AA_vs_EA

Tae Kim

3/23/2019

1. Data pre-processing

We use the RNA-seq expression level data provided by GTEx, which has been already pre-processed. We choose to study not-sun-exposed suprapubic skin tissue as a pilot study because we expect to see some ancestry effect on the expression level on the skin tissue. Genes were selected based on expression threshold of > 0.1 RPKM in at least 10 individuals and $\geq$ reads in at least 10 individuals. Then the expression values were quantile normalized to the average empirical distribution observed across samples. That means, each gene has the same distribution across the individuals. Then they were inverse quantile normalized to fit standard normal distribution.

(a)load and clean the data

We read the expression level data and match the individuals with the ancestry information.

```
## [1]   190 25192
```

(b) scale the data and regress out the mean

We code African Americans as 0 and European Americans as 1 to compare group by group coexpression difference. We also consider correlation matrix, so we scale the gene expression level matrix to have variance 1. Then we regress out ancestry from the gene expression level to remove the mean effect.

```
orig_A = X$A
X$A[X$A > 0] = 1
X$A = scale(X$A) * sqrt(length(X$A)) / sqrt(length(X$A)-1)
Y = resid(lm(Y~X$A))
# Y = scale(Y) * sqrt(length(X$A)) / sqrt(length(X$A)-1)
```

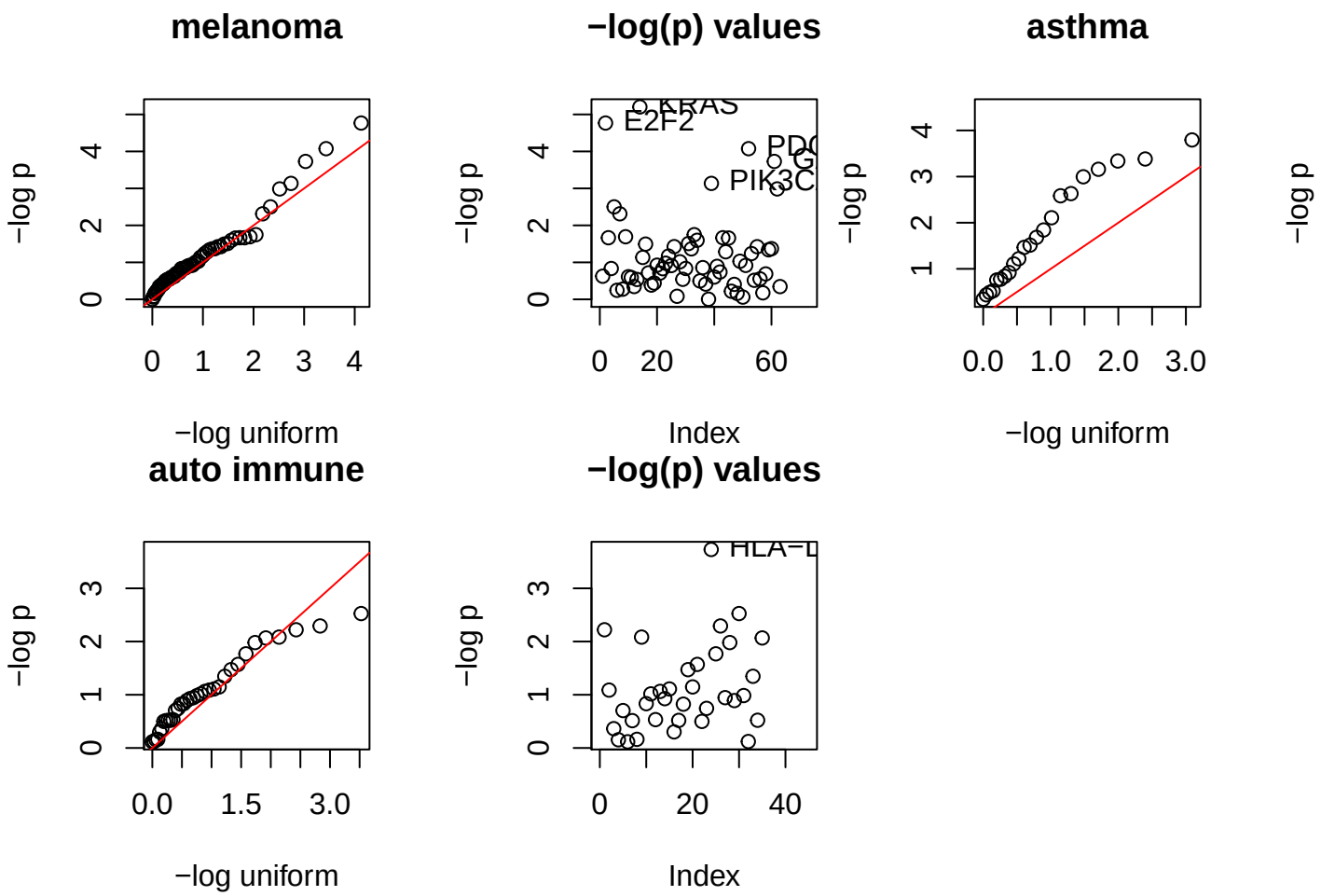
2. Gene selection

In order to avoid extensive multiple testing, we use prior information in biology.

