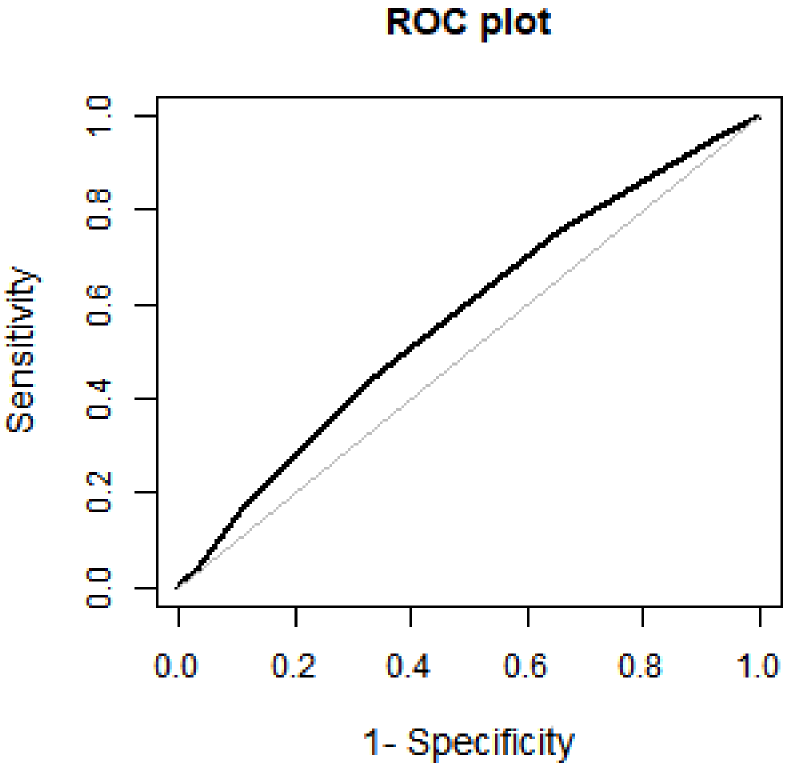


FIGURE 4.8
ROC curve of the genetic score used to predict case/control status in the asthma example

We observe that the risk of asthma increases 21% per each risk allele. The predictive power of the genetic score can be assessed by computing the area under the ROC curve (AUC)

```
> predrisk <- predRisk(mod.lin, dd.end.complete)
> plotROC(data=dd.end.complete, cOutcome=1,
+         predrisk = predrisk)
```

```
AUC [95% CI] for the model 1 :  0.574 [ 0.54  -  0.607 ]
```

Figure 4.8 shows that the predictive power of the genetic score is 57.4% with a confidence interval al 95% of (54.0 - 60.7).

4.5 Genome Wide Association Studies

Genome-wide association studies (GWASs) assess the association between the trait of interest and up to millions of SNPs. GWASs have been used to discover thousands of SNPs associated with several complex diseases [77]. The basic statistical methods are similar to those previously described, in particular, the massive univariate testing. The main issue with GWASs is data management and computation. Most publicly available data is in PLINK format, where genomic data is stored in a binary BED file, and phenotype and annotation data in text BIM and FAM files. PLINK data can be loaded into `R` with the Bioconductor's package *snpStats* (see section 3.2).

We illustrate the analysis of a GWAS including 100,000 SNPs that have been simulated using real data from a case-control study. Our phenotype of interest is obesity (0: no obese; 1:obese) that has been created using body mass index information of each individual. We start by loading genotype data that are in PLINK format (`obesity.bed`, `obesity.bim`, `obesity.fam` files) in our *brgedata* package

```
> library(snpStats)
> path <- system.file("extdata", package="brgedata")
> ob.plink <- read.plink(file.path(path, "obesity"))
```

The imported object is a list containing the genotypes, the family structure and the SNP annotation.

```
> names(ob.plink)
[1] "genotypes" "fam"       "map"
```

We store genotype, annotation and family data in different variables for downstream analyses

```
> ob.geno <- ob.plink$genotypes
> ob.geno
A SnpMatrix with  2312 rows and  100000 columns
Row names:  100 ... 998
Col names:  MitoC3993T ... rs28600179
> annotation <- ob.plink$map
> head(annotation)
           chromosome   snp.name cM position allele.1 allele.2
MitoC3993T         NA MitoC3993T NA     3993        T        C
MitoG4821A         NA MitoG4821A NA     4821        A        G
MitoG6027A         NA MitoG6027A NA     6027        A        G
MitoT6153C         NA MitoT6153C NA     6153        C        T
MitoC7275T         NA MitoC7275T NA     7275        T        C
MitoT9699C         NA MitoT9699C NA     9699        C        T
> family <- ob.plink$fam
> head(family)
     pedigree member father mother sex affected
100    FAM_OB    100     NA     NA   1        1
```

```
1001   FAM_OB   1001     NA     NA   1        1
1004   FAM_OB   1004     NA     NA   2        2
1005   FAM_OB   1005     NA     NA   1        2
1006   FAM_OB   1006     NA     NA   2        1
1008   FAM_OB   1008     NA     NA   1        1
```

