Starling-May18
Projects/Katarina
Stuart/KStuart.Starling-Aug18/Sv10_NZstarlings/Analysis/2024-04-26.WorkshopReanalysis

PDF Version generated by

Katarina Stuart (z5188231@ad.unsw.edu.au)

on

Sep 25, 2024 @12:48 PM AEST

Table of Contents

2024-04-26.WorkshopReanalysis

Katarina Stuart (z5188231@ad.unsw.edu.au) - Sep 25, 2024, 12:47 PM GMT+10

Workshop Reanalysis

MAP

```
module load R/4.1.0-gimkl-2020a
R

#install.packages("mapdata")
library(mapdata)
library(ggplot2)
library(data.table)
library(dplyr)

setwd("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/map")

burd <- fread("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools_old/Metadata_NZ_AU_UK_BE_ReplicatesSibRemoved2.csv",
sep=",", quote = "", fill=TRUE)
ANT <- burd %>% filter(pop2 == "ANT (BE)")
MKW <- burd %>% filter(pop2 == "MKW (UK)")
NWC <- burd %>% filter(pop2 == "NWC (UK)")

ORG <- burd %>% filter(pop2 == "ORG (AU)")
MLV <- burd %>% filter(pop2 == "MLV (AU)")

AUK <- burd %>% filter(pop2 == "AUK (NZ)")
UHT <- burd %>% filter(pop2 == "UHT (NZ)")
PLM <- burd %>% filter(pop2 == "PLM (NZ)")
BLM <- burd %>% filter(pop2 == "BLN (NZ)")
CAN <- burd %>% filter(pop2 == "CAN (NZ)")

w2hr <- map_data("worldHires")
dim(w2hr)

##### sorting out native and invasive ranges
Invasive <- read.csv("Invasive.txt", sep="\t", quote = "", header=FALSE)
Invasive <- as.vector(Invasive[,1])
Native <- read.csv("Native.txt", sep="\t", quote = "", header=FALSE)
Native <- as.vector(Native[,1])
ExpandingRange <- read.csv("ExpandingRange.txt", sep="\t", quote = "", header=FALSE)
ExpandingRange <- as.vector(ExpandingRange[,1])

starlingsfull <- read.csv("starlingsfull.csv",stringsAsFactors=TRUE,sep=",")
levels(starlingsfull$COUNTRY)
str(starlingsfull)

# Calculate differences between adjacent values
differences <- abs(diff(starlingsfull$LONGITUDE))
# Find birds where differences in long is greater than 0.2
indices_to_remove <- which(differences > 0.005) + 1
# Remove corresponding rows from the dataset
starlingsfull2 <- starlingsfull[-indices_to_remove, ]

# Calculate differences between adjacent values
differences <- abs(diff(starlingsfull$LATITUDE))
# Find birds where differences in long is greater than 0.2
indices_to_remove <- which(differences > 0.005) + 1
# Remove corresponding rows from the dataset
starlingsfull3 <- starlingsfull2[-indices_to_remove, ]

ExpandingRangePoints <- starlingsfull3[starlingsfull3$COUNTRY %in% ExpandingRange,]
```

```
InvasivePoints <- starlingsfull3[starlingsfull3$COUNTRY %in% Invasive,]
NativePoints <- starlingsfull3[starlingsfull3$COUNTRY %in% Native,]

#black outlined, grey filled
gg1 <- ggplot() +
  geom_polygon(data = w2hr, aes(x=long, y = lat, group = group), fill = "gray80", color = "gray80") +
  coord_fixed(1.3)

#Trying to add starlings
gg2 <- gg1 +
  geom_point(data = ExpandingRangePoints, aes(x = LONGITUDE, y = LATITUDE), color = "#634090", size = 0.25, alpha = 1) +
  geom_point(data = InvasivePoints, aes(x = LONGITUDE, y = LATITUDE), color = "#40905C", size = 0.25, alpha = 1) +
  geom_point(data = NativePoints, aes(x = LONGITUDE, y = LATITUDE), color = "#634090", size = 0.25, alpha = 1) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_blank())

#Non-blank map
pdf("World_Map_range.pdf")
gg2
dev.off()

png("World_Map_range.png",  width = 1050, height = 650, bg = "transparent")
gg2
dev.off()

#Blank Map
pdf("World_Map.pdf")
ggplot() +
  geom_polygon(data = w2hr, aes(x=long, y = lat, group = group), fill = "gray80", color = "gray80") +
  coord_fixed(1.3) +
   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
         panel.background = element_blank(), axis.line = element_blank())
dev.off()

#AU
eastern_coast <- subset(w2hr, long > 135 & long < 155 & lat > -45 & lat < -15)
InvasivePoints2 <- InvasivePoints %>% filter(COUNTRY == "Australia") %>% filter(LATITUDE > -50 & LONGITUDE < 155 & LONGITUDE > 135)

pdf("World_Map_AU.pdf")
ggplot() +
  geom_polygon(data = eastern_coast, aes(x = long, y = lat, group = group), fill = "gray80", color = "gray80") +
  coord_fixed(1.3) +
 geom_point(data = InvasivePoints2, aes(x = LONGITUDE, y = LATITUDE), color = "#40905C", size = 1, alpha = 1) +
  geom_point(data = ORG, aes(x = lon, y = lat), color = "#ffcc00", size = 5, alpha = 1) +
  geom_point(data = MLV, aes(x = lon, y = lat), color = "#ff6600", size = 5, alpha = 1) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_blank())
dev.off()

#NZ
eastern_coast <- subset(w2hr, long > 165 & long < 179 & lat > -48 & lat < -30)
InvasivePoints3 <- InvasivePoints %>% filter(COUNTRY == "New Zealand") %>% filter(LATITUDE > -48 & LONGITUDE < 179 & LONGITUDE > 165)

pdf("World_Map_NZ.pdf")
ggplot() +
  geom_polygon(data = eastern_coast, aes(x = long, y = lat, group = group), fill = "gray80", color = "gray80") +
  coord_fixed(1.3) +
 geom_point(data = InvasivePoints3, aes(x = LONGITUDE, y = LATITUDE), color = "#40905C", size = 1, alpha = 1) +
  geom_point(data = AUK, aes(x = lon, y = lat), color = "#4CBB17", size = 5, alpha = 1) +
  geom_point(data = UHT, aes(x = lon, y = lat), color = "#9FE2BF", size = 5, alpha = 1) +
  geom_point(data = PLM, aes(x = lon, y = lat), color = "#006600", size = 5, alpha = 1) +
  geom_point(data = BLM, aes(x = lon, y = lat), color = "#80ff80", size = 5, alpha = 1) +
  geom_point(data = CAN, aes(x = lon, y = lat), color = "#C9CC3F", size = 5, alpha = 1) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_blank())
dev.off()
```

```
#NATIVE
eastern_coast <- subset(w2hr, long > -20 & long < 10 & lat > 45 & lat < 60)

pdf("World_Map_NATIVE.pdf")
ggplot() +
  geom_polygon(data = eastern_coast, aes(x = long, y = lat, group = group), fill = "gray80", color = "gray80") +
  coord_fixed(1.3) +
```

```
  geom_point(data = ANT, aes(x = lon, y = lat), color = "blue", size = 5, alpha = 1) +
  geom_point(data = MKW, aes(x = lon, y = lat), color = "#800080", size = 5, alpha = 1) +
  geom_point(data = NWC, aes(x = lon, y = lat), color = "#CB65CB", size = 5, alpha = 1) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_blank())
dev.off()
```

