Chapter 14

Genome-Wide Association Analysis Using R

Julio Isidro-Sánchez, Deniz Akdemir, and Gracia Montilla-Bascón

Abstract

This chapter provides a practical overview of the statistical analysis using R [1] and genotype by sequencing (GBS) markers for genome-wide association studies (GWAS) in oats. Statistical analysis is performed by R package rrBLUP [2] and issues associated with the analysis are addressed along with the R code. The ultimate aim of this chapter is to provide a practical guideline to do GWAS analysis using R, rather than describe the theory in depth. For more details about the subject, readers are referred to the excellent resource book in GWAS [3]. A basic programming experience in R is assumed.

Key words GWAS, Population structure, Linkage disequilibrium, False discovery rate, Bonferroni correction

1 Introduction

The central goal of GWAS is to identify casual mutations that have an effect on a phenotype (any aspect of an organism that can be measured). A casual mutation is a position in the genome where an experimental manipulation of the DNA produces an effect on the phenotype on average. From a statistical point of view, a casual mutation occurs when $\mathrm{Cov}(Y,X) \neq 0$ where Y are the value of the phenotypes and X the value of the genotypes.

Genome-wide association analyses are aimed for detecting variants at genomic loci that are associated with complex traits in the population and, in particular, at detecting associations between common single-nucleotide polymorphisms (SNPs) and common diseases. Markers that are significantly associated with the phenotype are presumed to be in linkage disequilibrium (LD) with putative Quantitative Trait Loci (QTL). The goal of GWAS, is to test for association between the frequency of each of hundreds of thousands of common variants and a given phenotype, that exceed a

Electronic Supplementary Material: The online version of this chapter (doi: 10.1007/978-1-4939-6682-0_14) contains supplementary material, which is available to authorized users.

Sebastian Gasparis (ed.), *Oat: Methods and Protocols*, Methods in Molecular Biology, vol. 1536,
DOI 10.1007/978-1-4939-6682-0_14, © Springer Science+Business Media LLC 2017

conservative genome-wide threshold for association and then test these for evidence of replication. High statistical power, low probability of Type I error, use of covariates, and high resolution are the keys for success in GWAS.

From a multiple regression model, with a continuous phenotype Y, the rigorous formulation of GWAS analysis is

$$y = \beta_u + X_a\beta_a + X_d\beta_d + \varepsilon$$

where β_u is the mean, X_a and X_d are random markers variables with values of $A_1A_1 = -1, A_1A_2 = 0$ and $A_2A_2 = 1$ for additive Xa and $A_1A_1 = -1, A_1A_2 = 1$ and $A_2A_2 = -1$ for dominance X_d, β_a and β_d are associated with X_a and X_d respectively, and ε is the random error.

In GWAS we are testing for every marker the following hypothesis test:

$$\text{Ho} : \beta_a = 0 \cap \beta_d = 0$$
$$\text{Ha} : \beta_a \neq 0 \cap \beta_d \neq 0$$

In this chapter, the GWAS analysis is carried out using the R package rrBLUP (https://cran.r-project.org/web/packages/rrBLUP/rrBLUP.pdf). Within the package the *A.mat* and *GWAS* functions are used to perform the analysis based on the mixed model described by Yu et al. [4] as follow.

$$y = X\beta + Zg + S\tau + \varepsilon$$

where y is a vector of phenotype observations, β is a vector of fixed effects, g models the genetic background of each line as a random effect with $\text{Var}[g] = K\sigma^2$, τ is a vector of additive SNP effects as a fixed effect, and ε is a vector of residual effects with $\text{Var}[\varepsilon] = I\sigma^2_\varepsilon$.

It is important to point out that without additional evidence (experimental manipulation of the DNA) that the statistical models cannot inference with absolute certainty to identify a causal mutation. Genome-wide association studies are a starting point for additional experimental work.

2 Material

Our genetic system have two components that can be measured, the phenotypic and the genotypic component. The simulated data sets used here are an example modified from wheat dataset; with 932 phenotypic yield data observations, and 329 genotypic data with 3629 markers, which are mapped. Data can be downloaded from book repository (The data files "GWAS_Data.RData" and "GWAS_Script.R" can be downloaded from the link: 10.1007/978-1-4939-6682-0_14).

The analysis of GWAS is performed using genotype by sequencing (GBS), which is characterized for having many missing data (http://cbsu.tc.cornell.edu/lab/doc/GBS_Method_Overview1.pdf). Genotype by sequencing data normally generate hundred of thousands of markers. For the simplicity of the analysis, a subset of the GBS data has been selected in this study.

Although, phenotypes are considered following a normal probability model, there are a broad class of models that can apply to continuous and discrete phenotypes analysis [5]. The R software is free and open-source statistical software that can be downloaded for Windows, Mac OS X, or Linux from https://www.r-project.org/. From the left side of the website click on CRAN (Comprehensive R Archive Network), select the appropriate CRAN Mirror, and select the appropriate operating system and install it into your computer following on-screen prompts for installation. Once in your computer, rrBLUP package needs to be installed by using the next command.

>install.packages("rrBLUP") [1, 2]

3 Methods

3.1 Basic Guideline to Perform a GWAS Study

Main guidelines to do a GWAS study.

1. Read phenotypes and check the assumptions of the models.
   (a) Check outliers.
   (b) Normal distribution of errors. Possible transformation of the data.
2. Read genotypes and filter for:
   (a) Markers with a proportion of missing data more than a particular threshold set by the researcher.
   (b) Individuals with a high proportion of missing data.
   (c) Individuals with a high proportion of heterozygous.
   (d) Remove genotypes with a minor allele frequency (MAF) less than 5 %.
   (e) Remove genotypes that fail a Hardy-Weinberg test of equilibrium (Normally, use a conservative p-value cut-off of $<10^{-5}$).
3. Imputation of the genotype file, and performing the Kinship matrix.
4. Look for population structure effects.
5. Match phenotypes and genotypes.
6. Perform GWAS function from rrBLUP with and without population structure effects and Kinship matrix.
7. Manhattan and Q-Q plot graphs.
8. Interpretation and Validation.

