

```

# install.packages("devtools")
library(devtools)

## Warning: package 'devtools' was built under R version 3.4.1
# source("http://bioconductor.org/biocLite.R")
# biocLite("qvalue")
# install_github("whitlock/OutFLANK")
library(OutFLANK)

## Loading required package: qvalue
if(!( "adegenet" %in% installed.packages())){install.packages("adegenet")}

library(vcfR)

##
##      *****
## This is vcfR 1.5.0
##      browseVignettes('vcfR') # Documentation
##      citation('vcfR') # Citation
##      *****
library(adegenet)

## Loading required package: ade4
## Warning: package 'ade4' was built under R version 3.4.1
##
##      /// adegenet 2.0.1 is loaded ///////////
##      > overview: '?adegenet'
##      > tutorials/doc/questions: 'adegenetWeb()'
##      > bug reports/feature requests: adegenetIssues()
library(pcadapt)
library(lfmm)
library(LEA)

## devtools::install_github("privefl/bigsnpr")
library(bigsnpr)

## Loading required package: bigstatsr
library(bigstatsr)
library(vcfR)
library(RColorBrewer)
library(ggplot2)
library(fields)

## Loading required package: spam
## Loading required package: grid
## Spam version 1.4-0 (2016-08-29) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the

```

```

## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: maps

seed <- 1505550948364
vcf <- read.vcfR(paste0("evals/",seed,"_Invers_VCFallsim1.vcf.gz"))

##
Meta line 13 read in.
## All meta lines processed.
## Character matrix gt created.
## Character matrix gt rows: 13011
## Character matrix gt cols: 1009
## skip: 0
## nrows: 13011
## row_num: 0
##
##
Processed variant 1000
Processed variant 2000
Processed variant 3000
Processed variant 4000
Processed variant 5000
Processed variant 6000
Processed variant 7000
Processed variant 8000
Processed variant 9000
Processed variant 10000
Processed variant 11000
Processed variant 12000
Processed variant 13000
Processed variant: 13011
## All variants processed

ind0 <- read.table(paste0("evals/",seed,"_Invers_outputIndAll.txt"), header=TRUE)
muts <- read.table(paste0("evals/",seed,"_Invers_outputMuts.txt"), header=TRUE)
phen_env <- read.table(paste0("evals/",seed,"_Invers_outputPhenEnv.txt"), header=TRUE)
sim <- system(paste0("grep recomb evals/",seed,"_Invers_outputSim.txt"), TRUE)

```

## Count mutations that contribute at least 1% to genetic variance

```

muts$pa2 <- round(muts$selCoef^2*muts$freq*(1-muts$freq),3)
muts$prop=NA
muts$prop[muts$type=="m2"] <- muts$pa2[muts$type=="m2"]/sum(muts$pa2[muts$type=="m2"])
which(duplicated(muts$position))

## integer(0)

```

```

muts$count <- FALSE
muts$count [muts$prop>=0.01] <- TRUE
muts$count [muts$type=="m4" & muts$freq > 0.05] <- TRUE

muts

##   position    selCoef originGen type   freq   pa2      prop count
## 1      51475  0.1688910      5948 m2 0.0070 0.000 0.00000000 FALSE
## 2      51581 -0.3446080      5987 m2 0.0080 0.001 0.01098901 TRUE
## 3      52211  0.0336784     1650 m2 0.8740 0.000 0.00000000 FALSE
## 4      53012  0.2757600     5395 m2 0.0340 0.002 0.02197802 TRUE
## 5      53800 -1.0843600     5999 m2 0.0010 0.001 0.01098901 TRUE
## 6      54966 -0.4380020      172  m2 1.0000 0.000 0.00000000 FALSE
## 7      59393 -0.2582460     5997 m2 0.0005 0.000 0.00000000 FALSE
## 8      65350 -0.4411480     5999 m2 0.0005 0.000 0.00000000 FALSE
## 9      66103 -0.3656770     6000 m2 0.0005 0.000 0.00000000 FALSE
## 10     67017  0.1294500     2702 m2 0.8465 0.002 0.02197802 TRUE
## 11     67058  0.4800990       53  m2 0.5045 0.058 0.63736264 TRUE
## 12     69724  0.2111330     5999 m2 0.0005 0.000 0.00000000 FALSE
## 13     69766 -0.3176070     4318 m2 0.0770 0.007 0.07692308 TRUE
## 14     70769 -0.5431350     5988 m2 0.0080 0.002 0.02197802 TRUE
## 15     73015  0.2998850     5995 m2 0.0020 0.000 0.00000000 FALSE
## 16     73701  0.1616660     1623 m2 1.0000 0.000 0.00000000 FALSE
## 17     77187 -0.4676320     6000 m2 0.0005 0.000 0.00000000 FALSE
## 18     79305 -0.1617850     5975 m2 0.0020 0.000 0.00000000 FALSE
## 19     79805  0.6184820     5999 m2 0.0010 0.000 0.00000000 FALSE
## 20     79995 -0.1745930     3902 m2 0.5485 0.008 0.08791209 TRUE
## 21     81126  0.2498160     5937 m2 0.0270 0.002 0.02197802 TRUE
## 22     85950 -0.1117880     5915 m2 0.0210 0.000 0.00000000 FALSE
## 23     89096 -0.5685820     6000 m2 0.0005 0.000 0.00000000 FALSE
## 24     95659 -0.5738520     5997 m2 0.0010 0.000 0.00000000 FALSE
## 25     103054 0.2102690     4228 m2 0.0650 0.003 0.03296703 TRUE
## 26     105727 0.0738757     4943 m2 0.1240 0.001 0.01098901 TRUE
## 27     110242 -1.1506800     5999 m2 0.0005 0.001 0.01098901 TRUE
## 28     112525  0.0880859     2707 m2 0.5750 0.002 0.02197802 TRUE
## 29     113610  0.9898900     5999 m2 0.0005 0.000 0.00000000 FALSE
## 30     114325 -0.6368310     5996 m2 0.0010 0.000 0.00000000 FALSE
## 31     115003  0.4981430     5996 m2 0.0005 0.000 0.00000000 FALSE
## 32     122439  0.8204620     6000 m2 0.0005 0.000 0.00000000 FALSE
## 33     123604 -0.0105875     4019 m2 0.6595 0.000 0.00000000 FALSE
## 34     129666 -0.1067380     5994 m2 0.0035 0.000 0.00000000 FALSE
## 35     130660 -0.2060310      768  m2 1.0000 0.000 0.00000000 FALSE
## 36     137299  0.9273030     6000 m2 0.0005 0.000 0.00000000 FALSE
## 37     137919 -0.0938825     5800 m2 0.1820 0.001 0.01098901 TRUE
## 38     148281 -0.5184100     5998 m2 0.0005 0.000 0.00000000 FALSE
## 39     175000  0.8000000     5780 m4 1.0000 0.000          NA TRUE

```

## Set up genetic map for figures

```

#### Color recombination regions #####
lgs <- seq(50000, 500000, by=50000) # linkage groups recombination breakpoints 0.5
lg_whereplot <- lgs - 25000

```

```

(recom_rates <- as.numeric(unlist(strsplit(sim[1], " "))-1))

## [1] 1.00000e-05 5.00000e-01 1.00000e-05 5.00000e-01 1.00000e-05
## [6] 5.00000e-01 1.00000e-05 5.00000e-01 1.00000e-05 5.00000e-01
## [11] 1.00000e-05 5.00000e-01 1.00000e-05 5.00000e-01 1.00000e-05
## [16] 1.00000e-08 1.00000e-05 5.00000e-01 1.56098e-07 2.91295e-02
## [21] 2.85170e-03 3.92680e-06 2.92384e-05 5.58755e-08 4.77181e-05
## [26] 3.50242e-07 8.43974e-07 1.60727e-05 5.00000e-01 1.00000e-05

(recom_end <- as.integer(unlist(strsplit(sim[2], " "))-1))

## [1] 50000 50001 100000 100001 150000 150001 200000 200001 250000 250001
## [11] 300000 300001 350000 350001 370000 380000 400000 400001 400052 413592
## [21] 415805 418228 422879 428034 428082 433429 438943 450000 450001 500000

recom <- data.frame(recom_rates, recom_end)
recom$logrates <- log10(recom_rates)
# plot r=0.5 as black
# plot r=1e-11 as white
(brks <- with(recom, c(-12, -9, -6, -4, -2, 0.5)))

## [1] -12.0 -9.0 -6.0 -4.0 -2.0 0.5

grps <- with(recom, cut(logrates, breaks = brks, include.lowest = TRUE))
nlevels(grps)

## [1] 5

colfunc <- paste(colorRampPalette(colors=c( rgb(0,0,1,0.1), rgb(1,1,1,0), rgb(0,1,0,0.1)))(length(brks)))
recom$col <- colfunc[grps]
plot(recom$logrates, col=recom$col)

```

Index	recom\$logrates (approx.)	Group
2	-1.0	Green
3	-1.0	Green
6	-1.0	Green
8	-1.0	Green
10	-1.0	Green
12	-1.0	Green
14	-1.0	Green
17	-7.5	Dark Blue
19	-6.5	Medium Blue
20	-2.5	Light Blue
21	-3.5	Light Blue
24	-7.0	Medium Blue
26	-6.5	Medium Blue
27	-6.0	Medium Blue
28	-0.5	Green

```

### Replace chromosome 1 with actual chromosome positions #####
ends=c(0,lgs)
dim(vcf@gt)

```

```

## [1] 13011 1001
vcf@fix[,"CHROM"] <- NA
POS <- as.numeric(vcf@fix[,"POS"])

for (i in 1:(length(ends)-1)){
  cond <- POS >= ends[i] & POS < ends[i+1]
  print(c(ends[i], ends[i+1], sum(cond)))
  vcf@fix[cond,"CHROM"] = i
}

## [1]      0 50000 1333
## [1] 50000 100000 1311
## [1] 100000 150000 1248
## [1] 150000 200000 1272
## [1] 200000 250000 1290
## [1] 250000 300000 1289
## [1] 300000 350000 1377
## [1] 350000 400000 1316
## [1] 400000 450000 1226
## [1] 450000 500000 1349
table(vcf@fix[, "CHROM"])

##
##      1   10    2     3     4     5     6     7     8     9
## 1333 1349 1311 1248 1272 1290 1289 1377 1316 1226
my_ord <- order(as.numeric(vcf@fix[, "POS"]))

vcf2 <- vcf
vcf2 <- vcf[my_ord,]
head(vcf2)

## [1] "***** Object of class 'vcfR' *****"
## [1] "***** Meta section *****"
## [1] "##fileformat=VCFv4.2"
## [1] "##fileDate=20171003"
## [1] "##source=SLiM"
## [1] "##INFO=<ID=MID,Number=1>Type=Integer,Description=\"Mutation ID in SLiM\">"
## [1] "##INFO=<ID=S,Number=1>Type=Float,Description=\"Selection Coefficient\">"
## [1] "##INFO=<ID=DOM,Number=1>Type=Float,Description=\"Dominance\">"
## [1] "First 6 rows."
## [1]
## [1] "***** Fixed section *****"
##       CHROM POS    ID REF ALT QUAL FILTER
## [1,] "1"    "7"    NA "A"  "T"  "1000" "PASS"
## [2,] "1"    "19"   NA "A"  "T"  "1000" "PASS"
## [3,] "1"    "20"   NA "A"  "T"  "1000" "PASS"
## [4,] "1"    "70"   NA "A"  "T"  "1000" "PASS"
## [5,] "1"    "97"   NA "A"  "T"  "1000" "PASS"
## [6,] "1"    "100"  NA "A"  "T"  "1000" "PASS"
## [1]
## [1] "***** Genotype section *****"
##       FORMAT i0    i1    i2    i3    i4
## [1,] "GT"    "1|1" "1|1" "1|1" "1|1" "1|1"

```

```

## [2,] "GT"    "0|0"  "0|0"  "0|0"  "0|0"  "0|0"
## [3,] "GT"    "0|1"  "0|0"  "0|1"  "0|0"  "0|0"
## [4,] "GT"    "0|1"  "0|0"  "1|1"  "1|0"  "1|0"
## [5,] "GT"    "0|0"  "0|0"  "0|0"  "0|0"  "0|0"
## [6,] "GT"    "1|0"  "1|1"  "0|0"  "1|1"  "1|1"
## [1] "First 6 columns only."
## [1]
## [1] "Unique GT formats:"
## [1] "GT"
## [1]

head(vcf)

## [1] "***** Object of class 'vcfR' ****"
## [1] "***** Meta section *****"
## [1] "##fileformat=VCFv4.2"
## [1] "##fileDate=20171003"
## [1] "##source=SLiM"
## [1] "##INFO=<ID=MID,Number=1,Type=Integer,Description=\"Mutation ID in SLiM\">"
## [1] "##INFO=<ID=S,Number=1,Type=Float,Description=\"Selection Coefficient\">"
## [1] "##INFO=<ID=DOM,Number=1,Type=Float,Description=\"Dominance\">"
## [1] "First 6 rows."
## [1]
## [1] "***** Fixed section *****"
##      CHROM POS      ID REF ALT QUAL FILTER
## [1,] "7"   "320001" NA "A"  "T"  "1000" "PASS"
## [2,] "4"   "151786" NA "A"  "T"  "1000" "PASS"
## [3,] "8"   "367566" NA "A"  "T"  "1000" "PASS"
## [4,] "6"   "268461" NA "A"  "T"  "1000" "PASS"
## [5,] "4"   "165067" NA "A"  "T"  "1000" "PASS"
## [6,] "2"   "91930"  NA "A"  "T"  "1000" "PASS"
## [1]
## [1] "***** Genotype section *****"
##      FORMAT i0    i1    i2    i3    i4
## [1,] "GT"    "0|0"  "1|0"  "0|0"  "1|0"  "0|1"
## [2,] "GT"    "0|0"  "0|0"  "1|0"  "0|1"  "0|1"
## [3,] "GT"    "0|0"  "0|0"  "1|0"  "0|0"  "0|0"
## [4,] "GT"    "0|1"  "0|1"  "0|0"  "1|0"  "1|0"
## [5,] "GT"    "1|1"  "0|1"  "0|0"  "0|0"  "0|0"
## [6,] "GT"    "1|0"  "0|0"  "0|0"  "0|0"  "0|0"
## [1] "First 6 columns only."
## [1]
## [1] "Unique GT formats:"
## [1] "GT"
## [1]

```

Plotting functions Inversion is located at 320000 to 330000 bases the inversion “tracker” mutation is located at 320000

```

plot_r_legend <- function(){
  ##### Plot recombination legend
  xl <- 1
  yb <- 1
  xr <- 1.5
  yt <- 2

```

```

ncol = length(brks)-1
par(mar=c(5.1,2.5,3.1,0.5))
plot(NA,type="n",ann=FALSE,xlim=c(1,2),ylim=c(1,2),xaxt="n",yaxt="n",bty="n")
mtext("r", side=3, adj=0.2, cex=2)
rect(
  xl,
  head(seq(yb,yt,(yt-yb)/ncol),-1),
  xr,
  tail(seq(yb,yt,(yt-yb)/ncol),-1),
  col=colfunc
)
mtext(c(paste0("1e",round(brks[1:(length(brks)-1)],1)),0.5),side=2,at=seq(yb,yt,(yt-yb)/(ncol)),las=2)
}

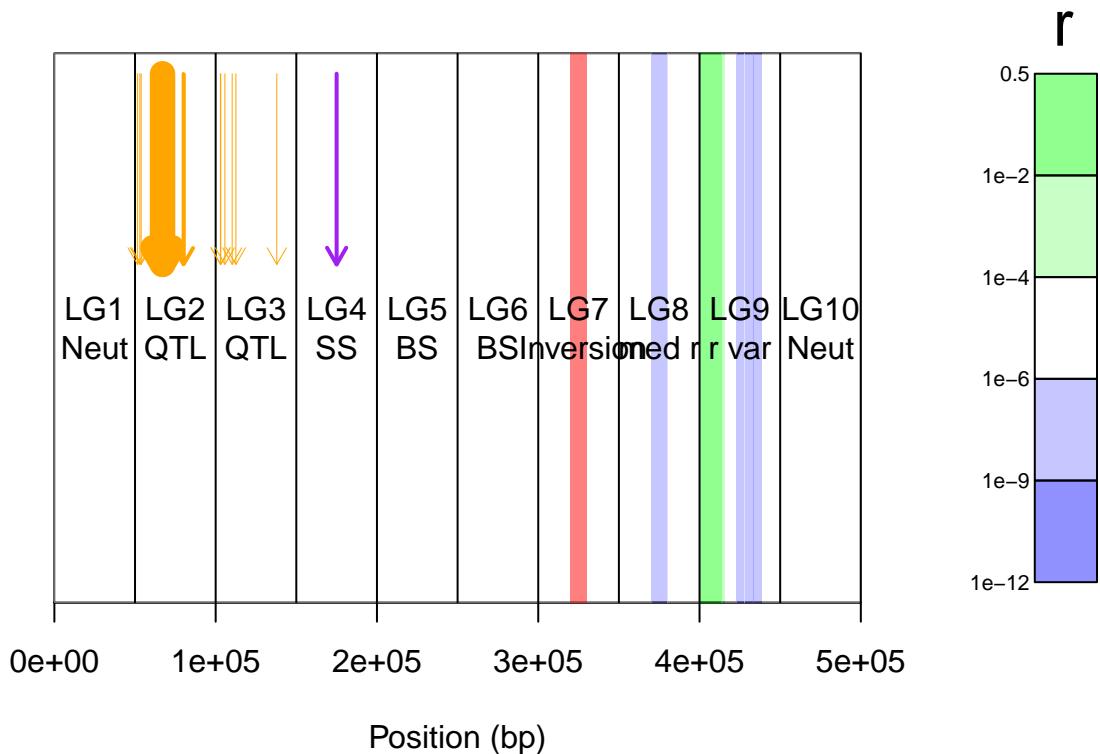
### Plot function
recom_end2 = c(0, recom_end)
plot_layers <- function(y_head=0, y_arrows=c(1,0.25), ...){
  ### Plot recombination variation
  for (i in 1:nrow(recom))
  {
    polygon(x = c(recom_end2[i], recom$recom_end[i], recom$recom_end[i], recom_end2[i]),
            y = c(-1000, -1000, 1000, 1000),
            col=as.character(recom$col[i]), border = NA)
  }
  polygon(x=c(320000, 330000, 330000, 320000),
          y = c(-1000, -1000, 1000, 1000),
          col=rgb(1,0,0,0.5), border=NA)
  abline(v=lgs)

  text(lg_whereplot, y = y_head,
        labels = c("LG1\nNeut", "LG2\nQTL", "LG3\nQTL",
                  "LG4\nSS", "LG5\nBS",
                  "LG6\nBS", "LG7\n Inversion",
                  "LG8\nmed r", "LG9\nr var", "LG10\nNeut"))

  ### Add QTLs and Sweep Location
  arrows(muts$position[muts$count], y_arrows[1], muts$position[muts$count], y_arrows[2], col="orange")
  arrows(muts$position[muts$type=="m4"], y_arrows[1], muts$position[muts$type=="m4"], y_arrows[2], col="green")
} #end plot function

layout(matrix(1:2,nrow=1),widths=c(0.8,0.2))
par(mar=c(5.1,3.1,3.1,1.9))
plot(0,0, col="white", xlim=c(0, 500000), ylim=c(-1,1), xaxs="i", yaxt="n", ylab="", xlab="Position (bp")
plot_layers()
plot_r_legend()

```



## Conversion script

```
# NB: Creates a vcfR object (stored in RAM) which size is twice as big as the original vcf file. So when you're done, don't forget to rm(vcf2)
```

```
geno <- vcf2@gt[,-c(1)] # Character matrix containing the genotypes
position <- getPOS(vcf2)-1 # Positions in bp, add one to line up with SLIM
which((position) %in% muts$position)

## [1] 1383 1388 1400 1417 1436 1464 1580 1737 1756 1776 1778 1854 1855 1881
## [15] 1954 1974 2047 2097 2108 2113 2145 2270 2364 2545 2717 2795 2904 2967
## [29] 2988 3003 3018 3183 3217 3382 3414 3581 3590 3847 4519

chromosome <- getCHROM(vcf2) # Chromosome information

G <- matrix(NA, nrow = nrow(geno), ncol = ncol(geno))

G[geno %in% c("0/0", "0|0")] <- 0
G[geno %in% c("0/1", "1/0", "1|0", "0|1")] <- 1
G[geno %in% c("1/1", "1|1")] <- 2
## Remove fixed loci or all heterozygotes ####
dim(G)

## [1] 13011 1000
head(G[1:10,1:10])

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,] 2 2 2 2 2 2 2 2 2 2
## [2,] 0 0 0 0 0 0 0 0 0 0
## [3,] 1 0 1 0 0 0 0 0 2 1
```

```

## [4,] 1 0 2 1 1 2 1 2 2 1
## [5,] 0 0 0 0 0 0 0 0 0 0
## [6,] 1 2 0 2 2 1 2 0 0 1

rem = c(which(rowSums(G)==0), which(rowSums(G-2)==0)) ## fixed loci
position[rem]

## [1] 54966 73701 130660 175000

training <- list(G = G[-rem,], position = position[-rem], chromosome = chromosome[-rem])
vcf_filt <- vcf2[-rem,]
dim(vcf_filt@gt)

## [1] 13007 1001

```

## Assign individuals to populations

```

### optional assignment to pops
toclust <- ind0[,c("x","y")]
d <- dist(toclust)
hc <- hclust(d, method="ward.D")
#fit <- kmeans(toclust, 30)
#plot(ind$x, ind$y, pch=fit$cluster, col=fit$cluster+1)
k <- 39
group <- cutree(hc, k=k)
table(group)

## group
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## 20 25 24 29 29 29 51 21 41 39 42 33 18 24 37 12 21 33 25 27 49 27 30 19 21
## 26 27 28 29 30 31 32 33 34 35 36 37 38 39
## 22 13 36 18 37 13 19 24 13 24 12 9 18 16
(group_env <- sort(round(tapply(ind0$envi, group, mean), 1)))

##      5   33   15   37   14    8   20   34    7   11   23   25    2   19   36
## -1.7 -1.5 -1.4 -1.4 -1.3 -1.2 -1.1 -1.1 -0.9 -0.9 -0.9 -0.9 -0.8 -0.8 -0.8
##   39   35    4   10   17    6   22    9   26    1   24   27   13   16   18
## -0.6 -0.5 -0.4 -0.4 -0.4 -0.3 -0.3 -0.3 -0.1 -0.1  0.0  0.0  0.1  0.2  0.2  0.3
##   28   29   30   32    3   12   21   38   31
##   0.3   0.4   0.4   0.4   0.5   0.6   0.9   0.9   1.0

(group_table <- data.frame(group=names(group_env), group_env, new_g = 1:39))

##      group group_env new_g
## 5       5     -1.7     1
## 33      33     -1.5     2
## 15      15     -1.4     3
## 37      37     -1.4     4
## 14      14     -1.3     5
## 8       8     -1.2     6
## 20      20     -1.1     7
## 34      34     -1.1     8
## 7       7     -0.9     9
## 11     11     -0.9    10
## 23     23     -0.9    11

```

```

## 25    25    -0.9    12
## 2     2     -0.8    13
## 19   19    -0.8    14
## 36   36    -0.8    15
## 39   39    -0.6    16
## 35   35    -0.5    17
## 4    4     -0.4    18
## 10   10    -0.4    19
## 17   17    -0.4    20
## 6    6     -0.3    21
## 22   22    -0.3    22
## 9    9     -0.1    23
## 26   26    -0.1    24
## 1    1      0.0    25
## 24   24    0.0     26
## 27   27    0.1     27
## 13   13    0.2     28
## 16   16    0.2     29
## 18   18    0.3     30
## 28   28    0.3     31
## 29   29    0.4     32
## 30   30    0.4     33
## 32   32    0.4     34
## 3    3     0.5     35
## 12   12    0.6     36
## 21   21    0.9     37
## 38   38    0.9     38
## 31   31    1.0     39

ind0$group <- group
ind2 <- merge(ind0, group_table)
head(ind2)

##   group id      x      y phenotype1      envi phenotype2 group_env
## 1     1  1 0.735243 0.184956  0.3655400 -0.06308660  0.3655400  0
## 2     1 578 0.706215 0.198238 -1.0801800  0.12929100 -1.0801800  0
## 3     1 108 0.739526 0.132165 -0.5631400 -0.04879510 -0.5631400  0
## 4     1 285 0.728465 0.184342 -0.0914679 -0.01696810 -0.0914679  0
## 5     1 827 0.733926 0.121190 -0.3357300 -0.00800448 -0.3357300  0
## 6     1 412 0.754067 0.123672 -0.5836690 -0.12760300 -0.5836690  0
##   new_g
## 1    25
## 2    25
## 3    25
## 4    25
## 5    25
## 6    25

# the merge puts individuals out of order relative to the genotype matrix
# the id should put them back into order
ind <- ind2[order(ind2$id),]
head(ind)

##   group id      x      y phenotype1      envi phenotype2 group_env
## 1     1  0 0.735243 0.184956  0.365540 -0.0630866  0.365540  0.0
## 23    2  1 0.862102 0.399151  0.224626 -0.8581350  0.224626 -0.8

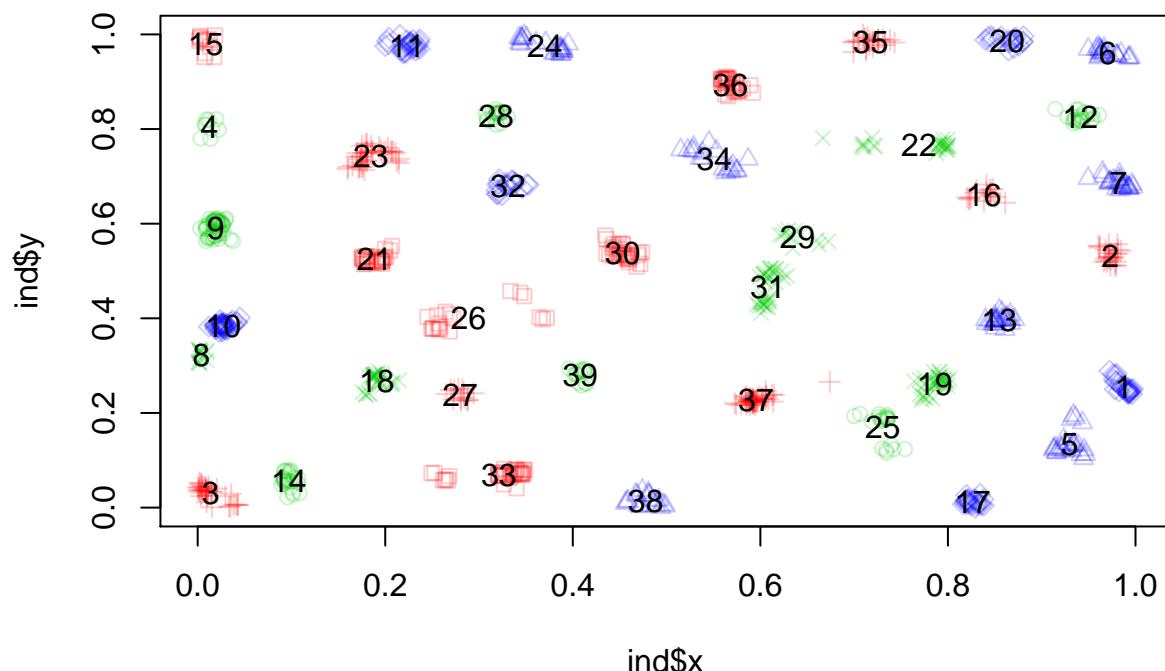
```

```

## 47      3 2 0.698954 0.983423   0.254235  0.5735460  0.254235   0.5
## 79      4 3 0.182200 0.241539  -0.365072 -0.4049860 -0.365072  -0.4
## 104     5 4 0.979798 0.267355  -1.484780 -1.6525600 -1.484780  -1.7
## 129     6 5 0.175654 0.525988  -0.823659 -0.3668130 -0.823659  -0.3
##      new_g
## 1       25
## 23      13
## 47      35
## 79      18
## 104     1
## 129     21

par(mfrow=c(1,1))
plot(ind$x, ind$y, pch=ind$group%%6, col=adjustcolor(ind$group%%3+2, alpha=0.2))
text(tapply(ind$x, ind$new_g, mean), tapply(ind$y, ind$new_g, mean), label=1:39)

```



```
#write.table(ind, "outputIndAll_pop.txt")
```

## LD pruned set of loci

```

G0<-add_code256(big_copy(t(training$G),type="raw"),code=bigsnpr:::CODE_012)

## Warning: Assignment will down cast from double to raw.
## Hint: To remove this warning, use options(bigstatsr.typecast.warning = FALSE).
#puts it in the raw format and stores likelihood genotype probability
dim(G0)

## [1] 1000 13007
str(G)

## num [1:13011, 1:1000] 2 0 1 1 0 1 0 1 0 0 ...