Notice that `geno` is as object of class *SnpMatrix* that stores the SNPs in binary (raw) format. While some basic phenotype data is usually available in the `fam` field of the *SnpMatrix* object, a more complete phenotypic characterization of the sample is usually distributed in additional text files. In our example, the complete phenotype data is in a tab-delimited file

```
> ob.pheno <- read.delim(file.path(path, "obesity.txt"))
> head(ob.pheno)
    id gender obese age   smoke country
1 4180   Male     1  41 Current      50
2 4880 Female    NA  35      Ex      51
3  435   Male     1  50      Ex      53
4 4938   Male     0  44 Current      53
5 2977   Male    NA  49   Never      53
6 1705   Male     0  40   Never      50
```

The file contains phenotypic information for a different set of individuals that overlap with those in the `ob.geno` object. Therefore, before analysis, we need to correctly merge and order the individuals across genomic and phenotype datasets. The row names of `ob.geno` correspond to the individual identifiers (id) variable of `ob.pheno`. Consequently, we also rename the the rows of `ob.pheno` with the `id` variable

```
> rownames(ob.pheno) <- ob.pheno$id
```

We can check if the row names of the datasets match

```
> identical(rownames(ob.pheno), rownames(ob.geno))
[1] FALSE
```

`FALSE` indicates that either there are different individuals in both objects or that they are in different order. This can be fixed by selecting common individuals.

```
> ids <- intersect(rownames(ob.pheno), rownames(ob.geno))
> geno <- ob.geno[ids, ]
> ob <- ob.pheno[ids, ]
> identical(rownames(ob), rownames(geno))
[1] TRUE
> family <- family[ids, ]
```

4.5.1 Quality control of SNPs

We now perform the quality control (QC) of genomic data at the SNP and individual levels, before association testing [3]. Different measures can be used to perform QC of SNPs: 1) SNPs with high rate of missing; 2) rare SNPS (e.g. having low minor allele frequency - MAF); and 3) SNPs that do not pass the HWE test.

Typically, markers with a call rate less than 95% are removed from association analyses, although some large studies chose higher call-rate thresholds (99%). Markers of low MAF (<5%) are also filtered. The significance threshold rejecting a SNPs for not being in HWE has varied greatly between studies, from thresholds between 0.001 and 5.7×10^{-7} [19]). Including SNPs with extremely low P-values for HWE test will require individual examination of the SNP genotyping process. A parsimonious threshold of 0.001 may be considered, though robustly genotyped SNPs below this threshold may remain in the study [3], as deviations from HWE may indeed arise from biological processes.

The function `col.summary` offers different summaries (at SNP level) that can be used in QC

```
> info.snps <- col.summary(geno)
> head(info.snps)
           Calls Call.rate Certain.calls       RAF          MAF
MitoC3993T  2286 0.9887543             1 0.9851269 0.0148731409
MitoG4821A  2282 0.9870242             1 0.9982472 0.0017528484
MitoG6027A  2307 0.9978374             1 0.9956654 0.0043346337
MitoT6153C  2308 0.9982699             1 0.9893847 0.0106152513
MitoC7275T  2309 0.9987024             1 0.9991338 0.0008661758
MitoT9699C  2302 0.9956747             1 0.9268028 0.0731972198
                   P.AA         P.AB      P.BB     z.HWE
MitoC3993T 0.0148731409 0.0000000000 0.9851269 -47.81213
MitoG4821A 0.0017528484 0.0000000000 0.9982472 -47.77028
MitoG6027A 0.0043346337 0.0000000000 0.9956654 -48.03124
MitoT6153C 0.0103986135 0.0004332756 0.9891681 -47.05069
MitoC7275T 0.0008661758 0.0000000000 0.9991338 -48.05206
MitoT9699C 0.0729800174 0.0004344049 0.9265856 -47.82555
```

snpStats does not compute P-values of HWE test but computes its z-scores. A P-value of 0.001 corresponds to a z-score of ±3.3 for a two-tail test. Strictly speaking, HWE test should should be applied to controls only (e.g. `obese = 0`), however, the default computation is for all samples.

We thus filter SNPs with a call rate > 95%, MAF of > 5% and $z.HWE < 3.3$ in controls

```
> controls <- ob$obese ==0 & !is.na(ob$obese)
> geno.controls <- geno[controls,]
> info.controls <- col.summary(geno.controls)
>
> use <- info.snps$Call.rate > 0.95 &
+        info.snps$MAF > 0.05 &
+        abs(info.controls$z.HWE < 3.3)
> mask.snps <- use & !is.na(use)
>
> geno.qc.snps <- geno[ , mask.snps]
> geno.qc.snps
A SnpMatrix with  2312 rows and  88723 columns
```

```
Row names:  4180 ... 277
Col names:  MitoT9699C ... rs28562204
> annotation <- annotation[mask.snps, ]
```