3.2 Analysis GWAS Study in R

1. Read phenotypes and check the assumptions of the models.

```
pheno <- read.csv("phenoat.csv",header=T);
dim(pheno)
```

```
## [1] 932 3
```

```
head(pheno) ## GID ENV Yield
```

```
## 1 Oat179 Env1 6317.606
## 2 Oat130 Env2 6335.475
## 3 Oat303 Env4 7259.274
## 4 Oat270 Env1 6916.124
## 5 Oat202 Env4 6845.943
## 6 Oat233 Env3 5750.001
```

```
str(pheno)
```

```
## 'data.frame': 932 obs. of 3 variables:
## $ GID : Factor w/ 330 levels "Oat1","Oat10",..:
89 36 228 191 116 150 205 25 153 264 …
## $ ENV : Factor w/ 4 levels "Env1","Env2",..:
1 2 4 1 4 3 2 2 3 2 …
## $ Yield: num 6318 6335 7259 6916 6846 …
```

The dataset contains 932 yield observations measured in four different environments. The structure function in R indicates that GID and ENV are factors, which is the correct class vector needed for the analysis. If they were not factors, you could change the class using "as.factor()" function in R.

Data showed a continuous distribution (Fig. 1), therefore we could make inference using regression models. Some assumptions have to be met in order to make inference. Observations have to be independent and identically distributed and errors (and therefore the phenotypes) have to follow a normal distribution. Data seems to meet the normality assumption (Fig. 1), although few outliers can be observed.

```
hist(pheno$Yield,xlab="Yield",main="Histogram Yield")
```

```
shapiro.test(pheno$Yield)
```

```
##
## Shapiro-Wilk normality test
##
## data: pheno$Yield
## W = 0.99392, p-value = 0.0007821
```

The code above checks the normality of the phenotypes. In general, if phenotypes are normal the residuals will be also normal. The Shapiro–Wilk test for normality indicates that data are not normal. The solution for this problem is to check for outliers and transform data in order to get the normality of the phenotypes.

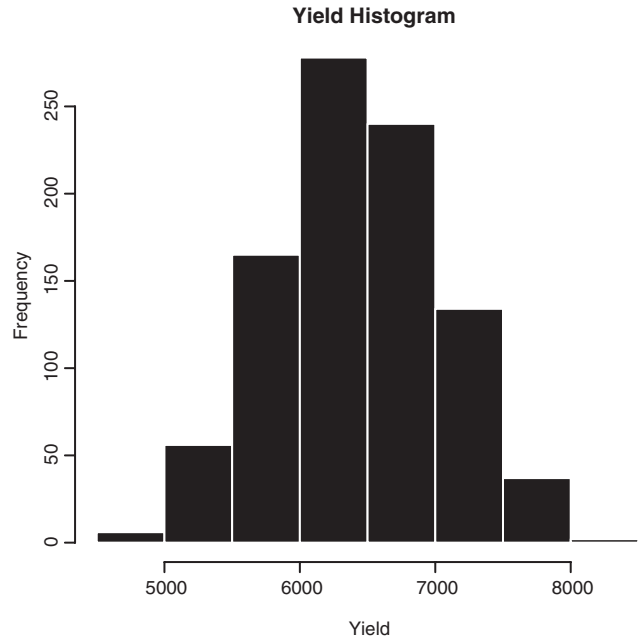

Fig. 1 Histogram of phenotypic data

```
boxplot.yield<- boxplot(pheno$Yield)
outliers <- boxplot.yield$out; outliers
## [1] 8979.334 3944.849 3947.001 3948.641 4576.506
3942.469 8166.286
## [8] 4616.835 4672.310 8920.055

pheno <- pheno[-which(pheno$Yield%in%outliers),]
shapiro.test(pheno$Yield)

##
## Shapiro-Wilk normality test
##
## data: pheno$Yield
## W = 0.99719, p-value = 0.1093

pheno <- na.omit(pheno)
```

The code above indicates that there are ten outliers on the yield phenotypic data. After removing the outliers, we cannot reject the Shapiro–Wilk normality test indicating that yield data are now normal. Last line in code is to eliminate any possible missing data (NA).

(*see* **Note 1**).

2. Read genotypes and filtering.

Genotypes and map information are read into the R working directory. Genotypic data contains 329 lines and 3629 markers. The map file contains 3354 markers and three variables.

```
geno <- read.csv("genoat.csv",header=T,row.names = 1);
dim(geno)

## [1] 329 3629

map <- read.csv("mapoat.csv",header=T,stringsAsFactors
=F,row.names=1); dim(map)

## [1] 3354 3

geno[1:5,1:5] ### View genotypic data.

## Marker1 Marker2 Marker3 Marker4 Marker5
## Oat1  NA   1   1  1   1
## Oat2  -1  NA   1  1   1
## Oat3  -1   1   1  1  NA
## Oat4  -1  NA  -1  1   1
## Oat5  -1  NA  -1  1   1

map[1:5,1:3] ###

## Markers chrom loc
## 1  Marker607  1A 16730
## 2 Marker2900  1A 16730
## 3 Marker1316  1A 25338
## 4 Marker2297  1A 26595
## 5 Marker1895  1A 27071
```

The next step is to filter the genotypic data. Filtering conditions will depend on researcher criteria. The code below represent a simple function to remove individuals and markers that does not met the criteria establish by the researcher. The function will remove individuals with more than a certain percentage of missing data, markers with a greater proportion of a threshold missing percentage, and also markers with a high proportion of heterozygous calls.

```
filter.fun <- function(geno,IM,MM,H){
   #Remove individuals with more than a certain %
   missing data  individual.missing <- apply(geno,1,
   function(x){
      return(length(which(is.na(x)))/ncol(geno))
   })

   #Remove markers with certain % missing data
   marker.missing <- apply(geno,2,function(x)
   {return(length(which(is.na(x)))/nrow(geno))

   })
   length(which(marker.missing>0.6))
   #Remove individuals with high heterozygous calls.
   heteroz <- apply(geno,1,function(x){
      return(length(which(x==0))/length(!is.na(x)))
   })
```

```
   filter1 <- geno[which(individual.missing<IM),
which(marker.missing<MM)]
      filter2 <- filter1[,(heteroz<H)]
      return(filter2)
   }
```