(a) KEGG pathway

We believe the disease pathways may reflect the gene-gene correlation structure or gene modules, so we choose three pathways that are related to either skin tissue or immunity - melonoma, asthma, and autoimmune disease.

For each pathway, we take a group of K genes and for each k in $1, \cdots, K$, we compute d_k and compare it to the empirical distribution by permuting the ancestry. Then we show qqplot of the p values against the uniform distribution after transformation $-log_{10}(p)$.

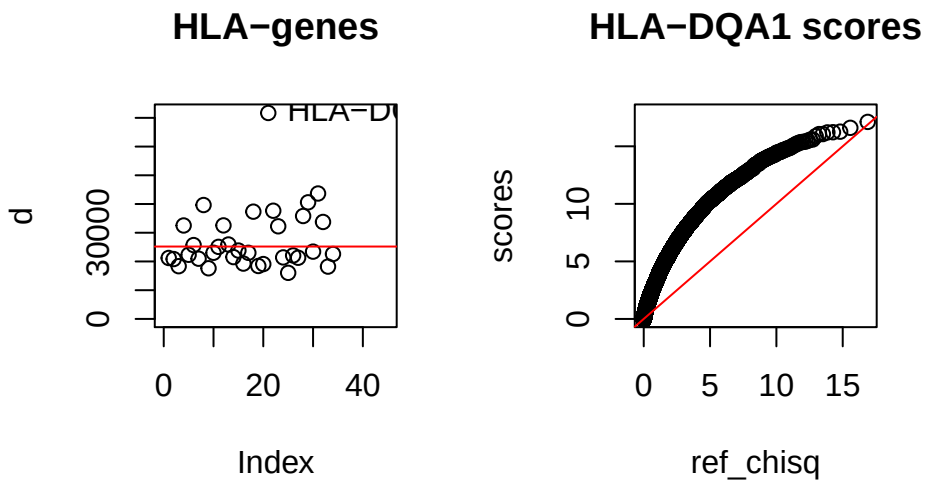

melanoma
−log(p) values
asthma
−log p
−log uniform
Index
KRAS
E2F2
PIK3C
auto immune
HLA−D

(b) HLA genes

We know that HLA genes are connected to many other genes and play critical role in regulating the gene transcription. We obtain the list of HLA genes and compute d_1 by adding up pair-wise test statistics across the entire genome.

```
hla_genes = c("HLA-A", "HLA-B", "HLA-C", "HLA-E",
              "HLA-F", "HLA-G", "HLA-H", "HLA-J",
              "HLA-K", "HLA-L", "HLA-N", "HLA-P",
              "HLA-S", "HLA-T", "HLA-U", "HLA-V",
              "HLA-W", "HLA-X", "HLA-Y", "HLA-Z",
              "HLA-DRA", "HLA-DRB2", "HLA-DRB3",
              "HLA-DRB4", "HLA-DRB5", "HLA-DRB6",
              "HLA-DRB7", "HLA-DRB8", "HLA-DRB9",
              "HLA-DQA1", "HLA-DQB1", "HLA-DQA2",
              "HLA-DQB2", "HLA-DQB3", "HLA-DOA",
              "HLA-DMA", "HLA-DMB", "HLA-DPA1", "HLA-DPB1",
              "HLA-DPA2", "HLA-DPB2", "HLA-DPA3",
              "HFE", "TAP1", "TAP2", "PSMB9", "PSMB8",
              "MICA", "MICB", "MICC", "MICD", "MICE")

results = getBM(attributes = c('hgnc_symbol','ensembl_gene_id'),
                filters = "hgnc_symbol",
                values = hla_genes,
                mart = ensembl)

genes = colnames(Y)[which(colnames(Y) %in% results$ensembl_gene_id)]
results = results[match(genes, results$ensembl_gene_id), ]

scores = newscores = matrix(0, length(genes), ncol(Y)-1)
coef = cubic_coeff_c(X$A, qchisq(1-0.05/ncol(Y), 1), 2, 2)
for (i in 1:length(genes)){
  ind = which(colnames(Y) == results$ensembl_gene_id[i])
  W = store_W_c(Y[,ind], Y[, -ind])
  scores[i,] = get_score_W_c(X$A, W)
  for (j in 1:ncol(scores)){
    roots = polyroot(c(-scores[i,j], coef))
    newscores[i,j] = Re(roots)[abs(Im(roots)) < 1e-6]
  }
}
```

And the result is plotted below, where the red line is the expected mean. HLA-DQA1 stands out in its test statistic.