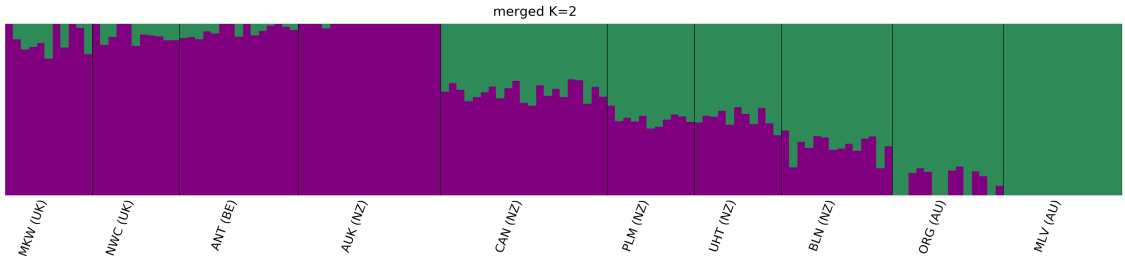
Admixture

Admixture: Full and NZ

```
#!/bin/bash -e

#SBATCH --job-name=2024_04_26.admixture_global_nz.sl
#SBATCH --account=uoa02613
#SBATCH --time=00-12:00:00
#SBATCH --mem=1GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --profile task

# load modules
module purge
module load PLINK/1.09b6.16
module load Miniconda3

#set up conda stuff
CONDA_BASE=$(conda info --base)
source ${CONDA_BASE}/etc/profile.d/conda.sh
conda activate /nesi/nobackup/uoa02613/kstuart_projects/programs/conda_pkgs/admixture

# Run admixture
cd  /nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/admixture_reanalysis

DIR=/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/final_filtering

#global
for K in {1..4};
do
admixture -s time -B -j16 --cv ${DIR}/starling_noduprel_qual_miss_filt.bed ${K} | tee starlings_global_${K}.out
done

#NZ only
for K in {1..4};
do
admixture -s time -B -j16 --cv ${DIR}/starling_noduprel_qual_miss_filt_NZ.bed ${K} | tee starlings_NZ_${K}.out
done


#global
for K in {5..10};
do
admixture -s time -B -j16 --cv ${DIR}/starling_noduprel_qual_miss_filt.bed ${K} | tee starlings_global_${K}.out
done

#NZ only
for K in {5..10};
do
admixture -s time -B -j16 --cv ${DIR}/starling_noduprel_qual_miss_filt_NZ.bed ${K} | tee starlings_NZ_${K}.out
done
```

Admixture plot: Full and NZ

```
cd  /nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/admixture_reanalysis

#Global Data
grep -h CV starlings_global_*.out > starlings_global_outall.txt

#CV PLOT
module load R-bundle-Bioconductor/3.15-gimkl-2022a-R-4.2.1
R

setwd("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/admixture_reanalysis")
CVs6 <- read.table("starlings_global_outall.txt", sep = " ")
CVs6 <- CVs6[, 3:4] ## drop the first two columns
## Remove the formatting around the K values:
CVs6[, 1] <- gsub(x = CVs6[, 1], pattern = "\\(K=",replacement = "")
CVs6[, 1] <- gsub(x = CVs6[, 1], pattern = "\\):", replacement = "")
head(CVs6)
CVs <- CVs6[c(2:10,1),]
pdf("starlings_global_outall.pdf", width=6, height=4)
plot(CVs6, xlab = "K", ylab = "CV error")
dev.off()
q()

#Rename files to meanQ from Q for compatibility with distruct
for file in starling_noduprel_qual_miss_filt.*.Q; do
  mv "$file" "${file/.Q/.meanQ}"
done

DIR=/nesi/nobackup/uoa02613/kstuart_projects/programs/distruct23
OUT_DIR=/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/admixture_reanalysis
cd ${OUT_DIR}

module load Python/2.7.16-gimkl-2018b #has to be this one
python ${DIR}/distruct2.3.py -K 2 --input=starling_noduprel_qual_miss_filt --output=starling_noduprel_qual_miss_filt_global_k2 --title="merged K=2" --popfile=distruct_pop_inds --
poporder=distruct_pop_order

python ${DIR}/distruct2.3.py -K 3 --input=starling_noduprel_qual_miss_filt --output=starling_noduprel_qual_miss_filt_global_k3 --title="merged K=3" --popfile=distruct_pop_inds --
poporder=distruct_pop_order
```