(*see* **Note 2**).

```
geno.filtered <- filter.fun(geno[,1:3629],0.4,0.60,0.02)
```

geno.filtered will be composed by those genotypes with less than 40% missing data, markers with less than 60% missing data and individuals with less than 2% of heterozygous calls. After filtering one individual have been removed and 361 markers.

```
geno.filtered[1:5,1:5];dim(geno.filtered)
```

```
##   Marker1 Marker2 Marker3 Marker4 Marker5
## Oat1   NA       1       1       1       1
## Oat2   -1      NA       1       1       1
## Oat3   -1       1       1       1      NA
## Oat4   -1      NA      -1       1       1
## Oat5   -1      NA      -1       1       1
```

```
## [1] 328 3268
```

The lower the minor allele frequency (MAF) the lower the statistical power, because MAF increases the variance of the phenotypes associated with the MAF alleles. Therefore, it is necessary to filter them and the most standard way is to eliminate markers with less than 5% of MAF. Minor allele frequency will be removed using the A.mat function within the rrBLUP package.

3. Imputation of the genotype file, and performing the Kinship matrix.

The main idea behind imputation is to predict (or 'impute') the missing data based upon the observed data. Imputation is now routinely used to facility genotyped studies by increasing the power of the analysis. Here, "A.mat" function from rrBLUP package is used for imputation. A.mat has two options for imputation (https://cran.r-project.org/web/packages/rrBLUP/rrBLUP.pdf). One is to replace missing data with the population mean for that marker, or using an expectation maximization (EM) algorithm based on the multivariate normal distribution [6].

```
library(rrBLUP)
Imputation     <-     A.mat(geno.filtered,impute.method="EM",
return.imputed=T,min.MAF=0.05)
```

```
## [1] "A.mat converging:"
## [1] 0.0431
## [1] 0.00647
```

```
K.mat <- Imputation$A ; dim(K.mat) ### KINSHIP matrix

## [1] 328 328

geno.gwas <- Imputation$imputed; dim(geno.gwas) #NEW geno
data.

## [1] 328 1962

geno.gwas[1:5,1:5]## view geno

##        Marker1 Marker3 Marker4 Marker5 Marker7
## Oat1 -1.44066       1       1 1.0000000    -1
## Oat2 -1.00000       1       1 1.0000000    -1
## Oat3 -1.00000       1       1 0.6753843    -1
## Oat4 -1.00000      -1       1 1.0000000    -1
## Oat5 -1.00000      -1       1 1.0000000    -1

K.mat[1:5,1:5]## view Kinship

##          Oat1        Oat2       Oat3     Oat4        Oat5
## Oat1 1.8191561  0.220454719 0.1009423 0.15218203 0.177660657
## Oat2 0.2204547 2.111943356 0.1266988 0.02978199 -0.007214392
## Oat3 0.1009423  0.126698834 1.8000414 -0.12678093 -0.126589635
## Oat4 0.1521820  0.029781989 -0.1267809 1.87520005 1.803271696
## Oat5 0.1776607  -0.007214392 -0.1265896 1.80327170 1.873595678
```

A.mat estimate the relationship matrix using the EM algorithm, and also return the genotypic information. In this function, we also removed the MAF markes having at the end of the filtering 328 genotypes and 1962 markers.

4. Look for population structure effects.

One crucial step in GWAS analysis is to study the population structure (PS). The main reason to perform this study is that, as a consequence of having different population genetic histories, distinct subpopulations could have differences in allele frequencies for many polymorphisms throughout the genome. If the populations have different overall values for the phenotype, any polymorphisms that differ in frequency between the two populations are associated with the phenotype even though they are not casual or in strong linkage disequilibrium with casual polymorphisms [7–9]. Principal component analysis (PCA) on genotypic data is used to visualize the structure of our populations using the function "*svd()*" in R.

Population structure accounted by PCA is limited to correcting for spurious associations on a global level of genetic variation. Thereby, PS does not adequately capture the relatedness between individuals, and this relationship between genotypes (K, kinship matrix) needs also be taking into account on the analysis. Not taking into account of PS, K as well as a potential confounding between the phenotype and the genotype effects, could lead to unrealistic assessments in GWAS analysis.

```
geno.scale <- scale(geno.gwas,center=T,scale=F) # Data needs to
be center.
Svdgeno <- svd(geno.scale)
PCA <- geno.scale%*%svdgeno$v #Principal components
colnames(PCA) <- paste("PCA",1:ncol(PCA),sep="")
PCA[1:5,1:5]
```

```
##            PCA1       PCA2     PCA3       PCA4          PCA5
## Oat1 -6.976549   -9.824339 5.290272 -1.184019    5.00415366
## Oat2 -8.745900  -10.545057 4.730541 -8.834242   -0.02504027
## Oat3 -6.292199    3.412671 4.774038  6.657838    8.93925754
## Oat4 -19.524147   7.689386 -7.963162 -13.644082  -6.19429306
## Oat5 -19.334593   8.011715 -7.624697 -13.033030  -6.53379527
```

Principal component analysis is calculated multiplying the scaled genotypes times the eigenvectors. The total number of PCA is equal to the total number of lines on the genotypic file. The total genetic variance explained by the PCA can be seen on Fig. 2.