It is common practice to report the number of SNPs that have been removed from the association analyses

```
> # number of SNPs removed for bad call rate
> sum(info.snps$Call.rate < 0.95)
[1] 888
> # number of SNPs removed for low MAF
> sum(info.snps$MAF < 0.05, na.rm=TRUE)
[1] 10461
> #number of SNPs that do not pass HWE test
> sum(abs(info.controls$z.HWE > 3.3), na.rm=TRUE)
[1] 80
> # The total number of SNPs do not pass QC
> sum(!mask.snps)
[1] 11277
```

4.5.2 Quality control of individuals

QC of individuals, or biological samples, comprises four main steps: 1) The identification of individuals with discordant reported and genomic sex, 2) the identification of individuals with outlying missing genotype or heterozygosity rate, 3) the identification of duplicated or related individuals, and 4) the identification of individuals of divergent ancestry from the sample [3].

We start by removing individuals with sex discrepancies, large number of missing genotypes and outlying heterozygosity. The function `row.summary` returns the call rate and the proportion of called SNPs which are heterozygous per individual.

```
> info.indv <- row.summary(geno.qc.snps)
> head(info.indv)
     Call.rate Certain.calls Heterozygosity
4180 0.9998873             1      0.3426781
4880 0.9998197             1      0.3539180
435  0.9958297             1      0.3392188
4938 0.9994928             1      0.3411782
2977 0.9985348             1      0.3426004
1705 0.9936657             1      0.3357721
```

Gender is usually inferred from the heterozygosity of chromosome X. Males have an expected heterozygosity of 0 and females of 0.30. Chromosome X heterozygosity can be extracted using `row.summary` function and and then plotted

```
> geno.X <- geno.qc.snps[,annotation$chromosome=="23" &
+                          !is.na(annotation$chromosome)]
> info.X <- row.summary(geno.X)
```

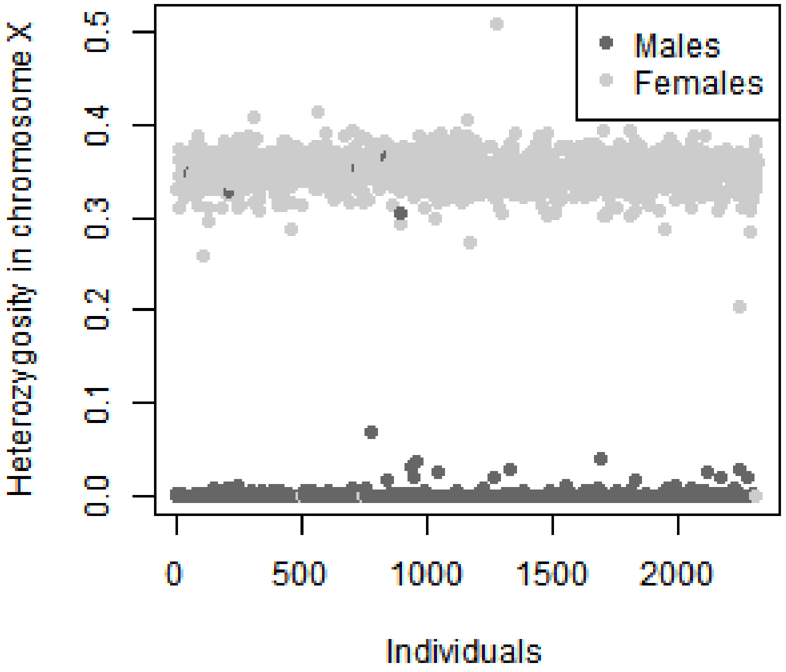


FIGURE 4.9
Heterozygosity in chromosome X by gender provided in the phenotypic data.

```
> mycol <- ifelse(ob$gender=="Male", "gray40", "gray80")
> plot(info.X$Heterozygosity, col=mycol,
+      pch=16, xlab="Individuals",
+      ylab="Heterozygosity in chromosome X")
> legend("topright", c("Males", "Females"), col=mycol,
+        pch=16)
```

Figure 4.9 shows that there are some reported males with non-zero X-heterozygosity and females with zero X-heterozygosity. These samples are located in `sex.discrep` for latter removal

```
> sex.discrep <- (ob$gender=="Male" & info.X$Heterozygosity > 0.2) |
+                (ob$gender=="Female" & info.X$Heterozygosity < 0.2)
```

Sex filtering based in X-heterozygosity is not sufficient to identify rare aneuploidies, like XXY in males. Alternatively, plots of the mean allelic intensities of SNPs on the X and Y chromosomes can identify mis-annotated sex as well as sex chromosome aneuploidies.