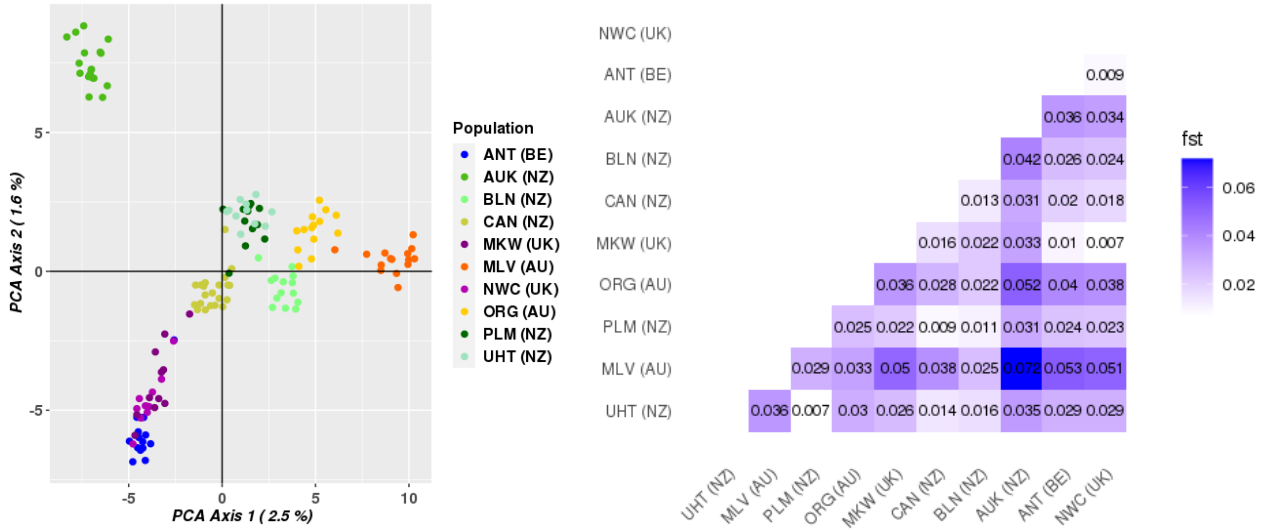


POPGEN: global

```
module load R-bundle-Bioconductor/3.15-gimkl-2022a-R-4.2.1
R
#Install Required packages (Only needs to be done once)
#load required packages
library(dartR)
library(vcfR)
library(dplyr)
library(tidyr)
library(networkD3)
library(ggplot2)

setwd("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis")
```

```
#library(tidyverse) #Won't install. Needed for gl.report.pa (private allele report)
#
# Read in the filtered VCF file, convert to Gen Light object, and run compleince check
readstarALLvcf <-
read.vcfR("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/final_filtering/starling_noduprel_qual_miss_filt.recode.vcf")
starALLgl <- vcfR2genlight(readstarALLvcf)
starALLgl <- gl.compliance.check(starALLgl) #Run Compliance check on new Gen Light object (Will be some orange but fixed after)

##Bring in Metadata file and Join (match) to genlight object
nameorder <- as.data.frame(indNames(starALLgl))
colnames(nameorder) <- c("id")
metanames <-
read.table("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/Metadata_NZ_AU_UK_BE_ReplicatesSibRemoved2.csv",header=T,sep
=",")
metacorrect <- right_join(nameorder, metanames, by = "id")
starALLgl$other$ind.metrics <- metacorrect
pop(starALLgl) <- starALLgl$other$ind.metrics$pop #add the population of each individual to the accessor pop
indNames(starALLgl) <- starALLgl$other$ind.metrics$id #add the individual names to the accessor indNames
pop(starALLgl) # Check pop data is there
names(starALLgl@other$ind.metrics) #list all available metrics for individuals to confirm integration to genlight object
#
#Create dataframe to add Lat/Lon coordinates to Genlight latlon slot
starALLgl@other$latlon <- data.frame(lat=starALLgl@other$ind.metrics$lat,lon=starALLgl@other$ind.metrics$lon)
#Housekeeping on Gen Light
starALLgl <- gl.compliance.check(starALLgl) # ReRun Compliance check on new Gen Light object after adding metadata. SHOULD BE ALL GREEN :)
starALLgl <- gl.recalc.metrics(starALLgl) #Re Calculate metrics
#
## Visual Outputs of dataset populations based on Regions
starALLgl <- gl.reassign.pop(starALLgl, as.pop="pop2") #Toggle pop name to the "pop2' (collapsed regions)
pca<-gl.pcoa(starALLgl,nfactors=5) # Performs Pearson Principal Component analysis (PCA) on SNP data
# Collapsed populations PCA with colour scheme (NZ = Green, BE = Blue, UK = Purple, AU – Orange)
pop_colors <- c("ANT (BE)" = "blue",
                "AUK (NZ)" = "#4CBB17",    # Kelly green
                "BLN (NZ)" = "#80ff80",    # Light green
                "CAN (NZ)" = "#C9CC3F",    # Pear
                "PLM (NZ)" = "#006600",    # CAMARONE, GREEN
                "MKW (UK)" = "#800080",    # Purple
                "NWC (UK)" = "#CB65CB",    # Dark purple
                "MLV (AU)" = "#ff6600",    # Orange
                "ORG (AU)" = "#ffcc00",    # Yellow-orange
                "UHT (NZ)" = "#9FE2BF")    # Seafoam Green


###PCA
# Use pop_colors in the gl.pcoa.plot function
pdf("Sv10_global_pca.pdf", width=8.5, height=5.0)
gl.pcoa.plot(pca, starALLgl, axis.label.size = 1.2, plevel = 0.99, pop.labels = 'legend',  pt.colors = pop_colors) #Run PCA plot
dev.off()

#
###FST Table
fstallr <- gl.fst.pop(starALLgl)
fstall <- round(fstallr$Fsts , 3)
 # Create FST table (note: 1 is completely dissimilar, 0 is identical)
# Create a matrix from the FST table
fst_matrix <- t(as.matrix(fstall ))

filled_matrix <- fst_matrix
filled_matrix[is.na(filled_matrix)] <- t(filled_matrix)[is.na(filled_matrix)]

# Set row names and column names based on populations
populations <- unique(starALLgl$pop)
fst_df <- data.frame(pop1 = rep(populations, each = length(populations)),
                     pop2 = rep(populations, times = length(populations)),
                     fst = as.vector(filled_matrix))
# Manually set the order of populations on the x and y axes
desired_order <- c("MKW (UK)", "NWC (UK)","ANT (BE)","ORG (AU)","MLV (AU)", "AUK (NZ)",  "UHT (NZ)" ,"PLM (NZ)", "BLN (NZ)",   "CAN (NZ)")
desired_order2 <- c("MKW (UK)", "NWC (UK)","ANT (BE)","ORG (AU)","MLV (AU)", "AUK (NZ)",  "UHT (NZ)" ,"PLM (NZ)", "BLN (NZ)",   "CAN (NZ)")
# Invert the order of 'desired_order'
desired_order3 <- rev(desired_order2)

# Convert 'pop1' and 'pop2' to factor with desired order
fst_df$pop1 <- factor(fst_df$pop1, levels = desired_order3)
fst_df$pop2 <- factor(fst_df$pop2, levels = desired_order)
```