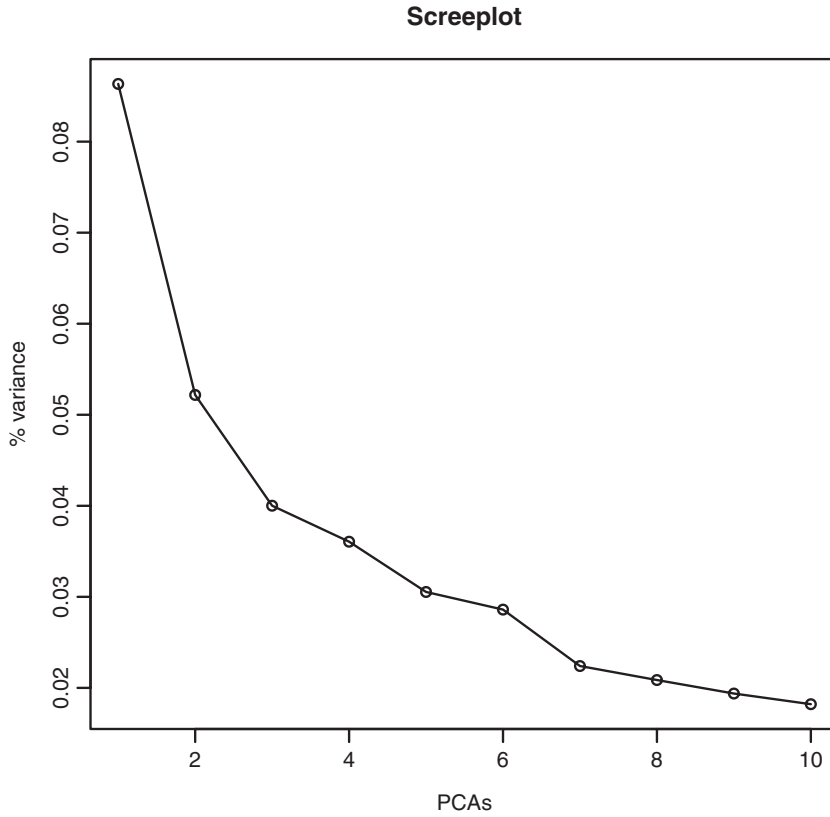

Fig. 2 Screeplot representing the percentage of the variance explained by the principal components

```
plot(round((svdgeno$d)^2/sum((svdgeno$d)^2)
,d=7)[1:10],type="o",main="Screeplot",xlab="PC
As",ylab="% variance")
```

The screeplot indicates that the first two PC explained around 14% of the genetic variance, indicating that population structure effects are mild in this dataset. Figure 3 shows the first two PC.

```
PCA1 <- 100*round((svdgeno$d[1])^2/sum((svdgeno$
d)^2),d=3); PCA1
## [1] 8.6
PCA2 <- 100*round((svdgeno$d[2])^2/sum((svdgeno$
d)^2),d=3); PCA2
## [1] 5.2
```

Depending on the genotype and breeder information, a clustering analysis we could highlight different clusters on the graph. The following script shows how to make clusters and plot the first tow clusters on the PC graph (Fig. 4).

```
Eucl <- dist(geno.gwas) # Euclinean distance
Fit <- hclust(Eucl,method="ward") # Ward crite-
rion makes clusters with same size.

## The "ward" method has been renamed to "ward.D";
note new "ward.D2"
```

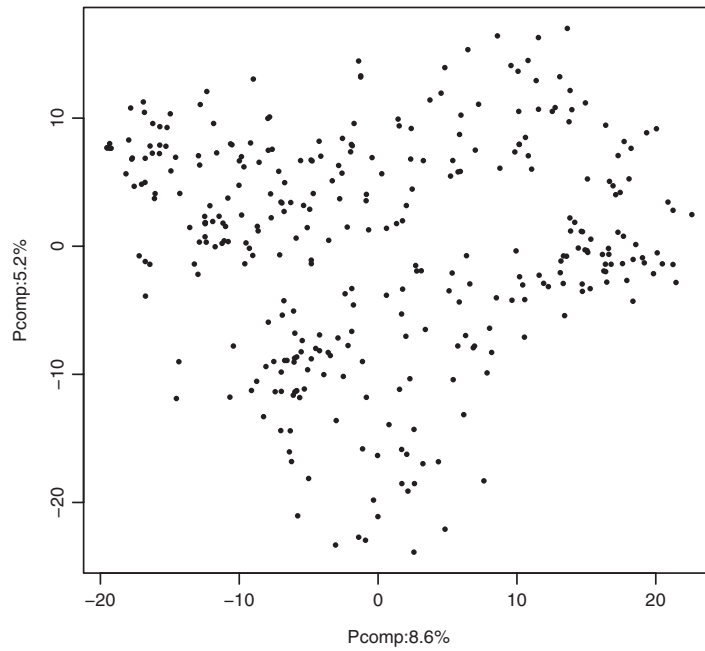

Fig. 3 First and second principal component analysis. *X* and *Y* values represent the proportion of variance explained by each PC

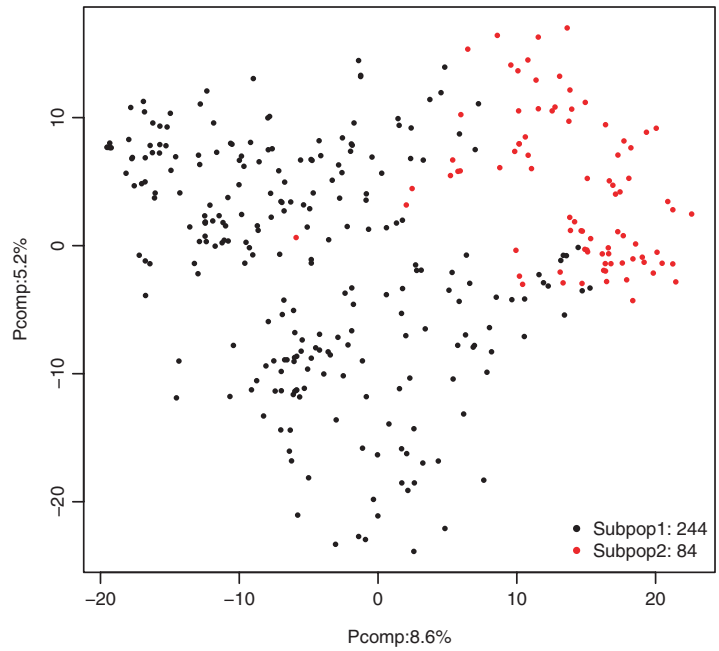

Fig. 4 Principal component analysis plots with the clustering information. *Red* and *black circles* represent two distinct subpopulations

```
groups2 <- cutree(fit,k=2) # Selecting two clusters.
table(groups2) #Number of individuals per cluster.
```

```
## groups2
## 1 2
## 244 84
```

```
plot(PCA[,1],PCA[ ,2],xlab=paste("Pcomp:",PCA1,"%",sep=""),ylab=paste("Pcomp:",PCA2,"%",sep=""),pch=20,cex=0.7,col=groups2)
```

