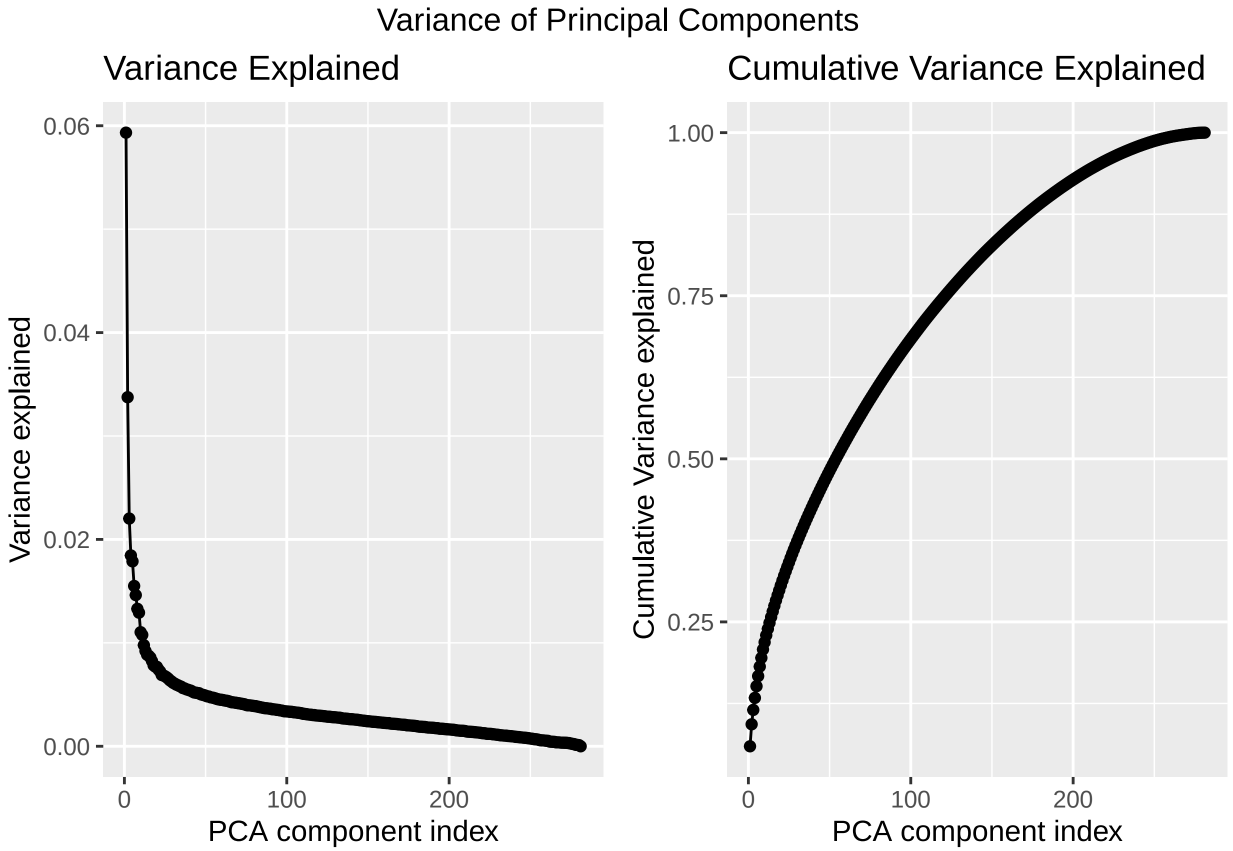
In order to fully demonstrate GWAStest functions, I am going to simulate some QTN effects using the function G2P. All G2Pis doing is generating effects and randomly assigning then to different SNPs. This will allow me to run GWAStest and be able to determine it's accuracy.