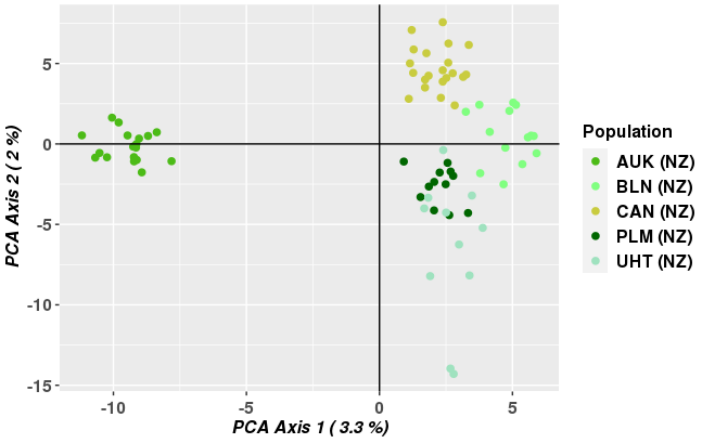
```
# Filter to keep only the lower triangle of the FST matrix
fst_df2 <- fst_df %>% filter(row_number(pop2) <= row_number(pop1))

# Create a ggplot triangular heatmap with FST values as text labels
pdf("Sv10_global_Fst.pdf", width=8.5, height=5.0)
ggplot(fst_df, aes(x = pop1, y = pop2, fill = fst, label = round(fst, 3))) +
  geom_tile() +
  geom_text(color = "black", size = 3) +  # Add FST values as text labels
  scale_fill_gradient(low = "white", high = "#2c64c4") +
  theme_minimal() +
  theme(panel.grid = element_blank(), axis.title = element_blank()) +
  coord_fixed(ratio = 1, xlim = c(0.5, length(desired_order) + 0.5), ylim = c(0.5, length(desired_order) + 0.5)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), axis.text.y = element_text(angle = 0, hjust = 1)) +
  geom_tile(data = subset(fst_df, pop1 == pop2), fill = "white")  # Set diagonal cells to white
dev.off()


#
#Genetic Indices Table
heteroall <- gl.report.heterozygosity(starALLgl, method = "pop")  #includes FIS (inbreeding)
heteroall2 <-  heteroall %>% select(pop, Ho, uHe, FIS)
write.table(heteroall2,"Global_heterozygosity_table.txt",row.names=FALSE,sep="\t", quote = FALSE,col.names=TRUE)
```



POPGEN: NZ

```
module load R-bundle-Bioconductor/3.15-gimkl-2022a-R-4.2.1
R
#Install Required packages (Only needs to be done once)
#load required packages
library(dartR)
library(vcfR)
library(dplyr)
library(tidyr)
library(networkD3)
library(ggplot2)

setwd("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis")

#library(tidyverse) #Won't install. Needed for gl.report.pa (private allele report)
#
```

```
# Read in the filtered VCF file, convert to Gen Light object, and run compliance check
readstarNZvcf <-
read.vcfR("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/final_filtering/starling_noduprel_qual_miss_filt_NZ.recode.vcf")

starNZgl <- vcfR2genlight(readstarNZvcf)
starNZgl <- gl.compliance.check(starNZgl) #Run Compliance check on new Gen Light object (Will be some orange but fixed after)

#

#Bring in Metadata file and Join (match) to genlight object
nameorder <- as.data.frame(indNames(starNZgl))

colnames(nameorder) <- c("id")
metanames <-
read.table("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/final_filtering/Metadata_NZReplicatesSibRemoved.csv",header=T,sep
=",")
metacorrect <- right_join(nameorder, metanames, by = "id")

starNZgl$other$ind.metrics <- metacorrect
pop(starNZgl) <- starNZgl$other$ind.metrics$pop #add the population of each individual to the accessor pop
indNames(starNZgl) <- starNZgl$other$ind.metrics$id #add the individual names to the accessor indNames
pop(starNZgl) # Check pop data is there
names(starNZgl@other$ind.metrics) #list all available metrics for individuals to confirm integration to genlight object

#
#Create dataframe to add Lat/Lon coordinates to Genlight latlon slot
starNZgl@other$latlon <- data.frame(lat=starNZgl@other$ind.metrics$lat,lon=starNZgl@other$ind.metrics$lon)
#Housekeeping on Gen Light
starNZgl <- gl.compliance.check(starNZgl) # ReRun Compliance check on new Gen Light object after adding metadata. SHOULD BE ALL GREEN :)
starNZgl <- gl.recalc.metrics(starNZgl) #Re Calculate metrics
#

## Visual Outputs of dataset populations based on NZ Regions
starNZgl <- gl.reassign.pop(starNZgl, as.pop="pop2") #Toggle pop name to the "pop2' (collapsed regions)
pca<-gl.pcoa(starNZgl,nfactors=5) # Performs Pearson Principal Component analysis (PCA) on SNP data
# Specify populations and corresponding shades
pop_colors <- c("AUK (NZ)" = "#4CBB17",    # Kelly green
                "BLN (NZ)" = "#80ff80",    # Light green
                "CAN (NZ)" = "#C9CC3F",    # Pear
                "PLM (NZ)" = "#006600",    # Another shade of green for PLM (NZ)
                "UHT (NZ)" = "#9FE2BF")    # Seafoam Green
# Use pop_colors in the gl.pcoa.plot function


###PCA
# Use pop_colors in the gl.pcoa.plot function
pdf("Sv10_global_nz.pdf", width=8.5, height=5.0)
gl.pcoa.plot(pca, starNZgl, axis.label.size = 1.2, plevel = 0.99, pop.labels = 'legend', pt.colors = pop_colors) #Run PCA plot
dev.off()
```



