Statistics compared

E. Oziolor

April 25, 2018

# Comparisons

* I am comparing fst data calculated with Weir&Cockerham 1984 (with correction for sample size) to one calculated from pi
* Comparing pi calculated by comparing individual snps to one calculated by SFS in ANGSD

## Fst statistic

* Weir & Cockerham Fst estimation includes correction for sample size
  + that has given some odd substructuring paterns for BB and SJ, which Noah has noticed are not present when Fst is calculated from pi (1-mean(pi1,pi2)/dxy)
* Let’s take Noah’s data where pi and dxy are calculated on a per SNP basis and take global and local Fst plots from that.

###### scp -P 2022 farm:/home/nreid/noah\_stats.RData ~/analysis/data/comparison/noah\_stats.RData

load("~/analysis/data/comparison/noah\_stats.RData")  
write.table(lift,"~/analysis/data/fst/noah.1kb.bed",row.names = FALSE,col.names = FALSE,quote = FALSE,sep='\t')

* starting with the PBS statistic comparisons
  + grabbing my data and plotting outlier windows

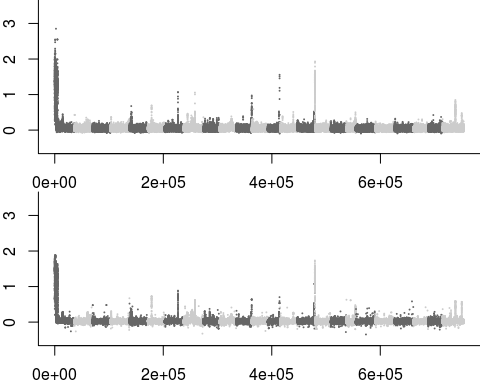
library(XML)  
library(magrittr)  
library(stringr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

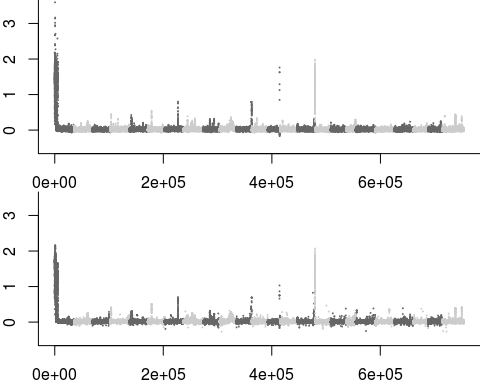
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

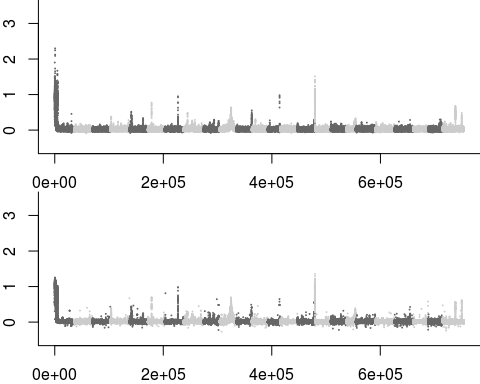
library(gtools)  
library(naturalsort)  
library(stringr)  
library(dplyr)  
library(gtools)  
  
#Reading in table and getting quantiles----  
pbs<-read.table("~/analysis/data/fst/allpbs5kb",header=FALSE,stringsAsFactors = FALSE)  
pbsname<-c("Scaf","start","end","BBpbs","VBpbs","PBpbs","SJpbs","BNPpbs","keep")  
colnames(pbs)<-pbsname  
  
pbsc<-pbs %>%   
 filter(str\_detect(Scaf,"chr"))  
subw<-pbsc[,9]>0  
  
#Reorder Noah's data by chromosome  
pbst<-cbind(lift[,1:3],pbstat[,4:9])  
ord<-mixedorder(pbst$V1)  
pbsn<-pbst[ord,]  
#Plotting a regression against the PBS statistics calculated by Noah  
  
pbsnc<-pbsn %>%   
 filter(str\_detect(V1,"chr"))  
  
  
pbsc$Scaf<-factor(pbsc$Scaf,levels=c("chr1","chr2","chr3","chr4","chr5","chr6","chr7","chr8","chr9","chr10",  
 "chr11","chr12","chr13","chr14","chr15","chr16","chr17","chr18","chr19",  
 "chr20","chr21","chr22","chr23","chr24"))  
  
pbsnc$V1<-factor(pbsnc$V1,levels=c("chr1","chr2","chr3","chr4","chr5","chr6","chr7","chr8","chr9","chr10",  
 "chr11","chr12","chr13","chr14","chr15","chr16","chr17","chr18","chr19",  
 "chr20","chr21","chr22","chr23","chr24"))  
  
palette(c("grey40","grey80"))  
par(mfrow=c(2,1),mar=c(2,2,0,0))  
plot(pbsc[subw,"BBpbs"],pch=20,cex=.2,col=as.factor(pbsc[subw,1]),bty='l',ylim=c(-0.5,3.5))  
plot(pbsnc[subw,"BB"],pch=20,cex=.2,col=as.factor(pbsnc[subw,1]),bty='l',ylim=c(-0.5,3.5))