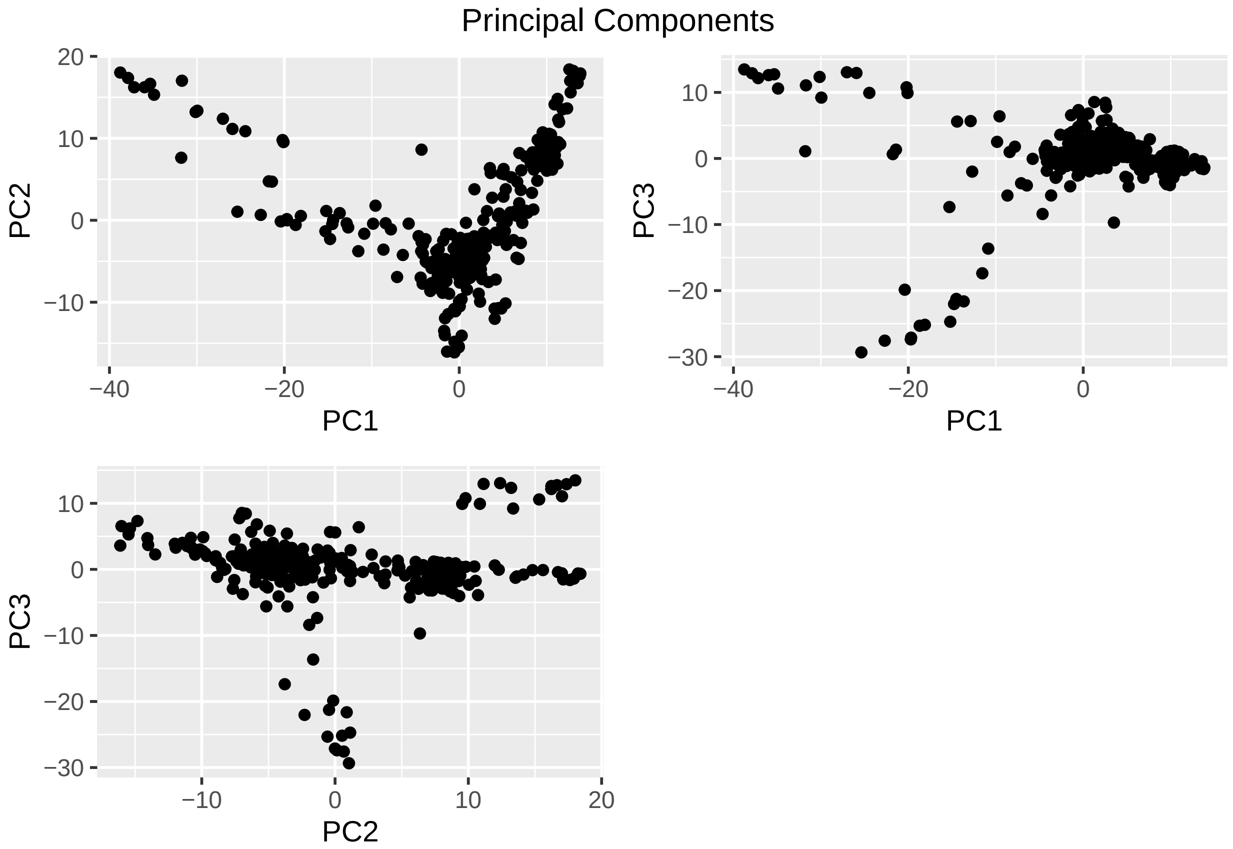
As we can see in the Manhattan plot and from the printed massages 8 significant SNPs were detected, the 4 of them in black are true positives 2 in blue are false positives and the two in red are false negatives. Note some of the lines are very close to each other and can be hard to see which is why it's nice to have messages = TRUE as that will prints the important findings into the R console.

