Genetic Analyses R Script Workflow for *Zostera marina* (Eelgrass)

About this resource:
This resource provides a comprehensive, script-based workflow for analyzing population genomics data in *Zostera marina* (eelgrass). It includes detailed R code and command-line steps for conducting principal component analysis (PCA), pairwise FST calculations, SNP pruning, admixture modeling, and landscape genomics analyses (e.g., LFMM and OutFLANK) to detect signatures of selection and population structure. The workflow also incorporates functional annotation of candidate genes and gene ontology enrichment using TopGO, as well as spatial visualization of genetic patterns.

This resource is intended for researchers and practitioners with experience in R and population genomics who are investigating adaptive variation, restoration planning, or conservation genetics in seagrass systems. It is particularly suited for researchers working with Zostera marina or other non- model organisms who aim to assess genetic structure, detect signals of local adaptation, or inform seed sourcing strategies for restoration. The resource assumes familiarity with genomic data formats (e.g., VCF, GDS), bioinformatic tools, and statistical methods used in landscape genomics and gene- environment association studies.

Citation: Kamel, S. & Scavo, K. (2025). *Genetic Analyses R Script Workflow for Zostera marina (Eelgrass).* NERRS Science Collaborative.

About the project:
This resource was developed through a 2022-2025 Collaborative Research project titled *Evaluating and Enhancing Eelgrass Resiliency and Restoration Potential in a Changing Climate*.

In the lower Chesapeake Bay, Virginia, warmer water temperatures in recent years have resulted in large scale diebacks of eelgrass meadows (Zostera marina). In contrast, many eelgrass populations in Back Sound, North Carolina appear to be more resilient to warming water temperatures. Understanding the drivers of these regional differences in eelgrass resilience could help more effectively restore eelgrass meadows in a changing climate.

With a network of the intended users from reserves, state agencies, and Chesapeake Bay nonprofits, this project compared resiliency traits of eelgrass populations in Virginia and North Carolina by conducting reciprocal restoration trials and genomic sequencing. The project results indicate the importance of seed sources in potential future eelgrass restoration, in addition to site selection.

This webpage provides more information about the project.

Genomic Analyses for *Zostera marina*

Population genetics/genomics analyses
****

```
#Set working directory
setwd

#Load relevant R packages (after installation)
library(gdsfmt)
library(SNPRelate)
library(SeqArray)
library(ggplot2)
library(tess3r)
library(LEA)
library(tidyverse)
library(OutFLANK)
library(vcfR)

#### Make Map of Sites#####
library("raster")
library("sf")
library("dplyr")
library("ggspatial")

# load in GPS coordinates of sites
sites = read.csv("sites_coords_new.csv")
sitesVA = read.csv("sites_coords_VA_new.csv")
sitesNC = read.csv("sites_coords_NC_new.csv")

# load map of USA
usa <- st_as_sf(raster::getData("GADM", country = "USA", level =
1))

east <- usa %>%
  filter(NAME_1 %in% c("North Carolina", "Virginia", "South
Carolina", "Florida", "Georgia", "Maryland", "Delaware", "New
Jersey", "Pennsylvania", "New York", "Connecticut", "Rhode
Island", "Massachusetts", "Maine", "New Hampshire", "Vermont",
"Ohio", "West Virginia", "Kentucky", "Alabama", "Tennessee",
"Indiana"))

# keep map of VA and NC
nc_va <- usa %>%
  filter(NAME_1 %in% c("North Carolina", "Virginia"))
```

```
# keep only map of VA
va <- usa %>%
  filter(NAME_1 == "Virginia")

# keep only map of NC
nc <- usa %>%
  filter(NAME_1 == "North Carolina")

# plot map of the southeast
ggplot() +
  geom_sf(data = east, fill = "white", color = "black", size =
10) +
  theme(panel.grid = element_blank(), panel.border =
element_blank(), panel.background = element_blank()) +
  coord_sf(xlim = c(-84.5, -67.5),  ylim = c(25, 50.5)) +
  annotation_scale(location = "br", width_hint = 0.2)

# plot map and sites for both NC and VA
ggplot() +
  geom_sf(data = nc_va, fill = "white", color = "black", size =
10) +
  #geom_point(data = sites, aes(Lon, Lat), size = 1.5, color =
"red", shape = 15) +
  theme_minimal() +
  coord_sf(xlim = c(-78, -75),  ylim = c(34, 38)) +
  annotation_scale(location = "br", width_hint = 0.2)

ggplot() +
  geom_sf(data = nc_va, fill = "white", color = "black", size =
10) +
  #geom_point(data = sites, aes(Lon, Lat), size = 1.5, color =
"red", shape = 15) +
  theme_minimal() +
  #coord_sf(xlim = c(-78, -75),  ylim = c(34, 38)) +
  annotation_scale(location = "br", width_hint = 0.2)

# better map with colors
ggplot() +
  geom_sf(data = nc_va, fill = "white", color = "black", size =
10) +
  geom_point(data = sites, aes(Lon, Lat, color =
as.factor(site)), size = 2.5, shape = 16) +
  scale_color_manual(values = c("purple3", "navyblue",
"dodgerblue4", "dodgerblue3", "dodgerblue2", "skyblue",
"darkgreen","lawngreen","seagreen4","seagreen1","coral4",
"firebrick4","firebrick3", "firebrick1",
```

```
"goldenrod4","goldenrod", "darkorange4","darkorange3",
"darkorange", "lightsalmon1")) +
  theme_minimal() +
  coord_sf(xlim = c(-78, -75),  ylim = c(34, 38)) +
  annotation_scale(location = "br", width_hint = 0.2)

# plot map and sites for VA
ggplot() +
  geom_sf(data = va, fill = "white", color = "black", size = 10)
+
  geom_point(data = sitesVA, aes(Lon, Lat, color =
as.factor(site)), size = 3, shape = 16) +
  scale_color_manual(values = c("coral4",
"limegreen","firebrick3", "firebrick1",
"goldenrod4","goldenrod", "skyblue","darkorange3", "darkorange",
"lightsalmon1")) +
  #scale_color_manual(values = c("coral4",
"firebrick4","firebrick3", "firebrick1",
"goldenrod4","goldenrod", "darkorange4","darkorange3",
"darkorange", "lightsalmon1")) +
  theme_minimal() +
  coord_sf(xlim = c(-76.8, -75.8),  ylim = c(36.9, 37.6)) +
  annotation_scale(location = "br", width_hint = 0.2)

# plot map and sites for NC
ggplot() +
  geom_sf(data = nc, fill = "white", color = "black", size = 10)
+
  geom_point(data = sitesNC, aes(Lon, Lat, color =
as.factor(site)), size = 3, shape = 16) +
  scale_color_manual(values = c("purple3", "navyblue",
"dodgerblue4", "dodgerblue3", "dodgerblue2", "skyblue",
"darkgreen","lawngreen","seagreen4","seagreen1")) +
  theme_minimal() +
  coord_sf(xlim = c(-78, -76),  ylim = c(34.2, 35.2)) +
  annotation_scale(location = "br", width_hint = 0.2)


#### Import VCF file into R ####
vcf.fn <- "Zm_QD7_Dp7_194_imputed_filt.vcf"

# Reformat VCF file to SNP GDS file (using package SeqArray)
# SNPRelate uses files in GDS (genomic data structure) format
# typical output files (VCF, plnk files) can be converted to GDS
format using the package "gdsfmt"

seqVCF2GDS(vcf.fn,"Zm_QD7_Dp7_194_imputed_filt.gds")
```

```
# import population metadata
meta <-
read.delim("metadata_Zm_QD7_Dp7_194_imputed_filt.txt",header=T)

# open gds file
genofile <- seqOpen("Zm_QD7_Dp7_194_imputed_filt.gds")
samp.id <- seqGetData(genofile, "sample.id")
ordermeta <- meta[match(samp.id,meta$INDV),]

# summarize missing data
g <- snpgdsGetGeno(genofile)
nas <- apply(g,1,function(x) sum(is.na(x))/ncol(g))
summary(nas)
#Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#0       0       0       0       0       0
# should be 0 because imputed

#### ......... LINKAGE DISEQUILIBRIUM (LD) BASED SNP PRUNING
.........####
# From resource - "It is suggested to use a pruned set of SNPs
which are in approximate linkage equilibrium with each other
# to avoid the strong influence of SNP clusters in principal
component analysis and relatedness analysis."