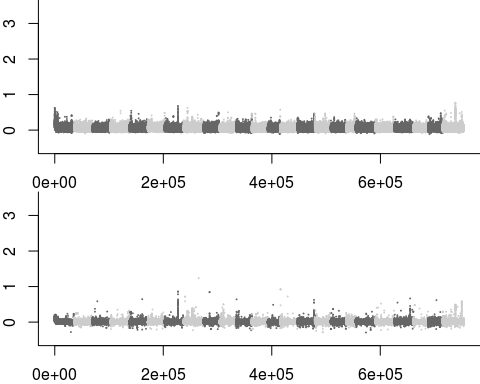
par(mfrow=c(2,1),mar=c(2,2,0,0))  
plot(pbsc[subw,"VBpbs"],pch=20,cex=.2,col=as.factor(pbsc[subw,1]),bty='l',ylim=c(-0.5,3.5))  
plot(pbsnc[subw,"VB"],pch=20,cex=.2,col=as.factor(pbsnc[subw,1]),bty='l',ylim=c(-0.5,3.5))



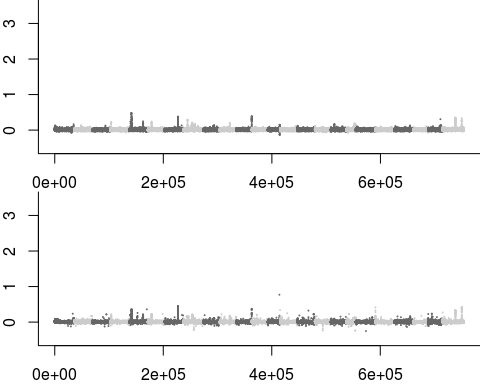
par(mfrow=c(2,1),mar=c(2,2,0,0))  
plot(pbsc[subw,"PBpbs"],pch=20,cex=.2,col=as.factor(pbsc[subw,1]),bty='l',ylim=c(-0.5,3.5))  
plot(pbsnc[subw,"PB"],pch=20,cex=.2,col=as.factor(pbsnc[subw,1]),bty='l',ylim=c(-0.5,3.5))



par(mfrow=c(2,1),mar=c(2,2,0,0))  
plot(pbsc[subw,"SJpbs"],pch=20,cex=.2,col=as.factor(pbsc[subw,1]),bty='l',ylim=c(-0.5,3.5))  
plot(pbsnc[subw,"SJSP"],pch=20,cex=.2,col=as.factor(pbsnc[subw,1]),bty='l',ylim=c(-0.5,3.5))



par(mfrow=c(2,1),mar=c(2,2,0,0))  
plot(pbsc[subw,"BNPpbs"],pch=20,cex=.2,col=as.factor(pbsc[subw,1]),bty='l',ylim=c(-0.5,3.5))  
plot(pbsnc[subw,"BNP"],pch=20,cex=.2,col=as.factor(pbsnc[subw,1]),bty='l',ylim=c(-0.5,3.5))



#Only works if you call the 1kb windows  
# par(mfrow=c(3,2),mar=c(2,2,0,0))  
# plot(pbsc[subw,"BBpbs"],pbsnc[subw,"BB"],pch=20,cex=.2)  
# plot(pbsc[subw,"VBpbs"],pbsnc[subw,"VB"],pch=20,cex=.2)  
# plot(pbsc[subw,"PBpbs"],pbsnc[subw,"PB"],pch=20,cex=.2)  
# plot(pbsc[subw,"SJpbs"],pbsnc[subw,"SJSP"],pch=20,cex=.2)  
# plot(pbsc[subw,"BNPpbs"],pbsnc[subw,"BNP"],pch=20,cex=.2)

### Calculating Genome-wide Fst and comparing

* Will use 2 ways of calculating genome wide Fst
  + Averaging Fst statistics from Weir & Cockerham windowed estimates of 1kb windows over the genome
  + Averaging Fst statistics with Hudson estimator from Noah’s pi and dxy
* Starting with my 1kb estimates of genome-wide Fst (Weir&Cockerham)

library("RColorBrewer")  
library("lattice")  
library("gplots")

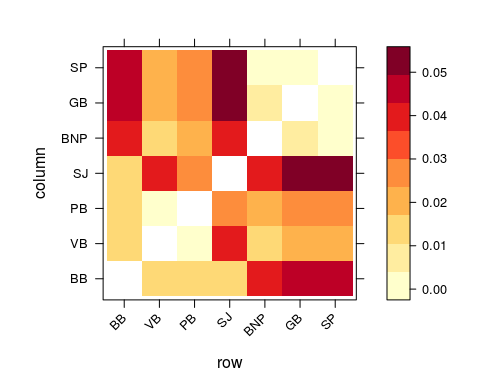
##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