(c) Transcription Factors

read TF data

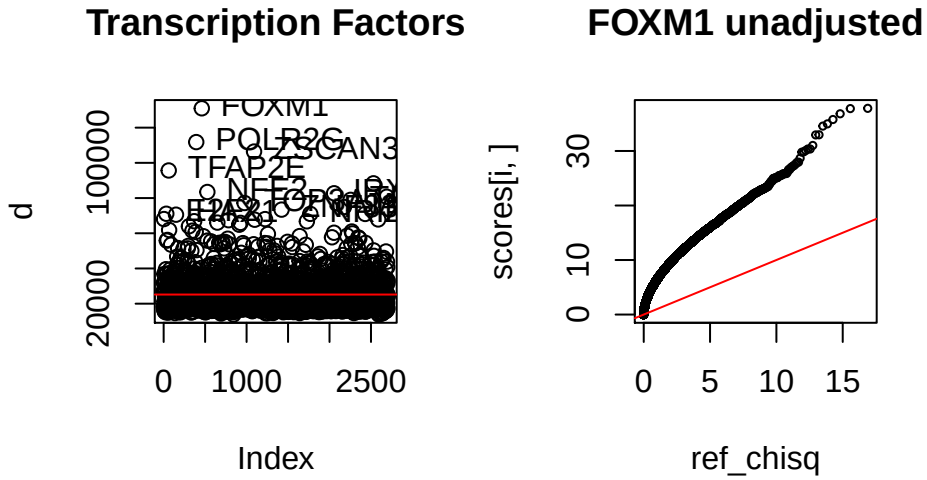
Since transcription factors usually regulate the transcription of multiple genes, we thought they were appropriate targets to study their variance in activity across different ancestry.

```
TF = fread("../data/TFcheckpoint.txt", header=TRUE)
TF = setDF(TF)
TF = TF[,1]
results = getBM(attributes = c('hgnc_symbol','ensembl_gene_id'),
                filters = "hgnc_symbol",
                values = TF,
                mart = ensembl)

genes = colnames(Y)[which(colnames(Y) %in% results$ensembl_gene_id)]
results = results[match(genes, results$ensembl_gene_id), ]
```

test all the TFs

```
par(mfrow = c(1,2))
scores = matrix(0, length(genes), ncol(Y)-1)
for (i in 1:length(genes)){
  ind = which(colnames(Y) == results$ensembl_gene_id[i])
  W = store_W_c(Y[,ind], Y[, -ind])
  scores[i,] = get_score_W_c(X$A, W)
}

d = rowSums(scores)
df = data.frame(d = d, genes = results$hgnc_symbol, xaxis = 1:length(d))
plot(d,
     ylab = 'd',
     main = 'Transcription Factors')
abline(h=25190, col='red')
ind = which(d > ncol(Y) + 200 * sqrt(2*ncol(Y)))
df = df[ind, ]
```

```r
with(df, text(d ~ xaxis, labels = genes, pos = 4))

ref_chisq = qchisq(seq(0,1, length=ncol(Y)-1), 1)
for (i in 1:length(genes)){
    if(results$hgnc_symbol[i] %in% c("FOXM1")){
      qqplot(ref_chisq, scores[i, ],
            main = paste(results$hgnc_symbol[i], 'unadjusted'),
            cex = 0.5)
      abline(0,1,col = 'red')
  }
}
```

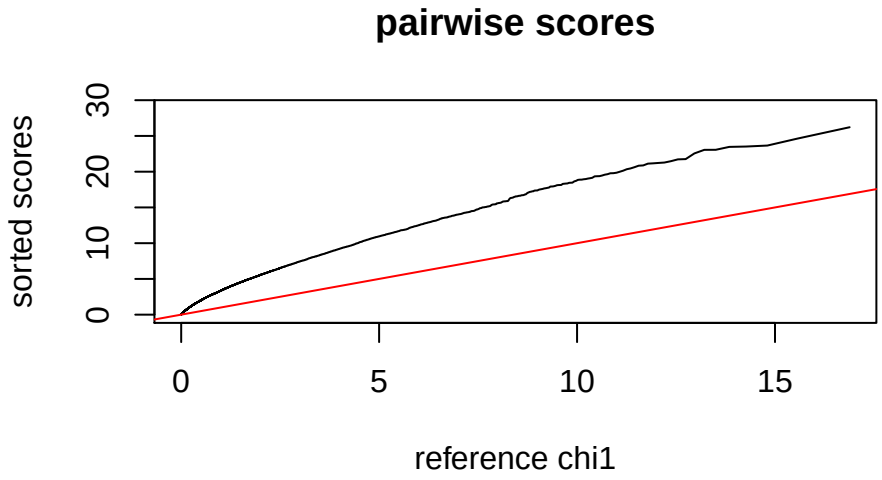


Investigating the top signals

HLA-DQA1

HLA-DQA1 Score test

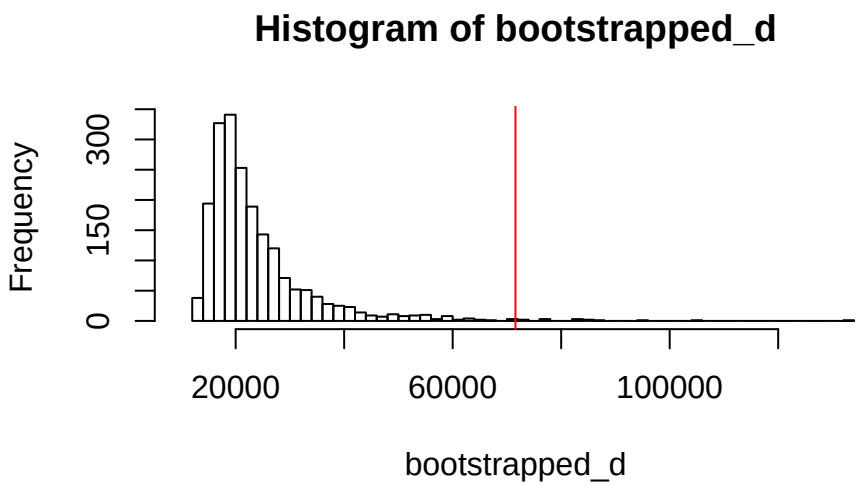
We re-cap what we have observed in HLA-DQA1 for the score tests.