samp.id <- seqGetData(genofile, "sample.id")
ld.snps <-
snpgdsLDpruning(genofile,ld.threshold=0.5,autosome.only=F)
good.snps <- unlist(unname(ld.snps))

##Write file with pruned positions to use in pop structure
analyses
#vcftools --vcf Zm_QD7_Dp7_194_imputed_filt.vcf --012 --out
Zm_QD7_Dp7_194_imputed_filt_genomat
pos <-
read.delim("Zm_QD7_Dp7_194_imputed_filt_genomat.012.pos",header=
F)
ld.pos <- pos[good.snps,]
write.table(ld.pos,"ld.pruned_Zm_QD7_Dp7_194_imputed_filt.pos",r
ow.names=F,col.names=F,sep="\t",quote=F) #makes file called
"ld.pruned_QD7_Dp7_0819_remdpblw3_imputed_filt_rem.pos"

#### ......... PRINCIPAL COMPONENT ANALYSIS (PCA) ......... ####
# Steps include calculating the genetic covariance matrix from
genotypes, computing the correlation coefficients between sample
loadings and genotypes for each SNP, calculating SNP
eigenvectors (loadings), and estimating the sample loadings of a
new dataset from specified SNP eigenvectors.
```

```
pca <- snpgdsPCA(genofile,snp.id=good.snps,autosome.only=F)

pc.percent <- pca$varprop*100
head(round(pc.percent, 2))
#3.78 1.64 0.96 0.72 0.68 0.67

frame <- data.frame(ID=samp.id,
                    POP=ordermeta$POP,
                    LOC=ordermeta$LOC,
                    PC1=pca$eigenvect[,1],
                    PC2=pca$eigenvect[,2],
                    PC3=pca$eigenvect[,3])

# Plot PCA with ggplot
pca_plot <- ggplot(frame, aes(x=PC1, y=PC2, shape=LOC,
color=POP)) +
  geom_point(size=3) +
  scale_color_manual(values = c("purple3", "navyblue",
"dodgerblue4", "dodgerblue3", "dodgerblue2", "skyblue",
"darkgreen","lawngreen","seagreen4","seagreen1","coral4",
"firebrick4","firebrick3", "firebrick1",
"goldenrod4","goldenrod", "darkorange4","darkorange3",
"darkorange", "lightsalmon1")) +
  theme_bw()

pca_plot

saveRDS(frame,"PCA_ld.prune.rds") # will use later in random
forest


#### ......... CALCULATE PAIRWISE FST ......... ####
unique.pops <- unique(ordermeta$POP)
fst.mat_filt <- matrix(nrow=length(unique.pops)^2,ncol=5)

count=1
for (i in 1:length(unique.pops)) {
  for (j in 1:length(unique.pops)) {
    if (i<j) {
      fst <- snpgdsFst(genofile,
                       snp.id=good.snps,

sample.id=ordermeta$INDV[ordermeta$POP%in%c(unique.pops[i],uniqu
e.pops[j])],

population=as.factor(ordermeta$POP[ordermeta$POP%in%c(unique.pop
s[i],unique.pops[j])]),
```

```
                        method="W&C84")
      fst.mat_filt[count,] <- c(unique.pops[i],unique.pops[j],

ordermeta$LOC[ordermeta$POP==unique.pops[i]][1],

ordermeta$LOC[ordermeta$POP==unique.pops[j]][1],
                                  round(fst$Fst,digits=6))
      count=count+1
    }
  }
}
fst.mat_filt <- na.omit(data.frame(fst.mat_filt))
names(fst.mat_filt) <- c("Pop1","Pop2","Loc1","Loc2","Fst")

fst.mat_filt$Fst = as.numeric(fst.mat_filt$Fst)
fst.mat_filt= subset(fst.mat_filt, select = -c(Loc1,Loc2))
fst.mat_filt=pmax(fst.mat_filt,0)
fst.mat_filt[is.na(fst.mat_filt)] <- 0

library(igraph)
G <- graph_from_data_frame(fst.mat_filt,directed=FALSE)
A <-
as_adjacency_matrix(G,names=TRUE,sparse=FALSE,attr="Fst",type='l
ower')

library(pheatmap)
pheatmap(A, cutree_rows = 4)

###Add geographic disances
library(fields)
geo <- read.delim("fst_latlong_new.txt")
dists <- fields::rdist.earth(geo[,c(2:3)])
fst.mat_filt$dist=NA

write.csv(dists, "dists_new.csv") #add sample pop ids then re-
import into R
dists <- read.csv("dists_new.csv", header = TRUE)

count=1
for (i in 1:length(unique.pops)) {
  for (j in 1:length(unique.pops)) {
    if (i<j) {
      fst.mat_filt$dist[count]=dists[i,j]
      count=count+1
    }
  }
}
```

```
write.csv(fst.mat_filt, "fst.mat_filt_new.csv") #write csv and
manually fix the distances based on the "dist" matrix
fst.mat_filt <- read.csv("fst.mat_filt_new.csv") # re-import the
correct fast.mat_filt with the correct distances

# needed to edit the matrix manually- add the distances to the
fst.mat_filt sheet for each population in excel then re-import
cor(fst.mat_filt$Fst,fst.mat_filt$dist,method="spearman")

# the above was overall, but what about separating pops within
each state and then between states
library(dplyr)
correlation_results <- fst.mat_filt %>%
  group_by(type) %>%
  summarise(correlation = cor(Fst, dist, method = "spearman"))

# other plot with different colors
my_colors <- c("dodgerblue4", "goldenrod3","seagreen")

p <- ggplot(fst.mat_filt,aes(x=dist,y=Fst,color=type)) +
  geom_point(size=2) +
  theme_bw() + xlab("Distance (km)") + ylab(expression('F'[ST]))
+
  scale_color_manual(values = my_colors)
p

#### ......... ADMIXTURE......... ####
library(LEA)

###Read in genotype data
##"ld.pruned_Zm_QD7_Dp7_194_imputed_filt.012" made with vcftools
using thinned positions from PCA script
(ld.pruned_Zm_QD7_Dp7_194_imputed_filt.pos)
## vcftools --vcf Zm_QD7_Dp7_194_imputed_filt.vcf --positions
ld.pruned_Zm_QD7_Dp7_194_imputed_filt.pos --012 --out
ld.pruned_Zm_QD7_Dp7_194_imputed_filt

gen <-
read.delim("ld.pruned_Zm_QD7_Dp7_194_imputed_filt.012",header=F,
sep="\t",na.strings="-1",row.names=1)
write.geno(gen, "genotypes_Zm_QD7_Dp7_194_imputed_filt.geno")

genB=gen
metaB <-
read.csv("metadata_Zm_QD7_Dp7_194_imputed_filt_copy.csv",
header= TRUE)
```

```
# Ensure the number of rows match before combining
if (nrow(metaB) == nrow(genB)) {
  # Add the new column from CSV at index 0
  combined_data <- cbind(metaB, genB)
} else {
  stop("The number of rows in the CSV and TSV files do not
match.")
