Estimating permutation p-values using MatrixEQTL

In our pipeline we first reformat the data per gene and then for each preprocessed gene run step4_MatrixEQTL script which runs multiple bootstraps.

First, load the data in TReCASE format. Arguments here are in the same style as original pipeline script. They give information about the chromosome on which gene is located, number of subsamples to be used for estimation (no more then total number of samples recorded in specification file), random seed, window size and which model to be used.

Specification file still will be used, since it is required at earlier steps linking in this pipeline. It is not necessary if you choose a different way to provide the path to the data.

The data in this example is simulated based on the GTEx dataset which allows to avoid distribution of the real data, but provides a dataset that is well represented of the GTEx dataset.

Once initial setup is done we read relevant gene information. Also, get a subset of 150 to make time to fit more feasible.

```
genepos_file_name = sprintf("%s/%s/geneInfo_prepr_%s.txt",
                    bas.dir, cnt.dir, model, echo=FALSE, message=F, results='hide')
geneInfo = read.table(genepos_file_name, as.is=T,header=T)



eChr = rep(chri, nrow(geneInfo))
ePos = as.numeric((geneInfo[,3] + geneInfo[,4])/2)


suffi = sprintf("%s_%s", chri, geni)
#genotypes_%s.dat
genfi = read.table(sprintf("%s/genotypes_%s.dat", int.dir, suffi), as.is=T, header=T)
infi = read.table(sprintf("%s/genotypei_%s.dat", int.dir, suffi), as.is=T, header=T)
#genfi[1:4,1:4]
table(unlist(genfi[,1]))
head(infi)

cnts = read.csv(sprintf("%s/counti_%s_%s.csv", int.dir, model, suffi), as.is=T, header=F)
#cnts[1:4,]
Xmatfil = sprintf("%s/Xmat_%s.csv", int.dir, model)
X = read.csv(Xmatfil, as.is=T, header=F)
nr = nrow(genfi)
genfi = matrix(unlist(genfi), nrow=nr)
nsam = nrow(X)
X = matrix(unlist(X), nrow=nsam)
dim(X)
#X[1:4,]

X = X[subs,]
cnts = cnts[subs,]
genfi = genfi[,subs]
```

```
#do an extra step to ensure that there is no 0-variance covariates
converge = 1e-4
vari = apply(X,2,var)

updvar = which(vari<converge)
for(i in updvar){
  if(length(vari[-updvar]>0)>0){
    correct = sqrt(median(vari[-updvar]))/sqrt(vari[i])
  }else{
    correct = 1/sqrt(vari[i])
  }
  xm = mean(X[,i])
  X[,i] = xm+(X[,i]-xm)*correct
}
vari
apply(X,2,var)

cnts = matrix(as.numeric(unlist(cnts)), nrow=nsub)
#cnts[1:4,]
```

Fit the model

```
mChr = rep(chri, nr)
#geni = 1
#message("geno: ", nrow(geno), " ", ncol(geno), " trecD: ", nrow(trecD), " ", ncol(trecD), " ", nrow(ge
#geni = 389

#geni = indi
  local.distances = as.numeric((geneInfo[,4] - geneInfo[,3]))/2+cis_window
#  kp = which(SNPInfo[,3]>=(ePos[geni]-local.distances[geni]) & SNPInfo[,3]<(ePos[geni]+local.distances
  kp = nrow(infi)
  sum(kp)

output.tagi     = sprintf("%s/%s",
                                outdir0, geneInfo[geni,1])#rownames(trecD)[geni])
    timj = sprintf("%s_time.txt", output.tagi)
    res.lon = sprintf("%s_eqtl.txt", output.tagi)

    time1 = proc.time()
    asSeq:::trecase(Y=cnts[,1,drop=F], Y1=cnts[,2,drop=F], Y2=cnts[,3,drop=F], X=X, Z=t(genfi), output.
                p.cut=1,  local.distance = local.distances[geni],
                eChr = eChr[geni],
                ePos = ePos[geni],
                mChr = as.numeric(infi[,2]), mPos = as.numeric(infi[,3]),
                maxit = 4000,
                min.AS.reads = 5, min.AS.sample = 5, min.n.het = 5)

    time2 = proc.time()

    write.table(time2[3]-time1[3], timj, row.names=F, col.names=F, quote=F)
    eqtl = read.table(res.lon, header=T, as.is=T)

    eqtl[,1] = infi[eqtl[,2],3]; colnames(eqtl)[1] = "Pos"
    eqtl[,2] = infi[eqtl[,2],1]
```

```
    eqtl$chr = chri
    write.table(eqtl, res.lon, row.names=F, col.names=T, quote=F, sep="\t")
```

Results are outputed in the following format:

```
eqtl[1:5,]
```

```
##     Pos         MarkerRowID          NBod       BBod    TReC_b TReC_Chisq
## 1 39457  chr9_39457_G_A_b38 4.940656e-323 -2.04e-268 -0.047935      0.226
## 2 39516  chr9_39516_C_T_b38 4.940656e-323 -2.04e-268  0.048005      0.226
## 3 39966  chr9_39966_A_G_b38 4.940656e-323 -2.04e-268 -0.140150      1.548
## 4 40997  chr9_40997_A_T_b38 4.940656e-323 -2.04e-268  0.175610      1.077
## 5 41644 chr9_41644_C_CT_b38 4.940656e-323 -2.04e-268  0.092514      0.322
##   TReC_df TReC_Pvalue ASE_b ASE_Chisq ASE_df ASE_Pvalue Joint_b Joint_Chisq
## 1       1       0.634    NA        NA     NA         NA      NA          NA
## 2       1       0.634    NA        NA     NA         NA      NA          NA
## 3       1       0.213    NA        NA     NA         NA      NA          NA
## 4       1       0.299    NA        NA     NA         NA      NA          NA
## 5       1       0.571    NA        NA     NA         NA      NA          NA
##   Joint_df Joint_Pvalue n_TReC n_ASE n_ASE_Het trans_Chisq trans_Pvalue
## 1       NA           NA    150     6         2          NA           NA
## 2       NA           NA    150     6         2          NA           NA
## 3       NA           NA    150     6         2          NA           NA
## 4       NA           NA    150     6         0          NA           NA
## 5       NA           NA    150     6         0          NA           NA
##   final_Pvalue chr
## 1        0.634   9
## 2        0.634   9
## 3        0.213   9
## 4        0.299   9
## 5        0.571   9
```

Time required to fit the model

```
time2 - time1
```

```
##    user  system elapsed
##  52.020   0.012  52.073
```