HLA-DQA1, permutation test

In order to verify if HLA-DQA truly has a signal, we perform permutation test. The p-value is indeed very small.

```
W = store_W_c(Y[,dqa_ind], Y[, -dqa_ind])
dqa_scores = get_score_W_c(X$A, W)
dqa_d = get_degree_c(X$A, Y[,dqa_ind], Y[,-dqa_ind])
# shuffled_A = shuffle_x_c(X$A, B)
storeW = store_W_c(Y[,dqa_ind], Y[,-dqa_ind])
out = bootstrap_c(X$A, B, storeW)
bootstrapped_d = rowSums(out)
dqa_p = sum(bootstrapped_d > dqa_d) / B
hist(bootstrapped_d, 50, xlim = c(10000, max(max(bootstrapped_d), dqa_d))+20)
abline(v=dqa_d, col = 'red')
```




```
print(dqa_p)
```

```
## [1] 0.0075
```

```
# par(mfrow = c(2,3))
# for (i in 1:12){
#   qqplot(ref_chisq, out[i, ], cex = 0.5, type = 'l', ylab = 'permuted score')
#   abline(0,1,col = 'red')
# }
```

Genes with the highest scores for HLA-DQA1

We make some diagnosis plot to observe what is driving the signal.

```
max_scores = sort(s, decreasing=TRUE, index.return=TRUE)
results = getBM(attributes = c('hgnc_symbol','ensembl_gene_id'),
                filters = "ensembl_gene_id",
                values = colnames(Y[,-dqa_ind])[max_scores$ix[1:10]],
                mart = ensembl)
results = results[match(results$ensembl_gene_id, colnames(Y[,-dqa_ind])[max_scores$ix[1:10]]), ]
top_genes = data.frame(gene_name = results$hgnc_symbol, score = max_scores$x[1:10],
                ensembl = results$ensembl_gene_id)
# print(top_genes)
AA_ind = which(orig_A > 0)
EA_ind = which(orig_A == 0)
diff_cor = data.frame(genes = top_genes$gene_name,
                      AA = as.numeric(cor(orig[AA_ind,dqa_ind],
                                          (orig[AA_ind,-dqa_ind])[, max_scores$ix[1:10]])),
                      EA = as.numeric(cor(orig[EA_ind,dqa_ind],
                                          (orig[EA_ind,-dqa_ind])[, max_scores$ix[1:10]])))
print(cbind(top_genes, diff_cor[,2:3]))
```

```
##     gene_name    score         ensembl        AA          EA
## 1        LRBA 28.85008 ENSG00000198589 0.3478526 -0.14944539
## 2     PLA2G4E 26.21730 ENSG00000188089 0.4189482 -0.10035988
## 3      OSBPL6 24.63537 ENSG00000079156 0.4538671 -0.12932814
## 4        PKP4 23.65608 ENSG00000144283 0.4588957 -0.15284831
## 5       EIF4B 23.51561 ENSG00000063046 0.3425440 -0.08661835
## 6       PRSS3 23.46787 ENSG00000010438 0.4111445 -0.09436705
```

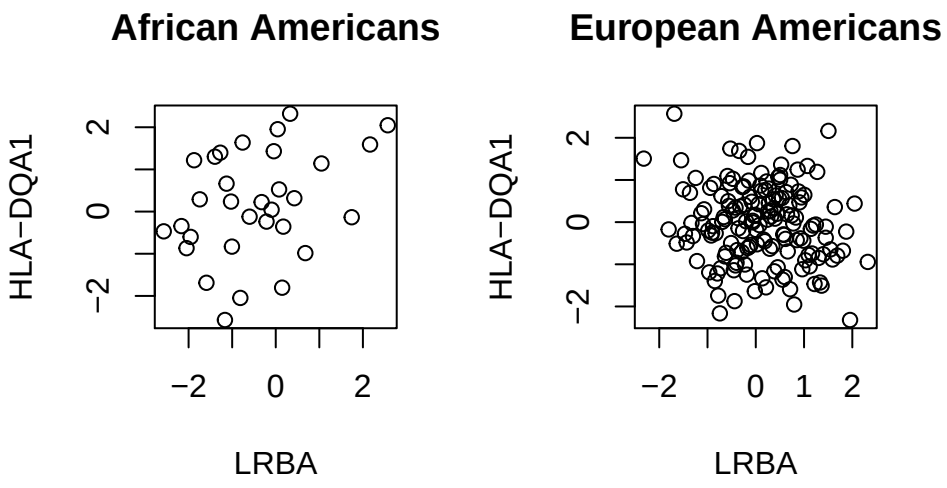
```
## 7             23.06378 ENSG00000272008 0.2536646 -0.01610219
## 8  SUCLA2-AS1 23.05285 ENSG00000227848 0.2267855 -0.14215591
## 9       UPK1B 22.55323 ENSG00000114638 0.4782435  0.04157245
## 10      SUDS3 21.76107 ENSG00000111707 0.1836475 -0.22015849
```