5. Match phenotypes and genotypes.

The following code prepares the files to be used on GWAS function. In order to do the analysis the same genotypes needs to be on phenotype and genotypes files. A matching function between files needs to be performed.

```
pheno=pheno[pheno$GID%in%rownames(geno.gwas),]
pheno$GID<-factor(as.character(pheno$GID), levels=rownames(geno.gwas)) #to assure same levels on both files
##Creating file for GWAS function from rrBLUP package X<-model.matrix(~-1+ENV, data=pheno)
```

```
pheno.gwas <- data.frame(GID=pheno$GID,X,Yield=
pheno$Yield) head(pheno.gwas)
```

```
## GID ENVEnv1 ENVEnv2 ENVEnv3 ENVEnv4 Yield
## 1 Oat179 1 0 0 0 6317.606
## 2 Oat130 0 1 0 0 6335.475
## 3 Oat303 0 0 0 1 7259.274
## 4 Oat270 1 0 0 0 6916.124
## 5 Oat202 0 0 0 1 6845.943
## 6 Oat233 0 0 1 0 5750.001
```

```
geno.gwas <- geno.gwas[rownames(geno.gwas)%in%
pheno.gwas$GID,]
pheno.gwas <- pheno.gwas[pheno.gwas$GID%in%
rownames(geno.gwas),]
geno.gwas <- geno.gwas[rownames(geno.gwas)%in%
rownames(K.mat),]
K.mat <- K.mat[rownames(K.mat)%in%rownames(geno.
gwas),colnames(K.mat)%in%rownames(geno.gwas)]
pheno.gwas <- pheno.gwas[pheno.gwas$GID%in%
rownames(K.mat),]
```

Likewise, same information needs to be between genotypes and map information. The next code matches both files.

```
geno.gwas <-geno.gwas[,match(map$Markers,colnam
es(geno.gwas))] head(map)
```

```
##       Markers  chrom     loc
## 1   Marker607     1A   16730
## 2  Marker2900     1A   16730
## 3  Marker1316     1A   25338
## 4  Marker2297     1A   26595
## 5  Marker1895     1A   27071
## 6  Marker2902     1A   27071
```

```
geno.gwas <- geno.gwas[,colnames(geno.gwas)%in%map
$Markers]
map <- map[map$Markers%in%colnames(geno.gwas),]
geno.gwas2<- data.frame(mark=colnames(geno.gwas),
chr=map$chrom,loc=map$loc,t(geno.gwas))
dim(geno.gwas2)
```

```
## [1] 1759 331
```

```
colnames(geno.gwas2)[4:ncol(geno.gwas2)]      <-
rownames(geno.gwas)
```

This is the final view of the three different files needed for the GWAS analysis in rrBLUP package.

```
head(pheno.gwas)
```

```
## GID ENVEnv1 ENVEnv2 ENVEnv3 ENVEnv4 Yield
## 1 Oat179 1 0 0 0 6317.606
```

```
## 2 Oat130 0 1 0 0 6335.475
## 3 Oat303 0 0 0 1 7259.274
## 4 Oat270 1 0 0 0 6916.124
## 5 Oat202 0 0 0 1 6845.943
## 6 Oat233 0 0 1 0 5750.001
```

```
geno.gwas2[1:6,1:6]
```

```
## mark chr loc Oat1 Oat2 Oat3
## Marker2297 Marker2297 1A 26595 -1 -1.0000000
-1
## Marker3125 Marker3125 1A 35232 1 -1.0000000
-1
## Marker2100 Marker2100 1A 35653 -1 1.0000000 1
## Marker1797 Marker1797 1A 51943 1 -0.4259651
-1
## Marker3191 Marker3191 1A 51943 -1 1.0000000 1
## Marker1403 Marker1403 1A 56393 -1 1.0000000 1
```

```
K.mat[1:6,1:6]
```

```
## Oat1 Oat2 Oat3 Oat4 Oat5
## Oat1 1.8191561 0.220454719 0.1009423 0.15218203
0.177660657
## Oat2 0.2204547 2.111943356 0.1266988 0.02978199
-0.007214392
##   Oat3   0.1009423   0.126698834   1.8000414
-0.12678093 -0.126589635
##   Oat4   0.1521820   0.029781989   -0.1267809
1.87520005 1.803271696
##   Oat5   0.1776607   -0.007214392   -0.1265896
1.80327170 1.873595678
##   Oat6   -0.1810759   -0.230823204   0.1734623
-0.40927660 -0.357916219
```

6. Perform GWAS function from rrBLUP with and without population structure effects and Kinship matrix.

   A statistically significant association between a genotypic marker and a particular trait is considered to be a proof of linkage between the phenotype and a casual locus. Generally, PS leads to spurious associations between markers and a trait, so that a statistical approach must account for PS [10].

   In this analysis, four different statistical models were performed.

   (a) Naïve model without controlling for PS or family relatedness (gwasresults).

   (b) Controlling for PS effects (Q model, gwasresults2).

   (c) Controlling just for relatedness (K model, gwasresults3).