```

```

head(G[,1:10])

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]     2     2     2     2     2     2     2     2     2     2
## [2,]     0     0     0     0     0     0     0     0     0     0
## [3,]     1     0     1     0     0     0     0     0     2     1
## [4,]     1     0     2     1     1     2     1     2     2     1
## [5,]     0     0     0     0     0     0     0     0     0     0
## [6,]     1     2     0     2     2     1     2     0     0     1

newpc<-snp_autoSVD(G=G0,infos.chr = training$chromosome,infos.pos = training$position)

## Phase of clumping at r2 > 0.2.. keep 7570 SNPs.
##
## Iteration 1:
## Computing SVD..
##
## Converged!

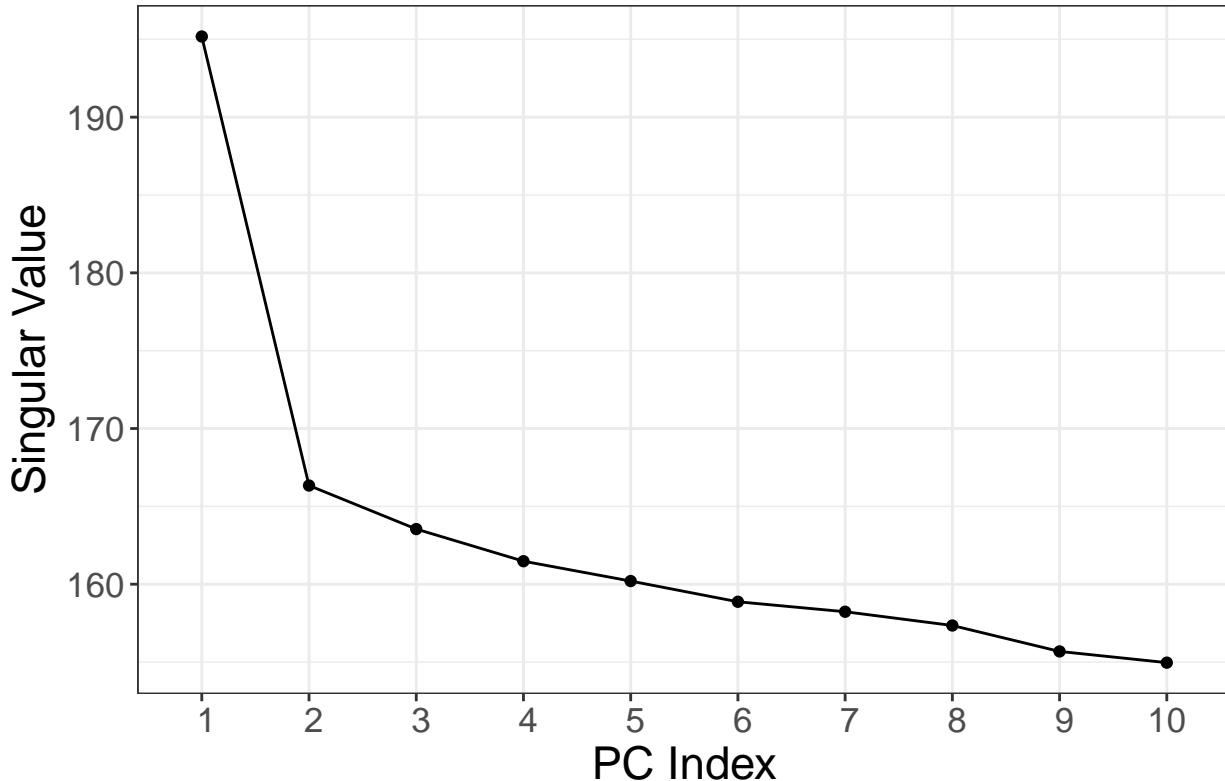
# this is doing SNP pruning - removing correlated SNPs
# take snps with highest MAF and correlate snps around it
# Snps with R^2 > 0.2 are removed
# the subset is the indexes of the remaining SNPs
str(newpc)

## List of 7
## $ d      : num [1:10] 195 166 164 161 160 ...
## $ u      : num [1:1000, 1:10] 0.0262 0.0247 0.038 -0.0185 -0.057 ...
## $ v      : num [1:7570, 1:10] 0.01034 0.00224 0.00278 0.00428 -0.00117 ...
## $ niter : int 8
## $ nops  : int 156
## $ center: num [1:7570] 1.986 0.001 0.001 0.898 0.009 ...
## $ scale : num [1:7570] 0.1179 0.0316 0.0316 0.7034 0.0947 ...
## - attr(*, "class")= chr "big_SVD"
## - attr(*, "subset")= int [1:7570] 1 5 7 8 9 12 13 15 16 17 ...
## - attr(*, "lrldr")='data.frame':   0 obs. of  3 variables:
##   ..$ Chr  : int(0)
##   ..$ Start: int(0)
##   ..$ Stop : int(0)

plot(newpc)

```

# Scree Plot



```
which_pruned <- attr(newpc, which="subset")
length(which_pruned)

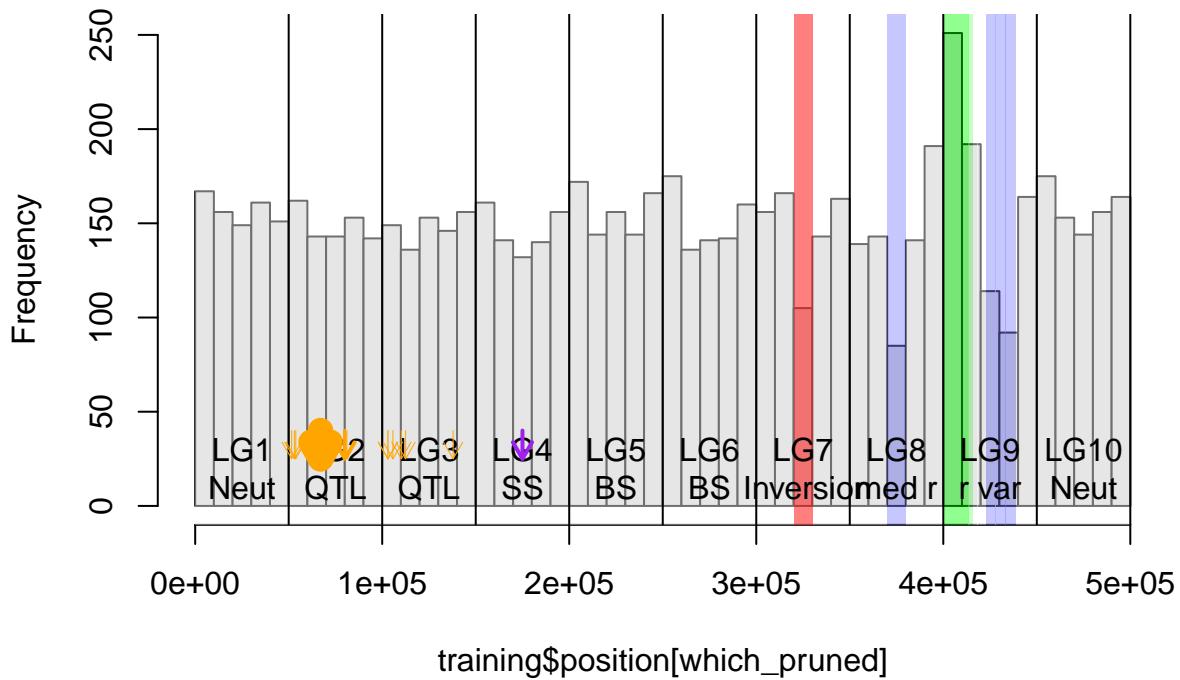
## [1] 7570

### Calculate average LD around each SNP #####
# LD <- LD2<- rep(NA, ncol(G0))
#   # the following is not my most efficient lines of code
# for(i in 26:(ncol(G0)-26)){
#   LD[i]=mean(cor(G0[, (i-25):(i+25)])[,25])
# }

layout(matrix(1))
# plot(training$position, abs(LD), pch=20, ylim=c(0,0.18), xaxs="i", col=rgb(0,0,0,0.5), xlab="Position")
# plot_layers(y_head=0.17, y_arrows=c(0.02, 0))

hist(training$position[which_pruned], breaks=seq(0,500000, by=10000), col="lightgrey")
plot_layers(y_head=20, y_arrows=c(40,25))
```

## Histogram of training\$position[which\_pruned]



### Inversion frequency across populations

```
(inv_index <- which(training$position==(320000))

## [1] 8279

head(training$position[training$position>319998])

## [1] 320000 320052 320059 320061 320070 320086

inv_all <- which(training$position>320000 & training$position<330000 & rowSums(training$G)>100)
  # remove low H individuals
inv_allG <- training$G[inv_all,]

dim(training$G)

## [1] 13007 1000

inv <- training$G[inv_index,]

# Genotype frequencies of inversion marker
table(inv)/(1000)

## inv
##   0     1     2
## 0.406 0.450 0.144
  # basically, expect individuals to be of type 0 or type 2

# Allele frequency of inversion marker
1-(sum(inv)/2000)
```

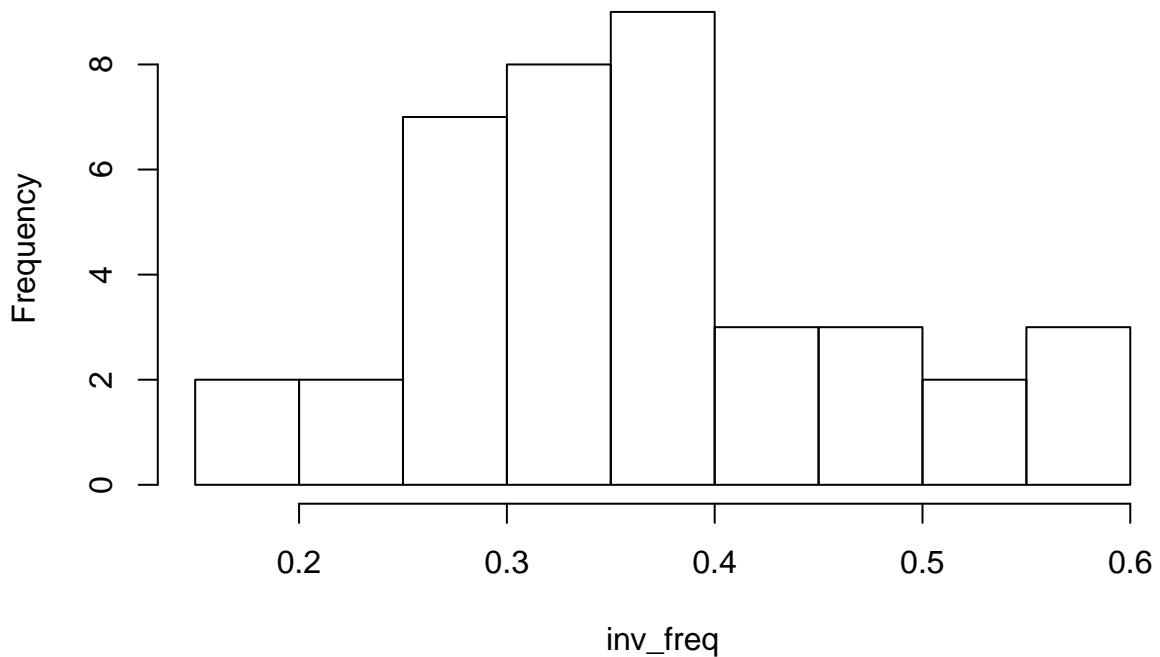
```

## [1] 0.631
## Haplotypes of inversion marker
gen <- apply(inv_allG, 1, function(x) (paste(x, collapse = "", sep="")))
nlevels(factor(gen))

## [1] 57
inv_freq <- tapply(inv, ind$new_g, FUN = function(x)(sum(x)/(2*length(x))))
hist(inv_freq)

```

**Histogram of inv\_freq**

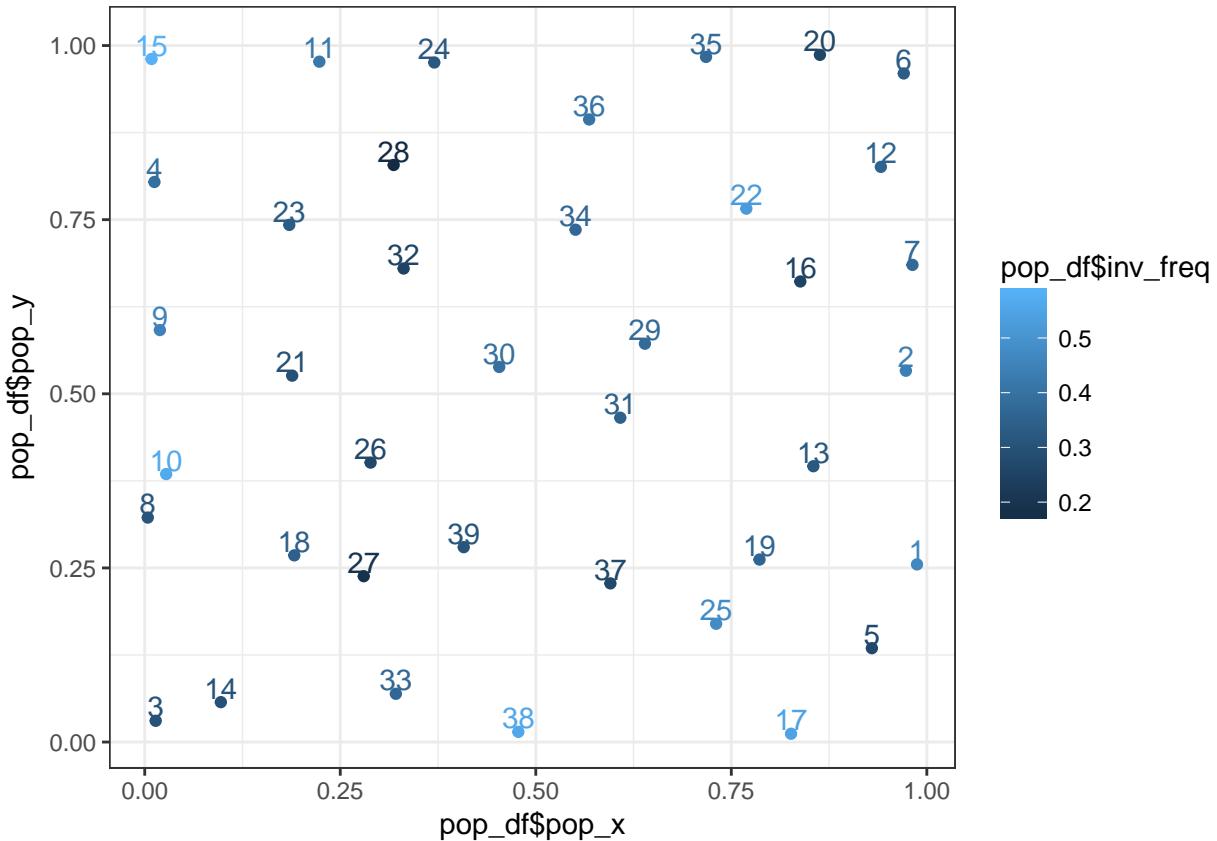


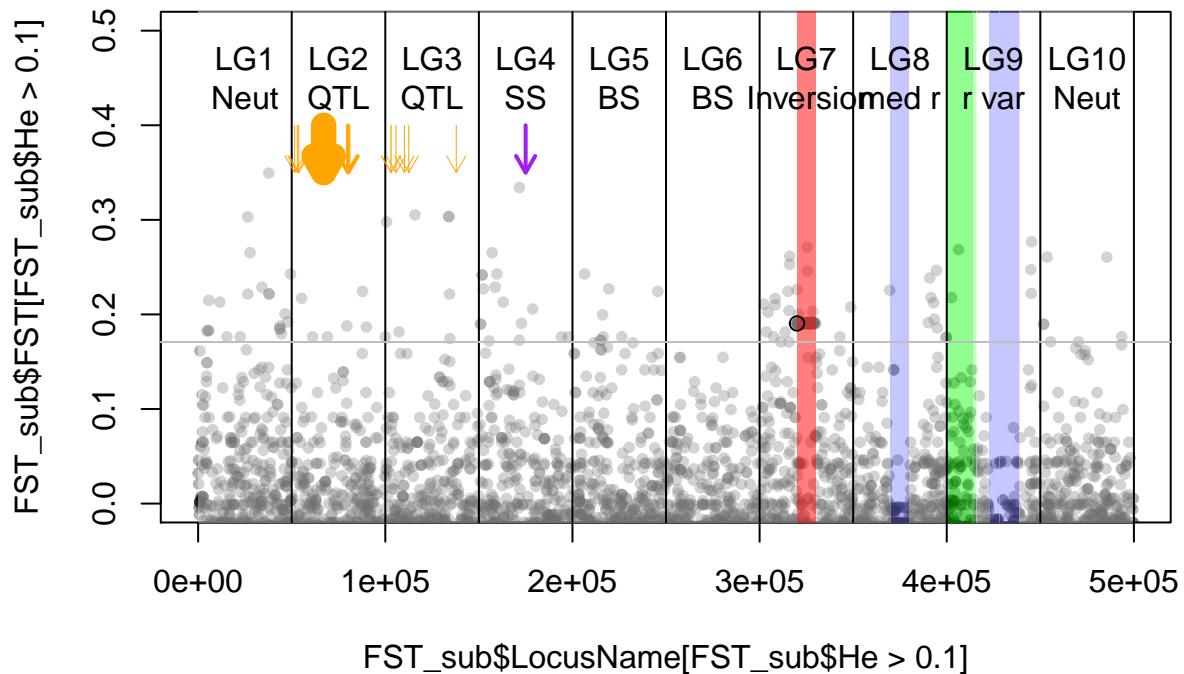
```

pop_df <- data.frame(new_g=rownames(inv_freq),
                      inv_freq=inv_freq,
                      pop_x =tapply(ind$x, ind$new_g, mean),
                      pop_y = tapply(ind$y, ind$new_g, mean),
                      pop_envi = tapply(ind$envi, ind$new_g, mean)
                     )

qplot(x = pop_df$pop_x, y = pop_df$pop_y, colour = pop_df$inv_freq) + theme_bw() + geom_text(aes(x = pop

```





## Population structure

Principle components based on all data

```

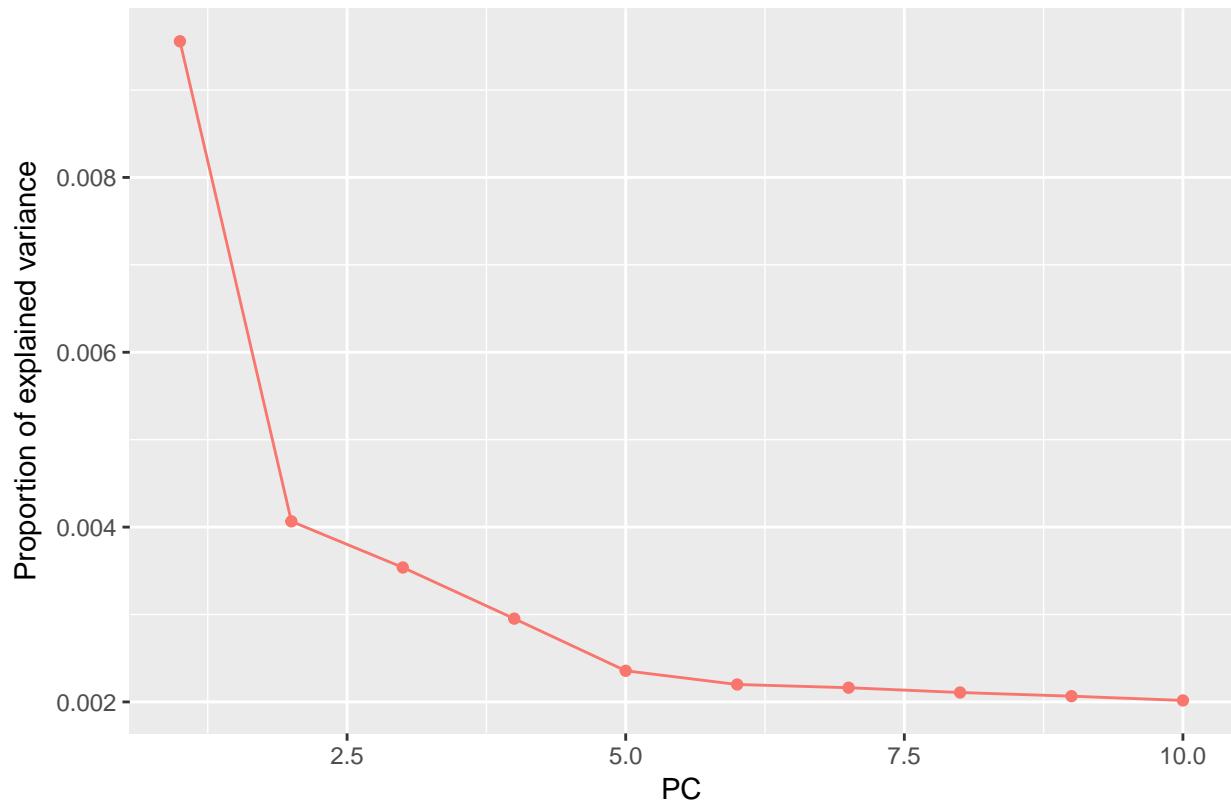
aux<-pcadapt(training$G,K=10)

## Number of SNPs: 13007
## Number of individuals: 1000
str(aux)

## List of 10
## $ scores      : num [1:1000, 1:10] 0.0332 -0.012 0.0363 -0.0107 -0.015 ...
## $ singular.values: num [1:10] 11.15 7.27 6.78 6.2 5.54 ...
## $ zscores      : num [1:13007, 1:10] 0 0 0.379 -3.048 0 ...
## $ loadings     : num [1:13007, 1:10] 0 0 0.128 -0.951 0 ...
## $ maf          : num [1:13007] 0.007 0.007 0.164 0.393 0.0005 0.423 0.0005 0.449 0.0045 0.2 ...
## $ missing       : num [1:13007] 0 0 0 0 0 0 0 0 0 ...
## $ stat         : num [1:13007(1d)] NA NA 1.57 22.57 NA ...
## $ gif          : num 1.19
## $ chi2.stat    : num [1:13007(1d)] NA NA 1.31 18.93 NA ...
## $ pvalues      : num [1:13007] NA NA 0.9994 0.0412 NA ...
## - attr(*, "class")= chr "pcadapt"
## - attr(*, "K")= num 10
## - attr(*, "data.type")= chr "genotype"
## - attr(*, "method")= chr "mahalanobis"
## - attr(*, "min.maf")= num 0.05
plot(aux,option="screeplot")

```

Scree Plot – K = 10

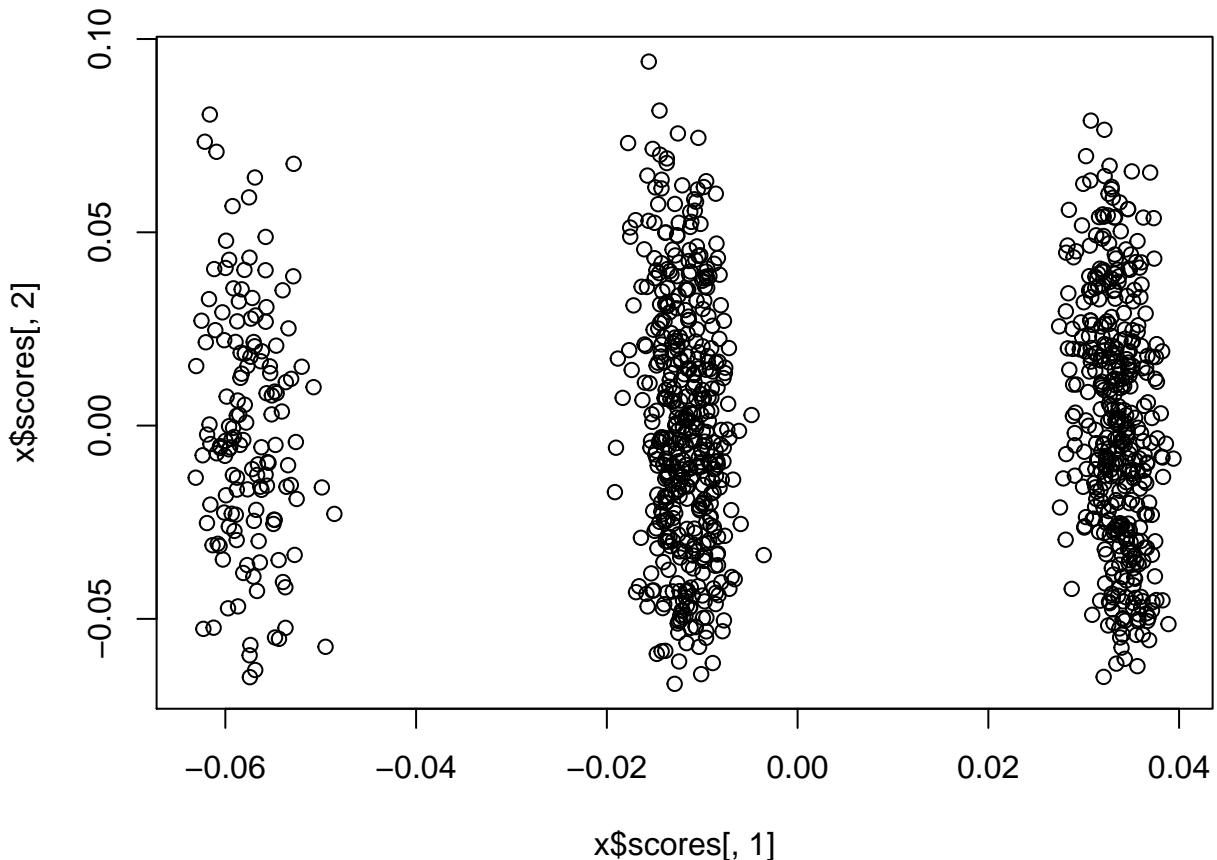


```
#num <- max(which(training$position<300000))
num <- length(training$position)
x <- pcadapt(training$G[1:num,], K=4)
```

```
## Number of SNPs: 13007
## Number of individuals: 1000
summary(x)
```

```
##                                     Length Class  Mode
## scores                  4000  numeric
## singular.values        4   numeric
## zscores                 52028  numeric
## loadings                52028  numeric
## maf                     13007  numeric
## missing                 13007  numeric
## stat                     13007  numeric
## gif                      1   numeric
## chi2.stat                13007  numeric
## pvalues                  13007  numeric
```

```
par(mar=c(4,4,1,1))
plot(x$scores[,1], x$scores[,2])
```

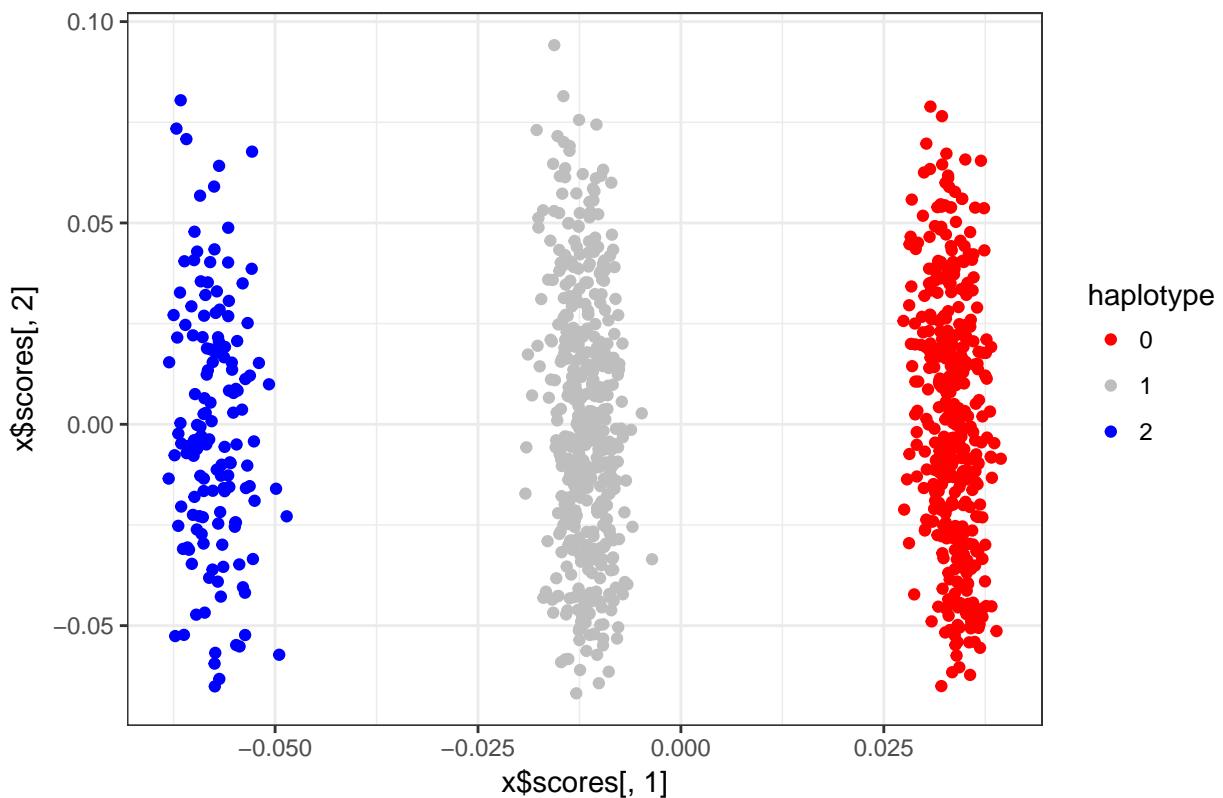


```
haplotype <- as.factor(training$G[inv_index,])
table(haplotype)
```

```
## haplotype
## 0 1 2
## 406 450 144
```

```
qplot(x$scores[,1], x$scores[,2], colour=haplotype, main="Individual scores without LD pruning") + scale
```