The expression level of HLA-DQA1's top signal LRBA

```
AA_ind = which(orig_A > 0)
EA_ind = which(orig_A == 0)
diff_cor = data.frame(genes = top_genes$gene_name,
                      AA = as.numeric(cor(orig[AA_ind,dqa_ind],
                                          (orig[AA_ind,-dqa_ind])[, max_scores$ix[1:10]])),
                      EA = as.numeric(cor(orig[EA_ind,dqa_ind],
                                          (orig[EA_ind,-dqa_ind])[, max_scores$ix[1:10]])))
print(diff_cor)
```

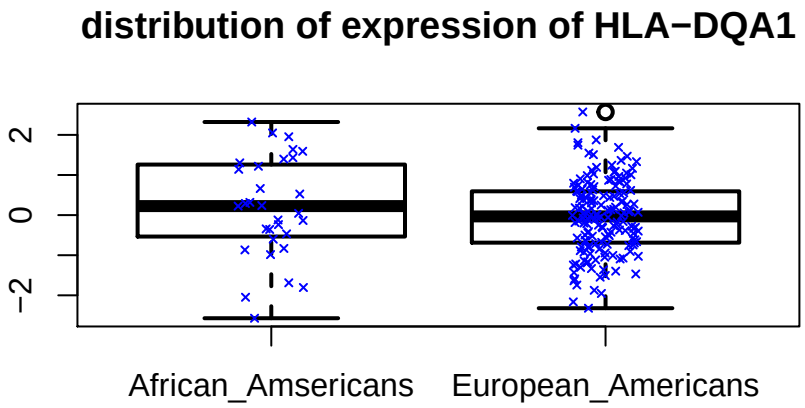
```
##         genes        AA          EA
## 1        LRBA 0.3478526 -0.14944539
## 2     PLA2G4E 0.4189482 -0.10035988
## 3      OSBPL6 0.4538671 -0.12932814
## 4        PKP4 0.4588957 -0.15284831
## 5       EIF4B 0.3425440 -0.08661835
## 6       PRSS3 0.4111445 -0.09436705
## 7             0.2536646 -0.01610219
## 8  SUCLA2-AS1 0.2267855 -0.14215591
## 9       UPK1B 0.4782435  0.04157245
## 10      SUDS3 0.1836475 -0.22015849
```

```
par(mfrow = c(1,2))
plot(orig[AA_ind, dqa_ind] ~ orig[AA_ind,-dqa_ind][, max_scores$ix[1]],
     main = 'African Americans',
     xlab = 'LRBA',
     ylab = 'HLA-DQA1')
plot(orig[EA_ind, dqa_ind] ~ orig[EA_ind,-dqa_ind][, max_scores$ix[1]],
     main = 'European Americans',
     xlab = 'LRBA',
     ylab = 'HLA-DQA1')
```



HLA-DQA1, box plot

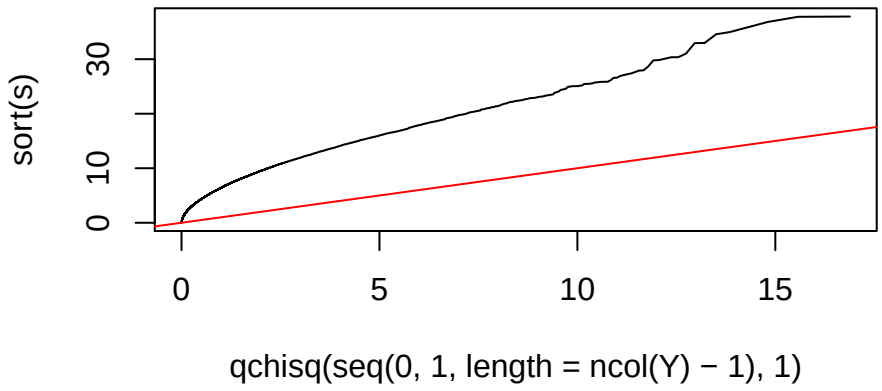
```
AA_ind = which(orig_A > 0)
EA_ind = which(orig_A == 0)
dqa_exp = as.matrix(orig[,dqa_ind])
dqa_exp_aa = dqa_exp[AA_ind]
dqa_exp_ea = dqa_exp[EA_ind]

df = data.frame(exp = dqa_exp, label = c(rep("African_Amsericans", length(AA_ind)),rep("European_America

boxplot(exp ~ label, data = df, lwd = 2, main = "distribution of expression of HLA-DQA1")
stripchart(exp ~ label, vertical = TRUE, data = df,
    method = "jitter", add = TRUE, pch = 4, col = 'blue', cex=0.5)
```