   (d) Controlling for both Q and K effects (Q+K model, gwasresults4).

```
## gwasresults<-GWAS(pheno.gwas,geno.gwas2, fixed=
colnames(pheno.gwas)[2:5], K=NULL, plot=T,n.PC=0)

##gwasresults2<-GWAS(pheno.gwas,geno.gwas2,
fixed=colnames(pheno.gwas)[2:5], K=NULL, plot=T,n.
PC=6)

##gwasresults3<-GWAS(pheno.gwas,geno.gwas2,
fixed=colnames(pheno.gwas)[2:5], K=K.mat, plot=T,n.
PC=0)

##gwasresults4<-GWAS(pheno.gwas,geno.gwas2,
fixed=colnames(pheno.gwas)[2:5], K=K.mat, plot=T,n.
PC = 6)
```

7. Manhattan and Q-Q plot graphs.

With the aim to provide more scrip for R learning readers, we report here the Q-Q and Manhattan plot using R. The readers must know that GWAS function has an option to plot both graphs (plot = T).

Once your GWAS analysis is complete, the minimum set of analysis one should undertake when presenting your GWAS analysis are the Quantile-Quantile (Q-Q) and the Manhattan plot.

The Q-Q plot is a visual diagnostic tool that can identify the effects of accounted for population structure, relatedness and other issues in GWAS analysis. A Q-Q plot is use to detect a difference between the distribution of *p*-values that we observe in GWAS compared to the *p*-values we would have expected to produce if the null hypothesis (no association) was true for every marker we tested. How the Q-Q plot differs from this null expectation will provide us with a considerable amount of information about whether or not something is amiss with our statistical analysis. In Fig. 5, the Q-Q plots for all models are presented.

As we checked on **step 4**, the PS effects were not strong, which indicates that the effect on the analysis will not be really significant. That is the reason why although including PS effects the change on the Q-Q plot were not substantial. We used *"n.PC = 6"* to account for PS because it will account for most of the variance explained by the genotypes.

In general, Q-Q plots that do not look like the null expectation indicate that something is wrong with our statistical analysis of our GWAs data (accounted for factors, large number of genotyping errors, violations of the assumptions of our statistical tests, etc.). In such cases, we cannot draw any conclusions concerning the detection of casual polymorphism.

A typical report visualization of GWAS of SNPs is the Manhattan plot of significance against chromosomal location (Fig. 6). In a GWAS, the goal is that any marker for which we have rejected the null hypothesis, to be in LD with a casual polymorphism. One of the reasons why a test we have rejected the null is

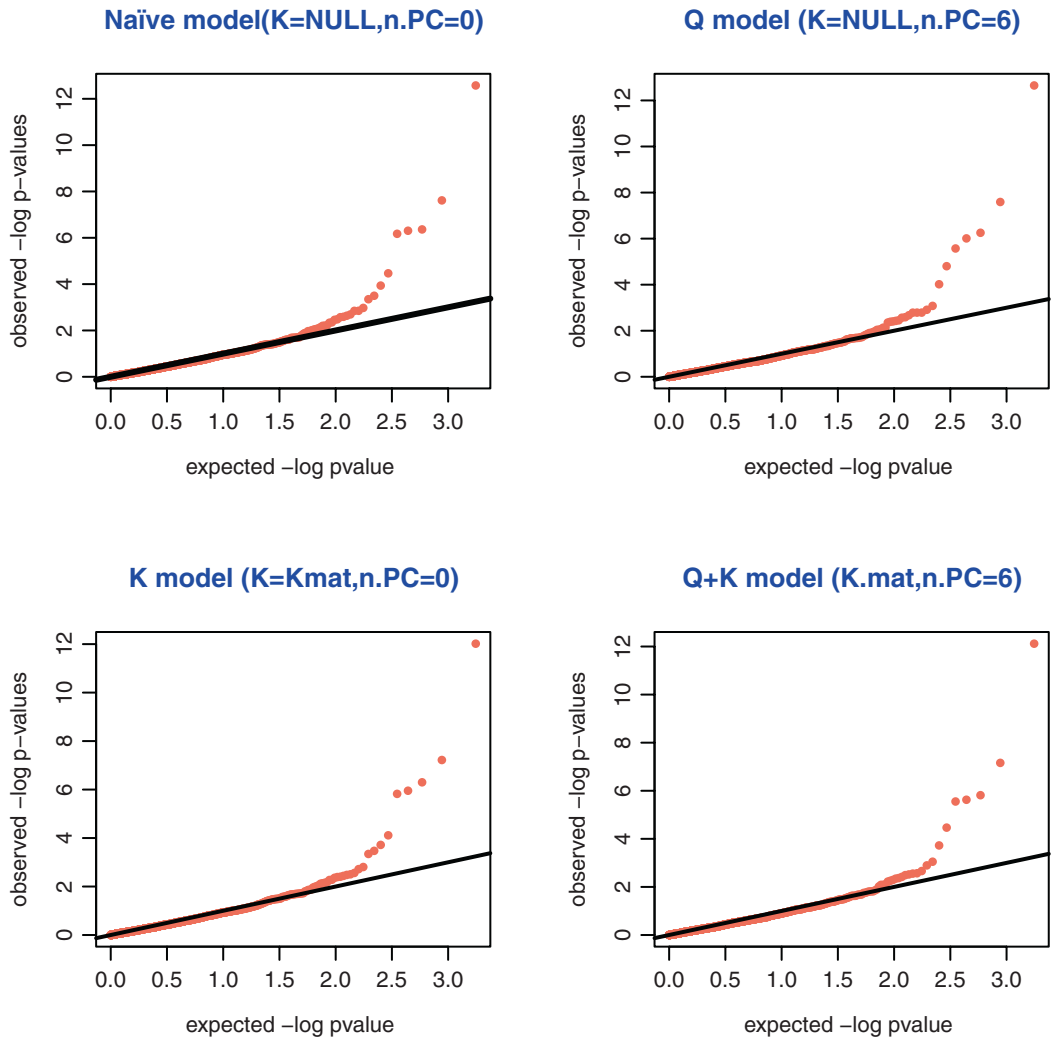

Fig. 5 Q-Q plots of the different models. The Q-Q plots for all models seem to follow the 1:1 ratio expectation. When accounting for PS on the analysis (Q model), more signal is captured by the model, as the *red line* follows closer the 1:1 expected ratio line, and in this sense the K model seems to fit better than Q model. The Q + K model did not improve significantly the analysis. This indicates the kinship matrix is enough to account for *solid line*. This makes sense since PS effects were mild as the PCA analysis shows on Fig. 3

not in LD with the casual polymorphism is the Type I Error (the probability of rejecting the null hypothesis when the null is true) from the multiple testing problem. Analysis needs to be corrected by Type I error because it would be very costly to track wrong significant casual polymorphisms. Type I error rate can be controlled by setting alpha lower. This is a trade-off: the lower the Type I error is set, the lower the power of our hypothesis. In this analysis, Bonferroni correction and False Discovered Rate (FDR) are presented. Those markers above the threshold are considered to be in LD with the casual polymorphism (Fig. 6).