```
source("http://www.zzlab.net/StaGen/2020/R/G2P.R")
NQTN = 6 #specify number of QTN
h2 = 0.75 #heritability
alpha = 0.6 #alpha value
distribution = "normal" #specify distribution
set.seed(66) # to get duplicatable results
mySim=G2P(genotypes, h2, alpha, NQTN, distribution)
Example2=GWASapply(pheno=mySim$y, geno=genotypes, GM=GM, PCA.M=3,QTN.position=mySim$QTN.position, plots
```

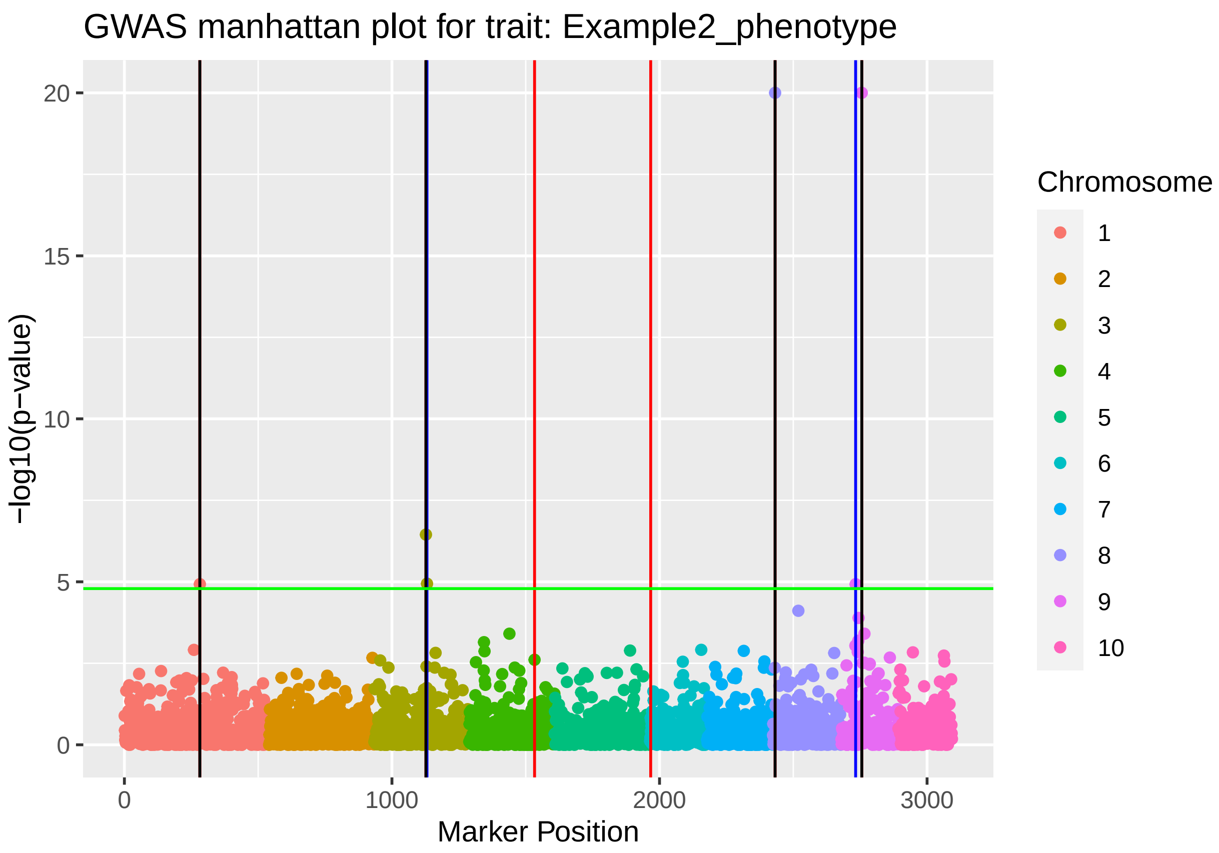
```
## [1] "GWASapply Starting"
## [1] "Principal Components have been calculated Successfully"
```