Now, we identify individuals with outlying heterozygosity from the overall genomic heterozigosity rate that is computed by `row.summary`. Heterozigosity, can also be computed from the statisitic $F = 1 - \frac{f(Aa)}{E(f(Aa))}$, where $f(Aa)$ is the observed proportion of heterozygous genotypes (Aa) of a given individual and $E(f(Aa))$ is the expected proportion of heterozygous genotypes. A subject's $E(f(Aa))$ can be computed from the MAF across all the subjects's non-missing SNPs

```
> MAF <- col.summary(geno.qc.snps)$MAF
> callmatrix <- !is.na(geno.qc.snps)
> hetExp <- callmatrix %*% (2*MAF*(1-MAF))
> hetObs <- with(info.indv, Heterozygosity*(ncol(geno.qc.snps))*Call.rate)
> info.indv$hetF <- 1-(hetObs/hetExp)
>
> head(info.indv)
     Call.rate Certain.calls Heterozygosity         hetF
4180 0.9998873             1      0.3426781  0.023324353
4880 0.9998197             1      0.3539180 -0.008701237
435  0.9958297             1      0.3392188  0.033025203
4938 0.9994928             1      0.3411782  0.027596273
2977 0.9985348             1      0.3426004  0.023487306
1705 0.9936657             1      0.3357721  0.042762824
```

In figure 4.10, we compare F statistic and the `Heterozygosity` field obtained from `row.summary`

Individuals whose F statistic is outside the band ±0.1 are considered sample outlyers (left panel Figure 4.10) and correspond to those having an Heterozygosity rate lower than 0.32.

GWASs are typically studies that are based on population samples. Therefore, close familial relatedness between individuals is not representative of the sample. We therefore search individuals whose relatedness is higher than expected. The package *SNPRelate* is used to perform identity-by-descent (IBD) analysis, computing kinship within the sample. The package requires a data in a GDS format that is obtained with the function `snpgdsBED2GDS`. In addition, IBD analysis requires SNPs that are not in LD (uncorrelated). The function `snpgdsLDpruning` iteratively removes adjacent SNPs that exceed an LD threshold in a sliding window

```
> library(SNPRelate)
>
> # Transform PLINK data into GDS format
> snpgdsBED2GDS("obesity.bed",
+               "obesity.fam",
+               "obesity.bim",
+               out="obGDS")
Start snpgdsBED2GDS ...
BED file: "obesity.bed" in the SNP-major mode (Sample X SNP)
```