### Individual scores without LD pruning



### Loading of genomic regions onto PC axes calculated from all data

```
#plot_layers(ylim=c(min(x$loadings[,1]), max(x$loadings[,1])), ylab="Loadings PC1")
layout(matrix(c(1,2,3,4,6,5,5,6),nrow=4),widths=c(0.8,0.2))
par(oma=c(3,3,1,0), mar=c(2,2,0,0))
## Top plot
summary(x$loadings[,1])

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## -10.372078  0.000000  0.000000  0.000195  0.000000  10.372078

plot(training$position,x$loadings[,1], xaxs="i", pch=20, ylim=c(-11, 15))
plot_layers(y_head = 12, y_arrows=c(-8, -11))
## Middle plot
summary(x$loadings[,2])

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## -13.746337  0.000000  0.000000  0.009282  0.000000  13.746337

plot(training$position,x$loadings[,2], xaxs="i", pch=20, ylim=c(-11, 13))
plot_layers(y_head = 20, y_arrows=c(-8, -11))
## Middle plot
summary(x$loadings[,3])

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## -16.204653  0.000000  0.000000 -0.005638  0.000000  8.877744
```

```

plot(training$position,x$loadings[,3], xaxs="i", pch=20, ylim=c(-15, 15))
plot_layers(y_head = 20, y_arrows=c(-12, -15))
## Bottom plot
summary(x$loadings[,4])

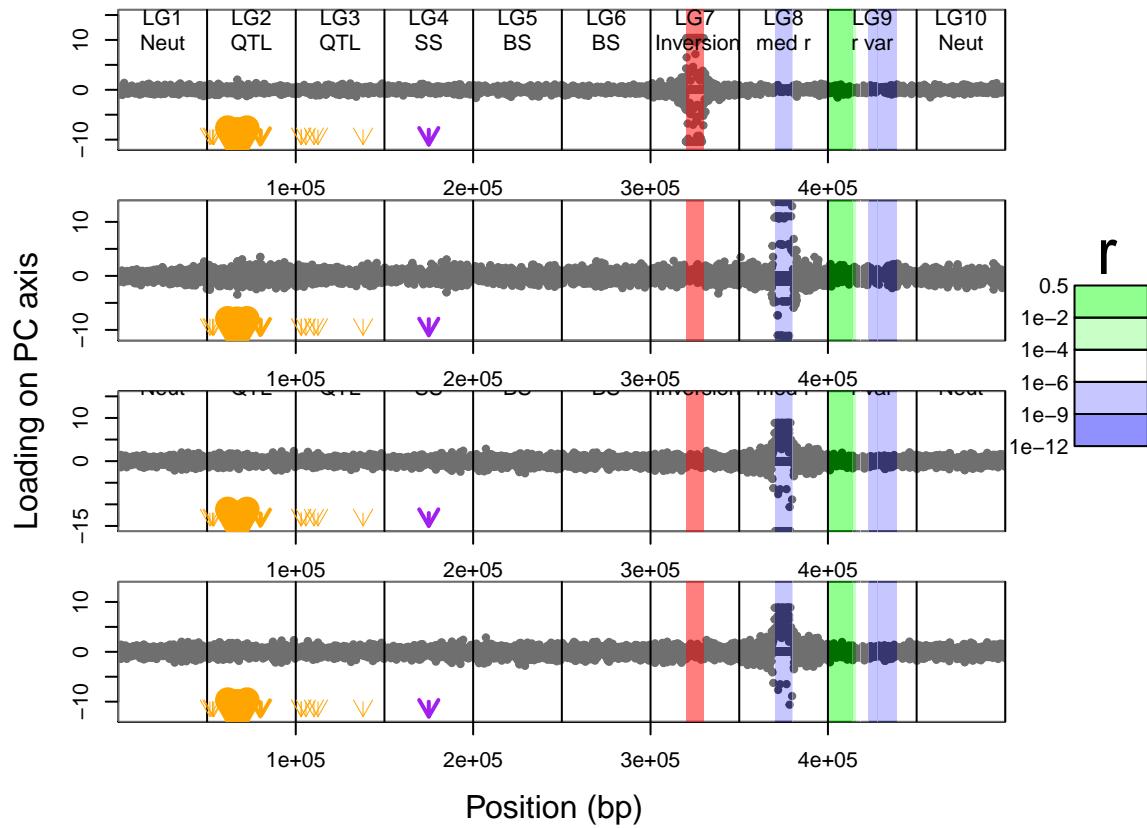
```

## Min. 1st Qu. Median Mean 3rd Qu. Max.  
## -15.616251 0.000000 0.000000 -0.000401 0.000000 7.853880

```

plot(training$position,x$loadings[,3], xaxs="i", pch=20, ylim=c(-13, 13))
plot_layers(y_head = 20, y_arrows=c(-10, -13))
## Right plot
plot_r_legend()
mtext("Position (bp)", outer=TRUE, side=1, line=1, adj=0.4)
mtext("Loading on PC axis", outer=TRUE, side=2, line=1, adj=0.5)

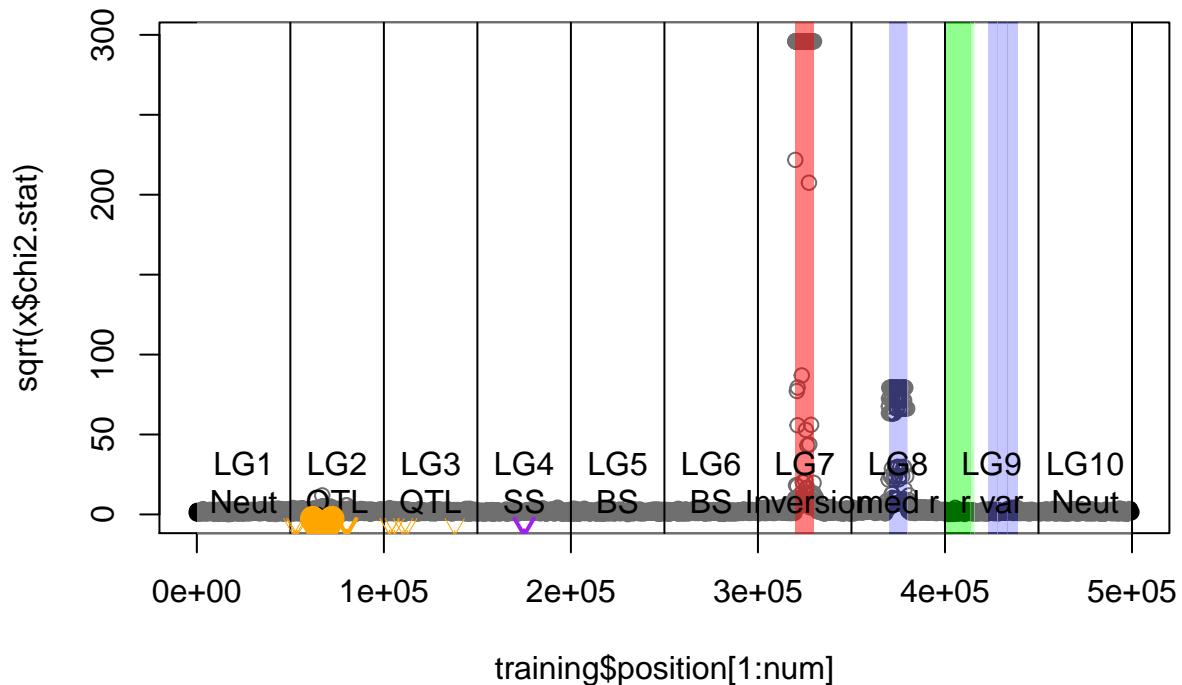
```



```

par(mifrow=c(1,1))
plot(training$position[1:num], sqrt(x$chi2.stat))
plot_layers(y_head = 20, y_arrows=c(-10, -13))

```



### Principle components based on pruned data

```

dim(training$G)

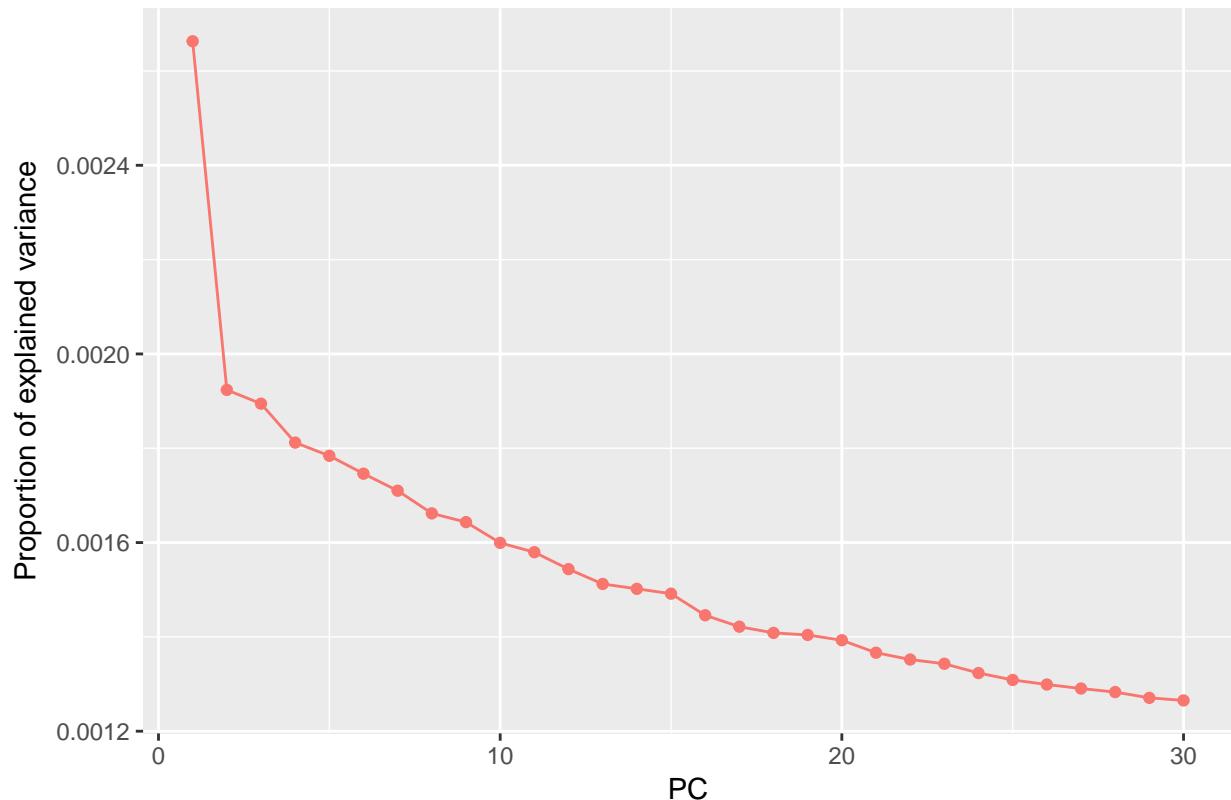
## [1] 13007 1000
aux<-pcadapt(training$G[which_pruned,],K=30)

## Number of SNPs: 7570
## Number of individuals: 1000
str(aux)

## List of 10
## $ scores      : num [1:1000, 1:30] 0.0226 0.036 0.0336 -0.0246 -0.0377 ...
## $ singular.values: num [1:30] 4.49 3.82 3.79 3.7 3.67 ...
## $ zscores      : num [1:7570, 1:30] 0 0 0 0.907 0 ...
## $ loadings     : num [1:7570, 1:30] 0 0 0 0.536 0 ...
## $ maf          : num [1:7570] 0.007 0.0005 0.0005 0.449 0.0045 ...
## $ missing       : num [1:7570] 0 0 0 0 0 0 0 0 0 0 ...
## $ stat         : num [1:7570(1d)] NA NA NA 27.1 NA ...
## $ gif          : num 1.1
## $ chi2.stat    : num [1:7570(1d)] NA NA NA 24.6 NA ...
## $ pvalues      : num [1:7570] NA NA NA 0.745 NA ...
## - attr(*, "class")= chr "pcadapt"
## - attr(*, "K")= num 30
## - attr(*, "data.type")= chr "genotype"
## - attr(*, "method")= chr "mahalanobis"
## - attr(*, "min.maf")= num 0.05
plot(aux,option="screeplot")

```

## Scree Plot – K = 30



```
x_LD <- pcadapt(training$G[which_pruned,], K=3)
```

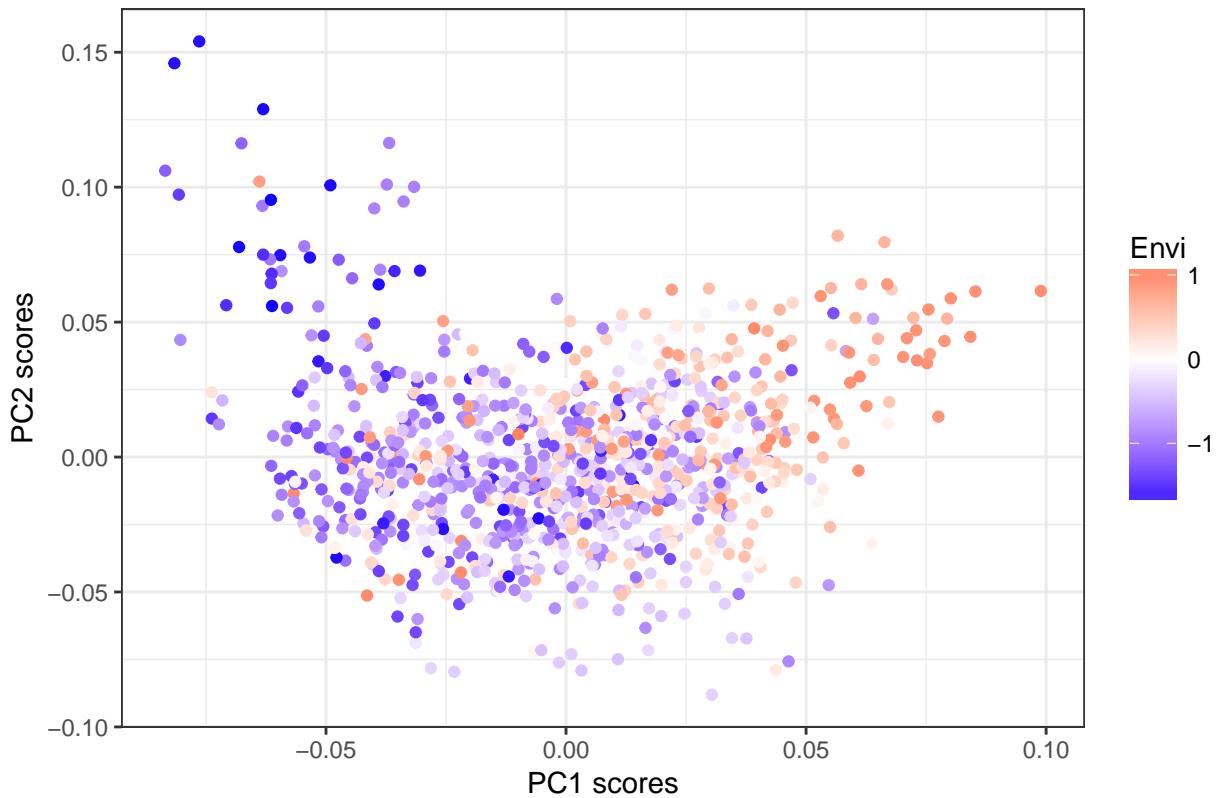
```
## Number of SNPs: 7570
## Number of individuals: 1000
summary(x_LD)
```

```
##          Length Class  Mode
## scores      3000 -none- numeric
## singular.values   3 -none- numeric
## zscores     22710 -none- numeric
## loadings    22710 -none- numeric
## maf        7570 -none- numeric
## missing     7570 -none- numeric
## stat        7570 -none- numeric
## gif          1 -none- numeric
## chi2.stat    7570 -none- numeric
## pvalues     7570 -none- numeric
```

```
par(mar=c(4,4,1,1))
```

```
qplot(x_LD$scores[,1], x_LD$scores[,2], colour=ind$envi, main="Individual scores with LD pruning", xlab=
high="red", space = "Lab" )
```

### Individual scores with LD pruning



### Loading of genomic regions onto PC axes calculated from pruned data

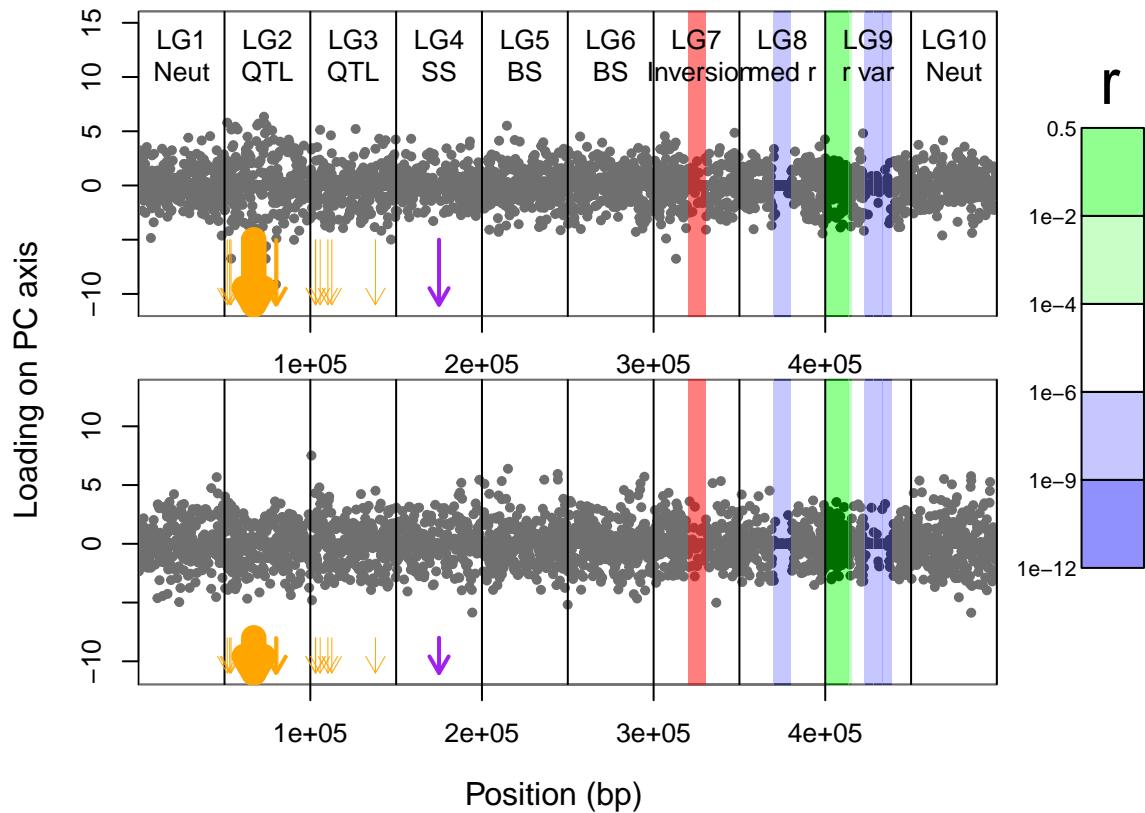
```
#plot_layers(ylim=c(min(x$loadings[,1]), max(x$loadings[,1])), ylab="Loadings PC1")
layout(matrix(c(1,2,3,3),nrow=2),widths=c(0.8,0.2))
par(oma=c(3,3,1,0), mar=c(2,2,0,0))
## Top plot
summary(x_LD$loadings[,1])

##      Min.    1st Qu.     Median      Mean    3rd Qu.    Max.
## -9.112180  0.000000  0.000000 -0.001167  0.000000 17.344448

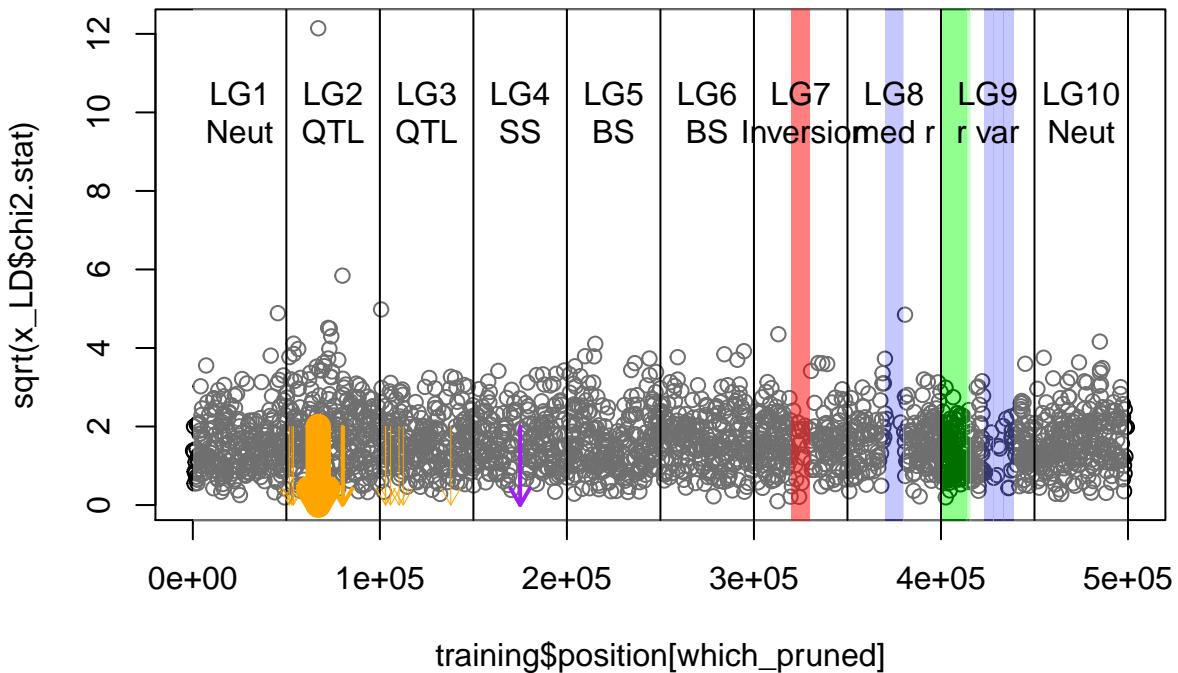
plot(training$position[which_pruned],x_LD$loadings[,1], xaxs="i", pch=20, ylim=c(-11, 15))
plot_layers(y_head = 12, y_arrows=c(-5, -11))
## Middle plot
summary(x_LD$loadings[,2])

##      Min.    1st Qu.     Median      Mean    3rd Qu.    Max.
## -5.866679  0.000000  0.000000  0.004825  0.000000  7.513840

plot(training$position[which_pruned],x_LD$loadings[,2], xaxs="i", pch=20, ylim=c(-11, 13))
plot_layers(y_head = 20, y_arrows=c(-8, -11))
## Right plot
plot_r_legend()
mtext("Position (bp)", outer=TRUE, side=1, line=1, adj=0.4)
mtext("Loading on PC axis", outer=TRUE, side=2, line=1, adj=0.5)
```



```
par(mfrow=c(1,1))
plot(training$position[which_pruned], sqrt(x_LD$chi2.stat))
plot_layers(y_head = 10, y_arrows=c(2, 0))
```



## Admixture/ancestry based on all data (snmf in LEA package)

```
write.geno(t(training$G), paste0("temp/",seed,"genotypes.geno"))

## [1] "temp/1505550948364genotypes.geno"
project = snmf(paste0("temp/",seed,"genotypes.geno"),
               K = 1:4,
               entropy = TRUE,
               repetitions = 3,
               project = "new")

## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] 1990314766
## [1] *****
## [1] "*          create.dataset      *"
## [1] *****
## summary of the options:
##
##           -n (number of individuals)      1000
##           -L (number of loci)            13007
##           -s (seed random init)        1990314766
##           -r (percentage of masked data) 0.05
##           -x (genotype file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp/
##           -o (output file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp/
##
## Write genotype file with masked data, /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.geno
##
## [1] *****
## [1] "* sNMF K = 1  repetition 1      *"
## [1] *****
## summary of the options:
##
##           -n (number of individuals)      1000
##           -L (number of loci)            13007
##           -K (number of ancestral pops)  1
##           -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.geno
##           -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.q
##           -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.g
##           -i (number max of iterations)  200
##           -a (regularization parameter) 10
##           -s (seed random init)        5280505121550
##           -e (tolerance error)         1E-05
##           -p (number of processes)     1
##           - diploid
##
```

```

## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/main.R
##
##
## Main algorithm:
##
## Least-square error: 2305067.624559
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364individualAncestry.snmf
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364ancestralFrequencies.snmf
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            13007
##          -K (number of ancestral pops)   1
##          -x (genotype file)           /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##          -q (individual admixture)     /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364individualAncestry.snmf
##          -g (ancestral frequencies)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364ancestralFrequencies.snmf
##          -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##          - diploid
##
## Cross-Entropy (all data):    0.306492
## Cross-Entropy (masked data): 0.308132
## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 2  repetition 1      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            13007
##          -K (number of ancestral pops)   2
##          -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##          -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364individualAncestry.snmf
##          -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364ancestralFrequencies.snmf
##          -i (number max of iterations)   200
##          -a (regularization parameter)   10
##          -s (seed random init)         140400176253710
##          -e (tolerance error)          1E-05
##          -p (number of processes)       1
##          - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/main.R
##

```

```

## 
## Main algorithm:
## [ ] 
## [=====]
## Number of iterations: 20
##
## Least-square error: 2262251.582654
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)           13007
##          -K (number of ancestral pops)   2
##          -x (genotype file)           /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##          -q (individual admixture)     /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##          -g (ancestral frequencies)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##          -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##          - diploid
##
## Cross-Entropy (all data):    0.300591
## Cross-Entropy (masked data): 0.303863
## The project is saved into :
## temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
## project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
## remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 3  repetition 1      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)           13007
##          -K (number of ancestral pops)   3
##          -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##          -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##          -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##          -i (number max of iterations)   200
##          -a (regularization parameter)  10
##          -s (seed random init)        140400176253710
##          -e (tolerance error)         1E-05
##          -p (number of processes)      1
##          - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject

```

```

## 
## Main algorithm:
## [
## [=====]
## Number of iterations: 21
##
## Least-square error: 2240844.020220
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)           13007
##          -K (number of ancestral pops)   3
##          -x (genotype file)           /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -q (individual admixture)     /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -g (ancestral frequencies)    /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          - diploid
##
## Cross-Entropy (all data):      0.296545
## Cross-Entropy (masked data):   0.301307
## The project is saved into :
## temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
## project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
## remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 4  repetition 1      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)           13007
##          -K (number of ancestral pops)   4
##          -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/15
##          -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/15
##          -g (ancestral frequencies file)/Users/katie/Desktop/RecombinationGenomeScans/temp/15
##          -i (number max of iterations)  200
##          -a (regularization parameter) 10
##          -s (seed random init)        4607182420790332174
##          -e (tolerance error)         1E-05
##          -p (number of processes)      1
##          - diploid
##

```

```

## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/main.R
##
##
## Main algorithm:
##   [
##   [=====]
## Number of iterations: 71
##
## Least-square error: 2231520.723917
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364individualAncestryCoefficients.snmf
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364ancestralAlleleFrequencyCoefficients.snmf
##
## [1] "*****"
## [1] "*      cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##       -n (number of individuals)      1000
##       -L (number of loci)            13007
##       -K (number of ancestral pops)  4
##       -x (genotype file)           /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##       -q (individual admixture)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364individualAdmixture.snmf
##       -g (ancestral frequencies)   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364ancestralAlleleFrequencies.snmf
##       -i (with masked genotypes)   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##       - diploid
##
## Cross-Entropy (all data):      0.295582
## Cross-Entropy (masked data):   0.300836
## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] 2006206821
## [1] "*****"
## [1] "*      create.dataset      *"
## [1] "*****"
## summary of the options:
##
##       -n (number of individuals)      1000
##       -L (number of loci)            13007
##       -s (seed random init)        2006206821
##       -r (percentage of masked data) 0.05
##       -x (genotype file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.geno
##       -o (output file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.geno
##
## Write genotype file with masked data, /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##
## [1] "*****"
## [1] "* sNMF K = 1  repetition 2      *"