```
##
## \begin{tabular}{l|r|r|r|r|r|r|r|r|r|r}
## \hline
##   & PC1 & PC2 & PC3 & PC4 & PC5 & PC6 & PC7 & PC8 & PC9 & PC10\\
## \hline
## Standard deviation & 10.28 & 7.76 & 6.27 & 5.73 & 5.65 & 5.26 & 5.11 & 4.87 & 4.80 & 4.43\\
## \hline
## Proportion of Variance & 0.06 & 0.03 & 0.02 & 0.02 & 0.02 & 0.02 & 0.01 & 0.01 & 0.01 & 0.01\\
## \hline
## Cumulative Proportion & 0.06 & 0.09 & 0.12 & 0.13 & 0.15 & 0.17 & 0.18 & 0.19 & 0.21 & 0.22\\
## \hline
## \end{tabular}
```

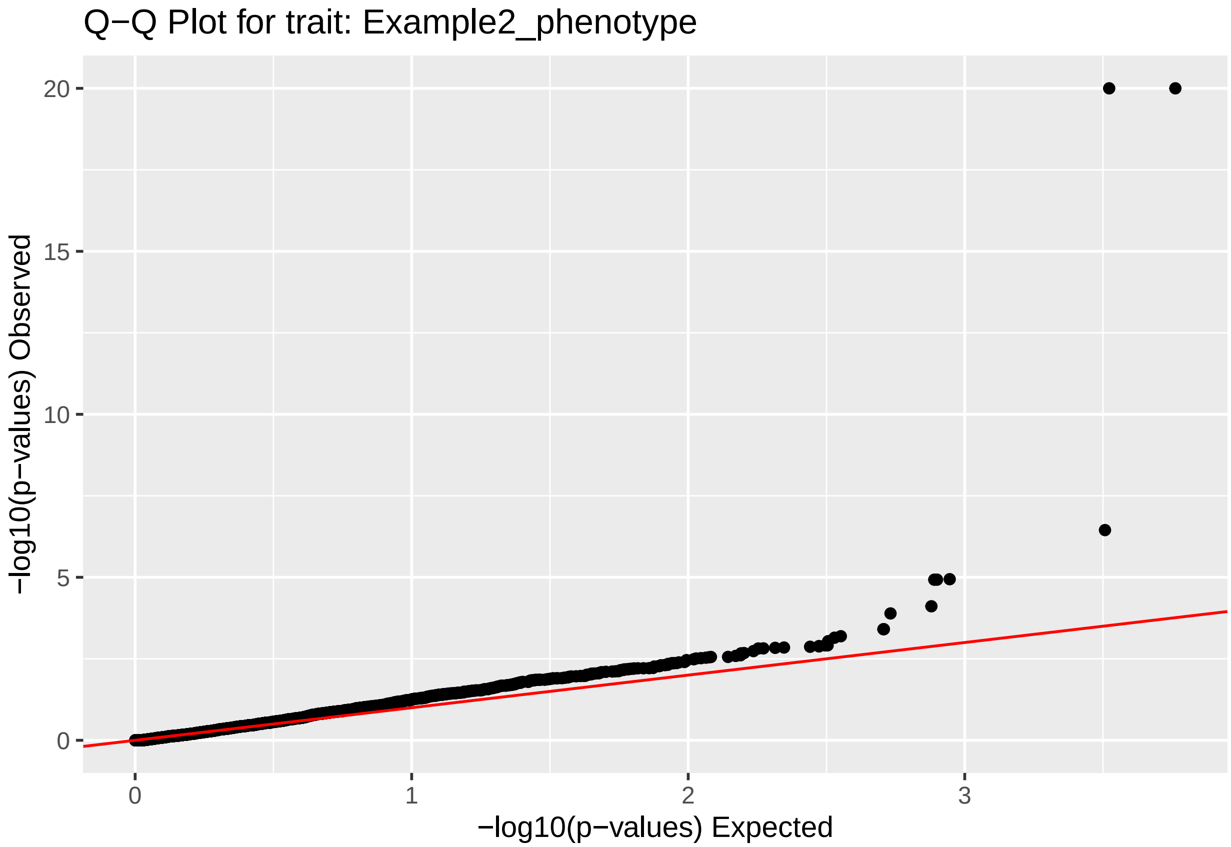