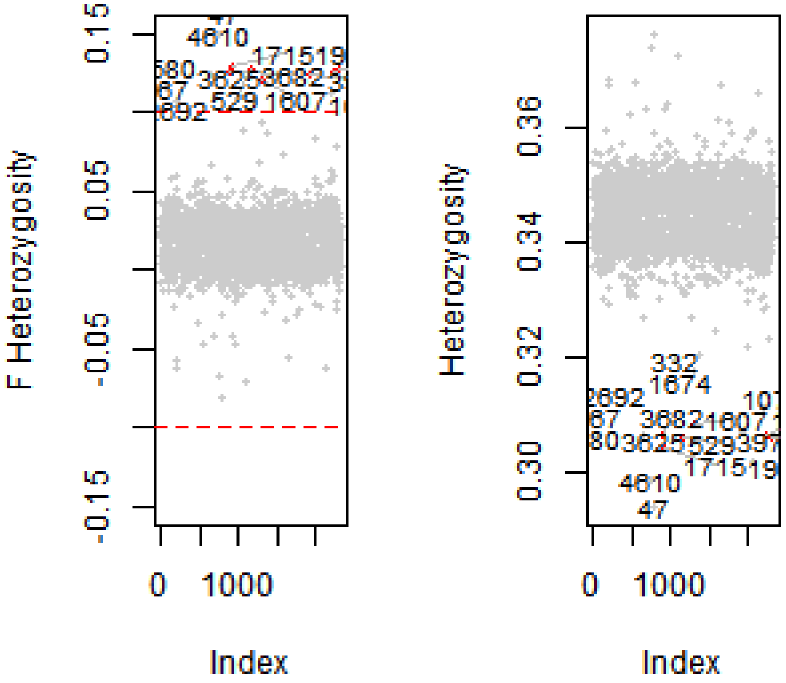


FIGURE 4.10
Heterozygosity computed using F statistic (left panel) and using t row.summary (right panel). The horizontal red line shows a suggestive value to detect individuals with outlier heterozygosity values.

```
FAM file: "obesity.fam", DONE.
BIM file: "obesity.bim", DONE.
Fri Jun 22 18:12:16 2018  store sample id, snp id, position, and chromosome.
start writing: 2312 samples, 100000 SNPs ...
  Fri Jun 22 18:12:16 2018 0%
  Fri Jun 22 18:12:17 2018 100%
Fri Jun 22 18:12:17 2018  Done.
Optimize the access efficiency ...
Clean up the fragments of GDS file:
    open the file 'obGDS' (55.7M)
    # of fragments: 39
    save to 'obGDS.tmp'
    rename 'obGDS.tmp' (55.7M, reduced: 252B)
    # of fragments: 18
> genofile <- snpgdsOpen("obGDS")
>
> #Prune SNPs for IBD analysis
> set.seed(12345)
> snps.qc <- colnames(geno.qc.snps)
> snp.prune <- snpgdsLDpruning(genofile, ld.threshold = 0.2,
+                            snp.id = snps.qc)
SNP pruning based on LD:
Excluding 13,410 SNPs (non-autosomes or non-selection)
Excluding 0 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
Working space: 2,312 samples, 86,590 SNPs
    using 1 (CPU) core
    sliding window: 500,000 basepairs, Inf SNPs
    |LD| threshold: 0.2
    method: composite
Chromosome 1: 31.51%, 2,433/7,721
Chromosome 2: 30.04%, 2,418/8,050
Chromosome 3: 30.84%, 2,059/6,676
Chromosome 4: 31.13%, 1,845/5,927
Chromosome 5: 30.87%, 1,875/6,074
Chromosome 6: 28.19%, 1,903/6,750
Chromosome 7: 31.24%, 1,673/5,356
Chromosome 8: 28.82%, 1,606/5,572
Chromosome 9: 31.52%, 1,487/4,718
Chromosome 10: 30.63%, 1,590/5,191
Chromosome 11: 31.12%, 1,485/4,772
Chromosome 12: 31.53%, 1,531/4,855
Chromosome 13: 31.19%, 1,136/3,642
Chromosome 14: 32.59%, 1,059/3,249
Chromosome 15: 31.80%, 973/3,060
Chromosome 16: 35.65%, 1,060/2,973
Chromosome 17: 36.97%, 1,006/2,721
Chromosome 18: 34.22%, 1,008/2,946
Chromosome 19: 40.80%, 694/1,701
Chromosome 20: 35.87%, 864/2,409
Chromosome 21: 34.62%, 485/1,401
Chromosome 22: 34.61%, 552/1,595
30,742 markers are selected in total.
> snps.ibd <- unlist(snp.prune, use.names=FALSE)
```

Note that this process is performed with SNPs that passed previous QC checks. IBD coefficients are then computed using the method of moments, implemented in snpgdsIBDMoM. The result of the analysis is a table indicating kinship among pairs of individuals

```
> ibd <- snpgdsIBDMoM(genofile, kinship=TRUE,
+                     snp.id = snps.ibd,
+                     num.thread = 1)
IBD analysis (PLINK method of moment) on genotypes:
Excluding 69,258 SNPs (non-autosomes or non-selection)
Excluding 0 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
Working space: 2,312 samples, 30,742 SNPs
    using 1 (CPU) core
PLINK IBD:    the sum of all selected genotypes (0,1,2) = 32844904
Fri Jun 22 18:12:22 2018    (internal increment: 6656)

[..................................................]  0%, ETC: ---
[==================================================] 100%, completed in 8s
Fri Jun 22 18:12:31 2018    Done.
> ibd.kin <- snpgdsIBDSelection(ibd)
> head(ibd.kin)
  ID1  ID2        k0         k1     kinship
1 100 1001 0.9926611 0.00191261 0.003191305
2 100 1004 1.0000000 0.00000000 0.000000000
3 100 1005 1.0000000 0.00000000 0.000000000
4 100 1006 1.0000000 0.00000000 0.000000000
5 100 1008 1.0000000 0.00000000 0.000000000
6 100 1013 1.0000000 0.00000000 0.000000000
```

A pair of individuals with higher than expected relatedness are considered with kinship score > 0.1

```
> ibd.kin.thres <- subset(ibd.kin, kinship > 0.1)
> head(ibd.kin.thres)
         ID1  ID2        k0        k1   kinship
46484   1049  188 0.2731024 0.5431008 0.2276736
232848  1202 1330 0.0000000 0.0000000 0.5000000
281069  1237  872 0.2747742 0.4556623 0.2486973
640474   155 1682 0.2410303 0.4506688 0.2668176
806337   170 2015 0.2548016 0.5399619 0.2376087
1158509 2055  825 0.0000000 0.0000000 0.5000000
```

The ids of the individuals with unusual kinship are located with related form the SNPassoc package

```
> ids.rel <-  related(ibd.kin.thres)
> ids.rel
 [1] "4364" "3380" "2999" "2697" "2611" "2088" "1202" "872"  "825"  "684"
[11] "188"  "170"  "155"  "2071"
```

Summing up, individuals with more than 3-7% missing genotypes[25, 125], with sex discrepancies, F absolute value > 1 and kinship coefficient > 0.1 are removed from the genotype and phenotype data

```
> use <- info.indv$Call.rate > 0.95 &
+        abs(info.indv$hetF) < 0.1 &
+        !sex.discrep &
+        !rownames(info.indv)%in%ids.rel
> mask.indiv <- use & !is.na(use)
> geno.qc <- geno.qc.snps[mask.indiv, ]
>
> ob.qc <- ob.pheno[mask.indiv, ]
> identical(rownames(ob.qc), rownames(geno.qc))
[1] TRUE
```