```

```

## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)      1000
##      -L (number of loci)           13007
##      -K (number of ancestral pops)  1
##      -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -i (number max of iterations)  200
##      -a (regularization parameter) 10
##      -s (seed random init)        140400192145765
##      -e (tolerance error)         1E-05
##      -p (number of processes)     1
##      - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/main.R
##
## Main algorithm:
##
## Least-square error: 2304915.502542
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/individual_coefficients.snmf
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/ancestral_coefficients.snmf
##
## [1] "*****"
## [1] "*   cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)      1000
##      -L (number of loci)           13007
##      -K (number of ancestral pops)  1
##      -x (genotype file)             /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -q (individual admixture)      /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -g (ancestral frequencies)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      - diploid
##
## Cross-Entropy (all data):      0.306429
## Cross-Entropy (masked data):   0.309376
## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 2  repetition 2      *"
## [1] "*****"
## summary of the options:

```

```

##          -n (number of individuals)      1000
##          -L (number of loci)           13007
##          -K (number of ancestral pops)    2
##          -x (input file)             /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##          -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##          -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##          -i (number max of iterations)     200
##          -a (regularization parameter)      10
##          -s (seed random init)           140400192145765
##          -e (tolerance error)            1E-05
##          -p (number of processes)          1
##          - diploid

## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/main.R
## 
## Main algorithm:
## [
## [=====]
## Number of iterations: 22
##
## Least-square error: 2261534.868705
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)           13007
##          -K (number of ancestral pops)    2
##          -x (genotype file)             /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##          -q (individual admixture)       /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##          -g (ancestral frequencies)     /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##          -i (with masked genotypes)      /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##          - diploid

## Cross-Entropy (all data):      0.300532
## Cross-Entropy (masked data):   0.304927
## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 3  repetition 2      *"
## [1] "*****"

```

```

## summary of the options:
##
##      -n (number of individuals)          1000
##      -L (number of loci)                 13007
##      -K (number of ancestral pops)       3
##      -x (input file)                   /Users/katie/Desktop/RecombinationGenomeScans/temp/15
##      -q (individual admixture file)    /Users/katie/Desktop/RecombinationGenomeScans/temp/15
##      -g (ancestral frequencies file)   /Users/katie/Desktop/RecombinationGenomeScans/temp/15
##      -i (number max of iterations)     200
##      -a (regularization parameter)    10
##      -s (seed random init)           140400192145765
##      -e (tolerance error)            1E-05
##      -p (number of processes)         1
##      - diploid

##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/m
##
##
## Main algorithm:
##  [
##  =====
## Number of iterations: 24
##
## Least-square error: 2240051.856662
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)          1000
##      -L (number of loci)                 13007
##      -K (number of ancestral pops)       3
##      -x (genotype file)                  /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##      -q (individual admixture)          /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##      -g (ancestral frequencies)         /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##      -i (with masked genotypes)        /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##      - diploid

##
## Cross-Entropy (all data):      0.296487
## Cross-Entropy (masked data):  0.302465
## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 4  repetition 2      *"

```

```

## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)          1000
##      -L (number of loci)                 13007
##      -K (number of ancestral pops)       4
##      -x (input file)                   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -q (individual admixture file)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -g (ancestral frequencies file)   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -i (number max of iterations)     200
##      -a (regularization parameter)    10
##      -s (seed random init)           140400192145765
##      -e (tolerance error)            1E-05
##      -p (number of processes)         1
##      - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/main.R
##
## Main algorithm:
## [
## [=====]
## Number of iterations: 87
##
## Least-square error: 2230952.379148
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)          1000
##      -L (number of loci)                 13007
##      -K (number of ancestral pops)       4
##      -x (genotype file)                  /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -q (individual admixture)          /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -g (ancestral frequencies)         /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      -i (with masked genotypes)        /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf
##      - diploid
##
## Cross-Entropy (all data): 0.295577
## Cross-Entropy (masked data): 0.302046
## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] 1433788466

```

```

## [1] "*****"
## [1] "*      create.dataset      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)      1000
##      -L (number of loci)           13007
##      -s (seed random init)        1433788466
##      -r (percentage of masked data) 0.05
##      -x (genotype file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp/
##      -o (output file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp/
##
## Write genotype file with masked data, /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/masked.genotype
##
## [1] "*****"
## [1] "* sNMF K = 1  repetition 3      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)      1000
##      -L (number of loci)           13007
##      -K (number of ancestral pops)   1
##      -x (input file)               /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/masked.genotype
##      -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/individual_admixture.txt
##      -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/ancestral_frequencies.txt
##      -i (number max of iterations)    200
##      -a (regularization parameter)    10
##      -s (seed random init)          1433788466
##      -e (tolerance error)           1E-05
##      -p (number of processes)        1
##      - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/masked.genotype
##
## Main algorithm:
##
## Least-square error: 2305116.392645
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/individual_coefficients.txt
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/ancestral_coefficients.txt
##
## [1] "*****"
## [1] "*      cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)      1000
##      -L (number of loci)           13007
##      -K (number of ancestral pops)   1
##      -x (genotype file)               /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/masked.genotype
##      -q (individual admixture)        /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/individual_admixture.txt
##      -g (ancestral frequencies)       /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/ancestral_frequencies.txt
##      -i (with masked genotypes)       /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/masked.genotype
##      - diploid

```

```

## Cross-Entropy (all data): 0.306514
## Cross-Entropy (masked data): 0.307711
## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 2  repetition 3      *"
## [1] "*****"
## summary of the options:
##
##           -n (number of individuals)          1000
##           -L (number of loci)                 13007
##           -K (number of ancestral pops)       2
##           -x (input file)                   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##           -q (individual admixture file)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##           -g (ancestral frequencies file)   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##           -i (number max of iterations)     200
##           -a (regularization parameter)    10
##           -s (seed random init)           140399619727410
##           -e (tolerance error)            1E-05
##           -p (number of processes)        1
##           - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##
## Main algorithm:
##   [
##   =====
## Number of iterations: 19
##
## Least-square error: 2261795.465163
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##           -n (number of individuals)          1000
##           -L (number of loci)                 13007
##           -K (number of ancestral pops)       2
##           -x (genotype file)                  /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##           -q (individual admixture)         /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##           -g (ancestral frequencies)        /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##           -i (with masked genotypes)        /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject

```

```

##      - diploid
##
## Cross-Entropy (all data):    0.300628
## Cross-Entropy (masked data): 0.30338
## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 3  repetition 3      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)          1000
##      -L (number of loci)                 13007
##      -K (number of ancestral pops)       3
##      -x (input file)                   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##      -q (individual admixture file)     /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##      -g (ancestral frequencies file)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##      -i (number max of iterations)      200
##      -a (regularization parameter)     10
##      -s (seed random init)            140399619727410
##      -e (tolerance error)              1E-05
##      -p (number of processes)          1
##      - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##
## Main algorithm:
##   [
##   =====
## Number of iterations: 23
##
## Least-square error: 2240716.589421
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##
## [1] "*****"
## [1] "*      cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)          1000
##      -L (number of loci)                 13007
##      -K (number of ancestral pops)       3
##      -x (genotype file)                  /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##      -q (individual admixture)          /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject
##      -g (ancestral frequencies)         /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmfProject

```

```

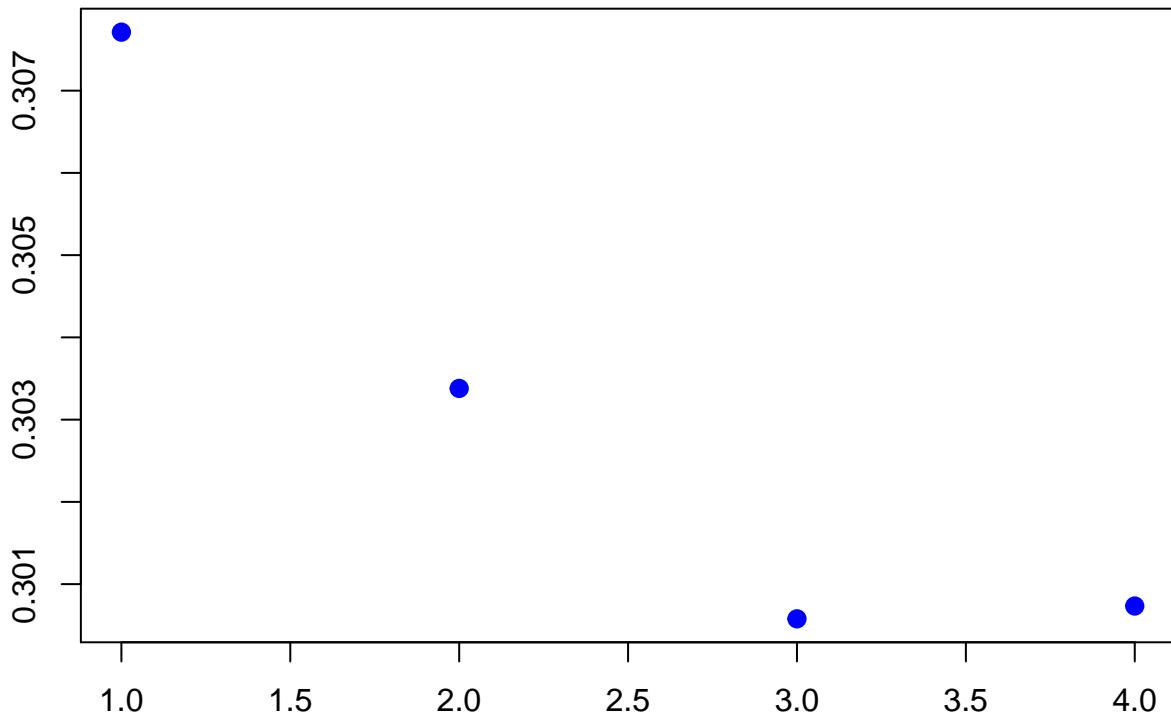
##          -i (with masked genotypes)      /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          - diploid
##
## Cross-Entropy (all data):    0.296581
## Cross-Entropy (masked data): 0.300578
## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 4  repetition 3      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            13007
##          -K (number of ancestral pops)  4
##          -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -i (number max of iterations)  200
##          -a (regularization parameter) 10
##          -s (seed random init)        140399619727410
##          -e (tolerance error)         1E-05
##          -p (number of processes)     1
##          - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes.snmf/mad
##
## Main algorithm:
##   [
##   =====
## Number of iterations: 200
##
## Least-square error: 2234108.211145
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            13007
##          -K (number of ancestral pops)  4
##          -x (genotype file)             /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -q (individual admixture)      /Users/katie/Desktop/RecombinationGenomeScans/temp/150555

```

```

##          -g (ancestral frequencies)      /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          - diploid
##
## Cross-Entropy (all data):      0.296285
## Cross-Entropy (masked data):   0.300734
## The project is saved into :
##   temp/1505550948364genotypes.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes.snmfProject")
par(mfrow=c(1,1), mar=c(3,3,3,1))
#project
# plot cross-entropy criterion of all runs of the project
plot(project, cex = 1.2, col = "blue", pch = 19)

```



```

# get the cross-entropy of all runs for K = 3
ce = cross.entropy(project, K = 2)
ce

##          K = 2
## run 1 0.3038627
## run 2 0.3049273
## run 3 0.3033796

# select the run with the lowest cross-entropy for K = 2
best = which.min(ce)

# display the Q-matrix

```

```

Q.matrix <- as.matrix(Q(project, K = 2, run = best))
dim(Q.matrix)

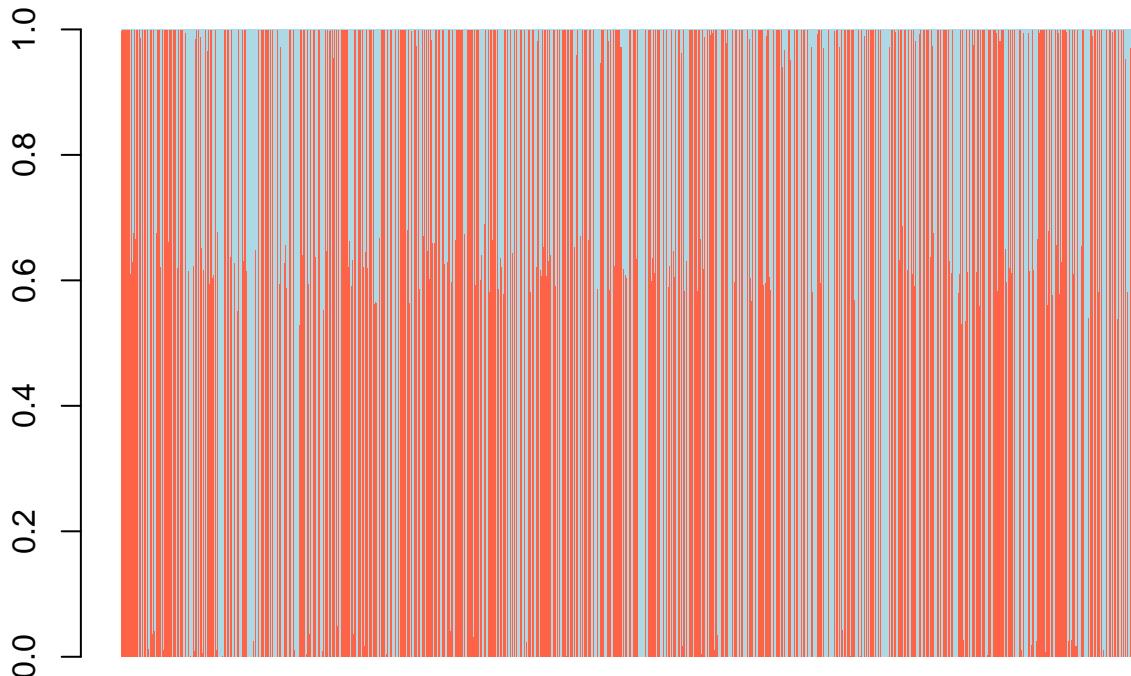
## [1] 1000      2
cluster <- apply(Q.matrix, 1, which.max)
my.colors <- c("tomato", "lightblue", "olivedrab")#, "gold")

ord <- order(ind$envi)
dim(Q.matrix)

## [1] 1000      2
bp <- barplot(t(Q.matrix[ord,]),
               border = NA,
               space = 0,
               col = my.colors,
               xlab = "Individuals",
               ylab = "Ancestry proportions",
               main = "Ancestry matrix")

```

**Ancestry matrix**



```
#axis(1, at = 1:nrow(Q.matrix), labels = bp$order, las = 3, cex.axis = .4)
```

```
# get the ancestral genotype frequency matrix, G, for the 2nd run for K = 4.
G.matrix = G(project, K = 3, run = 2)
```

Admixture/ancestry based on pruned data (snmf in LEA package)

```
write.geno(t(training$G[which_pruned,]), paste0("temp/", seed, "genotypes_LD.geno"))
```

```

## [1] "temp/1505550948364genotypes_LD.geno"
project_LD = snmf(paste0("temp/", seed, "genotypes_LD.geno"),
                  K = 1:4,
                  entropy = TRUE,
                  repetitions = 3,
                  project = "new")

## The project is saved into :
## temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
## project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
## remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] 1124381285
## [1] "*****"
## [1] "*          create.dataset           *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)          1000
##      -L (number of loci)                 7570
##      -s (seed random init)             1124381285
##      -r (percentage of masked data)    0.05
##      -x (genotype file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp/
##      -o (output file in .geno format)   /Users/katie/Desktop/RecombinationGenomeScans/temp/
##
## Write genotype file with masked data, /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.geno
##
## [1] "*****"
## [1] "* sNMF K = 1 repetition 1      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)          1000
##      -L (number of loci)                 7570
##      -K (number of ancestral pops)       1
##      -x (input file)                   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.geno
##      -q (individual admixture file)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.q
##      -g (ancestral frequencies file)   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.g
##      -i (number max of iterations)     200
##      -a (regularization parameter)     10
##      -s (seed random init)            140399310320229
##      -e (tolerance error)              1E-05
##      -p (number of processes)          1
##      - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
##
## Main algorithm:
##

```

```

## Least-square error: 1166322.377169
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            7570
##          -K (number of ancestral pops)   1
##          -x (genotype file)           /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -q (individual admixture)     /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -g (ancestral frequencies)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          - diploid
##
## Cross-Entropy (all data): 0.268156
## Cross-Entropy (masked data): 0.271961
## The project is saved into :
##   temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 2 repetition 1      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            7570
##          -K (number of ancestral pops)   2
##          -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -i (number max of iterations)   200
##          -a (regularization parameter)  10
##          -s (seed random init)         140399310320229
##          -e (tolerance error)          1E-05
##          -p (number of processes)       1
##          - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##
## Main algorithm:
##   [
##   [=====]
## Number of iterations: 67

```

```

## 
## Least-square error: 1160320.111864
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/
## 
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
## 
##          -n (number of individuals)      1000
##          -L (number of loci)           7570
##          -K (number of ancestral pops)  2
##          -x (genotype file)          /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -q (individual admixture)    /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -g (ancestral frequencies)   /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -i (with masked genotypes)   /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          - diploid
## 
## Cross-Entropy (all data):  0.265945
## Cross-Entropy (masked data):  0.271035
## The project is saved into :
##   temp/1505550948364genotypes_LD.snmfProject
## 
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
## 
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
## 
## [1] "*****"
## [1] "* sNMF K = 3  repetition 1      *"
## [1] "*****"
## summary of the options:
## 
##          -n (number of individuals)      1000
##          -L (number of loci)           7570
##          -K (number of ancestral pops)  3
##          -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -i (number max of iterations)  200
##          -a (regularization parameter) 10
##          -s (seed random init)        140399310320229
##          -e (tolerance error)         1E-05
##          -p (number of processes)     1
##          - diploid
## 
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## 
## Main algorithm:
## [
## =====

```

```

## Number of iterations: 200
##
## Least-square error: 1155945.216224
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/
##
## [1] "*****"
## [1] "* cross-entropy estimation *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)           7570
##          -K (number of ancestral pops)   3
##          -x (genotype file)           /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -q (individual admixture)     /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -g (ancestral frequencies)    /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          - diploid
##
## Cross-Entropy (all data): 0.264815
## Cross-Entropy (masked data): 0.270628
## The project is saved into :
##   temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 4  repetition 1      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)           7570
##          -K (number of ancestral pops)   4
##          -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##          -i (number max of iterations)  200
##          -a (regularization parameter)  10
##          -s (seed random init)        140399310320229
##          -e (tolerance error)          1E-05
##          -p (number of processes)       1
##          - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
## Main algorithm:
## [

```

```

## [=====]
## Number of iterations: 80
##
## Least-square error: 1151268.873457
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)      1000
##      -L (number of loci)            7570
##      -K (number of ancestral pops)   4
##      -x (genotype file)           /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##      -q (individual admixture)     /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##      -g (ancestral frequencies)    /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##      -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/150555
##      - diploid
##
## Cross-Entropy (all data): 0.263684
## Cross-Entropy (masked data): 0.270307
## The project is saved into :
##   temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] 1588698204
## [1] "*****"
## [1] "*          create.dataset      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)      1000
##      -L (number of loci)            7570
##      -s (seed random init)        1588698204
##      -r (percentage of masked data) 0.05
##      -x (genotype file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp
##      -o (output file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp
##
## Write genotype file with masked data, /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948
##
## [1] "*****"
## [1] "* sNMF K = 1 repetition 2      *"
## [1] "*****"
## summary of the options:
##
##      -n (number of individuals)      1000
##      -L (number of loci)            7570

```

```

## -K (number of ancestral pops) 1
## -x (input file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -i (number max of iterations) 200
## -a (regularization parameter) 10
## -s (seed random init) 10178632796
## -e (tolerance error) 1E-05
## -p (number of processes) 1
## - diploid

##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
##
## Main algorithm:
##
## Least-square error: 1166506.952719
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
## [1] "*****"
## [1] "* cross-entropy estimation *"
## [1] "*****"
## summary of the options:
##
## -n (number of individuals) 1000
## -L (number of loci) 7570
## -K (number of ancestral pops) 1
## -x (genotype file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -q (individual admixture) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -g (ancestral frequencies) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -i (with masked genotypes) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## - diploid

##
## Cross-Entropy (all data): 0.268267
## Cross-Entropy (masked data): 0.269712
## The project is saved into :
## temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
## project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
## remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 2 repetition 2 *"
## [1] "*****"
## summary of the options:
##
## -n (number of individuals) 1000
## -L (number of loci) 7570
## -K (number of ancestral pops) 2
## -x (input file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf

```

```

## -q (individual admixture file)          /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -g (ancestral frequencies file)        /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -i (number max of iterations)          200
## -a (regularization parameter)          10
## -s (seed random init)                 1285783919708
## -e (tolerance error)                  1E-05
## -p (number of processes)              1
## - diploid

##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
##
## Main algorithm:
## [
## [=====]
## Number of iterations: 44
##
## Least-square error: 1160217.004970
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
## [1] "*****"
## [1] "* cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
## -n (number of individuals)           1000
## -L (number of loci)                 7570
## -K (number of ancestral pops)       2
## -x (genotype file)                  /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -q (individual admixture)           /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -g (ancestral frequencies)          /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## -i (with masked genotypes)          /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## - diploid

##
## Cross-Entropy (all data):    0.266061
## Cross-Entropy (masked data):  0.268612
## The project is saved into :
##   temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 3  repetition 2      *"
## [1] "*****"
## summary of the options:
##
## -n (number of individuals)           1000
## -L (number of loci)                 7570
## -K (number of ancestral pops)       3

```

```

##      -x (input file)          /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##      -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##      -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##      -i (number max of iterations) 200
##      -a (regularization parameter) 10
##      -s (seed random init)        140399774637148
##      -e (tolerance error)         1E-05
##      -p (number of processes)     1
##      - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
##
## Main algorithm:
## [                                     ]
## [=====]
## Number of iterations: 200
##
## Least-square error: 1156194.975267
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
## [1] *****
## [1] * cross-entropy estimation      *
## [1] *****
## summary of the options:
##
##      -n (number of individuals)      1000
##      -L (number of loci)            7570
##      -K (number of ancestral pops)   3
##      -x (genotype file)             /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##      -q (individual admixture)       /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##      -g (ancestral frequencies)     /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##      -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##      - diploid
##
## Cross-Entropy (all data): 0.26483
## Cross-Entropy (masked data): 0.268587
## The project is saved into :
##   temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] *****
## [1] * sNMF K = 4  repetition 2      *
## [1] *****
## summary of the options:
##
##      -n (number of individuals)      1000
##      -L (number of loci)            7570

```

```

##          -K (number of ancestral pops)           4
##          -x (input file)                      /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -q (individual admixture file)        /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -g (ancestral frequencies file)       /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -i (number max of iterations)         200
##          -a (regularization parameter)        10
##          -s (seed random init)                140399774637148
##          -e (tolerance error)                 1E-05
##          -p (number of processes)              1
##          - diploid

##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##
##
## Main algorithm:
## [                                     ]
## [=====]
## Number of iterations: 32
##
## Least-square error: 1151774.282301
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##
## [1] "*****"
## [1] "*      cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            7570
##          -K (number of ancestral pops)  4
##          -x (genotype file)           /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -q (individual admixture)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -g (ancestral frequencies)   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -i (with masked genotypes)   /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          - diploid

##
## Cross-Entropy (all data):      0.263858
## Cross-Entropy (masked data):  0.268392
## The project is saved into :
##   temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] 1304616726
## [1] "*****"
## [1] "*          create.dataset      *"
## [1] "*****"
## summary of the options:
##

```

```

##          -n (number of individuals)      1000
##          -L (number of loci)            7570
##          -s (seed random init)        1304616726
##          -r (percentage of masked data) 0.05
##          -x (genotype file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp/
##          -o (output file in .geno format) /Users/katie/Desktop/RecombinationGenomeScans/temp/
##
## Write genotype file with masked data, /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
## [1] "*****"
## [1] "* sNMF K = 1  repetition 3      *"
## [1] "*****"
##
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            7570
##          -K (number of ancestral pops)   1
##          -x (input file)               /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##          -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##          -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##          -i (number max of iterations)   200
##          -a (regularization parameter)   10
##          -s (seed random init)         140399490555670
##          -e (tolerance error)          1E-05
##          -p (number of processes)       1
##          - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
## Main algorithm:
##
## Least-square error: 1166157.790949
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
## [1] "*****"
## [1] "*      cross-entropy estimation      *"
## [1] "*****"
##
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            7570
##          -K (number of ancestral pops)   1
##          -x (genotype file)               /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##          -q (individual admixture)        /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##          -g (ancestral frequencies)       /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##          -i (with masked genotypes)      /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##          - diploid
##
## Cross-Entropy (all data):    0.268169
## Cross-Entropy (masked data): 0.271663
## The project is saved into :
##   temp/1505550948364genotypes_LD.snmfProject

```

```

## 
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 2 repetition 3      *"
## [1] "*****"
## summary of the options:
##
##           -n (number of individuals)      1000
##           -L (number of loci)            7570
##           -K (number of ancestral pops)  2
##           -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##           -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##           -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##           -i (number max of iterations)  200
##           -a (regularization parameter) 10
##           -s (seed random init)        140399490555670
##           -e (tolerance error)         1E-05
##           -p (number of processes)     1
##           - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##
## 
## Main algorithm:
##   [
##   =====
## Number of iterations: 63
##
## Least-square error: 1160457.645513
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##
## [1] "*****"
## [1] "*    cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##           -n (number of individuals)      1000
##           -L (number of loci)            7570
##           -K (number of ancestral pops)  2
##           -x (genotype file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##           -q (individual admixture)       /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##           -g (ancestral frequencies)     /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##           -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##           - diploid
##
## Cross-Entropy (all data):      0.265971
## Cross-Entropy (masked data):   0.270489
## The project is saved into :

```

```

##  temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
##  project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
##  remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 3  repetition 3      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            7570
##          -K (number of ancestral pops)   3
##          -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -g (ancestral frequencies file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -i (number max of iterations)   200
##          -a (regularization parameter)  10
##          -s (seed random init)         9894551318
##          -e (tolerance error)          1E-05
##          -p (number of processes)       1
##          - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##
##
## Main algorithm:
##  [
##  =====
## Number of iterations: 200
##
## Least-square error: 1156137.323757
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##
## [1] "*****"
## [1] "*    cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##          -n (number of individuals)      1000
##          -L (number of loci)            7570
##          -K (number of ancestral pops)   3
##          -x (genotype file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -q (individual admixture)        /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -g (ancestral frequencies)      /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          -i (with masked genotypes)       /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmfProject
##          - diploid
##
## Cross-Entropy (all data):      0.26482
## Cross-Entropy (masked data):   0.270282

```

```

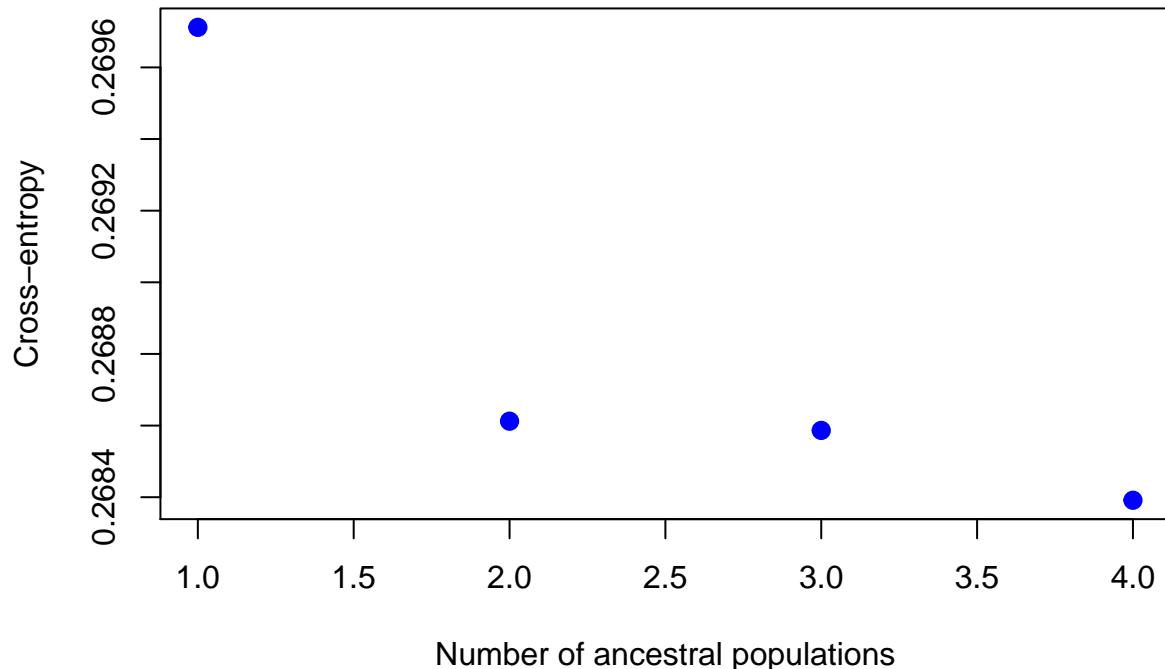
## The project is saved into :
##   temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## [1] "*****"
## [1] "* sNMF K = 4  repetition 3      *"
## [1] "*****"
## summary of the options:
##
##           -n (number of individuals)      1000
##           -L (number of loci)            7570
##           -K (number of ancestral pops)  4
##           -x (input file)              /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##           -q (individual admixture file) /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##           -g (ancestral frequencies)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##           -i (number max of iterations) 200
##           -a (regularization parameter) 10
##           -s (seed random init)        140399490555670
##           -e (tolerance error)         1E-05
##           -p (number of processes)     1
##           - diploid
##
## Read genotype file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
## Main algorithm:
##   [
##   =====]
## ] Number of iterations: 66
## Least-square error: 1151837.892485
## Write individual ancestry coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
## Write ancestral allele frequency coefficient file /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##
## [1] "*****"
## [1] "*      cross-entropy estimation      *"
## [1] "*****"
## summary of the options:
##
##           -n (number of individuals)      1000
##           -L (number of loci)            7570
##           -K (number of ancestral pops)  4
##           -x (genotype file)             /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##           -q (individual admixture)      /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##           -g (ancestral frequencies)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##           -i (with masked genotypes)    /Users/katie/Desktop/RecombinationGenomeScans/temp/1505550948364genotypes_LD.snmf
##           - diploid
##
## Cross-Entropy (all data):      0.263744