Private alleles

```
cd /nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/priv_alleles

module load Stacks/2.65-GCC-11.3.0

META=/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/Metadata_NZ_AU_UK_BE_ReplicatesSibRemoved2.csv
tail -n +2 $META | sed 's/,/\t/g' | cut -f1,3 > ind_pop_global.txt

populations -V /nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/final_filtering/starling_noduprel_qual_miss_filt.recode.vcf -M
 ind_pop_global.txt -t 2 --phylip --out-path ./
```

Population summary statistics (more detail in populations.sumstats_summary.tsv):

AUK (NZ): 16.656 samples per locus; pi: 0.18622; all/variant/polymorphic sites: 19340/19340/14166; private alleles: 16

CAN (NZ): 19.604 samples per locus; pi: 0.19381; all/variant/polymorphic sites: 19340/19340/16403; private alleles: 3

PLM (NZ): 11.198 samples per locus; pi: 0.19488; all/variant/polymorphic sites: 19340/19340/14447; private alleles: 1

UHT (NZ): 10.463 samples per locus; pi: 0.19221; all/variant/polymorphic sites: 19340/19340/13804; private alleles: 0

BLN (NZ): 13.492 samples per locus; pi: 0.19781; all/variant/polymorphic sites: 19340/19340/14960; private alleles: 4

ANT (BE): 14.324 samples per locus; pi: 0.18998; all/variant/polymorphic sites: 19340/19340/14875; private alleles: 4

MLV (AU): 13.774 samples per locus; pi: 0.18559; all/variant/polymorphic sites: 19340/19340/12904; private alleles: 0

MKW (UK): 9.826 samples per locus; pi: 0.19143; all/variant/polymorphic sites: 19340/19340/13578; private alleles: 0

NWC (UK): 10.1 samples per locus; pi: 0.19094; all/variant/polymorphic sites: 19340/19340/13513; private alleles: 0

ORG (AU): 13.31 samples per locus; pi: 0.18957; all/variant/polymorphic sites: 19340/19340/13780; private alleles: 4

Rarefied

```
module load R/4.1.0-gimkl-2020a
R
library(ggplot2)
library(data.table)
library(tidyr)
library(dplyr)



setwd("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/priv_alleles")
install.packages("poppr")



module load R-bundle-Bioconductor/3.15-gimkl-2022a-R-4.2.1
R
#Install Required packages (Only needs to be done once)
#load required packages
library(dartR)
library(vcfR)
library(dplyr)
library(tidyr)
library(networkD3)
library(ggplot2)



setwd("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/priv_alleles")

starALLgl

install.packages("poppr")
library(poppr)
```

```
data(Pinf)

private_alleles(Pinf, alleles ~ Country)
```