```
## [1] "Principal Components plots have been printed Successfully"
## [1] "The final cuttoff for a significant p-value is 1.61655350792111e-05"
## [1] "6 Significant SNPs were found"
## [1] "QTN's detected with 4 True Positives and 2 False Positives."
```



```
## [1] "Manhattan Plot Printed with QTNs and False and True Positives"
```



```
## [1] "QQ-Plot Printed"
## [1] "GWASapply results have printed"
## [1] "GWASapply ran successfully and took 1.83 seconds"
```

Quick run

If you want to reduce the output and speed up the run time you can run GWAStest with plots=FALSE which will still run the GWAS but does not output any plots

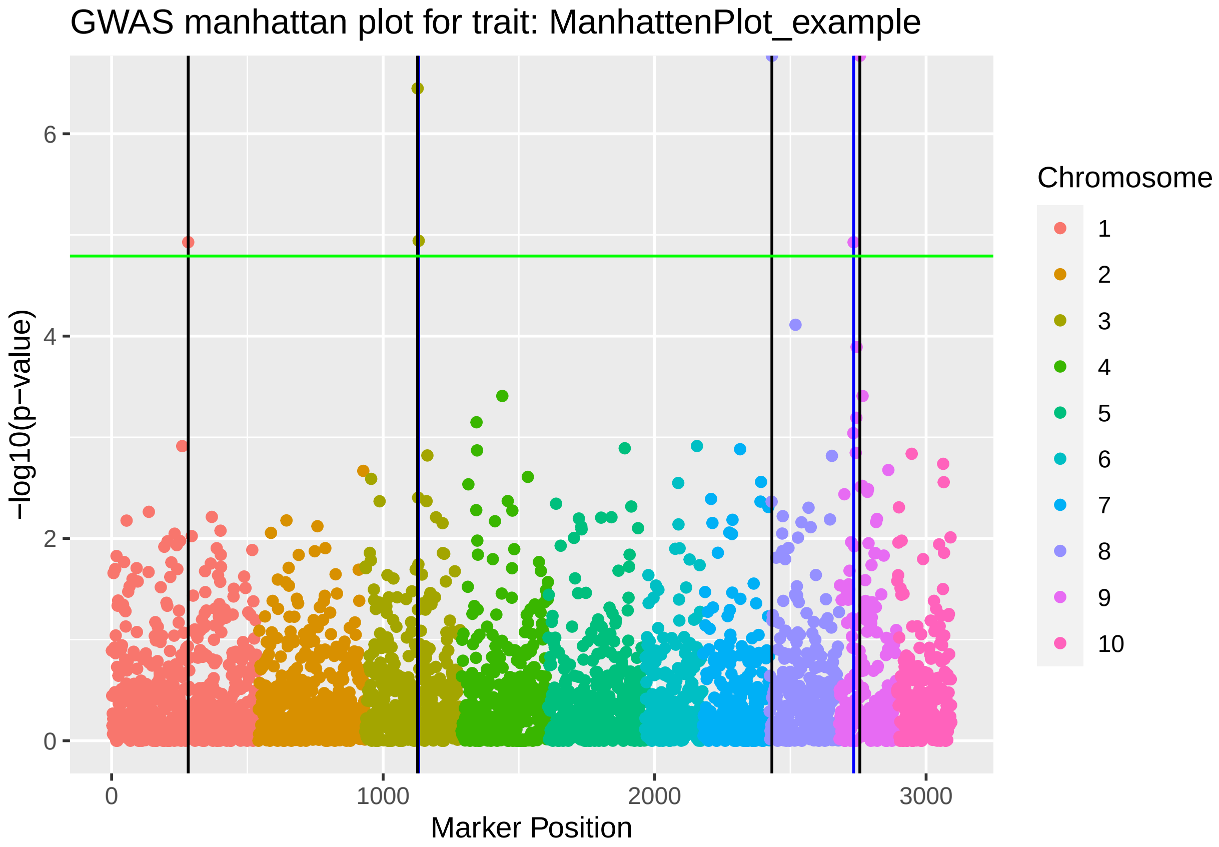
```
Example3=GWASapply(pheno=mySim$y, geno=genotypes, Cov=CV, GM=GM, PCA.M = 3,QTN.position=mySim$QTN.posit
```