}

# Step 1: Sort by the first column (index 0, which is the new
column from the CSV)
combined_data_sorted <- combined_data[order(combined_data[,1]),]

genB <- combined_data_sorted [,-1]

write.geno(genB, "genotypes_Zm_QD7_Dp7_194_imputed_filt_B.geno")
# SAMPLES RE_ORDERED FOR ADMIXTURE PLOT
project_filt = NULL
project_filt =
snmf("genotypes_Zm_QD7_Dp7_194_imputed_filt_B.geno", # SAMPLES
RE_ORDERED FOR ADMIXTURE PLOT
                    K = 1:6,
                    entropy = TRUE,
                    repetitions = 5,
                    project = "new")

#makes folder in working directory called
"genotypes_QD7_Dp7_0819_remdpblw3_imputed.snmf" and
"genotypes_QD7_Dp7_0819_remdpblw3_imputed.snmfProject"
plot(project_filt, col = "blue", pch = 19, cex = 1.2)
best = which.min(cross.entropy(project_filt, K = 3))

my.colors <- c("dodgerblue4", "goldenrod3", "tomato3")
barchart(project_filt, K = 3, run = best,
         border = TRUE, space = 0,
         col = my.colors,
         sort.by.Q = FALSE,
         xlab = "Individuals",
         ylab = "Ancestry proportions",
         main = "Ancestry matrix") -> bp
bp

axis(1, at = 1:length(bp$order),
     labels = bp$order, las=2,
     cex.axis = .5)
```

lfmm and OUTFLANK
****

```
##First, filtered to maf>0.05 with vcftools and output ped
#for some reason PLINK expects numeric chromosome values (1, 2,
3, etc.) so I had to re-name the chromosome before converting
#but first I made a copy of Zm_QD7_Dp7_194_imputed_filt.vcf
(Zm_QD7_Dp7_194_imputed_filt_copy.vcf), and then I renamed them
like this:
#sed -e 's/^Chr01/1/' -e 's/^Chr02/2/' -e 's/^Chr03/3/' -e
's/^Chr04/4/' -e 's/^Chr05/5/' -e 's/^Chr06/6/'
Zm_QD7_Dp7_194_imputed_filt.vcf >
Zm_QD7_Dp7_194_imputed_filt_renamed.vcf

#vcftools --vcf Zm_QD7_Dp7_194_imputed_filt_renamed.vcf --maf
0.05 --plink --out Zm_QD7_Dp7_194_imputed_filt_maf05.vcf #
filters out maf <0.05
#outputs a ped file

##Impute genotypes with LEA - do only once - DONT NEED TO DO
THIS SINCE I IMPUTED INITIALLY
##new data- filtered out maf with vcftools
ped2lfmm(input.file="Zm_QD7_Dp7_194_imputed_filt_renamed_maf05.p
ed",
output.file="Zm_QD7_Dp7_194_imputed_filt_renamed_maf05.lfmm")
#using ped file after filtering for maf 0.05

# makes a file called
Zm_QD7_Dp7_194_imputed_filt_renamed_maf05.lfmm
project <-
snmf("Zm_QD7_Dp7_194_imputed_filt_renamed_maf05.ped",K=3,
entropy=T, repetitions = 5, project="new")

# makes a folder and files in working directory called
"Zm_QD7_Dp7_194_imputed_filt_renamed_maf05.snmf",
"Zm_QD7_Dp7_194_imputed_filt_renamed_maf05.snmfProject",
"Zm_QD7_Dp7_194_imputed_filt_renamed_maf05.geno"
best=which.min(cross.entropy(project,K=3))

# DONT NEED TO DO THIS SINCE I IMPUTED INITIALLY
#Zm_QD7_Dp7_194_imputed_filt_maf05_imputed.lfmm=
impute(project,"Zm_QD7_Dp7_194_imputed_filt_maf05.lfmm",method="
mode",K=3,run=best)
# makes a file called
Zm_QD7_Dp7_194_imputed_filt_maf05_imputed.lfmm
```

```
###Read in imputed genotypes and metadata
# need to generate ind and pos files that match the ped/lfmm
files which have just gone through another filtering criteria
(maf>0.05)
#vcftools --vcf Zm_QD7_Dp7_194_imputed_filt_renamed.vcf --maf
0.05 --012 --out
Zm_QD7_Dp7_194_imputed_filt_renamed_maf05_genomat

# generates Zm_QD7_Dp7_194_imputed_filt_renamed_genomat.012.indv
and Zm_QD7_Dp7_194_imputed_filt_renamed_genomat.012.pos
inds <-
read.delim("Zm_QD7_Dp7_194_imputed_filt_renamed_maf05_genomat.01
2.indv",header=F)
#pos <-
read.delim("Zm_QD7_Dp7_194_imputed_filt_renamed_maf05_genomat.01
2.pos",header=F)
#names(pos) <- c("chr","pos")
posT <-
read.delim("Zm_QD7_Dp7_194_imputed_filt_renamed_maf05_genomat.01
2.pos",header=F) # made this to put the temp pvalues and such
names(posT) <- c("chr","pos")
gens <-
read.delim("Zm_QD7_Dp7_194_imputed_filt_renamed_maf05.lfmm",head
er=F,sep=" ")
meta <-
read.delim("metadata_Zm_QD7_Dp7_194_imputed_filt_pheno.txt",head
er=T)
identical(meta$IND,inds[,1])

# Analysis 1: Mean temperature (C)

#lfmm2
### Just NC #################

meta_with_temp <- meta[!is.na(meta$mean_tempC), ] # have to do
this because need to remove pops with no temp data for use in
lfmm
NC_T.gen <- gens[meta$State == "NC" & !is.na(meta$mean_tempC), ]

# only those with temp data
x = prcomp(NC_T.gen)
plot(x$sdev[1:20]^2, xlab = 'PC', ylab = "Variance explained")
points(5,x$sdev[6]^2, type = "h", lwd = 3, col = "blue") #K=5
looks like the correct number of factors
```

```
l2 <- apply(NC_T.gen,2,function(x)
sum(x,na.rm=t)/(2*length(na.omit(x))))
NC_T.poly <- NC_T.gen[,l2>0.05 & l2<0.95] #Remove SNPs that are
maf<0.05
dim(NC_T.poly) #261533 SNPs
NC_T.meta <- subset(meta_with_temp,State=="NC")