FOXM1

FOXM1 Score test

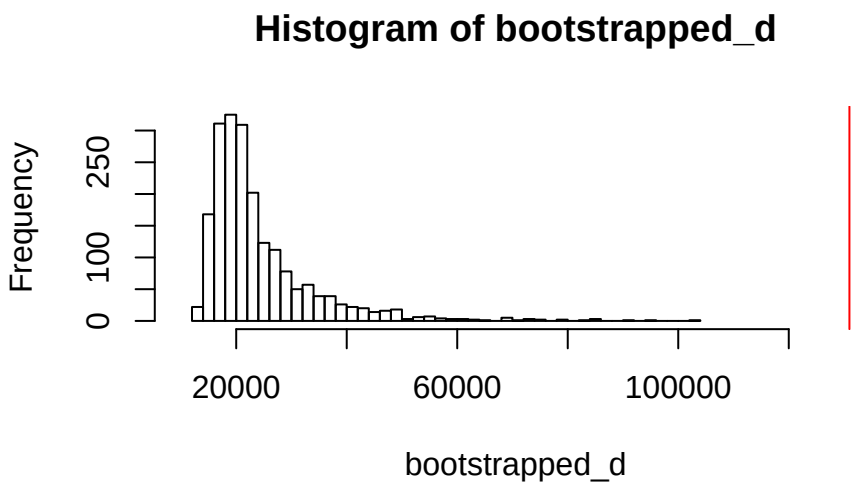
```
gene = "FOXM1"
results = getBM(attributes = c('hgnc_symbol','ensembl_gene_id'),
                filters = "hgnc_symbol",
                values = gene,
                mart = ensembl)

fox_ind = which(colnames(Y) %in% c(results$ensembl_gene_id))
smallY = Y[,-fox_ind]
s = news = rep(0, ncol(Y)-1)
for (i in 1:(ncol(Y)-1)){
  s[i] = get_score_c(X$A, Y[,fox_ind], smallY[,i])
}
plot(sort(s) ~ qchisq(seq(0,1,length=ncol(Y)-1), 1), type = 'l'); abline(0,1,col = 'red', main = 'adjus
```



FOXM1, permutation test

We conduct the permutation test to see if the signal is real.

```
fox_ind = which(colnames(Y) %in% results[results$hgnc_symbol=="FOXM1", 2])
W = store_W_c(Y[,fox_ind], Y[, -fox_ind])
fox_scores = get_score_W_c(X$A, W)
fox_d = get_degree_c(X$A, Y[,fox_ind], Y[,-fox_ind])
shuffled_A = shuffle_x_c(X$A, B)
storeW = store_W_c(Y[,fox_ind], Y[,-fox_ind])
out = bootstrap_c(X$A, B, storeW)
bootstrapped_d = rowSums(out)
fox_p = sum(bootstrapped_d > fox_d) / B
par(mfrow = c(1,1))
hist(bootstrapped_d, 50, xlim = c(10000, max(max(bootstrapped_d), fox_d))+100)
abline(v=fox_d, col = 'red')
```




```
print(fox_p)
```

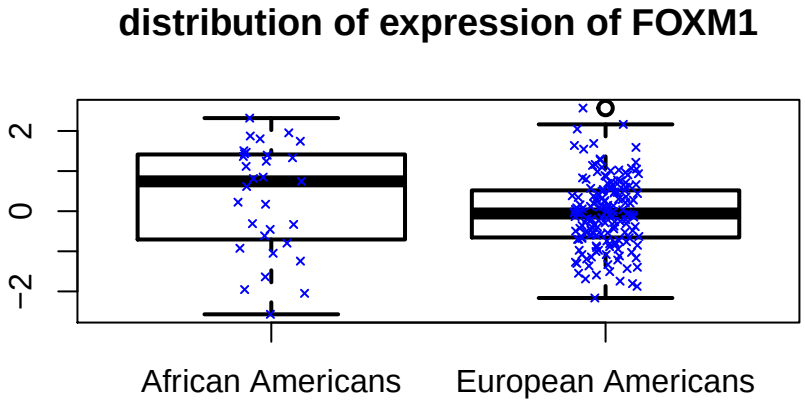
```
## [1] 0
```

```
# par(mfrow = c(2,3))
# for (i in 1:12){
#   qqplot(ref_chisq, out[i, ], cex = 0.5, type = 'l', ylab = 'permuted score')
#   abline(0,1,col = 'red')
# }
```