Selection Analysis

Example: https://github.com/Elahep/B.tryoni_PopGenomics/tree/main/3-BayPass

```
cd /nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/selection

PLINKDIR=/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/final_filtering/
#

#remove first 2 columns
cut -f 3- $PLINKDIR/starling_noduprel_qual_miss_filt.plink.ped > x.delete
#
META=/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/Metadata_NZ_AU_UK_BE_ReplicatesSibRemoved2.csv
tail -n +2 $META | sed 's/,/\t/g' | awk -F'\t' 'BEGIN {OFS="\t"} {print $3,$1}' | sed 's/ /_/g' > ind_pop_global.txt

#sed -e 's/ANT_(BE)/NATIVE/g' -e 's/AUK_(NZ)/NZ1/g' -e 's/BLN_(NZ)/NZ2/g' -e 's/CAN_(NZ)/NZ2/g' -e 's/MKW_(UK)/NATIVE/g' -e 's/MLV_(AU)/AU1/g' -e 's/NWC_(UK)/NATIVE/g' -
e 's/ORG_(AU)/AU2/g'  -e 's/PLM_(NZ)/NZ2/g'  -e 's/UHT_(NZ)/NZ2/g' ind_pop_global.txt > ind_pop_global_group.txt


paste   ind_pop_global.txt x.delete > starling_noduprel_qual_miss_filt.plink.ped
rm x.delete

#
cp $PLINKDIR/starling_noduprel_qual_miss_filt.plink.map .
cp $PLINKDIR/starling_noduprel_qual_miss_filt.plink.log .
#

# Run the population-based allele frequency calculations.
module load PLINK/1.09b6.16
plink --file starling_noduprel_qual_miss_filt.plink --allow-extra-chr --freq counts --family --out starling_noduprel_qual_miss_filt
# Manipulate file so it has BayPass format, numbers set for PLINK output file, and population number for column count.
tail -n +2 starling_noduprel_qual_miss_filt.frq.strat | awk '{ $9 = $8 - $7 } 1' | awk '{print $7,$9}' | tr "\n" " " | sed 's/ /\n/20; P; D' > starling_noduprel_qual_miss_filt_baypass.txt

#NOT GROUPED: ANT_(BE) AUK_(NZ) BLN_(NZ) CAN_(NZ) MKW_(UK) MLV_(AU) NWC_(UK) ORG_(AU) PLM_(NZ) UHT_(NZ)
awk '{print $1, $2, $9, $10, $13, $14, $11, $12}' starling_noduprel_qual_miss_filt_baypass.txt  > starling_AU1_native.txt
awk '{print $1, $2, $9, $10, $13, $14, $15, $16}' starling_noduprel_qual_miss_filt_baypass.txt  > starling_AU2_native.txt
awk '{print $1, $2, $9, $10, $13, $14, $3, $4}' starling_noduprel_qual_miss_filt_baypass.txt  > starling_NZ1_native.txt
awk '{print $1, $2, $9, $10, $13, $14, $5, $6, $7, $8, $17, $18, $19, $20}' starling_noduprel_qual_miss_filt_baypass.txt  > starling_NZ2_native.txt

nano starling_NAT_INV1.ecotype # 1 1 1 -1
nano starling_NAT_INV4.ecotype # 1 1 1 -1 -1 -1 -1
nano starling_NAT_INVall.ecotype# 1 -1 -1 -1 1 -1 1 -1 -1 -1

# Baypass
# load modules # Created Slurm job for these next 5 lines
module purge
module load BayPass/2.31-intel-2022a

i_baypass -gfile ./starling_noduprel_qual_miss_filt_baypass.txt -contrastfile starling_NAT_INVall.ecotype -efile starling_NAT_INVall.ecotype -outprefix starling_contrast -nthreads 2

i_baypass -gfile ./starling_AU1_native.txt -contrastfile starling_NAT_INV1.ecotype -efile starling_NAT_INV1.ecotype -outprefix starling_contrast_AU1 -nthreads 2
i_baypass -gfile ./starling_AU2_native.txt -contrastfile starling_NAT_INV1.ecotype -efile starling_NAT_INV1.ecotype -outprefix starling_contrast_AU2 -nthreads 2
i_baypass -gfile ./starling_NZ1_native.txt -contrastfile starling_NAT_INV1.ecotype -efile starling_NAT_INV1.ecotype -outprefix starling_contrast_NZ1 -nthreads 2
```

```
i_baypass -gfile ./starling_NZ2_native.txt -contrastfile starling_NAT_INV4.ecotype -efile starling_NAT_INV4.ecotype -outprefix starling_contrast_NZ2 -nthreads 2
```

Simulated thresholds for M_C2

```
module load R-bundle-Bioconductor/3.15-gimkl-2022a-R-4.2.1
R

setwd("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/selection")
source("/nesi/nobackup/uoa02613/kstuart_projects/programs/baypass_public/utils/baypass_utils.R")

library("ape")
library("mvtnorm")

#AU1
omega <- as.matrix(read.table("starling_contrast_AU1_mat_omega.out"))
pi.beta.coef <- read.table("starling_contrast_AU1_summary_beta_params.out", header = TRUE)
bta14.data <- geno2YN("starling_AU1_native.txt")
simu.bta <- simulate.baypass(omega.mat = omega, nsnp = 5000, sample.size = bta14.data$NN, beta.pi = pi.beta.coef$Mean, pi.maf = 0, suffix = "simulated_AU1")

#AU2
omega <- as.matrix(read.table("starling_contrast_AU2_mat_omega.out"))
pi.beta.coef <- read.table("starling_contrast_AU2_summary_beta_params.out", header = TRUE)
bta14.data <- geno2YN("starling_AU2_native.txt")
simu.bta <- simulate.baypass(omega.mat = omega, nsnp = 5000, sample.size = bta14.data$NN, beta.pi = pi.beta.coef$Mean, pi.maf = 0, suffix = "simulated_AU2")

#NZ1
omega <- as.matrix(read.table("starling_contrast_NZ1_mat_omega.out"))
pi.beta.coef <- read.table("starling_contrast_NZ1_summary_beta_params.out", header = TRUE)
bta14.data <- geno2YN("starling_NZ1_native.txt")
simu.bta <- simulate.baypass(omega.mat = omega, nsnp = 5000, sample.size = bta14.data$NN, beta.pi = pi.beta.coef$Mean, pi.maf = 0, suffix = "simulated_NZ1")

#NZ2
omega <- as.matrix(read.table("starling_contrast_NZ2_mat_omega.out"))
pi.beta.coef <- read.table("starling_contrast_NZ2_summary_beta_params.out", header = TRUE)
bta14.data <- geno2YN("starling_NZ2_native.txt")
simu.bta <- simulate.baypass(omega.mat = omega, nsnp = 5000, sample.size = bta14.data$NN, beta.pi = pi.beta.coef$Mean, pi.maf = 0, suffix = "simulated_NZ2")


ALL
omega <- as.matrix(read.table("starling_contrast_mat_omega.out"))
pi.beta.coef <- read.table("starling_contrast_summary_beta_params.out", header = TRUE)
bta14.data <- geno2YN("starling_noduprel_qual_miss_filt_baypass.txt")
simu.bta <- simulate.baypass(omega.mat = omega, nsnp = 5000, sample.size = bta14.data$NN, beta.pi = pi.beta.coef$Mean, pi.maf = 0, suffix = "simulated")
```

