Evaluating the Effects of Aspen Genetics on Insect Communities - Specimen of R code

Clay Morrow and Ting-Fung Ma

April 7, 2019

This document includes specimen of R code for fitting GLMM on the binary phenotype. The core computation is performed on CHTC which utilize a large number of threads for distributed computing.

R packages

```
library(data.table) # fast load data
library(lme4) # glmm
```

Data loading

```
phen.covar <- fread("phenos-and-covars.txt")
snp <- fread("gwas-flat.ped")[,-(2:6)] # drop some columns
bim <- fread("wisasp_gwa-data.bim")

# Rename column by SNP name
data.table::setnames(snp,old=names(snp), new = c("FID",
        unname(unlist(bim[,2]))))

genet.name <- unlist(snp[,1])

# Rename Genet name (FID), remove .bam
for (i in 1:458) {
tmp <- substr(genet.name[i],nchar(genet.name[i])-3,nchar(genet.name[i]))
if (tmp == ".bam") {
  genet.name[i] <- substr(genet.name[i], 1, nchar(genet.name[i])-4)
}
}

snp[,1] <- genet.name

# Remove extra row in phenos-and-covar
phen.covar <- phen.covar[phen.covar$FID %in% genet.name,]
```

We removed the phenotype records without SNP data and remove the ".bam" in the genet name.

```
dim(phen.covar)
```

```
[1] 5656   47
```

```
dim(snp)
```

```
[1]    458 114421
```

```
summary(as.numeric(table(phen.covar$FID)))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
   4.00    8.00   16.00   12.94   16.00   24.00
```

The entire dataset for analysis has 5656 trees within the 458 genets. The number of trees in a genets varies from 4 to 24. The number of SNPs for analysis is 114420.

GWAS for presence of insect

In this section, we adopt a GLMM model to perform a GWAS study. For simplicity, the phenotype is presence of a particular kind of insect. In particular, Harmandia is considered. The random effect controls the effect of common genetic material within the genet.

Also, we assume an additive effect on the SNPs.

Clay: Please check if I code correctly, I count how many of the allele equals the reference allele from the bim file. Moreover, it seems that the parallel loop does not speed up too much.

The number of allele is calculated by:

```
# Calculate no. allele
SNP.add <- matrix(nr=458, nc=114420)

for(i in 2:114421) {
  tmp <- unname(unlist(snp[,..i]))
  # Reference allele from .bim
  ref.allele <- unlist(bim[i-1,5])
  SNP.add[,i-1] <- (substr(tmp,start=1,stop=1) == ref.allele) +
    (substr(tmp,start=3,stop=3) == ref.allele)
}

SNP.add <- as.data.frame(SNP.add)
colnames(SNP.add) <- names(snp)[-1]
rownames(SNP.add) <- snp$FID
```

We further transform the covariate, such as age of tree is calculated by the difference between ("PlantingDate" - "Date")/365.25. Since majority of "Npct.all"" and "Cpct.all" (4252 out of 5656) are missing, we exclude these two covariates for much larger sample size.

```
# Harmandia, col 12
y <- 1*(phen.covar[,12]>0)

tmp.data <- phen.covar[,c(1,30,31,35:43)]

# Age calculation
Age <- as.numeric(as.Date(unlist(phen.covar[,32]),format="%Y-%m-%d") -
                    as.Date(unname(unlist(phen.covar[,34])),format="%m/%d/%Y"))/365.25

tmp.data <- cbind(y, tmp.data, Age)
colnames(tmp.data)[1] <- "y"

tmp.data <- as.data.frame(tmp.data)

standardize <- function(x) { (x - mean(x,na.rm=TRUE))/sd(x, na.rm=TRUE)}


for (i in 5:14) {
  tmp.data[,i] <- standardize(tmp.data[,i])
}
```

```
# save(tmp.data, SNP.add, file="harmandia.RData")
```

We fit mixed model and store the p-values for some SNPs.

```
snp.no <- 1

p.value <- rep(NA,5)

# Run first 5 SNP for a trial
for(snp.no in 1:5) {
lookup <- data.frame(FID=rownames(SNP.add),
                     snp=SNP.add[,snp.no])

tmp.data2 <- merge(lookup, tmp.data, by = 'FID')

m <- glmer(y ~ snp  + Volume + ALA.all +
        SLA.all + BBreakDegDay.all + EFNMean.all +
        CTsum + PGsum  + Age
        # Npct.all + Cpct.all # exclude these two variable due to missing
      + (1|survey.event/FID), family = binomial,
      data=tmp.data2)

p.value[snp.no] <- summary(m)$coef[2,4] # p-value of SNP
}
```

```
Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
control$checkConv, : Model failed to converge with max|grad| = 0.00255621
(tol = 0.001, component 1)
```

```
Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
control$checkConv, : Model failed to converge with max|grad| = 0.00113203
(tol = 0.001, component 1)
```

```
p.value
```

```
[1] 0.1914969 0.1854479 0.4052541 0.5111625 0.5890738
```

There are some warnings about convergence of model fitting, probably due to the difficulty of optimizer.