# Temperature
NC_T.lfmm <- lfmm2(input = NC_T.poly, env =
NC_T.meta$mean_tempC, K=5)
NC_T.pv <- lfmm2.test(NC_T.lfmm,NC_T.poly,NC_T.meta$mean_tempC)

##Check distribution
hist(NC_T.pv$pvalues)
NC_T.pvalues <- NC_T.pv$pvalues
qqplot(rexp(length(NC_T.pvalues), rate = log(10)),
       -log10(NC_T.pvalues), xlab = "Expected quantile",
       pch = 19, cex = .4)
abline(0,1)

##Add to dataframe
posT$NC_T <- NA
posT$NC_T[l2>0.05 & l2<0.95] <- NC_T.pvalues
posT$NC_T.q <- NA
posT$NC_T.q[l2>0.05 & l2<0.95] <-
p.adjust(NC_T.pvalues,method="fdr")

### Just VA #################

VA_T.gen <- gens[meta$State == "VA" & !is.na(meta$mean_tempC), ]
# only those with temp data

x = prcomp(VA_T.gen)
plot(x$sdev[1:20]^2, xlab = 'PC', ylab = "Variance explained")
points(2,x$sdev[6]^2, type = "h", lwd = 3, col = "blue")

l3 <- apply(VA_T.gen,2,function(x)
sum(x,na.rm=t)/(2*length(na.omit(x))))
VA_T.poly <- VA_T.gen[,l3>0.05 & l3<0.95] #Remove SNPs that are
maf<0.05
dim(VA_T.poly) #218238 SNPs
VA_T.meta <- subset(meta_with_temp,State=="VA")

# Temperature
VA_T.lfmm <- lfmm2(input = VA_T.poly, env =
VA_T.meta$mean_tempC, K=2)
VA_T.pv <- lfmm2.test(VA_T.lfmm,VA_T.poly,VA_T.meta$mean_tempC)
```

```
##Check distribution
hist(VA_T.pv$pvalues)
VA_T.pvalues <- VA_T.pv$pvalues
qqplot(rexp(length(VA_T.pvalues), rate = log(10)),
       -log10(VA_T.pvalues), xlab = "Expected quantile",
       pch = 19, cex = .4)
abline(0,1)

##Add to dataframe
posT$VA_T <- NA
posT$VA_T[l3>0.05 & l3<0.95] <- VA_T.pvalues
posT$VA_T.q <- NA
posT$VA_T.q[l3>0.05 & l3<0.95] <-
p.adjust(VA_T.pvalues,method="fdr")
saveRDS(posT,"lfmm_results_10.25.24_K5K2_meantempC.rds") #K=5,
K=2, mean_tempC

length(which(posT$NC_T<0.001)) #524
length(which(posT$NC_T.q<0.05)) #0
length(which(posT$VA_T<0.001)) #369
length(which(posT$VA_T.q<0.05)) #21
length(which(posT$NC_T<0.001 & posT$VA_T<0.001)) #0

##Plotting code ####################
posT <- readRDS("lfmm_results_10.25.24_K5K2_meantempC.rds")

palette(c("grey70","steelblue4"))
par(mfrow=c(2,1),mar=c(1,1,1,1))
plot(-
log10(posT$NC_T),col=as.factor(posT$chr),cex=0.5,pch=19,xlab="Po
sition",ylab="-log10(q)")
abline(h=-log10(0.001),col="red",lwd=2,lty="dashed")
summary(-log10(posT$NC_T))

plot(-log10(posT$VA_T),col=as.factor(posT$chr),cex=0.5,pch=19)
abline(h=-log10(0.001),col="red",lwd=2,lty="dashed")
summary(-log10(posT$VA_T))
```

```
# OutFLANK

library(OutFLANK)

###Paths to data
lfmm.path = "/Users/karinascavo/Library/Mobile
Documents/com~apple~CloudDocs/Research -
Eelgrass/PAPER_InPrep_NCVA_SeagrassGenomics/Seagrass_gen/may_sam
ples/Zm_QD7_Dp7_194_imputed_filt_renamed_maf05.lfmm"
meta <-
read.delim("metadata_Zm_QD7_Dp7_194_imputed_filt_pheno.txt",head
er=T)

#pos <-
read.delim("Zm_QD7_Dp7_194_imputed_filt_renamed_maf05_genomat.01
2.pos",header=F)
#names(pos) <- c("chr","pos")
posT <-
read.delim("Zm_QD7_Dp7_194_imputed_filt_renamed_maf05_genomat.01
2.pos",header=F)
names(posT) <- c("chr","pos")

###VCF to OutFLANK format
geno <- read.delim(lfmm.path,header=F,sep=" ")
#locusNames <- paste(pos$chr,"_",pos$pos,sep="")
locusNames <- paste(posT$chr,"_",posT$pos,sep="")


NC.G <- geno[meta$State == "NC" & !is.na(meta$mean_tempC), ]
NC.meta <- meta[meta$State == "NC" & !is.na(meta$mean_tempC), ]
NC.meta$group <- NA

#meanTempC
NC.meta$group[NC.meta$POP%in%c("DavisIsland","BogueA","Topsail",
"HarkersIsland")] <- "high"
NC.meta$group[NC.meta$POP%in%c("NorthRiver","BogueB")] <- "low"
table(NC.meta$group)

###Make FST matrix
NC.fst <-
MakeDiploidFSTMat(NC.G,locusNames=locusNames,popNames=NC.meta$gr
oup)
plot(NC.fst$FSTNoCorr,NC.fst$FST)
```

```
###Run OutFLANK
NC.OF <-
OutFLANK(NC.fst,LeftTrimFraction=0.05,RightTrimFraction=0.05,
                    Hmin=0.1,NumberOfSamples=4,qthreshold=0.05)
OutFLANKResultsPlotter(NC.OF,withOutliers=T,
                       NoCorr=T,Hmin=0.1,binwidth=0.005,

Zoom=F,RightZoomFraction=0.05,titletext=NULL) #Plot distribution
OutFLANKResultsPlotter(NC.OF,withOutliers=T,
                       NoCorr=T,Hmin=0.1,binwidth=0.005,

Zoom=T,RightZoomFraction=0.05,titletext=NULL) #Zoom in on right
tail


###Outliers
NC.P1 <-
pOutlierFinderChiSqNoCorr(NC.fst,Fstbar=NC.OF$FSTNoCorrbar,

dfInferred=NC.OF$dfInferred,qthreshold=0.05,Hmin=0.1)
NC.P1[,c("Chr","Pos")] <-
data.frame(do.call('rbind',strsplit(as.character(NC.P1$LocusName
),'_',fixed=TRUE)))
NC.P1 <- NC.P1[order(NC.P1$Chr,as.numeric(NC.P1$Pos)),]
NC.outliers <- NC.P1$OutlierFlag==TRUE
table(NC.outliers) #760
plot(NC.P1$He,NC.P1$FST,col=rgb(0,0,0,alpha=0.1))
points(NC.P1$He[NC.outliers],NC.P1$FST[NC.outliers],col="magenta
")

###Manhattan Plot