```

```

## Cross-Entropy (masked data): 0.270006
## The project is saved into :
##   temp/1505550948364genotypes_LD.snmfProject
##
## To load the project, use:
##   project = load.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
##
## To remove the project, use:
##   remove.snmfProject("temp/1505550948364genotypes_LD.snmfProject")
#project
# plot cross-entropy criterion of all runs of the project
plot(project_LD, cex = 1.2, col = "blue", pch = 19)

```



```

# get the cross-entropy of all runs for K = 3
ce = cross.entropy(project_LD, K = 2)
ce

##           K = 2
## run 1 0.2710351
## run 2 0.2686123
## run 3 0.2704886

# select the run with the lowest cross-entropy for K = 2
best = which.min(ce)

# display the Q-matrix
Q.matrix <- as.matrix(Q(project_LD, K = 2, run = best))
dim(Q.matrix)

## [1] 1000     2

cluster <- apply(Q.matrix, 1, which.max)
my.colors <- c("tomato", "lightblue")#, "olivedrab")#, "gold")

```

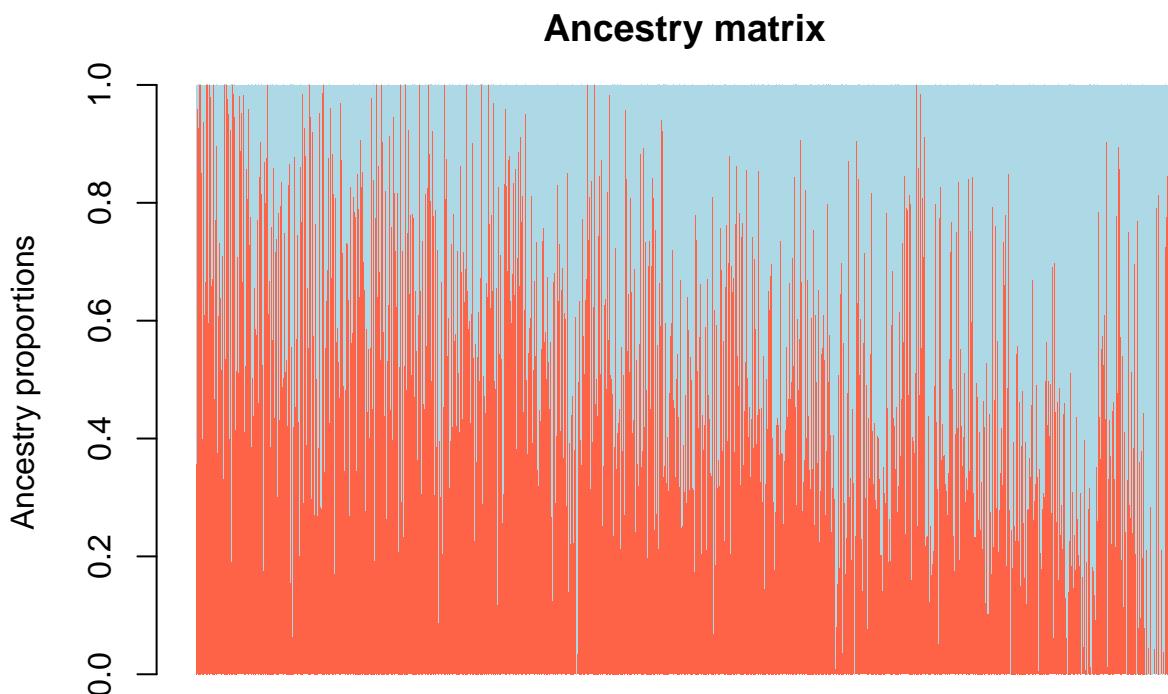
```

ord <- order(ind$envi)
dim(Q.matrix)

## [1] 1000    2

par(mfrow=c(1,1), mar=c(4,4,3,1))
bp <- barplot(t(Q.matrix[ord,]),
               border = NA,
               space = 0,
               col = my.colors,
               xlab = "Individuals (sorted by environment)",
               ylab = "Ancestry proportions",
               main = "Ancestry matrix")

```



Individuals (sorted by environment)

```

axis(1, at = 1:nrow(Q.matrix), labels = bp$order, las = 3, cex.axis = .4)

# get the ancestral genotype frequency matrix, G, for the 2nd run for K = 4.
G.matrix = G(project, K = 3, run = 2)

```

## OutFLANK

All data

```

dim(training$G)

## [1] 13007 1000

```

```

FstDataFrame <- MakeDiploidFSTMat(t(training$G), training$position, ind$group)

## Calculating FSTs, may take a few minutes...
## [1] "10000 done of 13007"

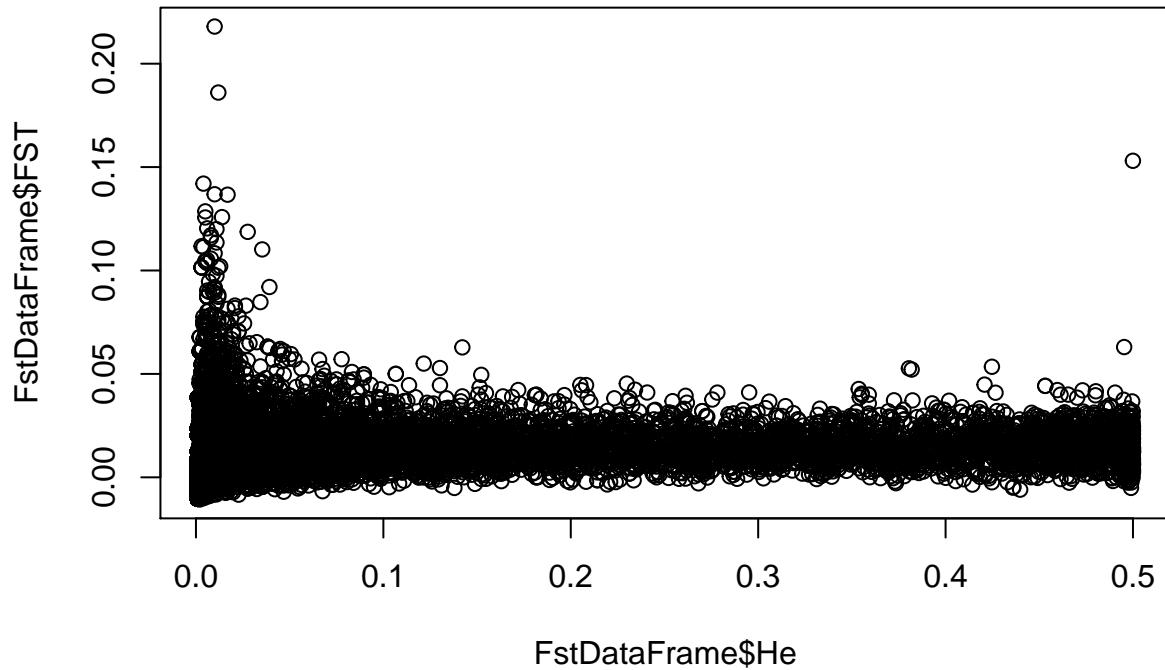
head(FstDataFrame)

##   LocusName      He        FST        T1        T2  FSTNoCorr
## 1       6 0.0139020 0.003361768 2.338152e-05 0.0069551250 0.02268250
## 2      18 0.0139020 0.003361768 2.338152e-05 0.0069551250 0.02268250
## 3      19 0.2742080 0.008486843 1.164493e-03 0.1372115720 0.02933242
## 4      69 0.4771020 0.024441181 5.837472e-03 0.2388375695 0.04304795
## 5      96 0.0009995 -0.001051497 -5.257326e-07 0.0004999849 0.01856812
## 6     99 0.4881420 0.016059153 3.923439e-03 0.2443117104 0.03599456

##   T1NoCorr      T2NoCorr meanAlleleFreq
## 1 1.577722e-04 0.0069556776      0.0070
## 2 1.577722e-04 0.0069556776      0.9930
## 3 4.025093e-03 0.1372233360      0.8360
## 4 1.028225e-02 0.2388558483      0.3930
## 5 9.284527e-06 0.0005000253      0.9995
## 6 8.794614e-03 0.2443317427      0.4230

#str(FstDataFrame)
par(mifrow=c(1,1))
plot(FstDataFrame$He, FstDataFrame$FST)

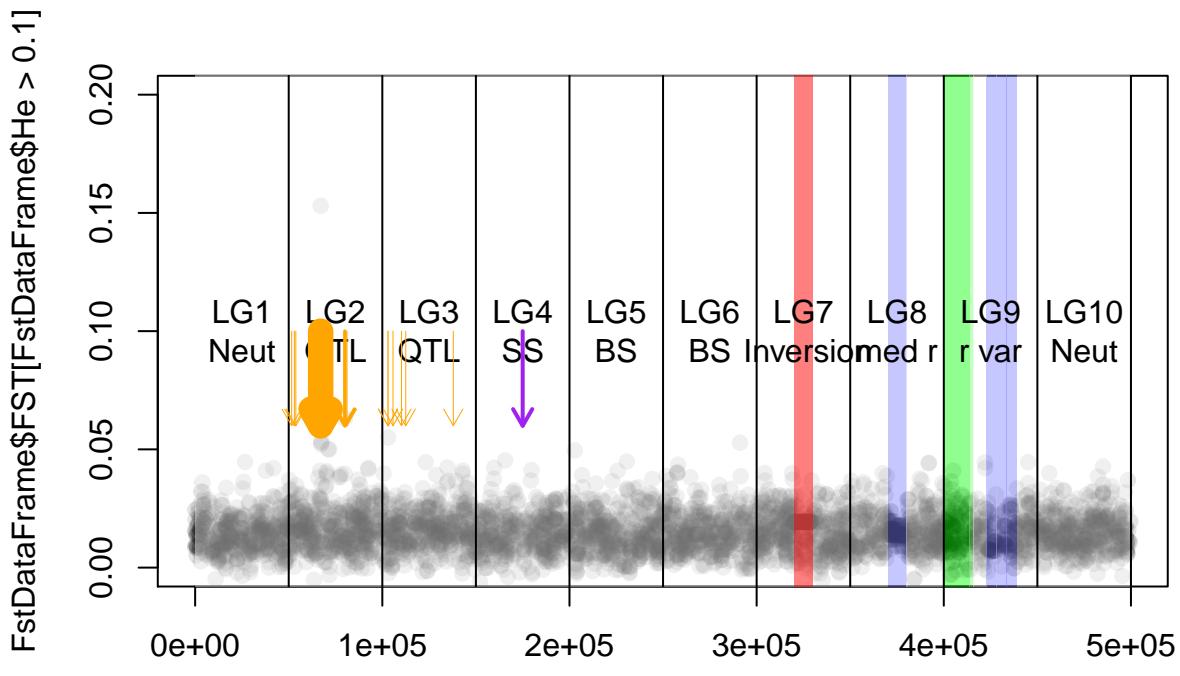
```



```

plot(as.numeric(FstDataFrame$LocusName)[FstDataFrame$He>0.1], FstDataFrame$FST[FstDataFrame$He>0.1], yl
plot_layers(y_head=0.1, y_arrows=c(0.1,0.06))

```



as.numeric(FstDataFrame\$LocusName)[FstDataFrame\$He > 0.1]

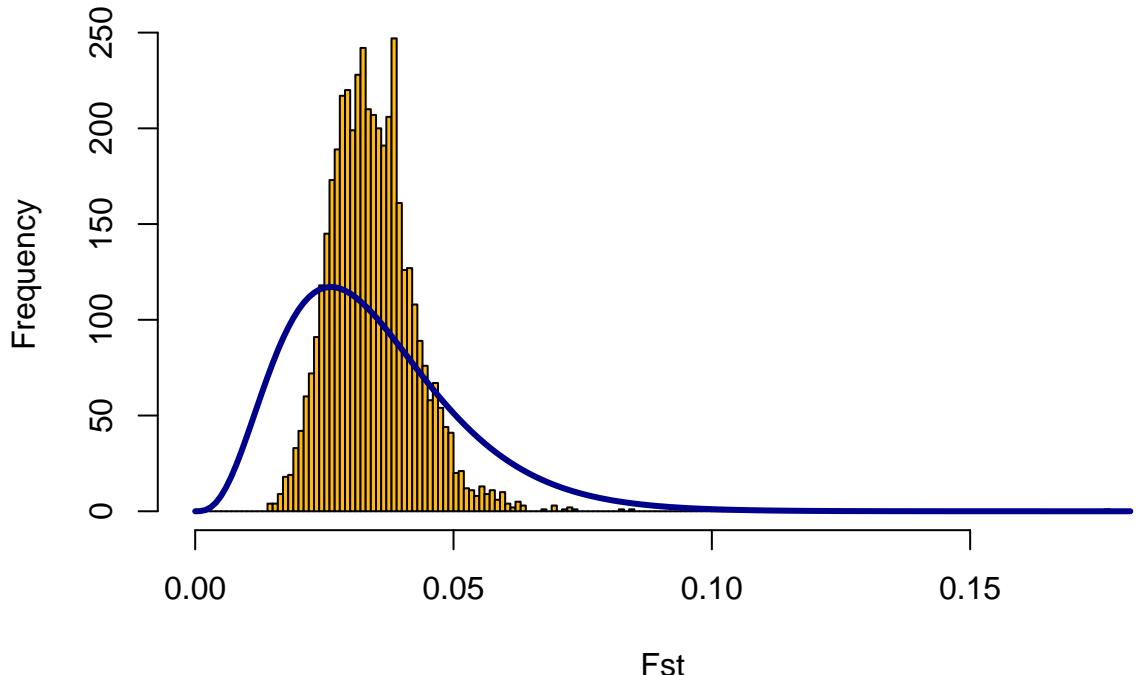
```

k <- 39 ## Number of pops
out_ini <- OutFLANK(FstDataFrame, NumberOfSamples=k)
## Run outflank on FST dataframe
#out_ini <- OutFLANK(FstDataFrame[FstDataFrame$He>0.05,], NumberOfSamples=k)
## Run outflank without low He loci

# Plot results to compare chi-squared distribution vs. actual FST distribution
OutFLANKResultsPlotter(out_ini, withOutliers = TRUE,
                       NoCorr = TRUE, Hmin = 0.1, binwidth = 0.001, Zoom =
                           FALSE, RightZoomFraction = 0.05, titletext = NULL)

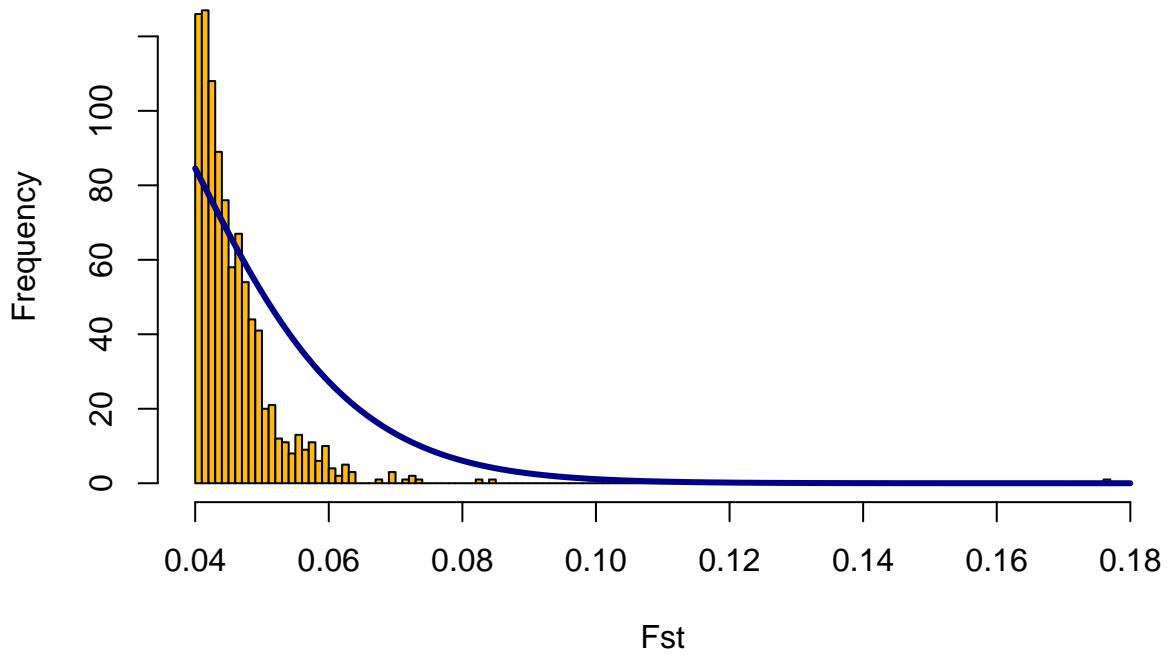
```

### Fst without sample size correction

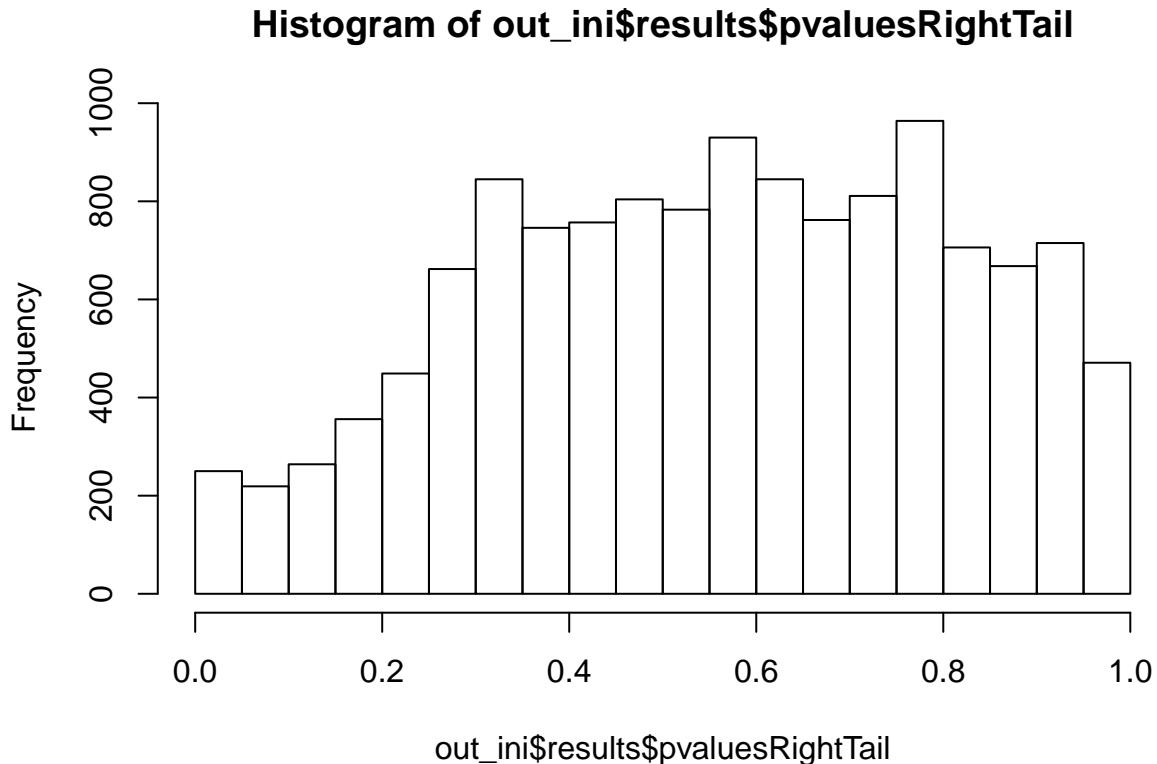


```
## Poor fit, particularly on right tail
OutFLANKResultsPlotter(out_ini, withOutliers = TRUE,
                       NoCorr = TRUE, Hmin = 0.1, binwidth = 0.001, Zoom =
TRUE, RightZoomFraction = 0.15, titletext = NULL)
```

### Fst without sample size correction

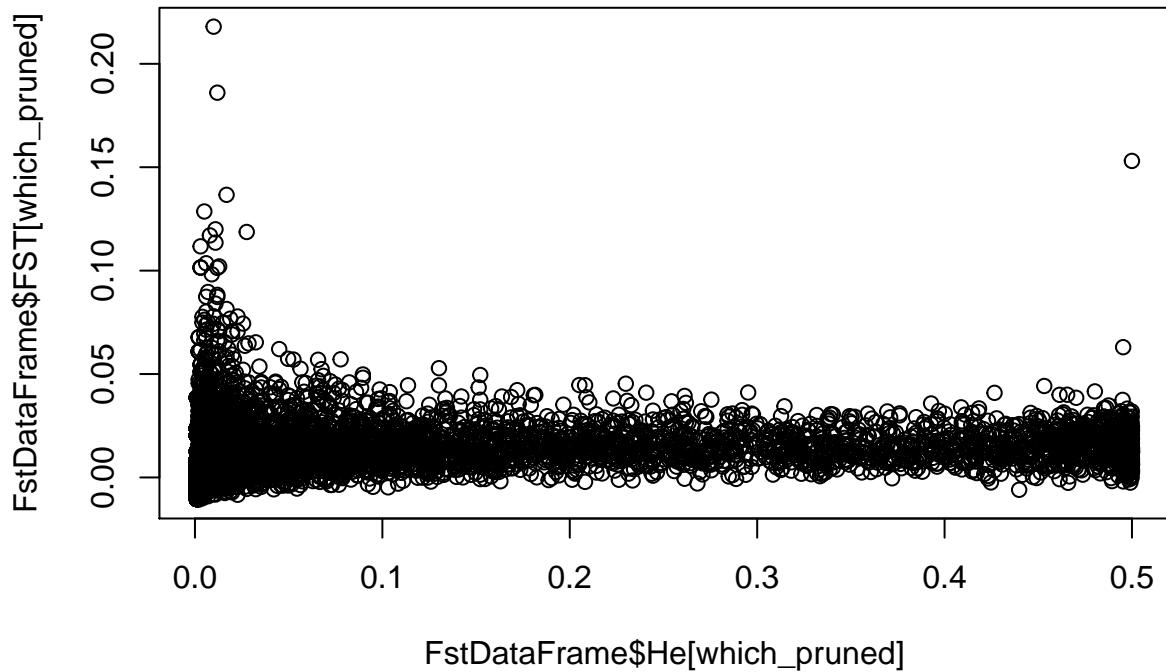


```
# Histogram of P-values weird  
hist(out_ini$results$pvaluesRightTail, breaks = 20)
```



With LD pruning

```
##### LD Pruning #####  
  
##### Evaluating OutFLANK with pruned data #####  
plot(FstDataFrame$He[which_pruned], FstDataFrame$FST[which_pruned])
```



```

Fstdf2 <- FstDataFrame[which_pruned,]
dim(Fstdf2)

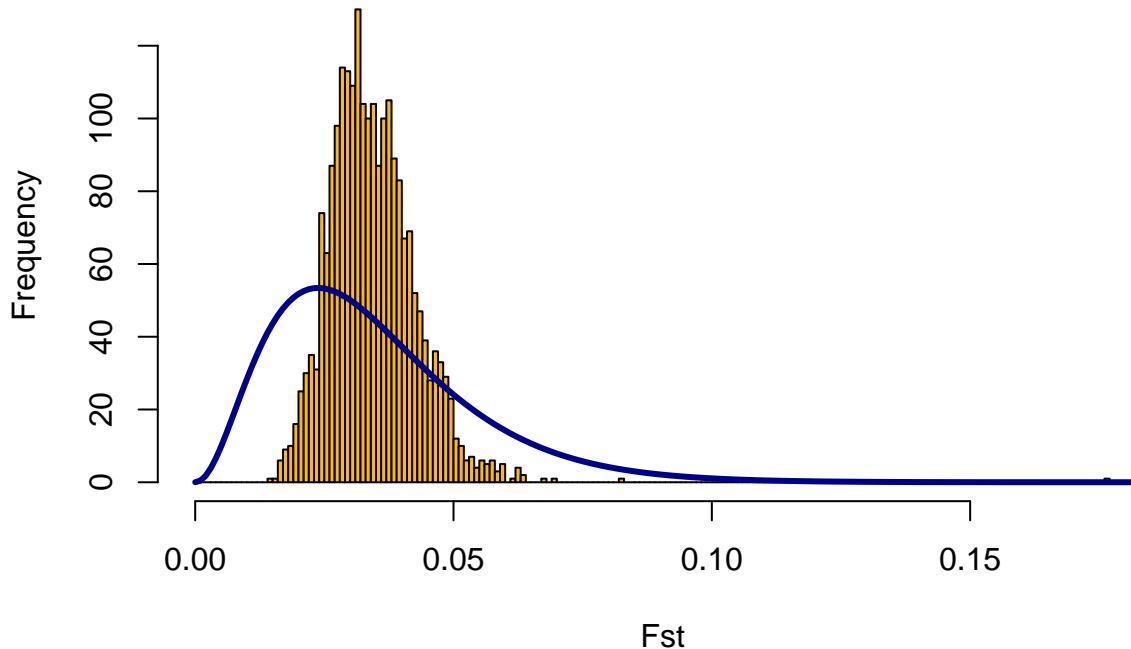
## [1] 7570     9

Fstdf3 <- Fstdf2[Fstdf2$He>0.05,]

### Trimming without He trimming
out_trim1 <- OutFLANK(Fstdf2, NumberOfSamples=k, Hmin = 0.05)
OutFLANKResultsPlotter(out_trim1, withOutliers = TRUE,
                      NoCorr = TRUE, Hmin = 0.1, binwidth = 0.001, Zoom =
                      FALSE, RightZoomFraction = 0.15, titletext = NULL)