#Loaidng list of fst files into a list object ----  
fs <- list.files("~/analysis/data/fst/raw/", "\*fst.1kb.bed",full.names=TRUE) # listing all the files for Fst calculated with W&C fst statistic  
  
fst <- list()  
  
for (i in 1:21){  
 fst[[i]] <- read.table(fs[i],stringsAsFactors=FALSE)  
 fst[[i]][,4] <- as.numeric(fst[[i]][,4])  
} #reading in those files

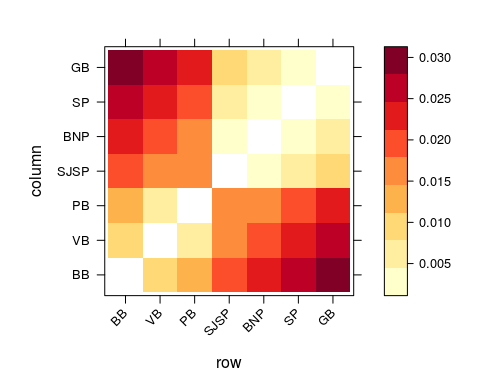
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion

nfs <- gsub(".\*\\/","",fs) #renaming the columns by removing the "." in the names  
nfs <- gsub(".fst.\*","",nfs) #renaming by removing ".fst.\*" from the name  
names(fst)<-nfs  
  
#selecting sites that ahave a minimum representation of 200 snps per region----  
nsnps <-fst[[1]][,5]  
  
for (i in 2:21){  
   
 nsnps <- nsnps + fst[[i]][,5]  
}  
  
nsnps <- nsnps/21  
  
subw <- nsnps > 20  
  
#calculating FST for all  
  
pops<-c("BB","VB","PB","SJ","BNP","GB","SP")  
fsth<-matrix(nrow = 7,ncol=7) #creating matrix to hold fst data  
colnames(fsth)<-pops  
rownames(fsth)<-pops  
  
for(i in pops){  
 for(j in pops){  
 if(i==j){next()}  
 if(which(pops %in% i) < which(pops %in% j)){  
 fsth[i,j]<-mean(fst[[paste(unique(c(i,j)),collapse=".")]][subw,4],na.rm=TRUE)  
 } else{  
 fsth[i,j]<-mean(fst[[paste(unique(c(j,i)),collapse=".")]][subw,4],na.rm=TRUE)  
 }  
 }  
} #global fst calculation for each pair  
  
#heatfst<-heatmap.2(fsth,Rowv=NA,Colv=NA,scale="none",margins=c(5,10),col=brewer.pal(9,"YlOrRd"),  
 #density.info="none", trace="none")  
levelplot(fsth,aspect="iso",col.regions=brewer.pal(9,"YlOrRd"),scale=list(x=list(rot=45)),cuts=8) #Better plot than above



* Calculating genome wide Hudson statistic

load("~/analysis/data/comparison/noah\_stats.RData")  
subw<-val[,4]>0  
neutsum<-colSums(fst[subw,4:94],na.rm=TRUE) #summing up columns of pi and dxy statistics  
snpsum<-sum(val[subw,4])  
neutbase<-neutsum/snpsum  
  
pops<-c("BB","VB","PB","SJSP","BNP","SP","GB")  
# dxy<-neutbase[c("BB.VB","BB.PB","BB.SJSP","BB.BNP","BB.SP","BB.GB",  
# "PB.VB","SJSP.VB","BNP.VB","SP.VB","GB.VB",  
# "PB.SJSP","BNP.PB","PB.SP","GB.PB",  
# "BNP.SJSP","SJSP.SP","GB.SJSP",  
# "BNP.SP","BNP.GB",  
# "GB.SP")]  
  
fsth<-matrix(nrow = 7,ncol=7) #creating matrix to hold fst data  
rownames(fsth)<- pops  
colnames(fsth)<- pops  
  
for(i in pops){  
 for(j in pops){  
 if(i==j){next()}  
 fsth[i,j]<-1-((neutbase[i]+neutbase[j])/2)/neutbase[paste(sort(unique(c(i,j))),collapse=".")]  
 }  
}  
  
  
levelplot(fsth,aspect="iso",col.regions=brewer.pal(9,"YlOrRd"),scale=list(x=list(rot=45)),cuts=8) #Better plot than above