plot(1:nrow(NC.P1),NC.P1$FST,xlab="Position",ylab="FST",col=as.f
actor(NC.P1$Chr),pch=19,cex=0.2)
points(which(NC.outliers==TRUE),NC.P1$FST[which(NC.outliers==TRU
E)],col="darkred")
summary(NC.P1$FST)

##### Virginia
VA.G <- geno[meta$State == "VA" & !is.na(meta$mean_tempC), ]
VA.meta <- meta[meta$State == "VA" & !is.na(meta$mean_tempC), ]
VA.meta$group <- NA
VA.meta$group[VA.meta$POP%in%c("GoodwinC","PoquosonA","AllensIsl
and","BigIsland")] <- "high"
VA.meta$group[VA.meta$POP%in%c("GoodwinA","GoodwinB")] <- "low"
table(VA.meta$group)
```

```
###Make FST matrix
VA.fst <-
MakeDiploidFSTMat(VA.G,locusNames=locusNames,popNames=VA.meta$gr
oup)
plot(VA.fst$FSTNoCorr,VA.fst$FST)

###Run OutFLANK
VA.OF <-
OutFLANK(VA.fst,LeftTrimFraction=0.05,RightTrimFraction=0.05,
                   Hmin=0.1,NumberOfSamples=4,qthreshold=0.05)
OutFLANKResultsPlotter(VA.OF,withOutliers=T,
                       NoCorr=T,Hmin=0.1,binwidth=0.005,

Zoom=F,RightZoomFraction=0.05,titletext=NULL) #Plot distribution
OutFLANKResultsPlotter(VA.OF,withOutliers=T,
                       NoCorr=T,Hmin=0.1,binwidth=0.005,

Zoom=T,RightZoomFraction=0.05,titletext=NULL) #Zoom in on right
tail

###Outliers
VA.P1 <-
pOutlierFinderChiSqNoCorr(VA.fst,Fstbar=VA.OF$FSTNoCorrbar,

dfInferred=VA.OF$dfInferred,qthreshold=0.05,Hmin=0.1)
VA.P1[,c("Chr","Pos")] <-
data.frame(do.call('rbind',strsplit(as.character(VA.P1$LocusName
),'_',fixed=TRUE)))
VA.P1 <- VA.P1[order(VA.P1$Chr,as.numeric(VA.P1$Pos)),]
VA.outliers <- VA.P1$OutlierFlag==TRUE
table(VA.outliers)
plot(VA.P1$He,VA.P1$FST,col=rgb(0,0,0,alpha=0.1))
points(VA.P1$He[VA.outliers],VA.P1$FST[VA.outliers],col="darkred
")

###Manhattan Plot
plot(1:nrow(VA.P1),VA.P1$FST,xlab="Position",ylab="FST",col=as.f
actor(VA.P1$Chr),pch=19,cex=0.2)
points(which(VA.outliers==TRUE),VA.P1$FST[which(VA.outliers==TRU
E)],col="darkred")
summary(VA.P1$FST)

######Add to lfmm dataframe
frame2 <- readRDS("lfmm_results_10.25.24_K5K2_meantempC.rds")
identical(VA.P1$Pos,as.character(frame$pos))
names(frame) <- c("chr","pos",
```

```
"NC.lfmm.p","NC.lfmm.q","VA.lfmm.p","VA.lfmm.q")
frame2$VA.out.p <- VA.P1$pvalues
frame2$VA.out.q <- VA.P1$qvalues
frame2$NC.out.p <- NC.P1$pvalues
frame2$NC.out.q <- NC.P1$qvalues

length(which(frame2$NC.out.p<0.001)) #4332
length(which(frame2$NC.out.q<0.05)) #760
length(which(frame2$VA.out.p<0.001)) #1142
length(which(frame2$VA.out.q<0.05)) #654
str(frame)

# plots grouping p and q values by chromosome
#lfmm2 p values
NC_p_frame <- frame[, c(1, 3)]
length(which(NC_p_frame$NC.lfmm.p<0.001)) #524

# Assuming NC_p_frame is your data frame
subset_NC_p_frame <- NC_p_frame[!is.na(NC_p_frame$NC.lfmm.p) &
NC_p_frame$NC.lfmm.p < 0.001, ]
length(subset_NC_p_frame$NC.lfmm.p) #524
subset_NC_p_frame2 <-
as.data.frame(table(subset_NC_p_frame$chr))

my_colors <-
c("grey70","steelblue4","grey70","steelblue4","grey70","steelblu
e4")
ggplot(subset_NC_p_frame2, aes(x = Var1, y = Freq, fill = Var1))
+
  geom_bar(stat = "identity") +
  scale_fill_manual(values = my_colors) +  # Use the custom
color scale
  theme_bw()

VA_p_frame <- frame[, c(1, 5)]
length(which(VA_p_frame$VA.lfmm.p<0.001)) #369

# Assuming VA_p_frame is your data frame
subset_VA_p_frame <- VA_p_frame[!is.na(VA_p_frame$VA.lfmm.p) &
VA_p_frame$VA.lfmm.p < 0.001, ]
length(subset_VA_p_frame$VA.lfmm.p) #369
subset_VA_p_frame2 <-
as.data.frame(table(subset_VA_p_frame$chr))
```

```
my_colors <-
c("grey70","steelblue4","grey70","steelblue4","grey70","steelblu
e4")
ggplot(subset_VA_p_frame2, aes(x = Var1, y = Freq, fill = Var1))
+
  geom_bar(stat = "identity") +
  scale_fill_manual(values = my_colors) +  # Use the custom
color scale
  theme_bw()

#OutFLANK q values

NC_q_out_frame <- frame2[, c(1, 10)]
length(which(NC_q_out_frame$NC.out.q<0.05)) #760
# Assuming NC_p_frame is your data frame
subset_NC_q_frame <-
NC_q_out_frame[!is.na(NC_q_out_frame$NC.out.q) &
NC_q_out_frame$NC.out.q < 0.05, ]
length(subset_NC_q_frame$NC.out.q) #760
subset_NC_q_frame2 <-
as.data.frame(table(subset_NC_q_frame$chr))

my_colors <-
c("grey70","steelblue4","grey70","steelblue4","grey70","steelblu
e4")
ggplot(subset_NC_q_frame2, aes(x = Var1, y = Freq, fill = Var1))
+
  geom_bar(stat = "identity") +
  scale_fill_manual(values = my_colors) +  # Use the custom
color scale
  theme_bw()

VA_q_out_frame <- frame2[, c(1, 8)]
length(which(VA_q_out_frame$VA.out.q<0.05)) #654
# Assuming NC_p_frame is your data frame
subset_VA_q_frame <-
VA_q_out_frame[!is.na(VA_q_out_frame$VA.out.q) &
VA_q_out_frame$VA.out.q < 0.05, ]
length(subset_VA_q_frame$VA.out.q) #654
subset_VA_q_frame2 <-
as.data.frame(table(subset_VA_q_frame$chr))