Baypass on simulated neutral data

```
module purge
module load BayPass/2.31-intel-2022a

i_baypass -gfile ./G.simulated_AU1 -contrastfile starling_NAT_INV1.ecotype -efile starling_NAT_INV1.ecotype -outprefix simulated_AU1 -nthreads 2
i_baypass -gfile ./G.simulated_AU2 -contrastfile starling_NAT_INV1.ecotype -efile starling_NAT_INV1.ecotype -outprefix simulated_AU2 -nthreads 2
i_baypass -gfile ./G.simulated_NZ1 -contrastfile starling_NAT_INV1.ecotype -efile starling_NAT_INV1.ecotype -outprefix simulated_NZ1 -nthreads 2
i_baypass -gfile ./G.simulated_NZ2 -contrastfile starling_NAT_INV4.ecotype -efile starling_NAT_INV4.ecotype -outprefix simulated_NZ2 -nthreads 2

i_baypass -gfile ./G.simulated -contrastfile starling_NAT_INVall.ecotype -efile starling_NAT_INVall.ecotype -outprefix simulated -nthreads 2
```

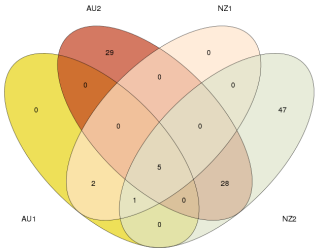
plots

```
cd /nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/selection
VCF=/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/final_filtering/starling_noduprel_qual_miss_filt.recode.vcf
cat $VCF | grep -v "#" | cut -f3 > scaffold_list.txt

cd /nesi/nobackup/uoa02613/kstuart_projects/programs/
git clone https://forgemia.inra.fr/mathieu.gautier/baypass_public.git

module load R/4.1.0-gimkl-2020a
R
library(ggplot2)
```

```
library(data.table)
library(tidyr)
library(dplyr)



setwd("/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/selection")
library("ape")
library("corrplot")
library(qvalue)
library(dplyr)
library(ggplot2)
```

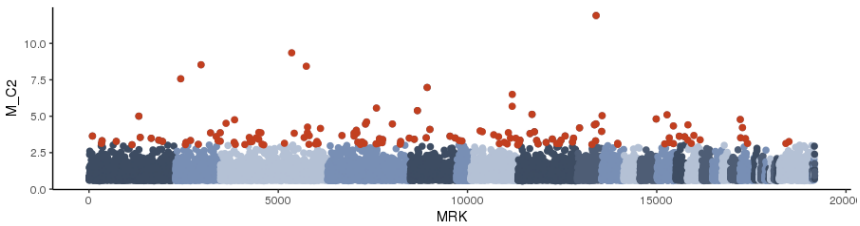
```
#CHROMS
scaffolds = read.table("scaffold_list.txt")
scaffolds$Chrom <- sub("\\..*$", "", scaffolds$V1)

#COLOURS
blue_palette <- colorRampPalette(c("#B4C1D5", "#4E5D75"))(length(unique(scaffolds$Chrom)))

colors <- c("#3D4C62", "#B4C1D5", "#4E5D75", "#788FB5")
blue_palette  <- rep(colors, length.out = (length(unique(scaffolds$Chrom))))

#NATIVE TO AU1
##import the BayPass output file and scaffold list
AU1.C2=read.table("starling_contrast_AU1_summary_contrast.out",h=T)
AU1.C2 = as.data.frame(cbind(AU1.C2, scaffolds))
AU1.C2 <- AU1.C2 %>%  mutate(snp_order = row_number())

##calculate threshold and note ones above
AU1.C2.sim <- read.table("simulated_AU1_summary_contrast.out", header = T)
AU1.C2.thresh <- quantile(AU1.C2.sim$M_C2 ,probs = 0.99)

selected_SNPs = AU1.C2[AU1.C2$M_C2 > AU1.C2.thresh, ]
str(selected_SNPs )
write.table(selected_SNPs$V1,"Outliers_NativeAU1.txt", sep = "\t", row.names=FALSE,  quote = FALSE,col.names=TRUE)

##Manhattan plot
pdf("Sv10_baypass_AU1.pdf", width=8.5, height=1.5)
ggplot(AU1.C2, aes(x=MRK, y = M_C2, color = Chrom)) +
  geom_point(show.legend = FALSE, alpha = 1, size = 2) +
    geom_point(selected_SNPs, mapping = aes(x=MRK, y = M_C2), color = "#C23F1F", size = 2) +
  theme_classic() + scale_color_manual(values = blue_palette )
dev.off()


#NATIVE TO AU2
##import the BayPass output file and scaffold list
AU2.C2=read.table("starling_contrast_AU2_summary_contrast.out",h=T)
AU2.C2 = as.data.frame(cbind(AU2.C2, scaffolds))
AU2.C2 <- AU2.C2 %>%  mutate(snp_order = row_number())

##calculate threshold and note ones above
AU2.C2.sim <- read.table("simulated_AU2_summary_contrast.out", header = T)
AU2.C2.thresh <- quantile(AU2.C2.sim$M_C2 ,probs = 0.99)

selected_SNPs = AU2.C2[AU2.C2$M_C2 > AU2.C2.thresh, ]
str(selected_SNPs )
write.table(selected_SNPs$V1,"Outliers_NativeAU2.txt", sep = "\t", row.names=FALSE,  quote = FALSE,col.names=TRUE)

##Manhattan plot
pdf("Sv10_baypass_AU2.pdf", width=8.5, height=1.5)
ggplot(AU2.C2, aes(x=MRK, y = M_C2, color = Chrom)) +
  geom_point(show.legend = FALSE, alpha = 1, size = 2) +
    geom_point(selected_SNPs, mapping = aes(x=MRK, y = M_C2), color = "#C23F1F", size = 2) +
  theme_classic() + scale_color_manual(values = blue_palette )
dev.off()


#NATIVE TO NZ1
##import the BayPass output file and scaffold list
NZ1.C2=read.table("starling_contrast_NZ1_summary_contrast.out",h=T)
NZ1.C2 = as.data.frame(cbind(NZ1.C2, scaffolds))
NZ1.C2 <- NZ1.C2 %>%  mutate(snp_order = row_number())
```