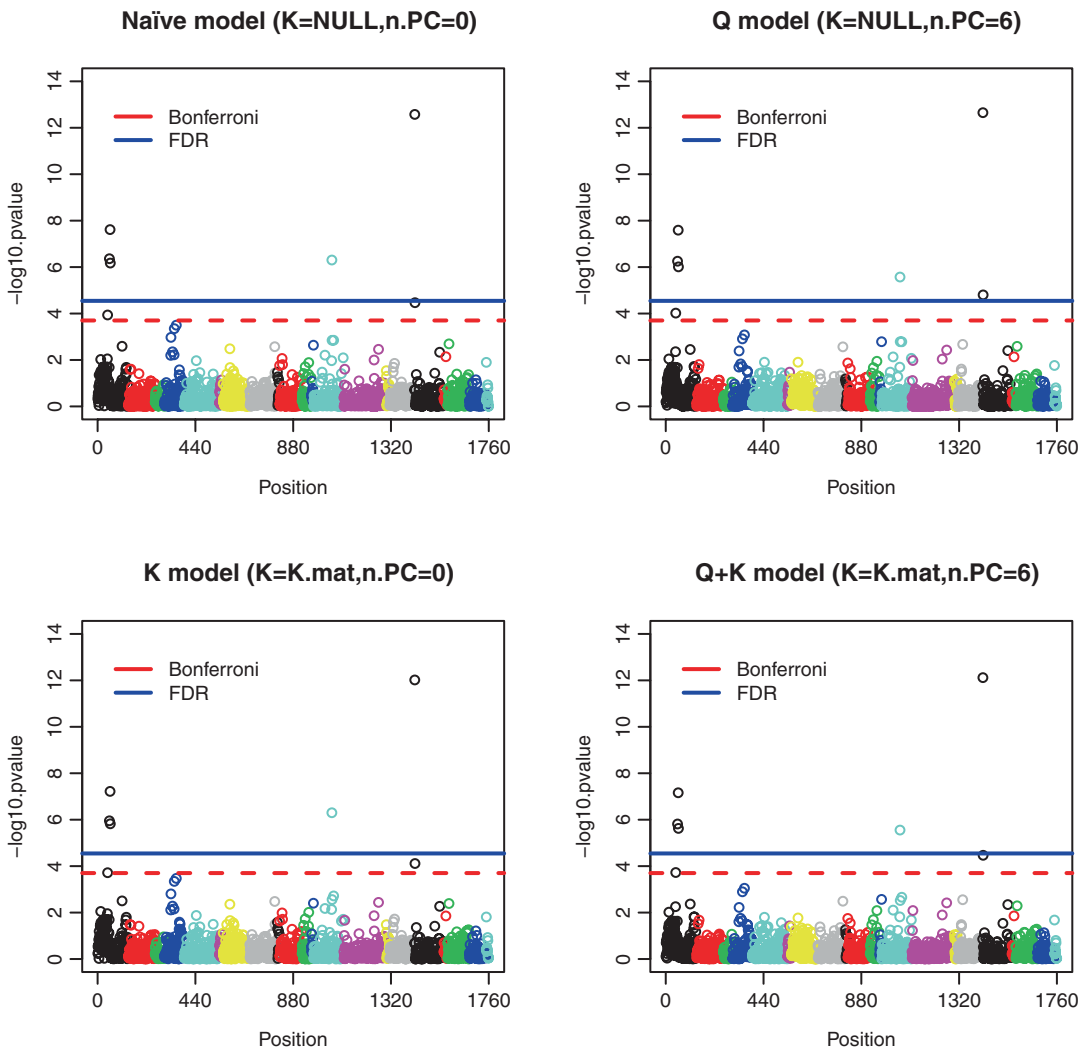

Fig. 6 Manhattan plots of the different statistical models. Threshold are indicated with *lines* in *red* (Bonferroni correction) and *blue* (False Discovery Rate). Chromosome positions are drawn with different colors on *x* axis)

(*see* **Note 3**).

The significant hits from the Manhattan plots on Q+K model are:

```
alpha_bonferroni=-log10(0.05/length(gwasresults
$Yield)) ###This is Bonferroni correcton
alpha_FDR_Yield<--log10(FDR(10^(-gwasresults$Yield),
0.05))## This is FDR cut off

which(gwasresults4$Yield>alpha_bonferroni)

## [1] 53 56 57 1054 1427

which(gwasresults4$Yield>alpha_FDR_Yield)
```