```
## [1] "GWASapply Starting"
## [1] "Principal Components have been calculated Successfully"
## [1] "The final cuttoff for a significant p-value is 1.61655350792111e-05"
## [1] "6 Significant SNPs were found"
## [1] "QTN's detected with 4 True Positives and 2 False Positives."
## [1] "GWASapply results have printed"
## [1] "GWASapply ran successfully and took 1.7 seconds"
```

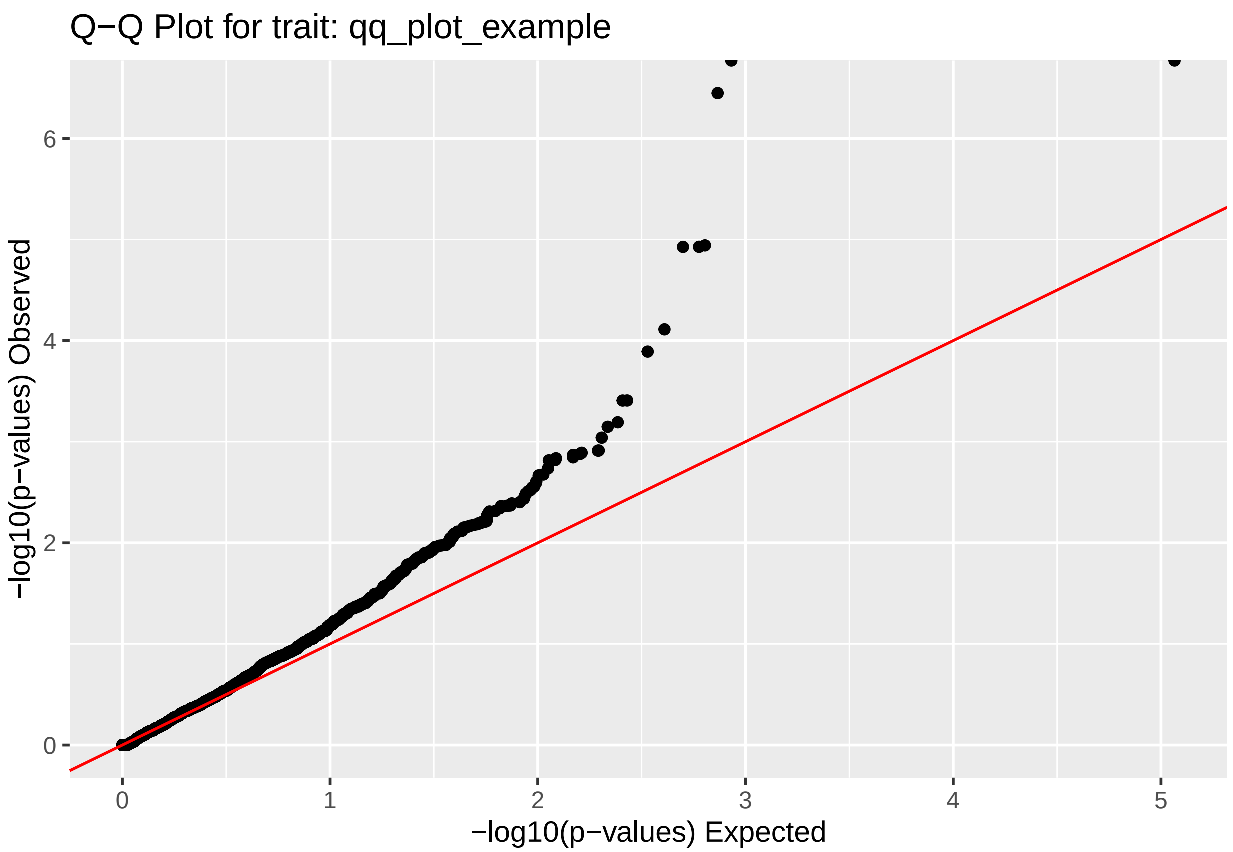
Quick run options

After running GWASapply with plots = FALSE you can always run the plots individual as shown below. This option gives you more flexibility over the output and can help for specific analyses.

```
manhattan_plot(GM,Example3$P.value.res, trait="ManhattenPlot_example",FP = Example3$False.Positive, TP
```



```
qq_plot(GM,Example3$P.value.res,QTN.position, trait = "qq_plot_example")
```



### Further Information

For more information on individual functions please see the “Referenc_Manual.pdf” or type ?FUNCITON_NAME into the R console, this will pull up specific information of each function inside GWAStest. For example typing ?manhatten_plot will pull of the help page with details about the function that creates the Manhattan plots.