These QC measures are usually reported

```
> # number of individuals removed to bad call rate
> sum(info.indv$Call.rate < 0.95)
[1] 34
> # number of individuals removed for heterozygosity problems
> sum(abs(info.indv$hetF) > 0.1)
[1] 15
> # number of individuals removed for sex discrepancies
> sum(sex.discrep)
[1] 8
> # number of individuals removed to be related with others
> length(ids.rel)
[1] 14
> # The total number of individuals that do not pass QC
> sum(!mask.indiv)
[1] 70
```

4.5.3 Population ancestry

As GWAS are studies based on general population samples, individual genetic differences between individuals need to be also representative of the population at large. The main source of genetic differences between individuals is ancestry. Therefore, it is important to check that there are not individuals with unexpected genetic differences in the sample. Ancestral differences can be inferred with principal component analysis (PCA) on the genomic data. Individuals with outlying ancestry can be removed from the study while smaller differences in ancestry can be adjusted in the association models, including the first principal components as covariates.

PCA on genomic data can be computed using the *SNPRelate* package with the `snpgdsPCA` function. Efficiency can be improved by removing SNPs that are in LD before PCA, see `snps.ibd`) in the prevoious IBD analysis. In addition `snpgdsPCA` allows parallelization with the argument `num.thread` that determines the number of computing cores to be used

```
> pca <- snpgdsPCA(genofile, sample.id = rownames(geno.qc),
+                            snp.id = snps.ibd,
+                            num.thread=1)
Principal Component Analysis (PCA) on genotypes:
```

```
Excluding 69,258 SNPs (non-autosomes or non-selection)
Excluding 0 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
Working space: 2,242 samples, 30,742 SNPs
    using 1 (CPU) core
PCA:    the sum of all selected genotypes (0,1,2) = 31854924
CPU capabilities: Double-Precision SSE2
Fri Jun 22 18:13:13 2018    (internal increment: 216)

[..................................................]  0%, ETC: ---
[==================================================] 100%, completed in 35s
Fri Jun 22 18:13:48 2018    Begin (eigenvalues and eigenvectors)
Fri Jun 22 18:13:52 2018    Done.
```

A PCA plot for the first two components can be obtained with

```
> with(pca, plot(eigenvect[,1], eigenvect[,2],
+                xlab="1st Principal Component",
+                ylab="2nd Principal Component",
+                main = "Ancestry Plot"))
```

Inspection of figure 4.11 can be used to identify individuals with unussual ancestry and remove them. Individuals with outlying values in the principal components will be considered for QC. In our example, we can see outlying individuals in the right side of the plot with 1st PC > 0.05. Smaller differences in ancestry are an important source of bias in association tests, as explained later. Therefore, we keep the first five principal components and add it to the phenotypic information that will be used in the association analyses

```
> ob.qc <- data.frame(ob.qc, pca$eigenvect[, 1:5])
```

After performing QC, the GDS file can be closed

```
> closefn.gds(genofile)
```

4.5.4 Genome-wise association analysis

Genome-wise association analysis involves regressing each SNP separately on our trait of interest. The analyses should be adjusted for clinical, environmental, and/or demographic factors as well as ancestral differences between the subjects. The analysis can be performed with a range of function in *snpStats* package. We first examine the unadjusted whole genome association of our obesity study

```
> res <- single.snp.tests(obese, data=ob.qc,
+                         snp.data=geno.qc)
> res[1:5,]
               N Chi.squared.1.df Chi.squared.2.df      P.1df     P.2df
MitoT9699C  2134        3.0263311               NA 0.08192307        NA
MitoA11252G 2090        0.3561812               NA 0.55063478        NA
```