my_colors <-
c("grey70","steelblue4","grey70","steelblue4","grey70","steelblu
e4")
ggplot(subset_VA_q_frame2, aes(x = Var1, y = Freq, fill = Var1))
+
```

```
  geom_bar(stat = "identity") +
  scale_fill_manual(values = my_colors) +  # Use the custom
color scale
  theme_bw()

#Compared NC sites and VA sites with OutFLANK (NC= warmer temps,
VA=cooler temps)
#For both NC and VA, only include rows where temperature data
(mean_tempC) is not missing
NC_VA_T.G <- geno[meta$State %in% c("NC", "VA") &
!is.na(meta$mean_tempC), ]
meta_with_temp <- meta[!is.na(meta$mean_tempC), ] # have to do
this bevause need to remove pops with no temp data for use in
lfmm
NC_VA_T.meta=meta_with_temp # just renaming it here
NC_VA_T.meta$group <- NA

#meanTempC
NC_VA_T.meta$group[NC_VA_T.meta$POP%in%c("BogueA","Topsail","Nor
thRiver","BogueB","HarkersIsland", "DavisIsland")] <- "high"
NC_VA_T.meta$group[NC_VA_T.meta$POP%in%c("PoquosonA","GoodwinA",
"GoodwinB","AllensIsland","BigIsland","GoodwinC")] <- "low"
table(NC_VA_T.meta$group)

###Make FST matrix
NC_VA.fst <-
MakeDiploidFSTMat(NC_VA_T.G,locusNames=locusNames,popNames=NC_VA
_T.meta$group)
plot(NC_VA.fst$FSTNoCorr,NC_VA.fst$FST)

###Run OutFLANK
NC_VA.OF <-
OutFLANK(NC_VA.fst,LeftTrimFraction=0.05,RightTrimFraction=0.05,
                      Hmin=0.1,NumberOfSamples=4,qthreshold=0.05)
OutFLANKResultsPlotter(NC_VA.OF,withOutliers=T,
                       NoCorr=T,Hmin=0.1,binwidth=0.005,

Zoom=F,RightZoomFraction=0.05,titletext=NULL) #Plot distribution
OutFLANKResultsPlotter(NC_VA.OF,withOutliers=T,
                       NoCorr=T,Hmin=0.1,binwidth=0.005,

Zoom=T,RightZoomFraction=0.05,titletext=NULL) #Zoom in on right
tail
```

```
###Outliers
NC_VA.P1 <-
pOutlierFinderChiSqNoCorr(NC_VA.fst,Fstbar=NC_VA.OF$FSTNoCorrbar
,

dfInferred=NC_VA.OF$dfInferred,qthreshold=0.05,Hmin=0.1)
NC_VA.P1[,c("Chr","Pos")] <-
data.frame(do.call('rbind',strsplit(as.character(NC_VA.P1$LocusN
ame),'_',fixed=TRUE)))
NC_VA.P1 <-
NC_VA.P1[order(NC_VA.P1$Chr,as.numeric(NC_VA.P1$Pos)),]
NC_VA.outliers <- NC_VA.P1$OutlierFlag==TRUE
table(NC_VA.outliers) #FALSE-287482
plot(NC_VA.P1$He,NC_VA.P1$FST,col=rgb(0,0,0,alpha=0.1))
points(NC_VA.P1$He[NC_VA.outliers],NC_VA.P1$FST[NC_VA.outliers],
col="magenta")

###Manhattan Plot
plot(1:nrow(NC_VA.P1),NC_VA.P1$FST,xlab="Position",ylab="FST",co
l=as.factor(NC_VA.P1$Chr),pch=19,cex=0.2)
points(which(NC_VA.outliers==TRUE),NC_VA.P1$FST[which(NC_VA.outl
iers==TRUE)],col="darkred")
summary(NC_VA.P1$FST)

##Add to lfmm dataframe
identical(NC_VA.P1$Pos,as.character(frame$pos))
frame$NC_VA.out.p <- NC_VA.P1$pvalues
frame$NC_VA.out.q <- NC_VA.P1$qvalues

length(which(frame$NC_VA.out.p<0.001)) #6210
length(which(frame$NC_VA.out.q<0.05)) #0
str(frame)

# no significant outliers, no need to make plot of outliers by
chromosome
#NC_VA_q_out_frame <- frame[, c(1, 8)]
#length(which(NC_VA_q_out_frame$NC_VA.out.q<0.05)) #0

saveRDS(frame2,"lfmm_results_10.25.24_K5K2_meantempC_warm4_cool2
.rds")
write.csv(frame2,
"lfmm_results_10.25.24_K5K2_meantempC_warm4_cool2.csv")
```

LDAnnot and Gene Intersect
****

```
##Create lists of outlier SNPs
##NC LFMM
#awk -F ',' '{if($4 < 0.05){print}}'
/projectnb/cnidaria/kscavo/Eelgrass/raw/23219Kam_N23139/cleanfas
tq/R_analysis/lfmm_results_10.25.24_K5K2_meantempC_warm4_cool2.c
sv | \
#cut -d "," -f1,2 | tr ',' '\t' > NCLFMM_cands_list_ChrPos.txt

##NC Outflank
#awk -F ',' '{if($10 < 0.05){print}}'
/projectnb/cnidaria/kscavo/Eelgrass/raw/23219Kam_N23139/cleanfas
tq/R_analysis/lfmm_results_10.25.24_K5K2_meantempC_warm4_cool2.c
sv | \
#cut -d "," -f1,2 | tr ',' '\t' > NCOut_cands_list_ChrPos.txt

##VA LFMM
#awk -F ',' '{if($6 < 0.05){print}}'
/projectnb/cnidaria/kscavo/Eelgrass/raw/23219Kam_N23139/cleanfas
tq/R_analysis/lfmm_results_10.25.24_K5K2_meantempC_warm4_cool2.c
sv | \
#cut -d "," -f1,2 | tr ',' '\t' > VALFMM_cands_list_ChrPos.txt

##VA Outflank
#awk -F ',' '{if($8 < 0.05){print}}'
/projectnb/cnidaria/kscavo/Eelgrass/raw/23219Kam_N23139/cleanfas
tq/R_analysis/lfmm_results_10.25.24_K5K2_meantempC_warm4_cool2.c
sv | \
#cut -d "," -f1,2 | tr ',' '\t' > VAOut_cands_list_ChrPos.txt

##NC_VA Outflank
#awk -F ',' '{if($12 < 0.05){print}}'
/projectnb/cnidaria/kscavo/Eelgrass/raw/23219Kam_N23139/cleanfas
tq/R_analysis/lfmm_results_10.25.24_K5K2_meantempC_warm4_cool2.c
sv | \
#cut -d "," -f1,2 | tr ',' '\t' > NCVAOut_cands_list_ChrPos.txt

############### Run LD-annot

##NC LFMM- dont need to do this - 0
#python3 LD-annot0.4.py Zm_QD7_Dp7_194_imputed_filt.vcf
Zmarina_668_v3.1.gene.gff3 NCLFMM_cands_list_ChrPos.txt mRNA 0.9
NCLFMM_cands_list_ChrPos_LDannot_mRNA_0.9.txt