```

## Fst without sample size correction



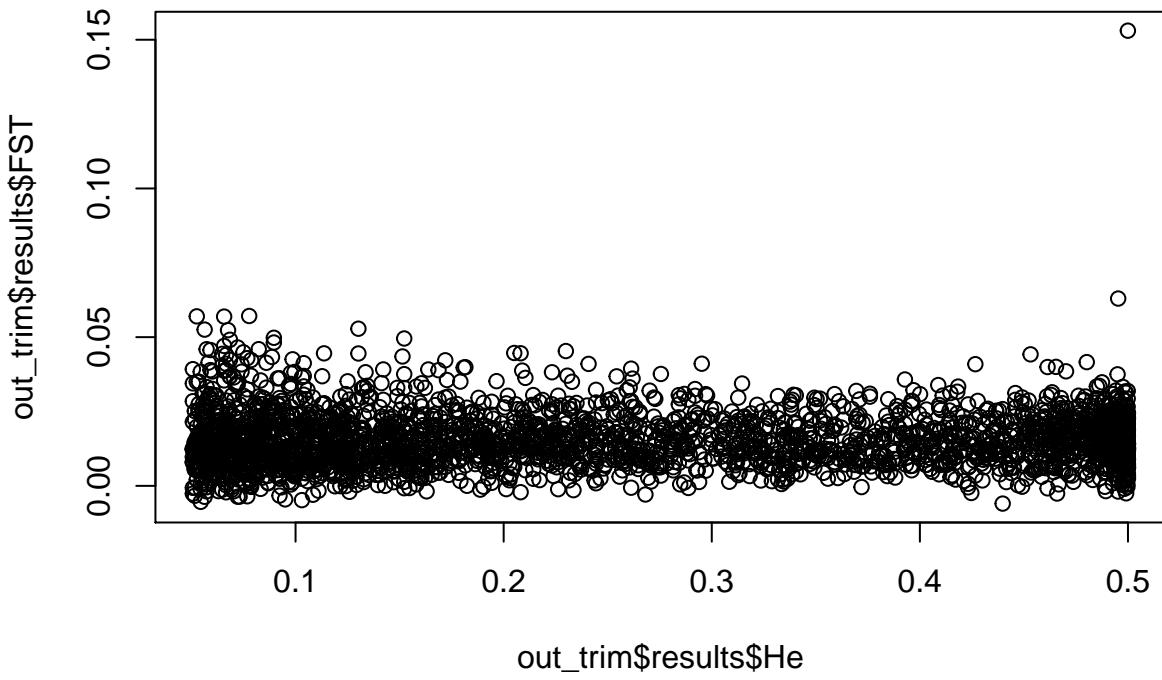
```

out_trim <- OutFLANK(Fstdf3, NumberOfSamples=k)
# I do not think that OutFLANK is removing low H loci correctly
# The fit is much better if I remove these manually than if I do not
head(out_trim$results)

##      LocusName       He        FST        T1        T2    FSTNoCorr
## 8       135 0.4947980 0.015079867 0.003734289 0.2476341 0.03474974
## 15      501 0.2647020 0.009371741 0.001241315 0.1324530 0.02864995
## 17      642 0.2983875 0.014339649 0.002141371 0.1493322 0.03398947
## 28      944 0.2528955 0.008735190 0.001105401 0.1265457 0.02890669
## 30      963 0.2239755 0.017437018 0.001954683 0.1120996 0.03626194
## 40     1384 0.4896320 0.014271419 0.003497093 0.2450417 0.03377313
##          T1NoCorr   T2NoCorr meanAlleleFreq indexOrder GoodH    qvalues
## 8  0.008605917 0.2476541           0.5510         1 goodH 0.9808741
## 15 0.003795074 0.1324635           0.8430         2 goodH 0.9808741
## 17 0.005076132 0.1493442           0.1825         3 goodH 0.9808741
## 28 0.003658322 0.1265562           0.8515         4 goodH 0.9808741
## 30 0.004065264 0.1121083           0.8715         5 goodH 0.9808741
## 40 0.008276489 0.2450614           0.5720         6 goodH 0.9808741
##          pvalues pvaluesRightTail OutlierFlag
## 8  0.9034933      0.4517466    FALSE
## 15 0.5198422      0.7400789    FALSE
## 17 0.9751864      0.4875932    FALSE
## 28 0.5423314      0.7288343    FALSE
## 30 0.7666788      0.3833394    FALSE
## 40 0.9958455      0.4979228    FALSE

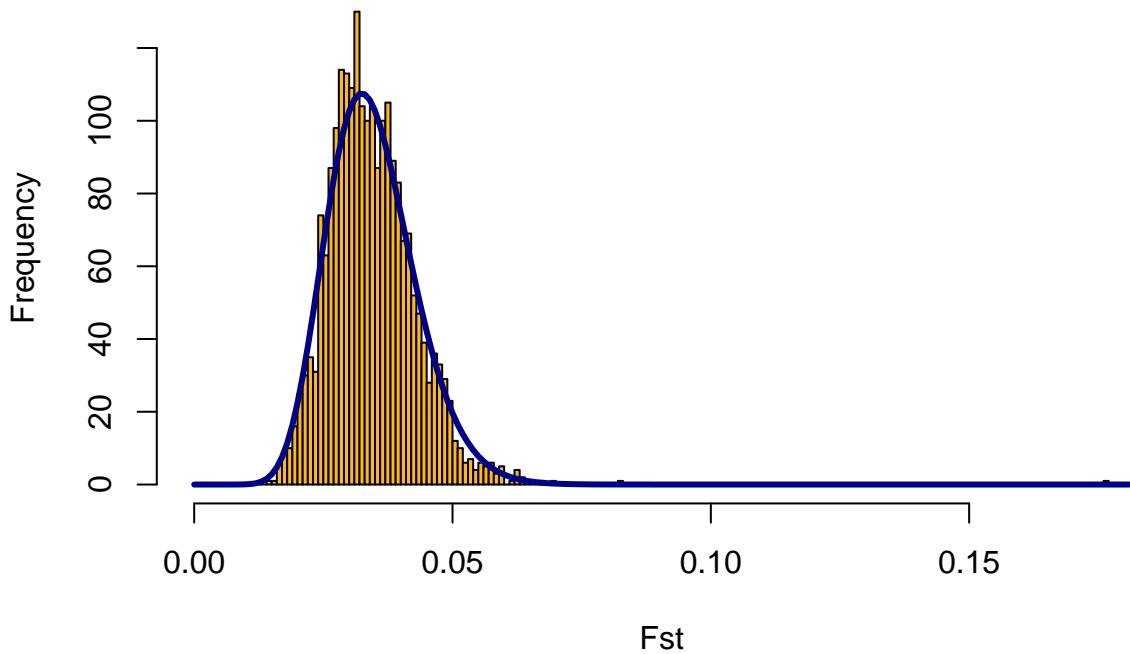
plot(out_trim$results$He, out_trim$results$FST)

```



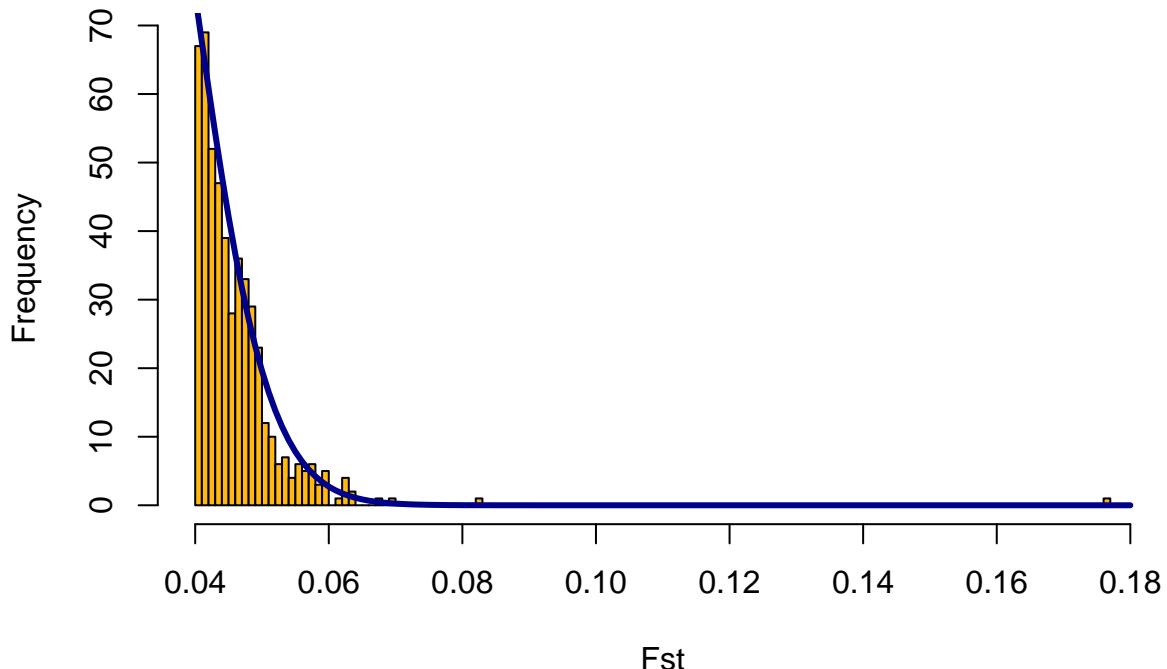
```
OutFLANKResultsPlotter(out_trim, withOutliers = TRUE,
                      NoCorr = TRUE, Hmin = 0.1, binwidth = 0.001, Zoom =
                      FALSE, RightZoomFraction = 0.15, titletext = NULL)
```

### Fst without sample size correction



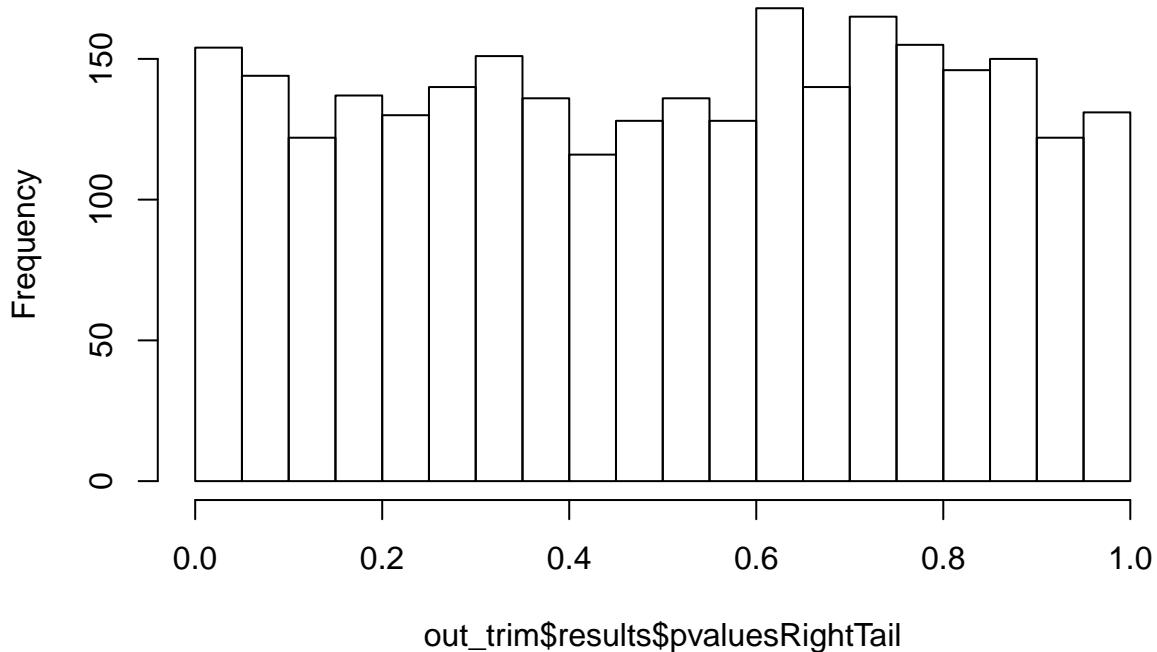
```
OutFLANKResultsPlotter(out_trim, withOutliers = TRUE,
                      NoCorr = TRUE, Hmin = 0.1, binwidth = 0.001, Zoom =
                      TRUE, RightZoomFraction = 0.10, titletext = NULL)
```

## Fst without sample size correction



```
# Decent distribution fit, no trimming needed.  
hist(out_trim$results$pvaluesRightTail, breaks = 20)
```

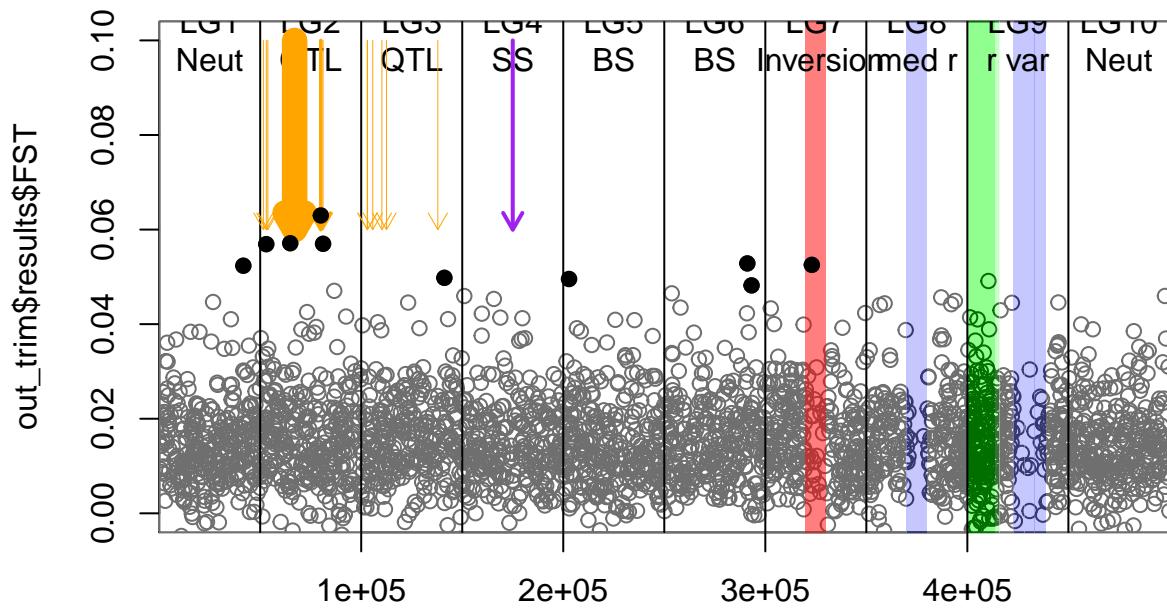
**Histogram of out\_trim\$results\$pvaluesRightTail**



```
# still a conservative histogram, which is typical of outflank  
outliers_crit <- out_trim$results$qvalues<0.2
```

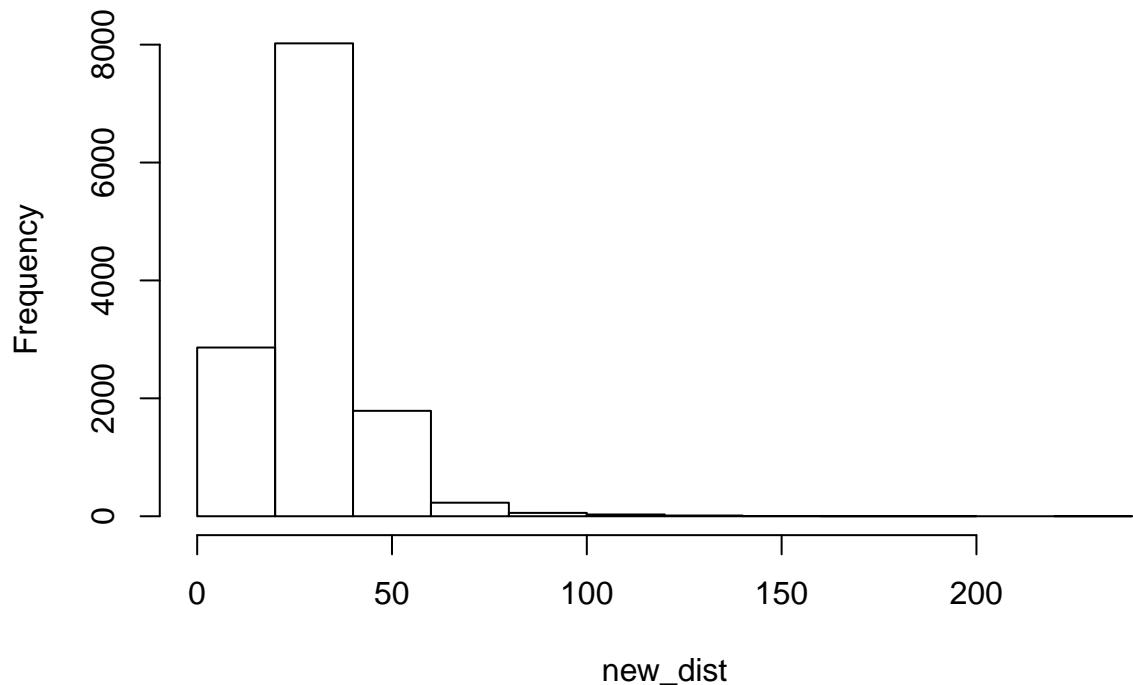
```
(outliers_LD <- out_trim$results$LocusName[outliers_crit])
## [1] 41662 53012 64875 67058 79995 81126 141163 202865 291111 293201
## [11] 323034
# outliers identified

### Plot trimmed data only
plot(out_trim$results$LocusName,out_trim$results$FST, ylim=c(0,0.1), xaxs="i")
plot_layers(y_head=0.1, y_arrows=c(0.1,0.06))
points(out_trim$results$LocusName[outliers_crit],
       out_trim$results$FST[outliers_crit], pch=19)
```



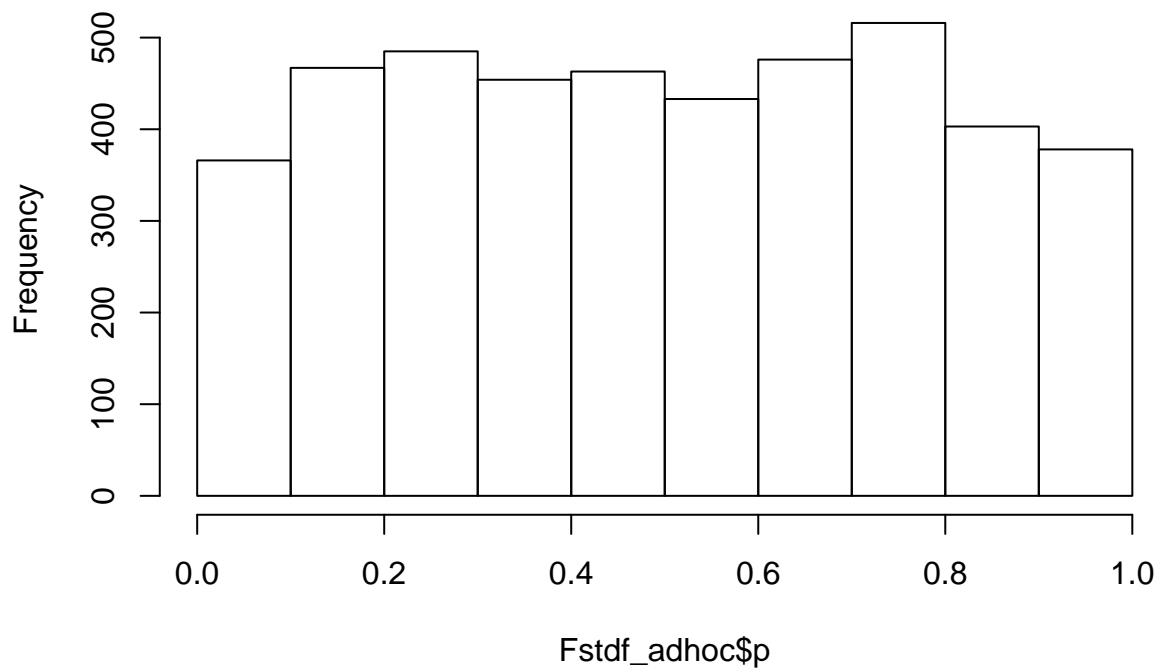
```
### Ad-hoc estimates of p-values for all data
Fstdf_adhoc <- FstDataFrame
new_dist <- FstDataFrame$FSTNoCorr*out_trim$dfInferred/out_trim$FSTNoCorrbar
hist(new_dist)
```

### Histogram of new\_dist

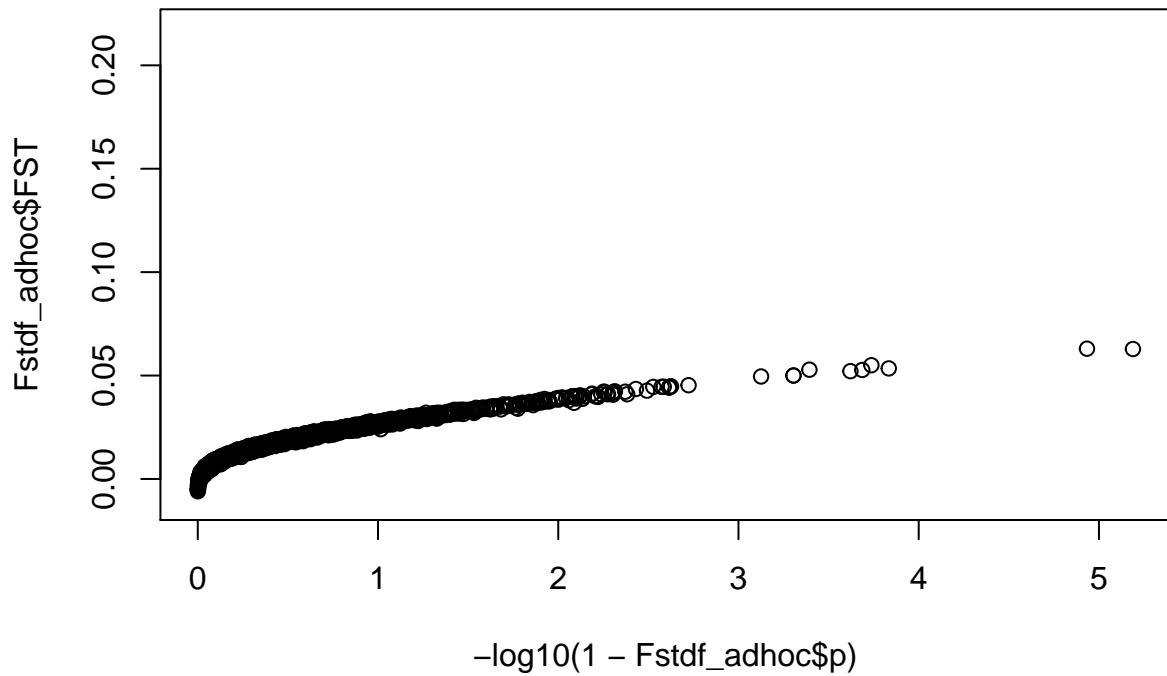


```
Fstdf_adhoc$p <- pchisq(new_dist, df = out_trim$dfInferred)
Fstdf_adhoc$p[Fstdf_adhoc$p < 0.1] <- NA
hist(Fstdf_adhoc$p)
```

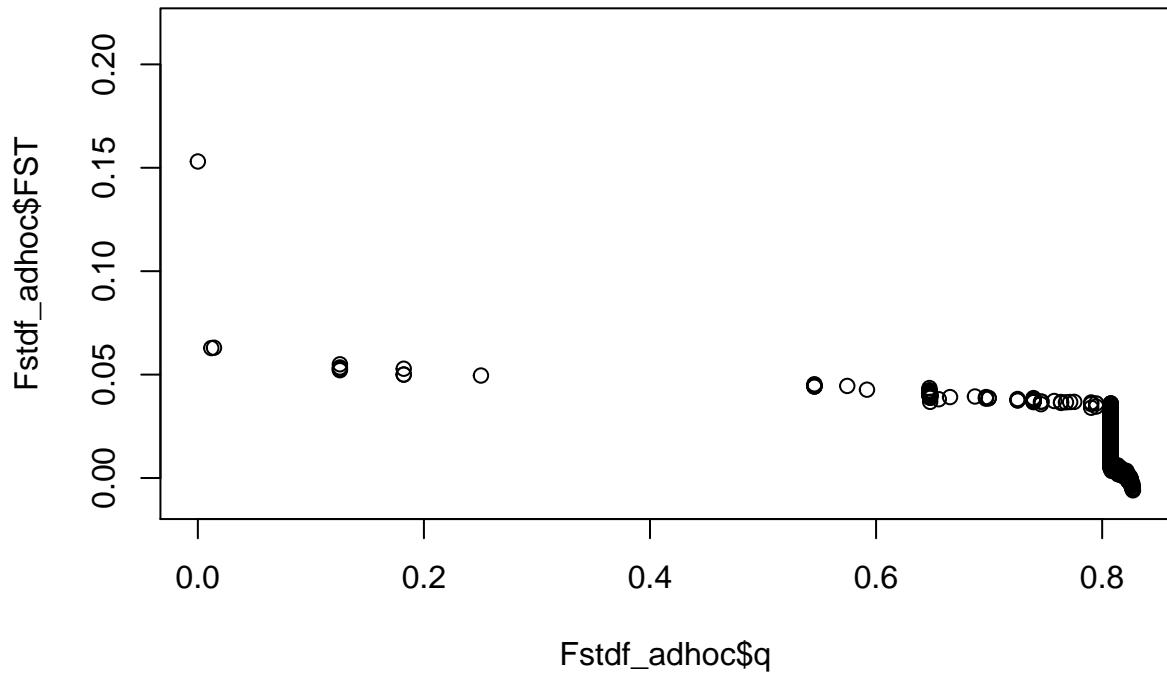
### Histogram of Fstdf\_adhoc\$p



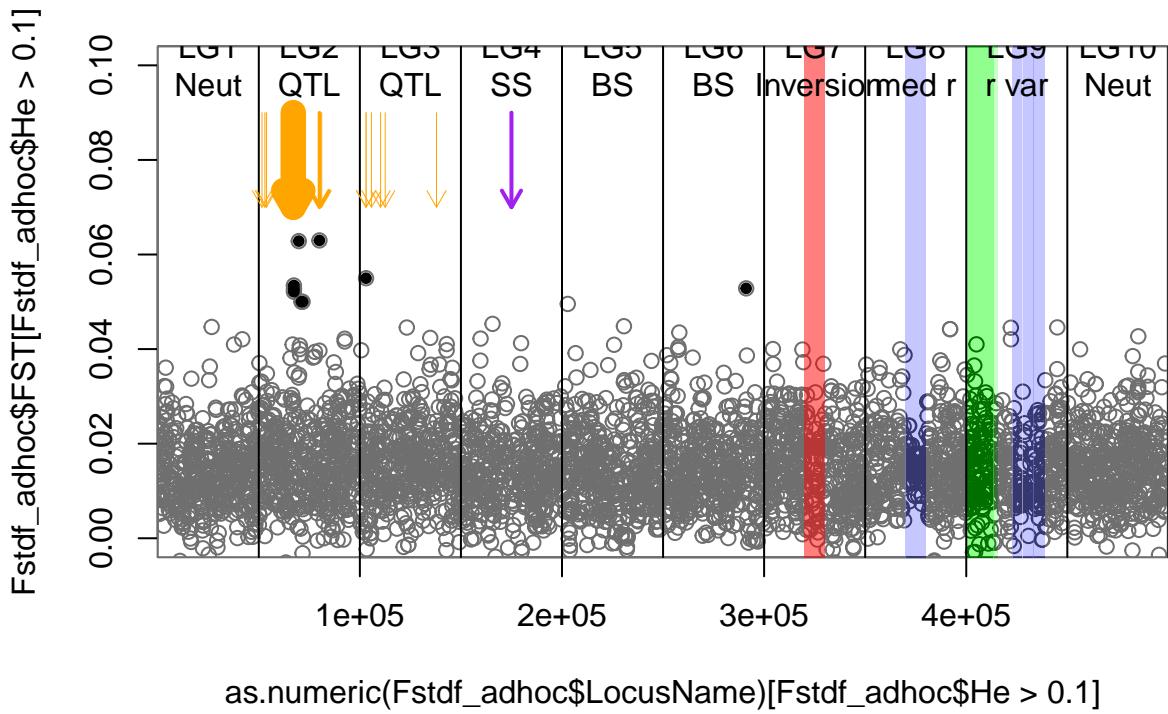
```
plot(-log10(1-Fstdf_adhoc$p) , Fstdf_adhoc$FST)
```



```
Fstdf_adhoc$q <- qvalue(1-Fstdf_adhoc$p)$qvalues  
plot(Fstdf_adhoc$q, Fstdf_adhoc$FST)
```



```
plot(as.numeric(Fstdf_adhoc$LocusName)[Fstdf_adhoc$He>0.1] , Fstdf_adhoc$FST[Fstdf_adhoc$He>0.1] , ylim=c(0,0.1)  
plot_layers(y_head=0.1, y_arrows=c(0.09, 0.07))  
points(Fstdf_adhoc$LocusName[Fstdf_adhoc$q<0.2] , Fstdf_adhoc$FST[Fstdf_adhoc$q<0.2] , pch=20)
```



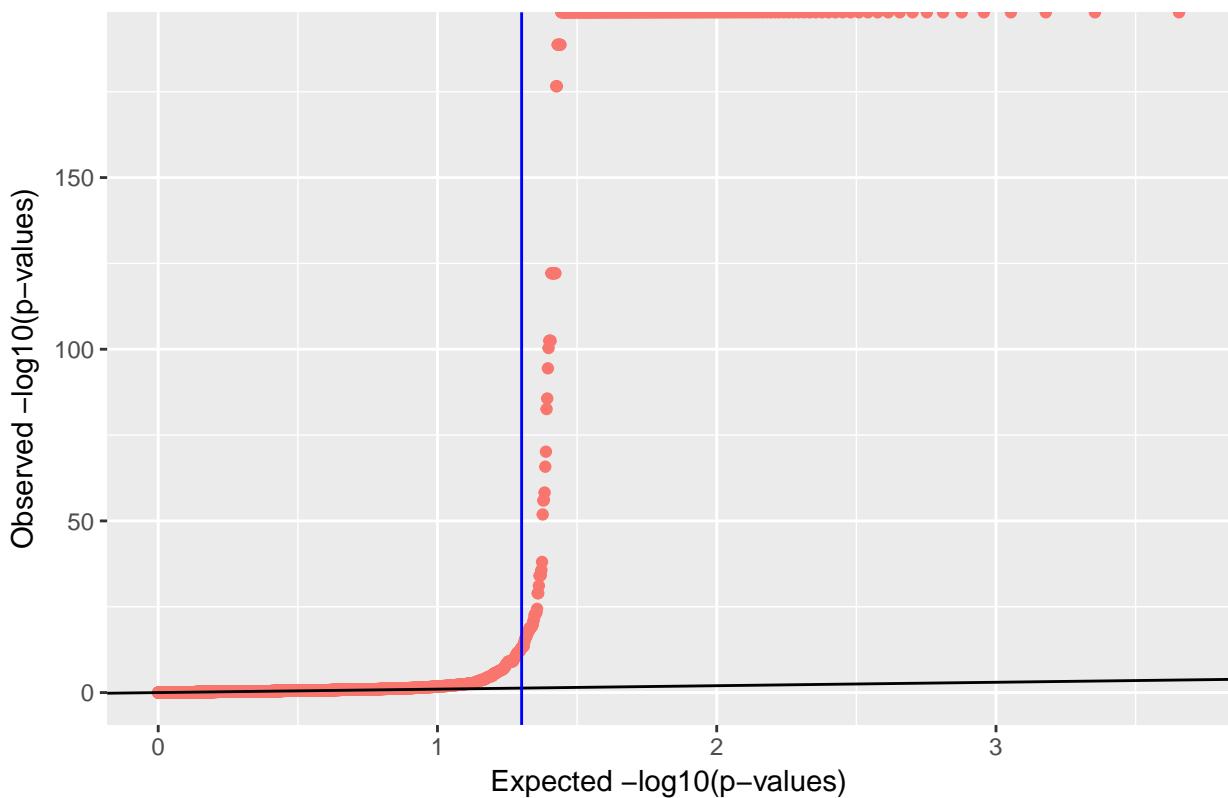
With individuals assigned to pops based on PCs from all data