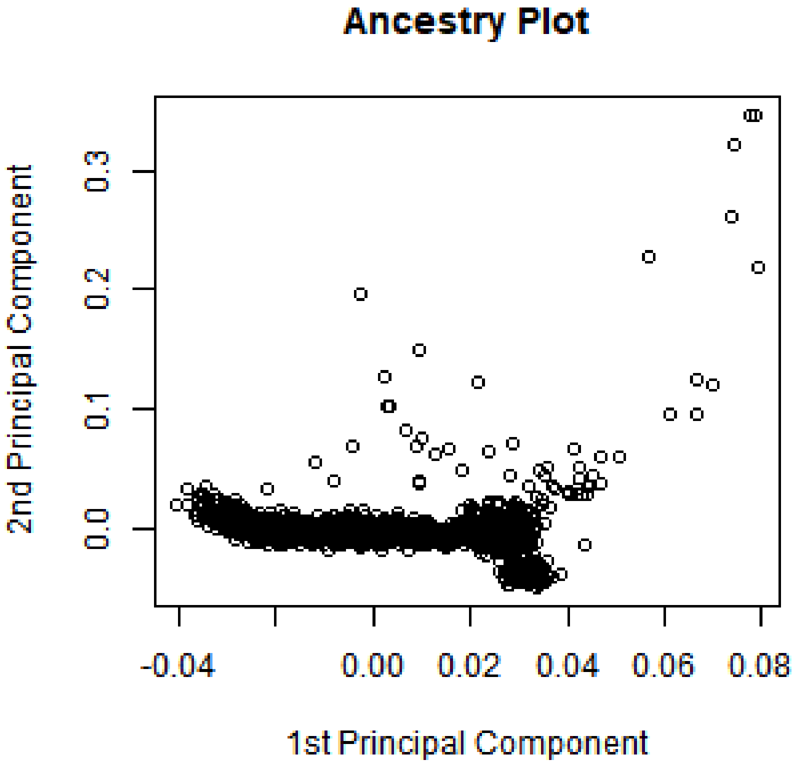


FIGURE 4.11
1st and 2nd principal components of obesity GWAS data example.

```
MitoA12309G 2136        0.1776464               NA 0.67340371        NA
MitoG16130A 2069        2.4766387        3.9480296 0.11554896 0.1388981
rs28705211  2125        0.7277258        0.7546827 0.39362135 0.6856820
```

This analysis is only available for the additive (χ^2(1.df)) and the codominant models (χ^2(2.df)). It requires the name variable phenotype (`obese`) in the `data` argument. Genomic data are given in the `snp.data` argument. It is important that the individuals in the rows of both datasets match. SNPs in the mitocondrial genome and gonosomes return NA for the χ^2C estimates. These variants should be analyzed separately. A common interest is to analyze autosomes only, and therefore these SNPs can be removed in the QC process.

A quantitative traits can also be analyzed setting the argument `family` equal to `Gaussian`

```
> res.quant <- snp.rhs.tests(age ~ 1,  data=ob.qc,
+                            snp.data=geno.qc,
+                            family="Gaussian")
> head(res.quant)
          Chi.squared Df  p.value
MitoT9699C 0.003422591  1 0.953348
```

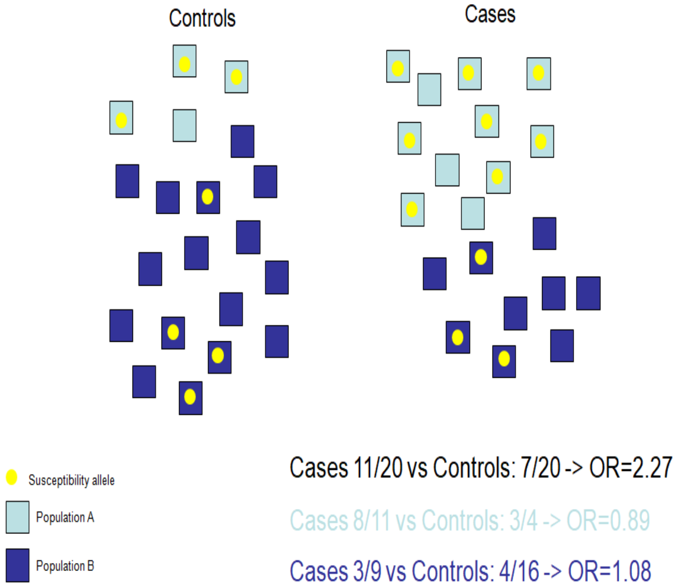
4.5.5 Adjusting for population stratification

Population stratification inflates the estimates of the χ^2 tests of association between the phenotype and the SNPs, and as a consequence the false positive rate increases. Figure 4.5.5 illustrate why population stratification may lead to false associations. In the hypothetical study in the figure, we compare 20 cases and 20 controls where individuals carrying a susceptibility allele are denoted by a yellow dot. The overall frequency of the susceptibility allele is much larger in cases (0.55 = 11/20) than in controls (0.35 = 7/20), the odds of being case in allele carriers is 2.3 times higher than the odds of being case in none carriers (OR= 2.27 = (0.55/0.45) / (0.35/0.65)). However the significant increase in susceptibility between the allele is misleading, as the OR in population A (light blue color) is 0.89 and in population B (dark blue color) is 1.08. The susceptibility allele strongly discriminates population A from B, and given the differences of the trait frequency between populations, it is likely that the association of the allele with the trait is through its links with population differences and not with the trait itself.

In genome-wide analyses the inflation of the associations due to undetected latent variables is assessed by quantile-quantile (Q-Q) plots where observed χ^2 values are plotted against the expected ones

```
> chi2 <- chi.squared(res, df=1)
> qq.chisq(chi2)
```

FIGURE 4.12
Illustrative example of population stratification. Read Section 4.2 for a detailed description.


Controls
Cases
Susceptibility allele
Population A
Population B
Cases 11/20 vs Controls: 7/20 -> OR=2.27
Cases 8/11 vs Controls: 3/4 -> OR=0.89
Cases 3/9 vs Controls: 4/16 -> OR=1.08

```
           N       omitted        lambda
88723.000000      0.000000      1.003853
```

Figure 4.13 shows, in particular, that the χ^2 estimates are not inflated (λ is also close to 1), as all quantile values fall in the confidence bands, meaning that most SNPs are not associated with obesity. In addition, the figure does not show any top SNP outside the confidence bands. A Q-Q plot with top SNPs outside the confidence bands indicate that those SNPs are truly associated with the disease and, hence, do not follow the null hypothesis. Therefore, the Q-Q plot of our examples reveals no significnat SNP associations.

