Effector\_Maps

## Overview

This is an R notebook that documents the creation of allele maps for *S. hermonthica*, based on sequencing data from 68 sequenced individuals from 6 different locations in western Kenya. We performed whole genome sequencing of 24 individuals each from 2 locations (Kisii and Homa Bay) and 5 individuals each from an additional 4 locations (Chemelil, Muhoroni, Mumias, and Kibos).

## Identification of SNPs in effector genes

To identify polymorphisms in effector genes in these populations, we mapped cleaned reads to 106 fasta sequences provided by the Runo lab and called SNPs. This involved:

**1. Cleaning fastq reads with bbduk version 38.76:**

bbduk.sh in=SH103\_S51\_L004\_R1\_001.fastq.gz in2=SH103\_S51\_L004\_R2\_001.fastq.gz out=SH103.R1.fastq.gz out2=SH03.R2.fastq.gz k=23 mink=11 hdist=1 tpe tbo ktrim=r qtrim=rl trimq=20 minlen=50 ref=adapters.fa -t 8

**2. Mapping to effector transcriptome sequences with bwa mem v0.7.17 and filtering bam files for regions of interest with samtools v1.9:**

bwa mem -t 8 DE\_Effectors.fasta SH103.R1.fastq.gz SH103.R2.fastq.gz >aln.pe.sam  
  
samtools view -b -F 4 aln.pe.sam | samtools sort - >aln.sort.bam

**3. Adding read groups with picard v2.23.1 so freebayes can process**

java -jar $PICARD AddOrReplaceReadGroups INPUT=aln.sort..bam OUTPUT=rh.bam RGID=$LINE RGLB=library1 RGPL=illumina RGPU=$LINE RGSM=$LINE

**4. Calling SNPs.** For this using freebayes v1.3.2. Tried GATK, but due to mapping to a transcriptome & short length of reference sequences there were a low number of reads mapping in proper pairs and GATK didn’t perform well.

freebayes -f ../DE\_Effectors.fasta -g 1500 -C 6 \*bam >freevar.vcf #2452 SNPs

**5. Filter variants with vcftools v0.1.17**

vcftools --vcf freevar.vcf --maf 0.05 --max-maf 0.98 --recode --stdout --max-missing 0.70 --minQ 30 >SH.effectors.vcf #1809 polymorphisms

## Mapping allele frequencies in R

**1. Load vcf file and convert to matrix:**

library('vcfR')  
library('dplyr')  
vcf <- read.vcfR("StherPMEI.vcf", verbose = FALSE )   
x <- vcfR2genlight(vcf) #1646 binary SNPs  
matx <- as.matrix(x) #convert to matrix

**2. Load GPS coordinates for all collected samples:**

gps <- read.csv('../4\_R\_Spatial\_Data/Striga\_GPS.csv', header=T)[,1:9] #get rid of some empty columns

**3. Create a function that will make a dataframe of all the SNPs found in a gene of interest:**

make\_gene\_df <- function(gene\_id) {   
 row.idx <- grep(gene\_id, colnames(matx)) #get indices of columns corresponding to SNPs in the gene of interest  
 gene.df <- as.data.frame(matx[,row.idx[1]:row.idx[length(row.idx)]]) #extract these SNPs from matrix and convert to df  
 gene.df$SampleID <- rownames(gene.df)  
 gene.df2 <- right\_join(gps, gene.df)   
 gene.df2 <- gene.df2 %>%  
 dplyr::select(-Host,-SampleID,-Elevation\_ft,-CollectionDate,-PlantsIn10x10m,-CollectionTime)  
 return(gene.df2)  
}

**4. Create a function that will condense the SNP dataframe into a summary table of allele frequencies**. We will group all Chemelil/Chemelil2, Homa Bay/Homa Bay 2, and Mumias/Mumias 2 samples, and samples from different hosts, even though they were collected in different fields.

library(stringr)  
get\_allele\_freqs <- function(genedf) {   
 # group samples from same general location  
 genedf$Site <- as.factor(str\_replace(genedf$Site, "2","") %>% str\_trim())  
  
 # calculate average allele per site (0 to 2 for all reference and all alternate allele, respectively)  
 gene.avg <- as.data.frame(genedf %>%  
 group\_by(Site) %>%  
 dplyr::summarise\_at(vars(-group\_cols()),mean,na.rm=TRUE))  
  
 # add a col for number of samples  
 gene.avg$n <- as.data.frame(genedf %>%   
 group\_by(Site) %>%  
 dplyr::summarise(n=n()) %>%  
 dplyr::select(n))  
 return(gene.avg)  
}

**5. Get western Kenya county boundaries for map:**

# partially following tutorial here https://rpubs.com/spoonerf/countrymapggplot2   
library(maptools)  
library(raster)  
library(plyr)  
library(ggplot2)  
library(rgdal)  
  
# get polygons for western Kenyan counties  
Kenya1<-getData("GADM", country="KE", level=1)  
Kenya1\_UTM<-spTransform(Kenya1, CRS("+init=EPSG:32737"))   
  
counties <-Kenya1\_UTM[Kenya1\_UTM@data$NAME\_1 == "Busia"| Kenya1\_UTM@data$NAME\_1 == "Kisumu" | Kenya1\_UTM@data$NAME\_1=="Homa Bay"| Kenya1\_UTM@data$NAME\_1 == "Kericho"|Kenya1\_UTM@data$NAME\_1 == "Nandi"|Kenya1\_UTM@data$NAME\_1 == "Vihiga"|Kenya1\_UTM@data$NAME\_1 == "Siaya"|Kenya1\_UTM@data$NAME\_1 == "Kakamega"|Kenya1\_UTM@data$NAME\_1 == "Kisii",]  
  
counties.ll <- spTransform(counties, CRS("+proj=longlat"))

**6. Create maps for allele frequency, one gene per output file, where allele frequencies at each location are represented by pie charts:**

library(mapplots)  
geneids <- unique(str\_replace(colnames(matx), "\_[0-9]+$", ""))  
  
for (j in 1:length(geneids)) {  
 tmp <- make\_gene\_df(geneids[j])  
 goi <- get\_allele\_freqs(tmp)  
 filename <- paste0("/Users/emilybellis/Desktop/",geneids[j],".pdf")  
  
 pdf(filename)  
 par(mfcol=c(5,3), mar=c(1,1,1,1))  
 for (i in 4:(length(goi)-1)) {  
 plot(counties.ll, main=paste(names(goi)[i]))  
 draw.pie(z=cbind(goi[,i]/2,1-(goi[,i]/2)), x=goi$Lon, y=goi$Lat, radius=0.1, col=c(alpha("orange", 0.6), alpha("blue", 0.6)), labels="")  
 }  
 dev.off()  
}