## PCADAPT

All data

```
plot(x, option = "qqplot", threshold = 0.05, main="pcadapt")
```

Q-Q plot

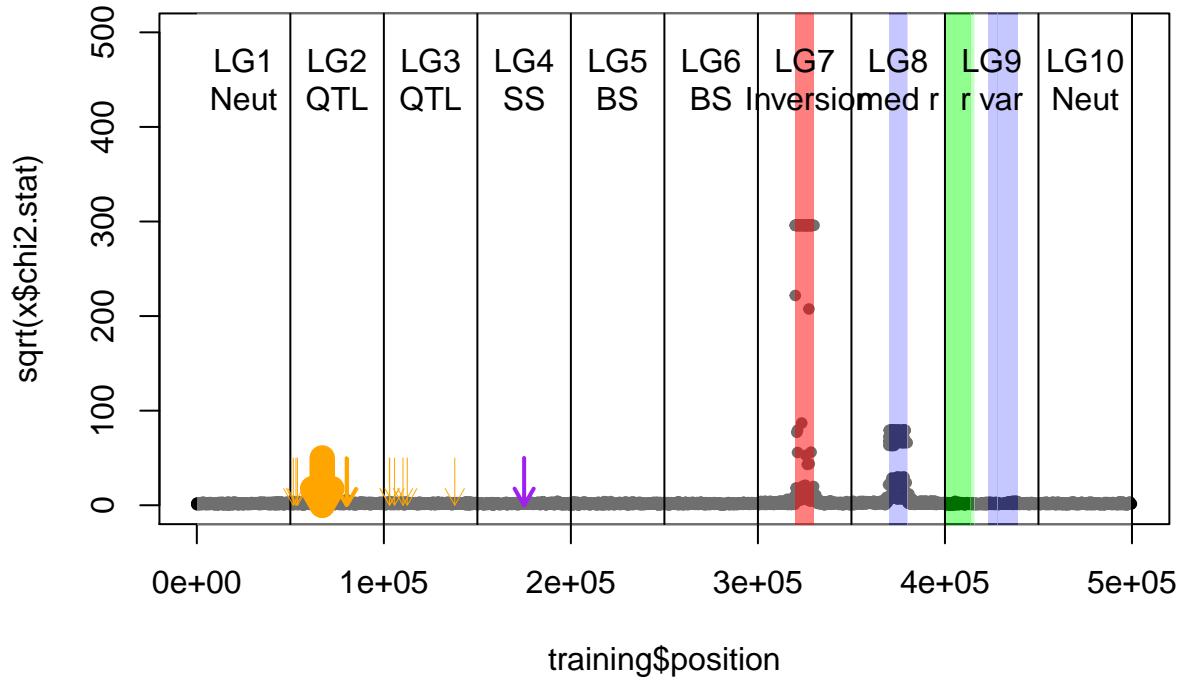


```
#plot(x, option = "stat.distribution")
summary(x$chi2.stat)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.    NA's
##      0.01     1.90     3.36  1907.64     6.02 87583.62    8483

# Default output from PCAdapt
par(mfrow=c(1,1))
plot(training$position, sqrt(x$chi2.stat), col="black", pch=20, main="pcadapt without LD pruning", yl...
```

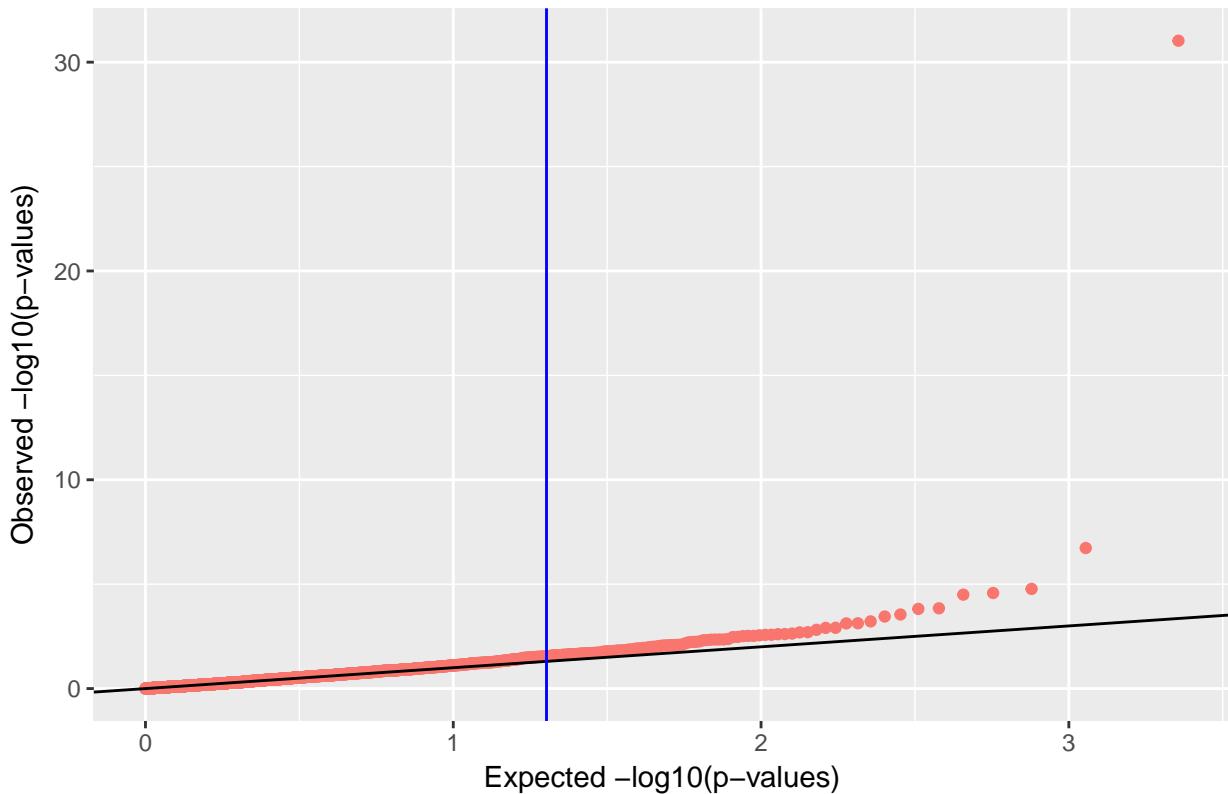
## pcadapt without LD pruning



With LD pruning

```
plot(x_LD, option = "qqplot", threshold = 0.05, main="pcadapt")
```

Q–Q plot



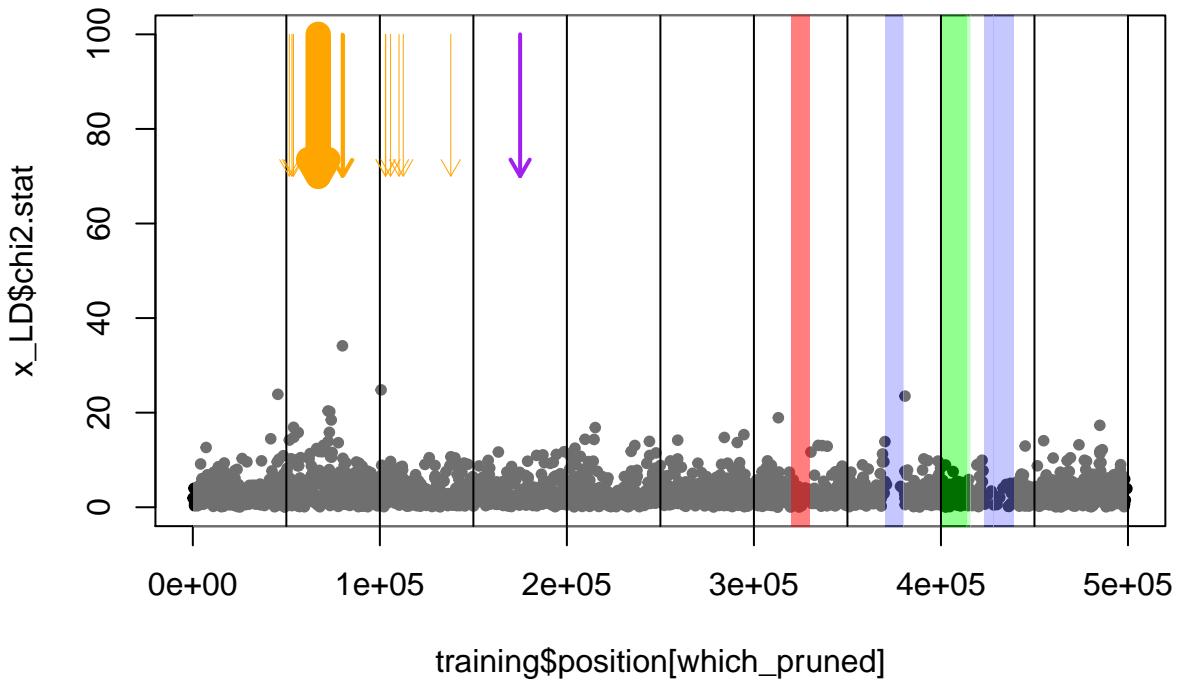
```
#plot(x, option = "stat.distribution")
summary(x_LD$chi2.stat)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.    NA's
## 0.010   1.190   2.366   3.289   4.306 147.442    5300
```

```
# Default output from PCAdapt
```

```
par(mfrow=c(1,1))
plot(training$position[which_pruned], x_LD$chi2.stat, col="black", pch=20, main="pcadapt without LD p")
plot_layers(y_head=450, y_arrows = c(100,70))
```

## pcadapt without LD pruning



## LFMM

Ridge regression model phenotype ~ genotype

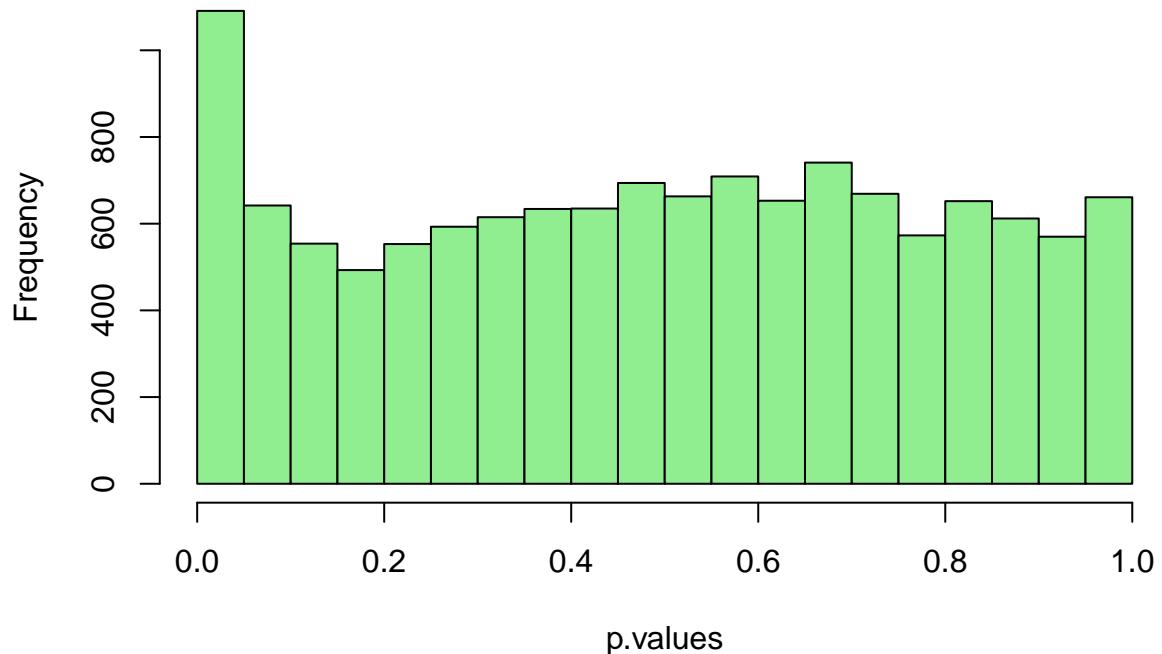
```
# extract scaled genotypes
scaled.genotype <- scale(as.matrix(t(training$G)))
#scaled.genotype <- as.matrix(t(sim1$G))
# extract scaled phenotypes
phen <- scale(as.matrix(ind$phenotype1))
# centering is important to remove mean
# x <- scale(as.matrix(sim1$phenotype1), center=TRUE, scale=FALSE)
# x <- as.matrix(sim1$phenotype1)
# to do mean and not SD. this might make it possible to get effect sizes
#pc <- prcomp(scaled.genotype,)
#plot(pc$sdev[1:20]^2)
#points(5,pc$sdev[5]^2, type = "h", lwd = 3, col = "blue")

# ridge regression
lfmm.ridge <- lfmm:::lfmm_ridge(Y = scaled.genotype, X = phen, K = 3, lambda = 1e-4)
#The lfmm.ridge object contains estimates for the latent variables and for the effect sizes. Here, the
lfmm.test <- lfmm:::lfmm_test(Y = scaled.genotype, X = phen, lfmm = lfmm.ridge, calibrate = "gif")
p.values <- lfmm.test$calibrated.pvalue
lfmm.test$gif

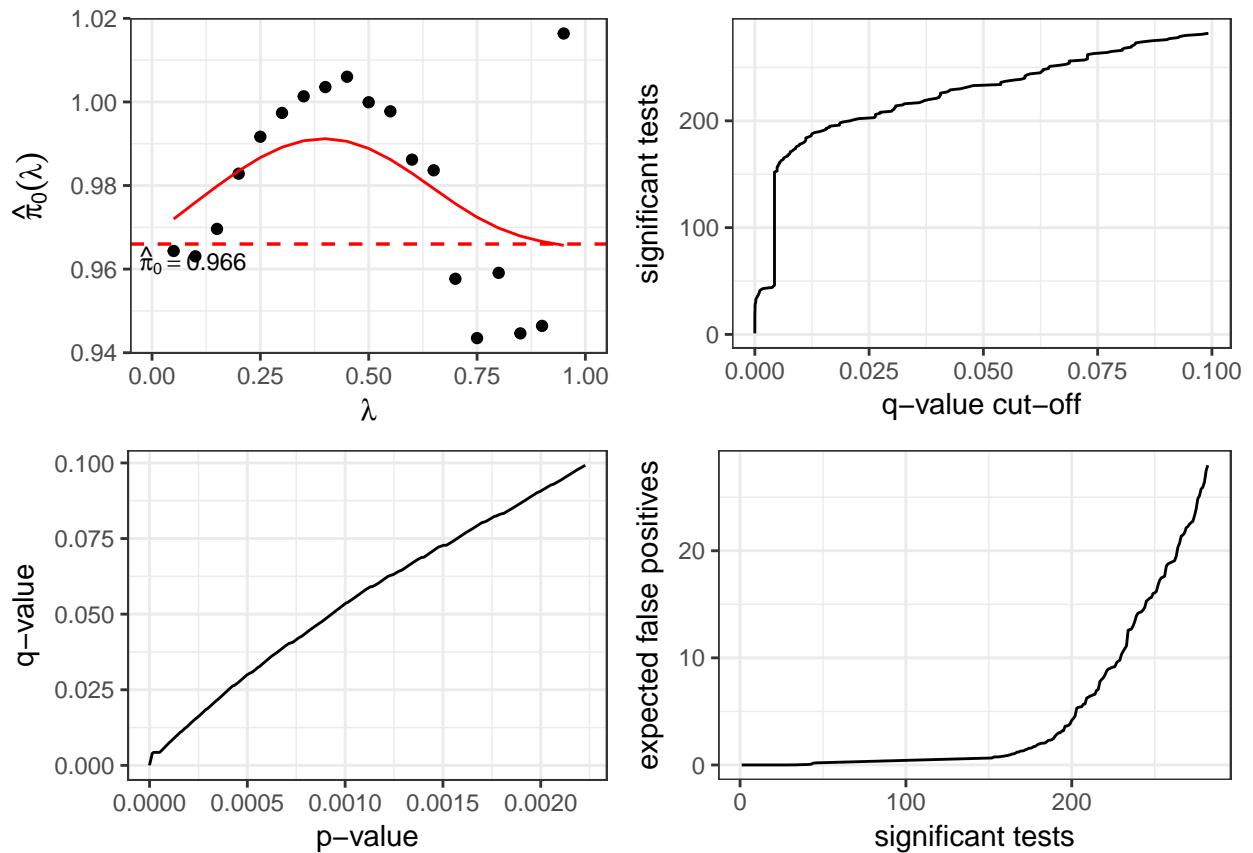
## [1] 3.388228
```

```
hist(p.values, col = "lightgreen", main="LFMM ridge")
```

**LFMM ridge**



```
qval <- qvalue::qvalue(p.values)
plot(qval)
```

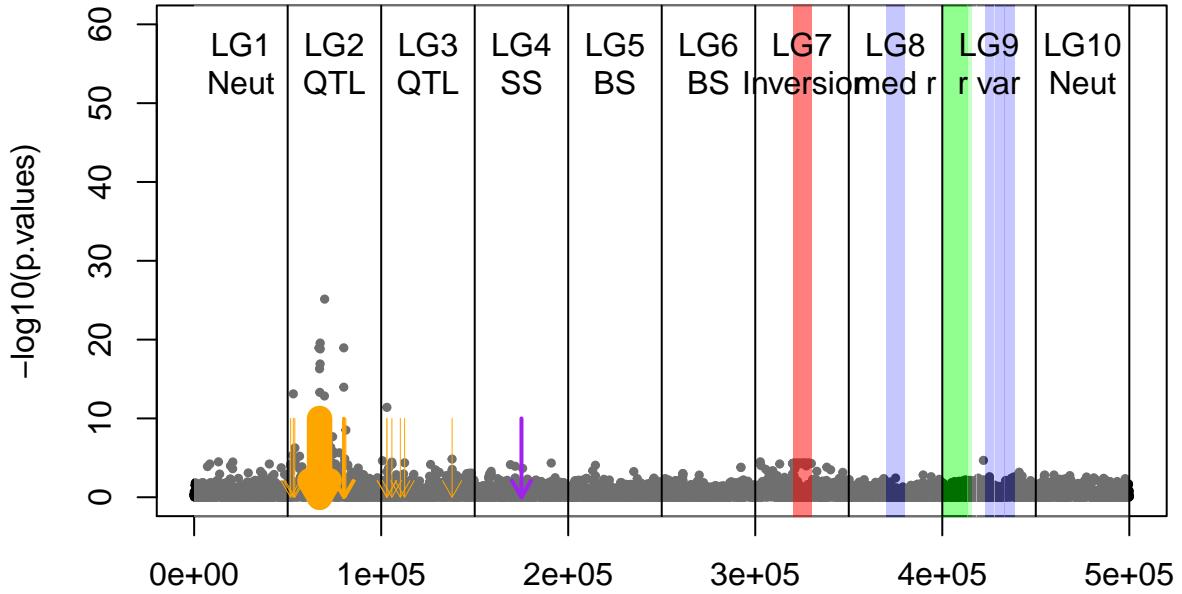


#The plot suggests that setting `fdr.level = 0.025` warrant few false positives.

```
qval <- qvalue::qvalue(p.values, fdr.level = 0.005)
candidates <- which(qval$significant)
```

```
plot(training$position, -log10(p.values), cex = .5, pch = 19, col = "black", main="LFMM ridge", ylim=c(0, 55))
plot_layers(y_head=55, y_arrows=c(10, 0))
```

## LFMM ridge



training\$position

#### Effect sizes ridge regression step 1: run lfmm\_ridge (or any lfmm model) and get the estimated latent factors from the U matrix (obj.lfmm\$U). When lfmm is run with K factors and n individuals, U is an n by K matrix.

step 2: perform a standard linear regression analysis of the phenotype on the SNP frequency (in the direction opposite to LFMM) by adding U as covariate to the model. This will estimate the LFMM effect size for each SNP. The R command should look like this: lm( y ~ . , data = data.frame(genotype[,i], U))

```
str(lfmm.ridge)
```

```
## List of 5
## $ K      : num 3
## $ lambda: num 1e-04
## $ U      : num [1:1000, 1:3] 10.87 -5.18 12.88 -3.15 -4.62 ...
## $ V      : num [1:13007, 1:3] -0.000918 0.000918 0.000393 -0.008504 -0.001573 ...
## $ B      : num [1:13007, 1] 0.0469 -0.0469 0.037 0.0174 0.011 ...
## - attr(*, "class")= chr "ridgeLFMM"
m2 <- which(training$position %in% (mut$position[mut$type=="m2" & mut$count==TRUE]))
dim(G)

## [1] 13011 1000
effects <- data.frame(position=training$position[m2], est_coef_ridge=NA, est_coef_PC=NA)
### Try Olivier's suggestion
for (i in 1:length(m2)){
  effects$est_coef_ridge[i] <- lm(phen ~., data = data.frame(gen = training$G[m2[i],], lfmm.ridge$U))$co
}

### Use PC axes as covariates
for (i in 1:length(m2)){
  effects$est_coef_PC[i] <- lm(phen ~., data = data.frame(gen = training$G[m2[i],], x_LD$scores))$coef[
```

```

}

effects

##      position est_coef_ridge est_coef_PC
## 1      51581    -1.26567006 -0.54155789
## 2      53012     1.49269679  0.47778962
## 3      53800    -0.68882563 -1.09829457
## 4      67017     0.03244131  0.07870101
## 5      67058     1.00111307  0.69022729
## 6      69766    -1.25495464 -0.59235010
## 7      70769    -1.25212009 -0.43574525
## 8      79995    -0.63273782 -0.19321557
## 9      81126     1.38199308  0.55501515
## 10     103054    1.05941901  0.35069708
## 11     105727    0.46937414  0.03574035
## 12     110242    -3.67957137 -2.78866071
## 13     112525    0.33567171  0.16265880
## 14     137919    -0.43395260 -0.19213616

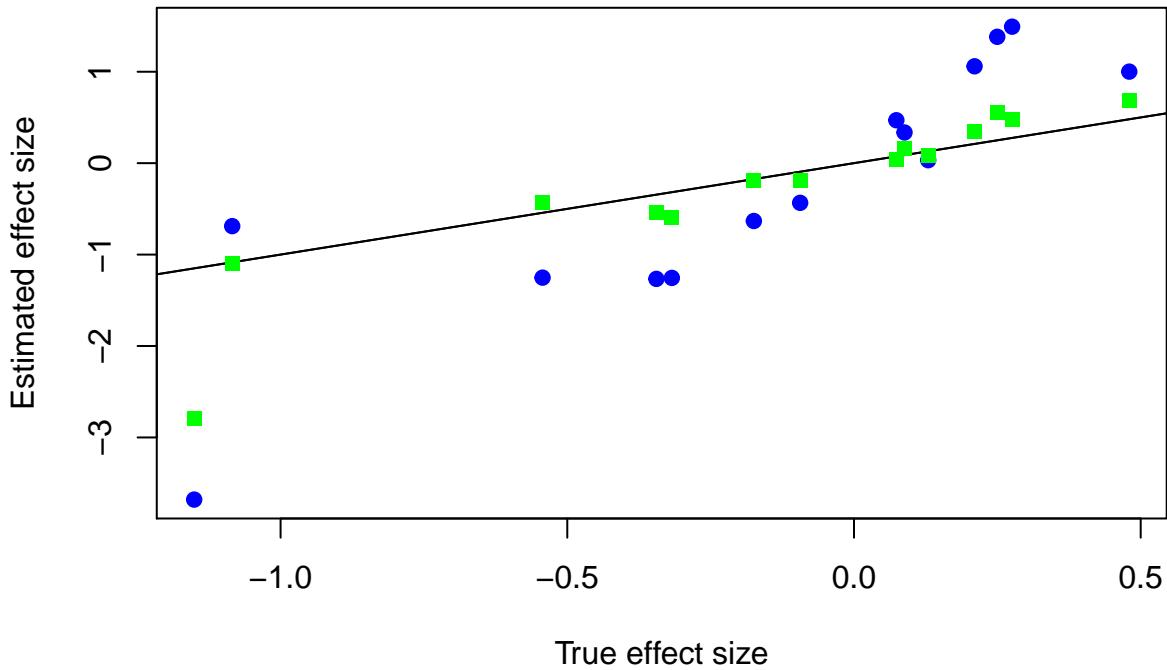
(new_mutns <- merge(muts, effects))

##      position   selCoef originGen type   freq   pa2      prop count
## 1      51581 -0.3446080      5987   m2 0.0080 0.001 0.01098901 TRUE
## 2      53012  0.2757600      5395   m2 0.0340 0.002 0.02197802 TRUE
## 3      53800 -1.0843600      5999   m2 0.0010 0.001 0.01098901 TRUE
## 4      67017  0.1294500      2702   m2 0.8465 0.002 0.02197802 TRUE
## 5      67058  0.4800990       53   m2 0.5045 0.058 0.63736264 TRUE
## 6      69766 -0.3176070      4318   m2 0.0770 0.007 0.07692308 TRUE
## 7      70769 -0.5431350      5988   m2 0.0080 0.002 0.02197802 TRUE
## 8      79995 -0.1745930      3902   m2 0.5485 0.008 0.08791209 TRUE
## 9      81126  0.2498160      5937   m2 0.0270 0.002 0.02197802 TRUE
## 10     103054  0.2102690      4228   m2 0.0650 0.003 0.03296703 TRUE
## 11     105727  0.0738757      4943   m2 0.1240 0.001 0.01098901 TRUE
## 12     110242 -1.1506800      5999   m2 0.0005 0.001 0.01098901 TRUE
## 13     112525  0.0880859      2707   m2 0.5750 0.002 0.02197802 TRUE
## 14     137919 -0.0938825      5800   m2 0.1820 0.001 0.01098901 TRUE

##      est_coef_ridge est_coef_PC
## 1      -1.26567006 -0.54155789
## 2       1.49269679  0.47778962
## 3      -0.68882563 -1.09829457
## 4       0.03244131  0.07870101
## 5       1.00111307  0.69022729
## 6      -1.25495464 -0.59235010
## 7      -1.25212009 -0.43574525
## 8      -0.63273782 -0.19321557
## 9       1.38199308  0.55501515
## 10      1.05941901  0.35069708
## 11      0.46937414  0.03574035
## 12     -3.67957137 -2.78866071
## 13      0.33567171  0.16265880
## 14     -0.43395260 -0.19213616

plot(new_mutns$selCoef, new_mutns$est_coef_ridge, abline(0,1), col="blue", pch=19, xlab="True effect size"
points(new_mutns$selCoef, new_mutns$est_coef_PC, abline(0,1), col="green", pch=15)

```



### LASSO model

```
#LMM parameters can alternatively be estimated by solving regularized least-squares minimization, with
lfmm.lasso <- lfmm:::lfmm_lasso(Y = scaled.genotype, X = phen, K = 3, nozero.prop = 0.02)

## It = 1/100, err2 = 0.999000000053069
## It = 2/100, err2 = 0.992443936559843
## It = 3/100, err2 = 0.992448956476981
## === lambda = 0.272089379187123, no zero B proportion = 0.00238333205197201
## It = 1/100, err2 = 0.992449244390459
## It = 2/100, err2 = 0.992443611515161
## === lambda = 0.259722496977145, no zero B proportion = 0.00276774044745137
## It = 1/100, err2 = 0.992443276991036
## It = 2/100, err2 = 0.992437233935841
## === lambda = 0.247917708649891, no zero B proportion = 0.00315214884293073
## It = 1/100, err2 = 0.992436872552334
## It = 2/100, err2 = 0.992430809570413
## === lambda = 0.236649466170892, no zero B proportion = 0.00338279388021834
## It = 1/100, err2 = 0.992430455598705
## It = 2/100, err2 = 0.992424324472974
## === lambda = 0.225893382703272, no zero B proportion = 0.00392096563388944
## It = 1/100, err2 = 0.992423970653498
```