FOXM1, box plot

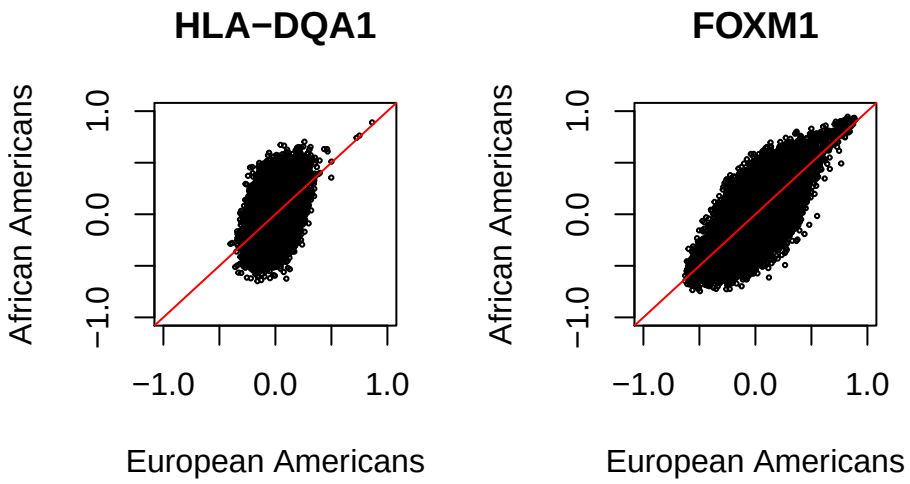
We inspect the raw data to see any patterns between Afircan Americans and European Americans.

```
AA_ind = which(orig_A > 0)
EA_ind = which(orig_A == 0)
fox_exp = as.matrix(orig[,fox_ind])
fox_exp_aa = fox_exp[AA_ind]
fox_exp_ea = fox_exp[EA_ind]

df = data.frame(exp = fox_exp, label = c(rep("African Americans", length(AA_ind)),rep("European Americar

boxplot(exp ~ label, data = df, lwd = 2, main = "distribution of expression of FOXM1")
stripchart(exp ~ label, vertical = TRUE, data = df,
    method = "jitter", add = TRUE, pch = 4, col = 'blue', cex=0.5)
```



Correlation of HLA-DQA1 and FOXM1 against all other genes

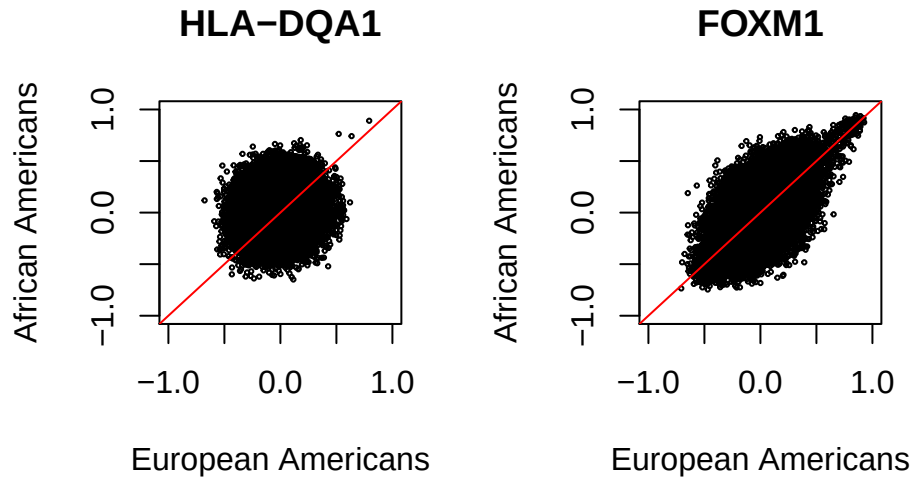
```
diff_cor_fox = data.frame(genes = colnames(orig)[-fox_ind],
                      AA = as.numeric(cor(orig[AA_ind,fox_ind],
                                          (orig[AA_ind,-fox_ind]))),
                      EA = as.numeric(cor(orig[EA_ind,fox_ind],
                                          (orig[EA_ind,-fox_ind]))))
diff_cor_dqa = data.frame(genes = colnames(orig)[-dqa_ind],
                      AA = as.numeric(cor(orig[AA_ind,dqa_ind],
                                          (orig[AA_ind,-dqa_ind]))),
                      EA = as.numeric(cor(orig[EA_ind,dqa_ind],
                                          (orig[EA_ind,-dqa_ind]))))
par(mfrow = c(1,2))
plot(diff_cor_dqa$AA ~ diff_cor_dqa$EA, cex=0.3, ylim = c(-1,1), xlim = c(-1,1), main='HLA-DQA1',
     ylab = 'African Americans',
     xlab = 'European Americans'); abline(0,1,col = 'red')

plot(diff_cor_fox$AA ~ diff_cor_fox$EA, cex=0.3, ylim = c(-1,1), xlim = c(-1,1), main='FOXM1',
     ylab = 'African Americans',
     xlab = 'European Americans'); abline(0,1,col = 'red')
```



What happens when I reduce the sample size of European Americans?