Q-Q plots are used to inspect population stratification. In particular, when population stratification is present, most SNP Q-Q values will be found outside the confidence bands, suggesting that the overall genetic structure of the sample can discriminate differences between subject traits. The λ value is a measure of the degree of inflation. The main source of population stratification that is derived from genomic data is ancestry. Therefore, in the cases of inflated Q-Q plots, it is ancestry differences and not individual SNP differences that explain the differences in the phenotype. Population stratification may be corrected by genomic control, mixed models or EIGENSTRAT method [102]. However, the most common approach is to use the infer ancestry from genomic data as covariates in the association analyses [101]. Genome-wide as-

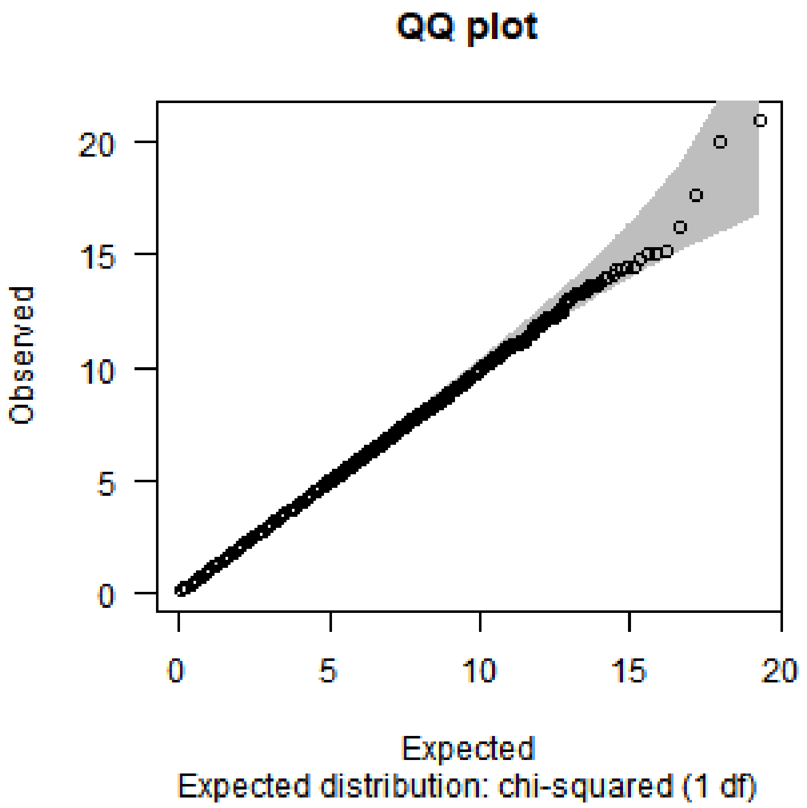


FIGURE 4.13
QQ-plot corresponding to obesity GWAS data example

sociation analysis typically adjust for population stratification using the PCs on genomic data to infer ancestral differences in the sample. Covariates are easily incorporated in the model of `snp.rhs.tests`

```
> res.adj <- snp.rhs.tests(obese ~ X1 + X2 + X3 + X4 + X5,
+                          data=ob.qc, snp.data=geno.qc)
> head(res.adj)
          Chi.squared Df    p.value
MitoT9699C    3.108601  1 0.07787985
```

This function only computes the additive model, adjusting for the first five genomic PCs. The resulting $-\log_{10}(P)$-values of association for each SNP are then extracted

```
> pval.log10 <- -log10(p.value(res.adj))
```

These transformed P-values are used to create a Manahattan plot to visualize which SNPs are significantly associated with obesity. Manahattan plots are implemented in the *qqman* package.

```
> library(qqman)
> # Create the required data frame
> pvals <- data.frame(SNP=annotation$snp.name,
+                    CHR=annotation$chromosome,
+                    BP=annotation$position,
+                    P=p.value(res.adj))
> # missing data is not allowed
> pvals <- subset(pvals, !is.na(CHR) & !is.na(P))
>
> manhattan(pvals, suggestiveline=TRUE, genomewideline=TRUE,
+          annotatePval = 1e-4, annotateTop = FALSE,
+          main="GWAS obesity", col=c("black", "gray"))
```

Significance at Bonferroni level is set at $10^{-7} = 0.05/10^5$, as we are testing 100,000 SNPs. The level corresponds to $-\log_{10}(P) = 6.30$. Therefore, we confirm, as expected form the Q-Q plot, that no SNP in our study is significanlty associated with obesity, as observed in figure 4.14.

With our obesity example, we illustrate the common situation of finding no significant associations in small studies (thousands of subjects) with small genomic data (100,000 SNPs). This situation motivates multi-center studies with larger samples sizes, where small effects can be inferred with sufficient power and consistency.

4.6 Post-GWAS visualization and interpretation

The main aim of genomic association studies is the identification of *any* variant that is significanlty associated with phenotype differences. The analysis does