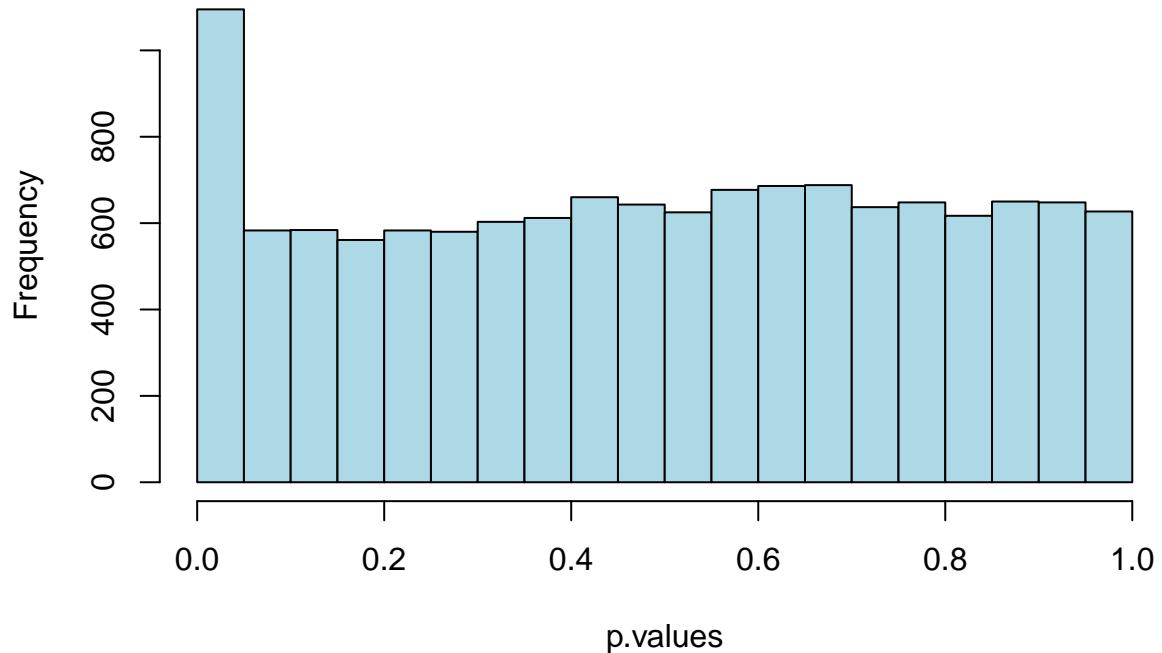
```

## It = 2/100, err2 = 0.992417351879739
## === lambda = 0.215626179829529, no zero B proportion = 0.00522795417851926
## It = 1/100, err2 = 0.99241694474686
## It = 2/100, err2 = 0.992409506261531
## === lambda = 0.205825637172164, no zero B proportion = 0.00668870608134082
## It = 1/100, err2 = 0.992409025379002
## It = 2/100, err2 = 0.992401111444979
## === lambda = 0.196470544304128, no zero B proportion = 0.00799569462597063
## It = 1/100, err2 = 0.992400571751582
## It = 2/100, err2 = 0.992391842262289
## === lambda = 0.187540654845016, no zero B proportion = 0.00961020988698393
## It = 1/100, err2 = 0.992391257156359
## It = 2/100, err2 = 0.992381580774635
## === lambda = 0.17901664264366, no zero B proportion = 0.0117628969016683
## It = 1/100, err2 = 0.992380906011195
## It = 2/100, err2 = 0.992369454530003
## === lambda = 0.170880059952288, no zero B proportion = 0.0141462289536403
## It = 1/100, err2 = 0.992368631542013
## It = 2/100, err2 = 0.992351987408391
## It = 3/100, err2 = 0.992349921758794
## === lambda = 0.163113297501739, no zero B proportion = 0.0172214961174752
## It = 1/100, err2 = 0.99234978725299
## It = 2/100, err2 = 0.992318464074598
## It = 3/100, err2 = 0.992316379116006
## === lambda = 0.155699546391307, no zero B proportion = 0.0216037518259399
#The lfmm.lasso object contains new estimates for the latent variables and for the effect sizes. Note
lfmm.test <- lfmm::lfmm_test(Y = scaled.genotype, X = phen, lfmm = lfmm.lasso, calibrate = "gif")
p.values <- lfmm.test$calibrated.pvalue
lfmm.test$gif

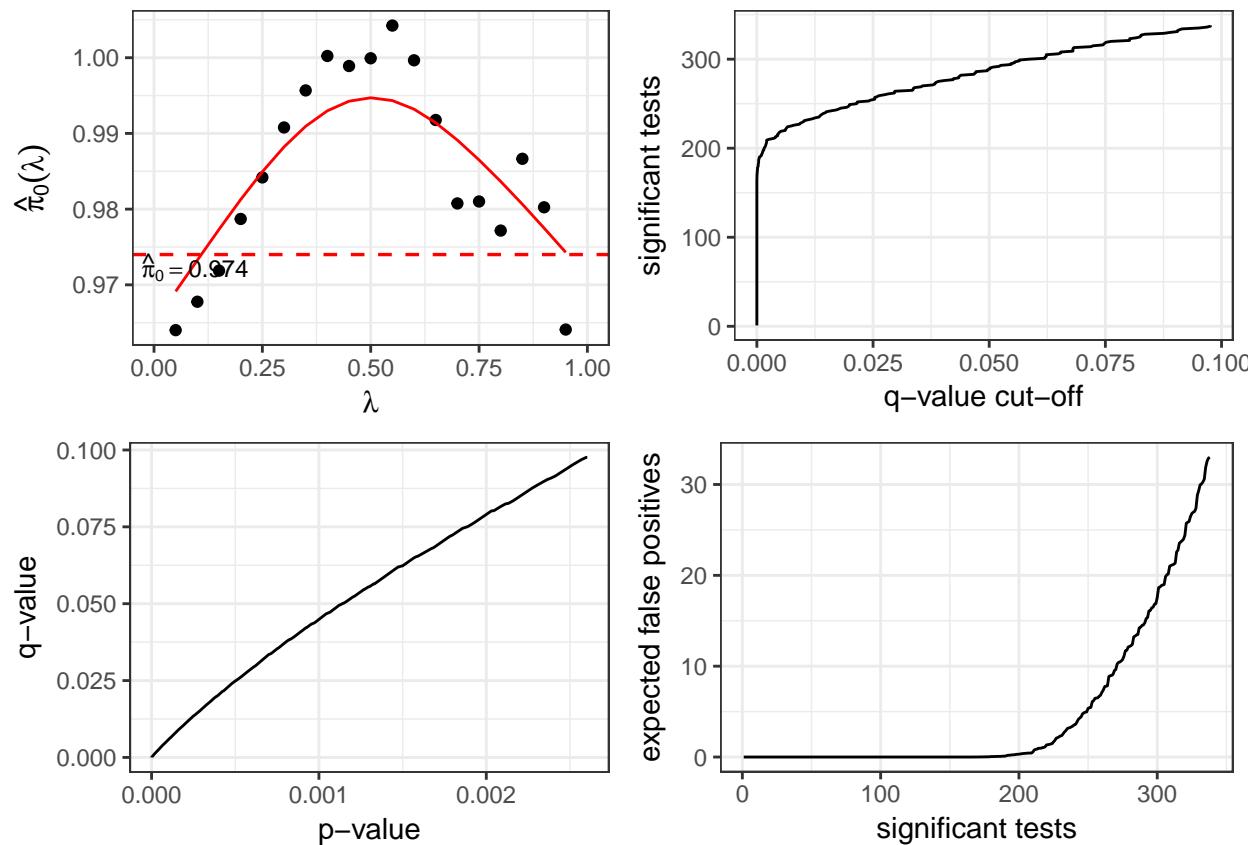
## [1] 2.687169
hist(p.values, col = "lightblue")

```

## Histogram of p.values



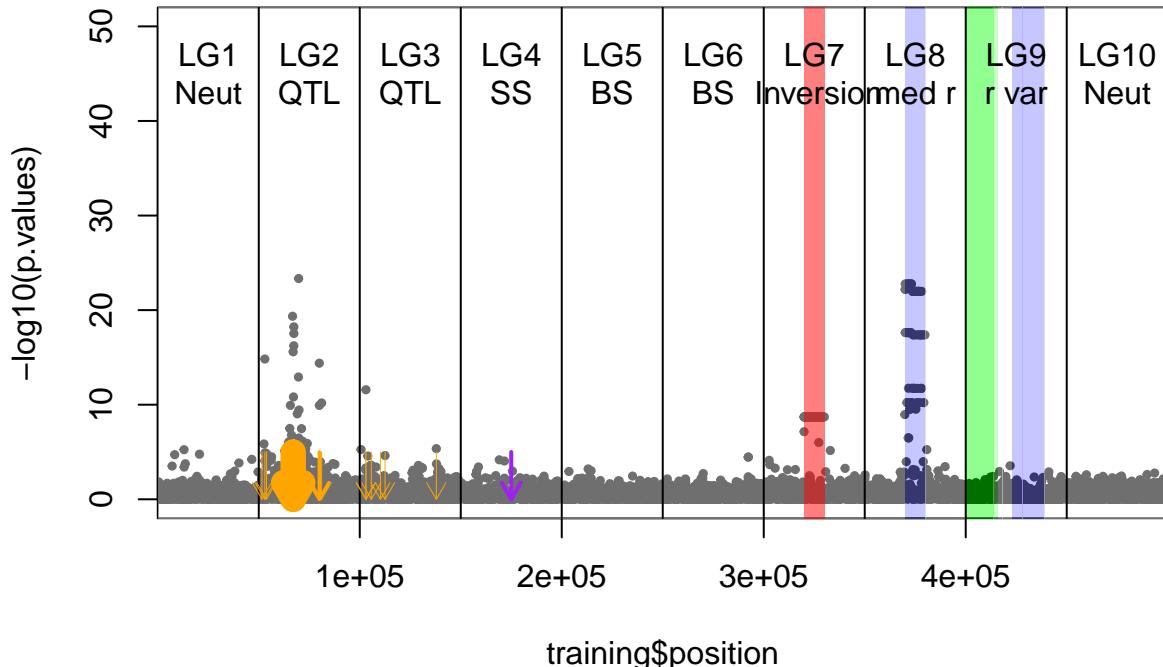
```
qval <- qvalue::qvalue(p.values)
plot(qval)
```



```
qval <- qvalue::qvalue(p.values, fdr.level = 0.005)
candidates <- which(qval$significant)
```

```
plot(training$position, -log10(p.values), cex = .5, pch = 19, col = "black", main="LFMM lasso", ylim=c(0,50))
plot_layers(y_head=45, y_arrows=c(5,0))
```

## LFMM lasso



## Bayesian (LEA) model

iHS

## Conversion scripts

```
library(rehh)
```

```
## Loading required package: rehh.data
## Loading required package: gplots
##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##      lowess
### Convert vcf@gt to haplotype format .thap
# one file for each chromosome
#SNP1 SNP2 SNP3
#IND1 hap1 A T A
#IND1 hap2 A C T
#IND2 hap1 G C T
```

```

#IND2    hap2      A      T      A
nlociperchrom <- table(vcf@fix[, "CHROM"])

substring("0|1", 1, last=1)

## [1] "0"
substring("0|1", 3, last=3)

## [1] "1"
head(vcf_filt)

## [1] "***** Object of class 'vcfR' *****"
## [1] "***** Meta section *****"
## [1] "##fileformat=VCFv4.2"
## [1] "##fileDate=20171003"
## [1] "##source=SLiM"
## [1] "##INFO=<ID=MID,Number=1>Type=Integer,Description=\"Mutation ID in SLiM\">"
## [1] "##INFO=<ID=S,Number=1>Type=Float,Description=\"Selection Coefficient\">"
## [1] "##INFO=<ID=DOM,Number=1>Type=Float,Description=\"Dominance\">"
## [1] "First 6 rows."
## [1]
## [1] "***** Fixed section *****"
##      CHROM POS    ID REF ALT QUAL FILTER
## [1,] "1"    "7"    NA "A"  "T"  "1000" "PASS"
## [2,] "1"    "19"   NA "A"  "T"  "1000" "PASS"
## [3,] "1"    "20"   NA "A"  "T"  "1000" "PASS"
## [4,] "1"    "70"   NA "A"  "T"  "1000" "PASS"
## [5,] "1"    "97"   NA "A"  "T"  "1000" "PASS"
## [6,] "1"    "100"  NA "A"  "T"  "1000" "PASS"
## [1]
## [1] "***** Genotype section *****"
##      FORMAT i0    i1    i2    i3    i4
## [1,] "GT"    "1|1"  "1|1"  "1|1"  "1|1"  "1|1"
## [2,] "GT"    "0|0"  "0|0"  "0|0"  "0|0"  "0|0"
## [3,] "GT"    "0|1"  "0|0"  "0|1"  "0|0"  "0|0"
## [4,] "GT"    "0|1"  "0|0"  "1|1"  "1|0"  "1|0"
## [5,] "GT"    "0|0"  "0|0"  "0|0"  "0|0"  "0|0"
## [6,] "GT"    "1|0"  "1|1"  "0|0"  "1|1"  "1|1"
## [1] "First 6 columns only."
## [1]
## [1] "Unique GT formats:"
## [1] "GT"
## [1]

rem2 <- which(duplicated(vcf_filt@fix[, "POS"]))
sort(vcf_filt@fix[rem2, "POS"])

## [1] "100878" "104011" "111996" "123483" "12587"  "127743" "128827"
## [8] "129938" "130331" "131420" "131651" "134116" "13559"  "138688"
## [15] "144856" "14747"  "148472" "152404" "15534"  "156750" "163099"
## [22] "163461" "167236" "167289" "172799" "178820" "187362" "194105"
## [29] "19414"  "199409" "201469" "202057" "203799" "204330" "204588"
## [36] "208737" "209274" "210510" "214954" "215863" "216112" "218783"
## [43] "223313" "224110" "226535" "228151" "23648"  "241372" "242446"

```

```

## [50] "243145" "247310" "2489"    "24899"   "250703" "250990" "262897"
## [57] "263557" "264659" "269330" "269691" "27467"  "27616"  "280934"
## [64] "286410" "290483" "297128" "300889" "302106" "307137" "309410"
## [71] "312748" "312748" "314819" "314925" "315235" "318617" "319178"
## [78] "321220" "321499" "325493" "325644" "328433" "329409" "331733"
## [85] "335201" "335999" "336442" "342570" "346320" "346320" "346644"
## [92] "349640" "352217" "353892" "35594"  "356458" "363991" "368855"
## [99] "373487" "375440" "378360" "384195" "384475" "386505" "386722"
## [106] "387167" "391793" "397539" "398342" "39953"  "401462" "403569"
## [113] "408133" "409128" "41165"  "412681" "414380" "415457" "415719"
## [120] "423089" "423133" "426774" "427028" "437326" "437431" "438833"
## [127] "439863" "445645" "454481" "46069"  "461170" "462916" "464914"
## [134] "465274" "466353" "47216"  "472243" "472639" "475424" "477029"
## [141] "482652" "483984" "484992" "489377" "489782" "490838" "49160"
## [148] "493373" "495022" "495910" "497929" "57233"  "59284"  "61671"
## [155] "64217"  "70053"  "70722"  "71016"  "71245"  "7293"   "76650"
## [162] "79040"  "80838"  "81168"  "86386"  "88992"  "93587"

vcf_filt2 = vcf_filt[-rem2,]

#### Get into right format
for (i in 1:length(nlociperchrom)){
  keep <- which((vcf_filt2@fix[, "CHROM"]==i))
  head(vcf_filt2@gt[keep,1:10], 10)
  hap1 <- apply(vcf_filt2@gt[keep,-1], 1, FUN=function(x) substring(x,1,1))
  dim(hap1)
  #head(hap1[,1:10])
  hap2 <- apply(vcf_filt2@gt[keep,-1], 1, FUN=function(x) substring(x,3,3))
  #head(hap2[,1:10])

  hapt_out <- matrix(NA, nrow=2*1000, ncol=length(keep))
  odd <- seq(1,(2*1000), by=2)
  even <- odd +1
  hapt_out[odd,] <- as.numeric(hap1)
  rownames(hapt_out) <- rep("", nrow(hapt_out))
  rownames(hapt_out)[odd] <- rownames(hap1)
  rownames(hapt_out)[even] <- rownames(hap2)
  hapt_out[even,] <- as.numeric(hap2)
  #head(hapt_out[,1:10])
  write.table(cbind(rownames(hapt_out), hapt_out+1), paste0("temp/",seed,"chrom",i,".thap"), row.names=FALSE)
}
#a<- read.table("chrom1.thap")

#### Also need to convert map.inp
#Each line contains five columns corresponding to:
#1. the SNP name
#2. the SNP chromosome (or scaffold) of origin
#3. the SNP position on the chromosome (or scaffold). Note that it is up to the user to choose either
#physical or genetic map positions (if available).
#4. the SNP ancestral allele (as coded in the haplotype input file)
#5. the SNP derived alleles (as coded in the haplotype input file)
map <- data.frame(name=1:nrow(vcf_filt2), chrom=as.numeric(vcf_filt2@fix[, "CHROM"]),
                    pos=as.numeric(vcf_filt2@fix[, "POS"]), anc=1, derived=2)
# setting anc=0 and derived = 1 thinks missing data

```

```

head(map)

##   name chrom pos anc derived
## 1    1     1   7   1     2
## 2    2     1  19   1     2
## 3    3     1  20   1     2
## 4    4     1  70   1     2
## 5    5     1  97   1     2
## 6    6     1 100   1     2

which(duplicated(map$pos))

## integer(0)

write.table(map, paste0("temp/",seed,"map.inp"), row.names=F, col.names=F)

```

## iHS Analysis

```

cnt=0
for(i in 1:length(nlociperchrom)){
  cnt=cnt+1
  tmp.hapfile=paste0("temp/",seed,"chrom",i,".thap")

  tmp.hap=data2haplohh(hap_file=tmp.hapfile, map_file=paste0("temp/",seed,"map.inp"), chr.name=i,haplotype=1)

  tmp.scan=scan_hh(tmp.hap,threads=4)

  if(cnt==1){wgscan=tmp.scan}else{wgscan=rbind(wgscan,tmp.scan)}
}

## Map file seems OK: 1316 SNPs declared for chromosome 1
## Standard rehh input file assumed
## Discard Haplotype with less than 100 % of genotyped SNPs
## No haplotype discarded
## Discard SNPs genotyped on less than 100 % of haplotypes
## No SNP discarded
## Data consists of 2000 haplotypes and 1316 SNPs
## Map file seems OK: 1294 SNPs declared for chromosome 2
## Standard rehh input file assumed
## Discard Haplotype with less than 100 % of genotyped SNPs
## No haplotype discarded
## Discard SNPs genotyped on less than 100 % of haplotypes
## No SNP discarded
## Data consists of 2000 haplotypes and 1294 SNPs
## Map file seems OK: 1233 SNPs declared for chromosome 3
## Standard rehh input file assumed
## Discard Haplotype with less than 100 % of genotyped SNPs
## No haplotype discarded
## Discard SNPs genotyped on less than 100 % of haplotypes
## No SNP discarded
## Data consists of 2000 haplotypes and 1233 SNPs
## Map file seems OK: 1260 SNPs declared for chromosome 4
## Standard rehh input file assumed
## Discard Haplotype with less than 100 % of genotyped SNPs
## No haplotype discarded

```

```

## Discard SNPs genotyped on less than 100 % of haplotypes
## No SNP discarded
## Data consists of 2000 haplotypes and 1260 SNPs
## Map file seems OK: 1270 SNPs declared for chromosome 5
## Standard rehh input file assumed
## Discard Haplotype with less than 100 % of genotyped SNPs
## No haplotype discarded
## Discard SNPs genotyped on less than 100 % of haplotypes
## No SNP discarded
## Data consists of 2000 haplotypes and 1270 SNPs
## Map file seems OK: 1278 SNPs declared for chromosome 6
## Standard rehh input file assumed
## Discard Haplotype with less than 100 % of genotyped SNPs
## No haplotype discarded
## Discard SNPs genotyped on less than 100 % of haplotypes
## No SNP discarded
## Data consists of 2000 haplotypes and 1278 SNPs
## Map file seems OK: 1351 SNPs declared for chromosome 7
## Standard rehh input file assumed
## Discard Haplotype with less than 100 % of genotyped SNPs
## No haplotype discarded
## Discard SNPs genotyped on less than 100 % of haplotypes
## No SNP discarded
## Data consists of 2000 haplotypes and 1351 SNPs
## Map file seems OK: 1300 SNPs declared for chromosome 8
## Standard rehh input file assumed
## Discard Haplotype with less than 100 % of genotyped SNPs
## No haplotype discarded
## Discard SNPs genotyped on less than 100 % of haplotypes
## No SNP discarded
## Data consists of 2000 haplotypes and 1300 SNPs
## Map file seems OK: 1209 SNPs declared for chromosome 9
## Standard rehh input file assumed
## Discard Haplotype with less than 100 % of genotyped SNPs
## No haplotype discarded
## Discard SNPs genotyped on less than 100 % of haplotypes
## No SNP discarded
## Data consists of 2000 haplotypes and 1209 SNPs
## Map file seems OK: 1329 SNPs declared for chromosome 10
## Standard rehh input file assumed
## Discard Haplotype with less than 100 % of genotyped SNPs
## No haplotype discarded
## Discard SNPs genotyped on less than 100 % of haplotypes
## No SNP discarded
## Data consists of 2000 haplotypes and 1329 SNPs

dim(wgscan)

## [1] 12840      7

ihs=ihh2ihs(wgscan,minmaf=0.05,freqbin=0.05)
head(ihs$iHS,25)

##      CHR POSITION iHS -log10(p-value)
## 3      1          20    NA            NA

```

```

## 4    1     70  NA      NA
## 6    1    100  NA      NA
## 8    1    136  NA      NA
## 10   1    260  NA      NA
## 11   1    285  NA      NA
## 15   1    502  NA      NA
## 17   1    643  NA      NA
## 24   1    842  NA      NA
## 25   1    861  NA      NA
## 27   1    905  NA      NA
## 28   1    945  NA      NA
## 30   1    964  NA      NA
## 31   1    982  NA      NA
## 33   1   1036  NA      NA
## 40   1   1385  NA      NA
## 42   1   1500  NA      NA
## 43   1   1516  NA      NA
## 50   1   1937  NA      NA
## 51   1   1947  NA      NA
## 55   1   2038  NA      NA
## 56   1   2088  NA      NA
## 60   1   2207  NA      NA
## 64   1   2278  NA      NA
## 71   1   2415  NA      NA

tail(ihs$iHS)

##      CHR POSITION iHS -log10(p-value)
## 12803 10    498237  NA      NA
## 12804 10    498347  NA      NA
## 12809 10    498593  NA      NA
## 12815 10    498934  NA      NA
## 12825 10    499395  NA      NA
## 12831 10    499675  NA      NA

ihs$frequency.class

##           Number of SNPs mean of the log(iHHA/iHHD) ratio
## 0.05 - 0.1          81       1.10640109
## 0.1 - 0.15         73       0.82773810
## 0.15 - 0.2          73       0.49641375
## 0.2 - 0.25         87       0.34196810
## 0.25 - 0.3          105      0.32007144
## 0.3 - 0.35         104      0.16180145
## 0.35 - 0.4          152      -0.08759543
## 0.4 - 0.45         120      -0.01889971
## 0.45 - 0.5          134      -0.07102058
## 0.5 - 0.55         155      -0.09114274
## 0.55 - 0.6          201      -0.17661796
## 0.6 - 0.65         272      -0.25309133
## 0.65 - 0.7          222      -0.37881039
## 0.7 - 0.75         274      -0.58096588
## 0.75 - 0.8          348      -0.69135457
## 0.8 - 0.85         422      -0.79691182
## 0.85 - 0.9          582      -1.01064003

```

```

## 0.9 - 0.95          1079          -1.32595023
##           sd of the log(iHHA/iHHD) ratio
## 0.05 - 0.1           0.5250669
## 0.1 - 0.15          0.4895879
## 0.15 - 0.2          0.4143042
## 0.2 - 0.25          0.4236345
## 0.25 - 0.3          0.4136948
## 0.3 - 0.35          0.3455953
## 0.35 - 0.4          0.3876334
## 0.4 - 0.45          0.3935355
## 0.45 - 0.5          0.3526019
## 0.5 - 0.55          0.3859168
## 0.55 - 0.6          0.3616476
## 0.6 - 0.65          0.4468767
## 0.65 - 0.7          0.3850021
## 0.7 - 0.75          0.4842224
## 0.75 - 0.8          0.4284628
## 0.8 - 0.85          0.4311297
## 0.85 - 0.9          0.4899910
## 0.9 - 0.95          0.5832333

#distribplot(ihs$iHS$iHS)

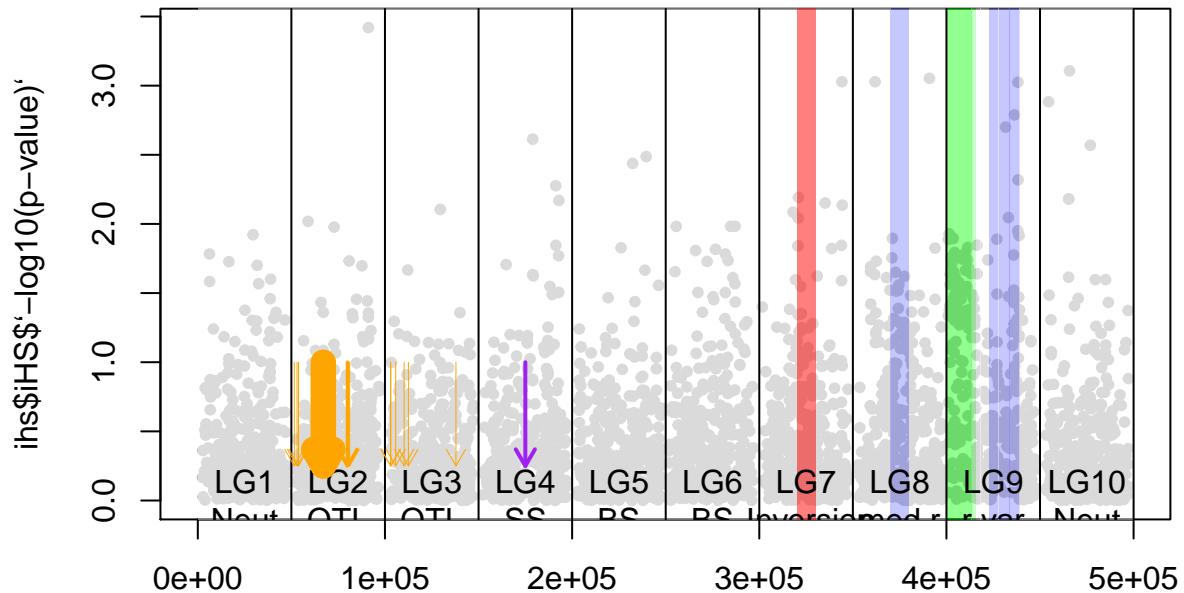
#ihsplot(ihs)
ihs$iHS[which(ihs$iHS[,4]>2),]

##      CHR POSITION      iHS -log10(p-value)
## 1545    2    58932 -2.590425    2.018373
## 2390    2    91113  3.553271    3.419677
## 3337    3   129544 -2.658302    2.104933
## 4556    4   178967  3.030838    2.612830
## 4882    4   191235 -2.788518    2.276136
## 4933    4   192910 -2.708436    2.170046
## 5907    5   232321  2.907133    2.437995
## 6095    5   239622  2.942344    2.487131
## 8127    7   318097 -2.643086    2.085370
## 8218    7   320805 -2.608879    2.041726
## 8219    7   320901  2.725242    2.192097
## 8621    7   335188  2.694161    2.151403
## 8839    7   344138 -3.308831    3.028323
## 8849    7   344366 -2.682280    2.135951
## 9287    8   361896  3.308501    3.027812
## 10023   8   390942  3.324688    3.052971
## 11064   9   431569 -3.090875    2.699911
## 11101   9   433103 -2.612160    2.045892
## 11167   9   436359 -3.150746    2.788202
## 11215   9   438205 -3.306212    3.024262
## 11219   9   438304  2.820079    2.318652
## 11635  10   454563 -3.214270    2.883469
## 11926  10   465324  2.716292    2.180340
## 11927  10   465328  2.716292    2.180340
## 11938  10   465797 -3.359070    3.106764
## 12221  10   476890  3.000196    2.568948

```

```
plot(ihs$iHS$POSITION,ihs$iHS$`-log10(p-value)`, col="grey", pch=20, main="REHH iHS")
plot_layers()
```

**REHH iHS**



```
plot(ihs$iHS$POSITION,ihs$iHS, col="grey", pch=20, main="REHH iHS")
plot_layers()
```

**REHH iHS**