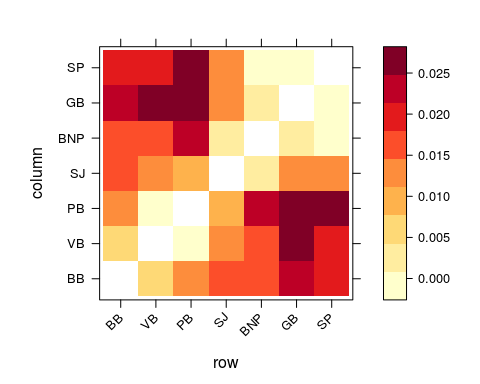


* Calculating W&C with only 24 randomly selected individuals per populations to see if it’s a problem of correcting for sample number

library("RColorBrewer")  
library("lattice")  
library("gplots")  
#Loaidng list of fst files into a list object ----  
fs <- list.files("~/analysis/data/fst/raw/subsample/", "\*fst.1kb.bed",full.names=TRUE) # listing all the files for Fst calculated with W&C fst statistic  
  
fst <- list()  
  
for (i in 1:21){  
 fst[[i]] <- read.table(fs[i],stringsAsFactors=FALSE)  
 fst[[i]][,4] <- as.numeric(fst[[i]][,4])  
} #reading in those files

## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion

nfs <- gsub(".\*\\/","",fs) #renaming the columns by removing the "." in the names  
nfs <- gsub(".fst.\*","",nfs) #renaming by removing ".fst.\*" from the name  
names(fst)<-nfs  
  
#selecting sites that ahave a minimum representation of 200 snps per region----  
nsnps <-fst[[1]][,5]  
  
for (i in 2:21){  
   
 nsnps <- nsnps + fst[[i]][,5]  
}  
  
nsnps <- nsnps/21  
  
subw <- nsnps > 20  
  
#calculating FST for all  
  
pops<-c("BB","VB","PB","SJ","BNP","GB","SP")  
fsth<-matrix(nrow = 7,ncol=7) #creating matrix to hold fst data  
colnames(fsth)<-pops  
rownames(fsth)<-pops  
  
for(i in pops){  
 for(j in pops){  
 if(i==j){next()}  
 if(which(pops %in% i) < which(pops %in% j)){  
 fsth[i,j]<-mean(fst[[paste(unique(c(i,j)),collapse=".")]][subw,4],na.rm=TRUE)  
 } else{  
 fsth[i,j]<-mean(fst[[paste(unique(c(j,i)),collapse=".")]][subw,4],na.rm=TRUE)  
 }  
 }  
} #global fst calculation for each pair  
  
#heatfst<-heatmap.2(fsth,Rowv=NA,Colv=NA,scale="none",margins=c(5,10),col=brewer.pal(9,"YlOrRd"),  
 #density.info="none", trace="none")  
levelplot(fsth,aspect="iso",col.regions=brewer.pal(9,"YlOrRd"),scale=list(x=list(rot=45)),cuts=8) #Better plot than above

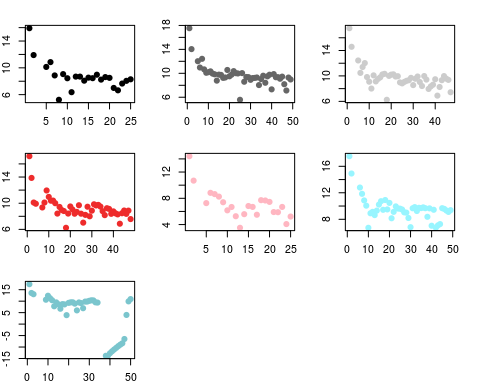


* When sampling the same number of individuals, now we have a more IBD like global Fst pattern, pointing to sampling number correction being the cuplrit here
* This suggests that W&C has a genome wide effect, but it still points to similar outliers
* Not a huge problem overall, but to do this properly I will repeat outlier analyses with Hudson estimator instead. It will take a bit, but oh well.

## Neutrality statistics

* Pi and Theta were calculated with SFS as prior for ANGSD
  + priors look a bit messed up (clumpy), suggesting they are not the best

sf<-list.files("~/analysis/data/angsd/raw/","\*.sfs",full.names=TRUE)  
cols<-c("black","grey40","grey80","firebrick2","lightpink","cadetblue1","cadetblue3")  
pop<-list("bb","vb","pb","sj","bnp","sp","gb")  
  
for(i in 1:7){  
 pop[[i]]<-scan(sf[[i]])  
}  
  
par(mfrow=c(3,3),mar=c(2,2,2,2))  
for(i in 1:7){  
 plot(log(pop[[i]]),col=cols[i],pch=20,lwd=3)  
}



* this is not great. The SFS is used in further estimations of SAF as a prior. We do see genome-wide shifts in theta and pi, but it is possible that those are due to errors in estimating SAF.
* Noah’s has calculated pi and dxy on a per site basis, which doesn’t rely on SFS. This doesn’t show the same pattern of decreasing pi.