##NC OutFlank
```

```
#python3 LD-annot0.4.py Zm_QD7_Dp7_194_imputed_filt.vcf
Zmarina_668_v3.1.gene.gff3 NCOut_cands_list_ChrPos.txt mRNA 0.9
NCOut_cands_list_ChrPos_LDannot_mRNA_0.9.txt

##VA LFMM
#python3 LD-annot0.4.py Zm_QD7_Dp7_194_imputed_filt.vcf
Zmarina_668_v3.1.gene.gff3 VALFMM_cands_list_ChrPos.txt mRNA 0.9
VALFMM_cands_list_ChrPos_LDannot_mRNA_0.9.txt

##VA OutFlank
#python3 LD-annot0.4.py Zm_QD7_Dp7_194_imputed_filt.vcf
Zmarina_668_v3.1.gene.gff3 VAOut_cands_list_ChrPos.txt mRNA 0.9
VAOut_cands_list_ChrPos_LDannot_mRNA_0.9.txt

##Generate lists of gene models
#cat VALFMM_cands_list_ChrPos_LDannot_mRNA_0.9.txt | \
#cut -f 7 | cut -d ";" -f 2 | cut -d "=" -f 2 | sort | uniq |
sed '$d' > \
#VALFMM_cands_list_ChrPos_LDannot_mRNA_0.9_geneList_name_final_u
niq.txt

#cat NCOut_cands_list_ChrPos_LDannot_mRNA_0.9.txt | \
#cut -f 7 | cut -d ";" -f 2 | cut -d "=" -f 2 | sort | uniq |
sed '$d' > \
#NCOut_cands_list_ChrPos_LDannot_mRNA_0.9_geneList_name_final_un
iq.txt

#cat VAOut_cands_list_ChrPos_LDannot_mRNA_0.9.txt | \
#cut -f 7 | cut -d ";" -f 2 | cut -d "=" -f 2 | sort | uniq |
sed '$d' > \
#VAOut_cands_list_ChrPos_LDannot_mRNA_0.9_geneList_name_final_un
iq.txt

###Generate gene lists
#grep -wFf
VALFMM_cands_list_ChrPos_LDannot_mRNA_0.9_geneList_name_final_un
iq.txt Zmarina_668_v3.1.annotation_info.txt > VA_LFMM_genes.txt
##grep -wFf
NCOut_cands_list_ChrPos_LDannot_mRNA_0.9_geneList_name_final_uni
q.txt Zmarina_668_v3.1.annotation_info.txt > NC_Out_genes.txt
#grep -wFf
VAOut_cands_list_ChrPos_LDannot_mRNA_0.9_geneList_name_final_uni
q.txt Zmarina_668_v3.1.annotation_info.txt > VA_Out_genes.txt
```

```
VA.lfmm <-
read.delim("VALFMM_cands_list_ChrPos_LDannot_mRNA_0.9_geneList_n
ame_final_uniq.txt",header=F)[,1]
NC.out <-
read.delim("NCOut_cands_list_ChrPos_LDannot_mRNA_0.9_geneList_na
me_final_uniq.txt",header=F)[,1]
VA.out <-
read.delim("VAOut_cands_list_ChrPos_LDannot_mRNA_0.9_geneList_na
me_final_uniq.txt",header=F)[,1]

length(VA.lfmm) #4
length(NC.out) #77
length(VA.out) #94
length(intersect(VA.lfmm,NC.out)) #1
length(intersect(VA.lfmm,VA.out)) #2
length(intersect(NC.out,VA.out)) #3

genes <- unique(c(VA.lfmm,NC.out,VA.out))
frame2 <- data.frame(genes,
                     VA.lfmm=genes%in%VA.lfmm,
                     NC.out=genes%in%NC.out,
                     VA.out=genes%in%VA.out)

saveRDS(list(all=frame2),file="GeneIntersect_meantempC_warm4_coo
l2.rds")

#labels = c("VA LFMM","NC OutFLANK","VA OutFLANK")
#plot(euler(frame[,2:4],shape="ellipse"), fills =
c("darkslateblue", "cadetblue", "seagreen"), quantities = T,
labels=labels)
#plot(euler(frame[,2:4],shape="ellipse"), fills =
c("darkslateblue", "cadetblue", "seagreen"), quantities = T,
legend = list(labels = c("VA LFMM","NC OutFLANK","VA OutFLANK"),
col = "black"))
#plot(euler(frame[,2:4],shape="ellipse"), edges = fills =
c("darkslateblue", "cadetblue", "seagreen"), quantities = F,
legend = list(labels = c("VA LFMM","NC OutFLANK","VA OutFLANK"),
col = "black"))

library(eulerr)
library(cowplot)
library(grid)
library(viridis)

cols <- viridis(3,option="D")
labs <- c("VA LFMM","NC OutFLANK","VA OutFLANK")
```

```
###SNP Level Overlap
dat <-
readRDS("lfmm_results_10.25.24_K5K2_meantempC_warm4_cool2.rds")
lfmm <- dat[,grep("lfmm.q",names(dat))]
out <- dat[,grep("out.q",names(dat))]

lfmm.thresh=0.05
out.thresh=0.05

lfmm.tf <- lfmm<lfmm.thresh
out.tf <- out<out.thresh

all.tf <- data.frame(lfmm.tf,out.tf)
all.tf[is.na(all.tf)] <- FALSE

par(mar=c(6,6,6,6))

###Plot
plot(euler(all.tf[,2:4],shape="ellipse"),quantities=list(cex =
1.5),
     edges="black",fills =
c("darkslateblue","seagreen","steelblue"), #list(fill = cols,
alpha = 0.5),
     labels=T,lwd=2)
plot(euler(all.tf[,2:4],shape="ellipse"),
     edges="black",fills =
c("darkslateblue","seagreen","steelblue"), #list(fill = cols,
alpha = 0.5),
     labels=F,lwd=2)
colSums(all.tf)

##Gene level
gene <- readRDS("GeneIntersect_meantempC_warm4_cool2.rds")
gene.tf <- gene[[1]]

###Plot both
plot(euler(gene.tf[,2:4],shape="ellipse"),quantities=list(cex =
1.5),
     edges="black",fills = c("darkslateblue", "steelblue",
"seagreen"), #list(fill = cols, alpha = 0.5),
     labels=T,lwd=2)
plot(euler(gene.tf[,2:4],shape="ellipse"),
     edges="black",fills = c("darkslateblue", "steelblue",
"seagreen"), #list(fill = cols, alpha = 0.5),
     labels=F,lwd=2)

colSums(gene.tf[,2:4])
```

```
TopGO Gene Ontology
**************************************************

library(topGO)
geneID2GO <- readMappings(file
="Zmarina_668_v3.1.annotation_info_GO.txt", sep = "\t", IDsep =
",")
geneNames <- names(geneID2GO)