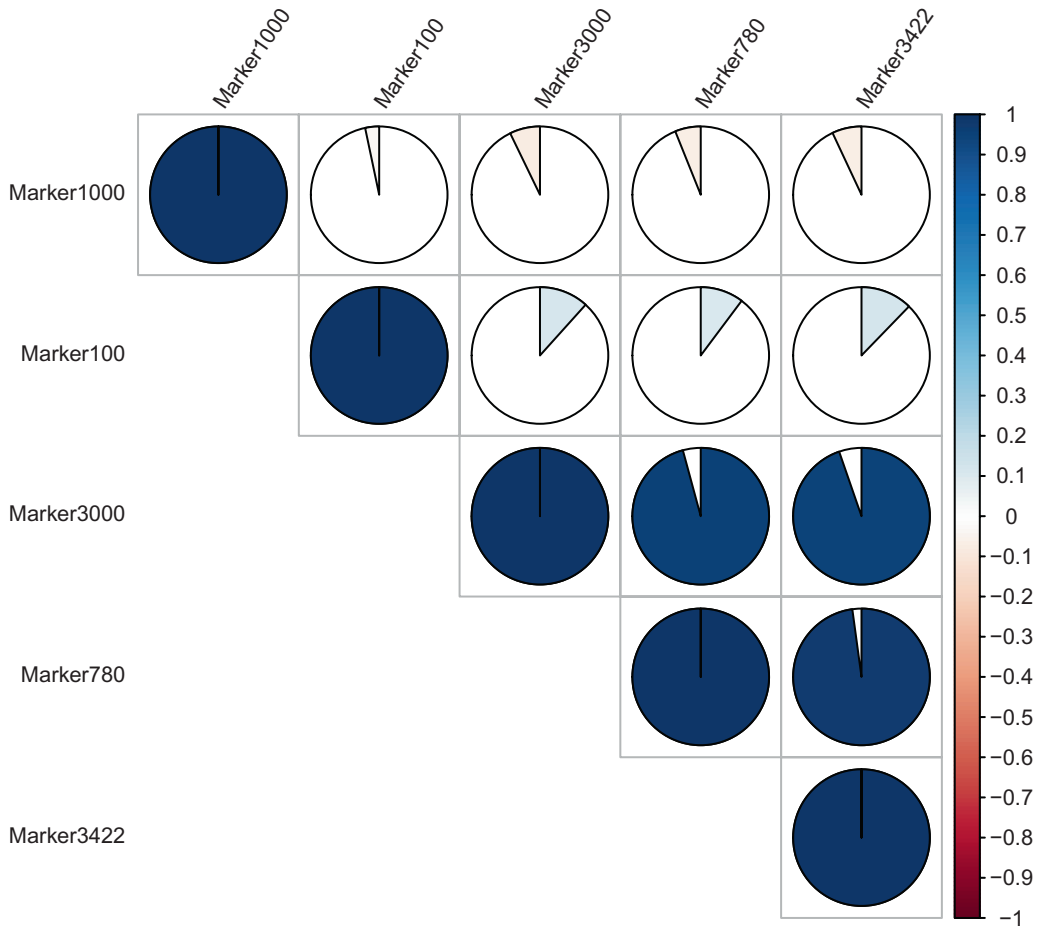

Fig. 7 Correlation among significant hits on Q + K model. High correlations are expressed with high intensity *blue* or *red color*

```
## [1]   45   53   56   57 1054 1427 1428
```

These are the significant *p*-values. We expect a set of polymorphism that are close to each other and high correlated in the genome to the causal polymorphism. Figure 7 shows the correlation between the significant markers. The highest correlation appears to be between SNPs closer to the casual mutation, since there are in high LD with each other and therefore with a putative QTL.

```
library(corrplot)
markers.gwasresults4.bonf<-geno.gwas[,c(53,56,57,
1054,1427)]
corr_sign  <-  cor(markers.gwasresults4.bonf,
use="complete.obs") par(oma=c(2,2,4,2))
corrplot(corr_sign,order="hclust",method="pie",
tl.pos="lt", type="upper",
```

```
        tl.col="black", tl.cex=0.8, tl.srt=55,
        sig.level=0.90,cl.length=21,insig = "blank")
mtext("Correlation Significant hits",outer=TRUE,
line=1)
```

(*see* **Note 4**).

8. Interpretation and Validation.

Results show that there are slight differences between the naïve model and the rest. This is expected since the screeplot in Figs. 2 and 3 showed that the first few PC does not explain much of the variability of the data. Nevertheless, when we account for PS and K matrix, the Q-Q plot showed that data followed better the uniform null distribution, making Q + K the desirable model. Five loci were significantly associated with the casual polymorphism:

These significant markers should be validated in another independent GWAS experiment (with more lines and more markers to increase the power of detection) to be able to confirm with more certain that in fact there is a true casual polymorphism in the same region of the chromosomes and more QTL.

4 Notes

1. When data are not normal, you can improve normality of the original phenotypic data using logit, inverse, square root, arcsine, and log transformation methods.
2. You can check how many individuals with more that 40% of missing date will be removed by running: length(which(individual.missing > 0.40).
3. Bonferroni correction is very conservative in comparison with FDR. A Bonferroni correction of alpha = 0.01 decrease the power of detection considerably. The choice between Bonferroni and FDR depends on researcher.
4. This does not indicate that we found the casual polymorphism, but it is a food approximation of the region where the casual polymorphism might be. Information from Q-Q plot guide to use the K or Q + K model as the guideline to show the Manhattan plots results.

References

1. R Core Team (2013) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org/
2. Endelman JB (2011) Ridge regression and other kernels for genomic selection with R package rrBLUP. Plant Genome 4(3):250–255
3. Gondro C, Van der Werf J, Hayes B (eds) (2013) Genome-wide association studies and genomic prediction. Humana, New York
4. Yu J, Pressoir G, Briggs WH, Bi IV, Yamasaki M, Doebley JF, McMullen MD, Gaut BS, Nielsen DM, Holland JB, Kresovich S, Buckler ES (2006) A unified mixed-model method for

association mapping that accounts for multiple levels of relatedness. Nat Genet 38:203–208

5. Cantor RM, Lange K, Sinsheimer JS (2010) Prioritizing GWAS results: a review of statistical methods and recommendations for their application. Am J Hum Genet 86:6–22
6. Poland JA, Rife TW (2012) Genotyping-by-sequencing for plant breeding and genetics. Plant Genome 5:92–102
7. Pritchard JK, Donnelly P (2001) Case-control studies of association in structured or admixed populations. Theor Popul Biol 60:227–237
8. Marchini J, Cardon LR, Phillips MS, Donnelly P (2004) The effects of human population structure on large genetic association studies. Nat Genet 36:512–517
9. Price AL, Zaitlen NA, Reich D, Patterson N (2010) New approaches to population stratification in genome-wide association studies. Nat Rev Genet 11:459–463
10. Larsson SJ, Lipka AE, Buckler ES (2013) Lessons from Dwarf8 on the strengths and weaknesses of structured association mapping. PLoS Genet 9:e1003246