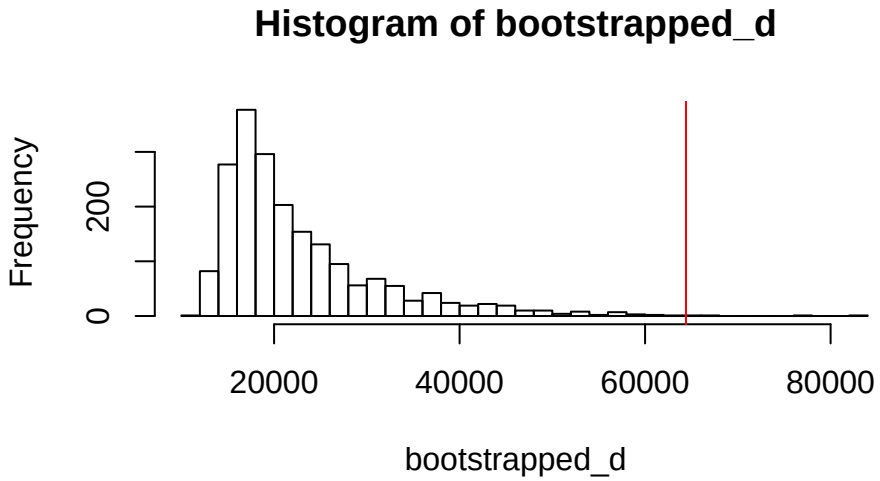
```
EA_subset = sample(EA_ind, size=31)

diff_cor_fox = data.frame(genes = colnames(orig)[-fox_ind],
                      AA = as.numeric(cor(orig[AA_ind,fox_ind],
                                          (orig[AA_ind,-fox_ind]))),
                      EA = as.numeric(cor(orig[EA_subset,fox_ind],
                                          (orig[EA_subset,-fox_ind]))))
diff_cor_dqa = data.frame(genes = colnames(orig)[-dqa_ind],
                      AA = as.numeric(cor(orig[AA_ind,dqa_ind],
                                          (orig[AA_ind,-dqa_ind]))),
                      EA = as.numeric(cor(orig[EA_subset,dqa_ind],
                                          (orig[EA_subset,-dqa_ind]))))
par(mfrow = c(1,2))
```

```r
plot(diff_cor_dqa$AA ~ diff_cor_dqa$EA, cex=0.3, ylim = c(-1,1), xlim = c(-1,1), main='HLA-DQA1',
     ylab = 'African Americans',
     xlab = 'European Americans'); abline(0,1,col = 'red')

plot(diff_cor_fox$AA ~ diff_cor_fox$EA, cex=0.3, ylim = c(-1,1), xlim = c(-1,1), main='FOXM1',
     ylab = 'African Americans',
     xlab = 'European Americans'); abline(0,1,col = 'red')
```



Repeat HLA-DQA1 with sampled Euopreans

```r
W = store_W_c(Y[c(AA_ind, EA_subset),dqa_ind], Y[c(AA_ind, EA_subset), -dqa_ind])
dqa_scores = get_score_W_c(X$A[c(AA_ind, EA_subset)], W)
dqa_d = get_degree_c(X$A[c(AA_ind, EA_subset)], Y[c(AA_ind, EA_subset),dqa_ind], Y[c(AA_ind, EA_subset)
storeW = store_W_c(Y[c(AA_ind, EA_subset),dqa_ind], Y[c(AA_ind, EA_subset),-dqa_ind])
out = bootstrap_c(X$A[c(AA_ind, EA_subset)], B, storeW)
bootstrapped_d = rowSums(out)
dqa_p = sum(bootstrapped_d > dqa_d) / B
hist(bootstrapped_d, 50, xlim = c(10000, max(max(bootstrapped_d), dqa_d))+20)
abline(v=dqa_d, col = 'red')
```

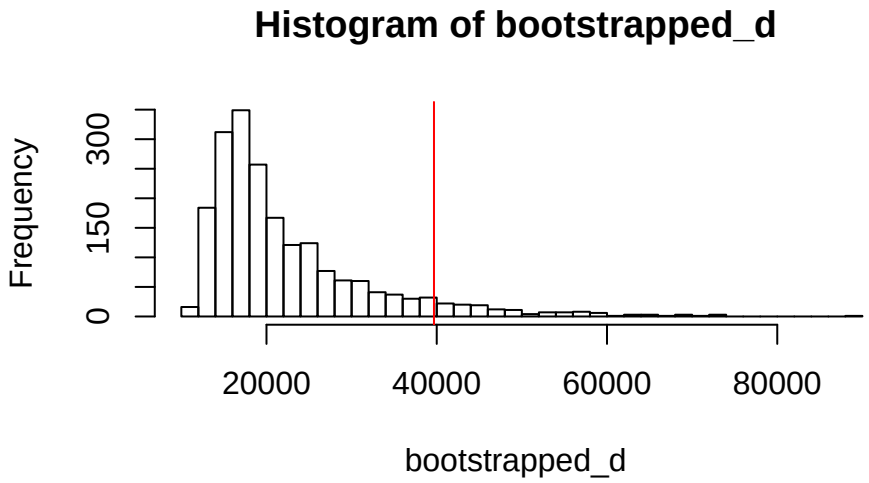


```r
print(dqa_p)
```

```
## [1] 0.002
```

Repeat FOX-M1 with sampled Euopreans

```r
W = store_W_c(Y[c(AA_ind, EA_subset),fox_ind], Y[c(AA_ind, EA_subset), -fox_ind])
fox_scores = get_score_W_c(X$A[c(AA_ind, EA_subset)], W)
fox_d = get_degree_c(X$A[c(AA_ind, EA_subset)], Y[c(AA_ind, EA_subset),fox_ind], Y[c(AA_ind, EA_subset)
storeW = store_W_c(Y[c(AA_ind, EA_subset),fox_ind], Y[c(AA_ind, EA_subset),-fox_ind])
out = bootstrap_c(X$A[c(AA_ind, EA_subset)], B, storeW)
bootstrapped_d = rowSums(out)
fox_p = sum(bootstrapped_d > fox_d) / B
hist(bootstrapped_d, 50, xlim = c(10000, max(max(bootstrapped_d), fox_d))+20)
abline(v=fox_d, col = 'red')
```




```r
print(fox_p)
```

```
## [1] 0.069
```