#need to manually edit the columns in these files before
importing- get rid of the column with"PAC:50095054" and the
duplicate column "Zosma01g28420"
# read in one of these "cands" at a time, read one in, then
proceed with following code
cands <- read.delim("VA_LFMM_genes.txt",header=F)[,1]
#cands <- read.delim("NC_Out_genes.txt",header=F)[,1]
#cands <- read.delim("VA_Out_genes.txt",header=F)[,1]
myIG <- cands
length(cands)

geneList <- factor(as.integer(geneNames%in%myIG))
names(geneList) <- geneNames

###Molecular Function
GOdata_MF <- new("topGOdata",ontology="MF",allGenes=geneList,
                 annot=annFUN.gene2GO,gene2GO=geneID2GO)
test.stat <- new("classicCount", testStatistic = GOFisherTest,
name = "Fisher test")
resultFisher <- getSigGroups(GOdata_MF, test.stat)
test.stat2 <- new("weightCount", testStatistic = GOFisherTest,
name = "Fisher test", sigRatio = "ratio")
resultWeight <- getSigGroups(GOdata_MF, test.stat2)
allRes <- GenTable(GOdata_MF, classic = resultFisher,
                   weight = resultWeight, orderBy = "weight",
                   ranksOf =
"classic",topNodes=length(resultFisher@score),numChar=100)
filtRes.MF <- allRes[allRes$classic<0.05 &
allRes$Significant>2,]
filtRes.MF$Ontology <- "MF"

##Biological Process
GOdata_BP <- new("topGOdata",ontology="BP",allGenes=geneList,
                 annot=annFUN.gene2GO,gene2GO=geneID2GO)
resultFisher <- getSigGroups(GOdata_BP, test.stat)
resultWeight <- getSigGroups(GOdata_BP, test.stat2)
allRes <- GenTable(GOdata_BP, classic = resultFisher,
                   weight = resultWeight, orderBy = "weight",
```

```
                    ranksOf =
"classic",topNodes=length(resultFisher@score),numChar=100)
filtRes.BP <- allRes[allRes$classic<0.05 &
allRes$Significant>2,]
filtRes.BP$Ontology <- "BP"

##Cellular Component
GOdata_CC <- new("topGOdata",ontology="CC",allGenes=geneList,
                 annot=annFUN.gene2GO,gene2GO=geneID2GO)
resultFisher <- getSigGroups(GOdata_CC, test.stat)
resultWeight <- getSigGroups(GOdata_CC, test.stat2)
allRes <- GenTable(GOdata_CC, classic = resultFisher,
                   weight = resultWeight, orderBy = "weight",
                   ranksOf = "classic",
topNodes=length(resultFisher@score))
filtRes.CC <- allRes[allRes$classic<0.05 &
allRes$Significant>2,]
filtRes.CC$Ontology <- "CC"

##Output all three
filt.all <- rbind(filtRes.BP,filtRes.MF,filtRes.CC)
write.csv(filt.all,"VA_Out_GOenrichment_meantempC_warm4_cool2.cs
v")

## GO Intersect
###Read in GO enrichment for each
NCOut <-
read.csv("NC_Out_GOenrichment_meantempC_warm4_cool2.csv")
VAOut <-
read.csv("VA_Out_GOenrichment_meantempC_warm4_cool2.csv")
VAlfmm <- read.csv("VA_LFMM_GOenrichment.csv")

GOs <- unique(c(NCOut$GO.ID,VAOut$GO.ID))

##Including inversion
frame <- data.frame(GOs,
                    NC1.out=GOs%in%NCOut$GO.ID,
                    VA1.out=GOs%in%VAOut$GO.ID)
saveRDS(list(all=frame),file="GOIntersect_meanTempC_warm4_cool2.
rds")

labels = c("NC OutFLANK","VA OutFLANK")
plot(euler(frame[,2:3],shape="ellipse"),quantities =
T,labels=labels)
plot(euler(frame[,2:3],shape="ellipse"),
     edges="black",fills = c("white", "seagreen",
"darkslateblue"),
```

```
     labels=T,lwd=2)


library(viridis)
custom_colors <- viridis(5)  # Generate 20 colors using viridis
ggplot(NCOut, aes(x = reorder(Term, -classic), y = -
log10(classic), fill = GO.ID)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Significant GO Terms in Biological Process
(BP)", x = "GO Terms", y = "-log10(p-value)") +
  scale_fill_manual(values = custom_colors) +  # Use custom
viridis colors
  theme_minimal() +
  theme(legend.position = "none")

ggplot(VAOut, aes(x = reorder(Term, -classic), y = -
log10(classic), fill = GO.ID)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Significant GO Terms in Biological Process
(BP)", x = "GO Terms", y = "-log10(p-value)") +
  scale_fill_manual(values = custom_colors) +  # Use custom
viridis colors
  theme_minimal() +
  theme(legend.position = "none")
```

RClone
****

```
####Load packages####
library(vcfR)
library(RClone)
library(adegenet)

setwd

VA_genind <- vcfR2genind(VA.vcf, sep = "[|/]")
VA_geninddf <- genind2df(VA_genind, sep="/")

write.table(VA_geninddf,"zm_all_VA10_geninddf.txt", row.names =
TRUE)
VA.txt.import <- read.table("zm_all_VA10_geninddf.txt", header =
TRUE)

VA_rclone <- convert_GC(as.data.frame(VA.txt.import), 1, "/")

VAres <- genet_dist(VA_rclone, alpha1 = 0.05)

plot <- hist(VAres$distance_matrix, freq = FALSE, col =
"darkgray", main = "VA10", xlab = "Genetic distances", breaks =
seq(0, max(VAres$distance_matrix)+1, 1))

VAres$potential_clones
```