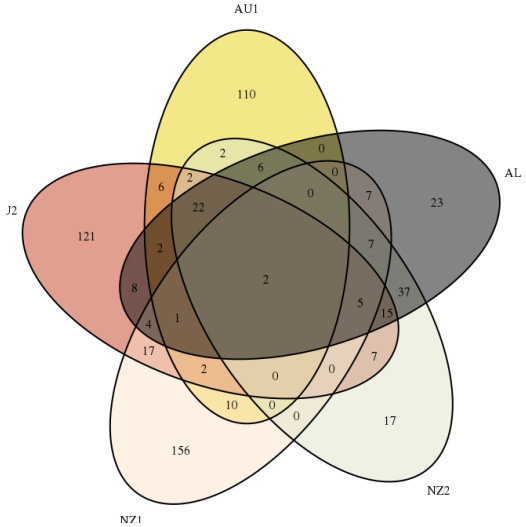
```
##calculate threshold and note ones above
NZ1.C2.sim <- read.table("simulated_NZ1_summary_contrast.out", header = T)
NZ1.C2.thresh <- quantile(NZ1.C2.sim$M_C2 ,probs = 0.99)

selected_SNPs = NZ1.C2[NZ1.C2$M_C2 > NZ1.C2.thresh, ]
str(selected_SNPs )
write.table(selected_SNPs$V1,"Outliers_NativeNZ1.txt", sep = "\t", row.names=FALSE,  quote = FALSE,col.names=TRUE)

##Manhattan plot
pdf("Sv10_baypass_NZ1.pdf", width=8.5, height=1.5)
ggplot(NZ1.C2, aes(x=MRK, y = M_C2, color = Chrom)) +
  geom_point(show.legend = FALSE, alpha = 1, size = 2) +
    geom_point(selected_SNPs, mapping = aes(x=MRK, y = M_C2), color = "#C23F1F", size = 2) +
  theme_classic() + scale_color_manual(values = blue_palette )
dev.off()


#NATIVE TO NZ2
##import the BayPass output file and scaffold list
NZ2.C2=read.table("starling_contrast_NZ2_summary_contrast.out",h=T)
NZ2.C2 = as.data.frame(cbind(NZ2.C2, scaffolds))
NZ2.C2 <- NZ2.C2 %>%  mutate(snp_order = row_number())

##calculate threshold and note ones above
NZ2.C2.sim <- read.table("simulated_NZ2_summary_contrast.out", header = T)
NZ2.C2.thresh <- quantile(NZ2.C2.sim$M_C2 ,probs = 0.99)

selected_SNPs = NZ2.C2[NZ2.C2$M_C2 > NZ2.C2.thresh, ]
str(selected_SNPs )
write.table(selected_SNPs$V1,"Outliers_NativeNZ2.txt", sep = "\t", row.names=FALSE,  quote = FALSE,col.names=TRUE)

##Manhattan plot
pdf("Sv10_baypass_NZ2.pdf", width=8.5, height=1.5)
ggplot(NZ2.C2, aes(x=MRK, y = M_C2, color = Chrom)) +
  geom_point(show.legend = FALSE, alpha = 1, size = 2) +
    geom_point(selected_SNPs, mapping = aes(x=MRK, y = M_C2), color = "#C23F1F", size = 2) +
  theme_classic() + scale_color_manual(values = blue_palette )
dev.off()

## we will also make a Venn diagram to check the common SNPs between the three comparisons
AU1 = read.table("./Outliers_NativeAU1.txt", header=T)
AU2 = read.table("./Outliers_NativeAU2.txt", header=T)
NZ1 = read.table("./Outliers_NativeNZ1.txt", header=T)
NZ2 = read.table("./Outliers_NativeNZ2.txt", header=T)


library(VennDiagram)
# Define custom fill colors
fill_colors <- c("#C23F1F","#E7D210",  "#93AF76", "#DEE3CA")

# Create a list of sets
x <- list(  AU1 = AU1$x,  AU2 = AU2$x,  NZ1 = NZ1$x,  NZ2 = NZ2$x)

# Create the Venn diagram with custom fill colors
venn.plot <- venn.diagram(  x,
  category.names = c("AU1", "AU2", "NZ1", "NZ2"),
  fill = fill_colors,
  filename = NULL
)

pdf("Sv10_baypass_venn.pdf", width=8.5, height=8)
# Plot the Venn diagram
grid.draw(venn.plot)
dev.off()
```



#NATIVE TO ALL

```
##import the BayPass output file and scaffold list
ALL.C2=read.table("starling_contrast_summary_contrast.out",h=T)
ALL.C2 = as.data.frame(cbind(ALL.C2, scaffolds))
ALL.C2 <- ALL.C2 %>% mutate(snp_order = row_number())

##calculate threshold and note ones above
ALL.C2.sim <- read.table("simulated_summary_contrast.out", header = T)
ALL.C2.thresh <- quantile(ALL.C2.sim$M_C2 ,probs = 0.99)

selected_SNPs = ALL.C2[ALL.C2$M_C2 > ALL.C2.thresh, ]
str(selected_SNPs )
write.table(selected_SNPs$V1,"Outliers_NativeALL.txt", sep = "\t", row.names=FALSE,  quote = FALSE,col.names=TRUE)

##Manhattan plot
pdf("Sv10_baypass_ALL.pdf", width=8.5, height=1.5)
ggplot(ALL.C2, aes(x=MRK, y = M_C2, color = Chrom)) +
  geom_point(show.legend = FALSE, alpha = 1, size = 2) +
    geom_point(selected_SNPs, mapping = aes(x=MRK, y = M_C2), color = "#C23F1F", size = 2) +
  theme_classic() + scale_color_manual(values = blue_palette )
dev.off()

ALL = read.table("./Outliers_NativeALL.txt", header=T)

## we will also make a Venn diagram to check the common SNPs between the three comparisons
library(VennDiagram)
# Define custom fill colors
fill_colors <- c("#C23F1F","#E7D210",  "#93AF76", "#DEE3CA", "#090909")

# Create a list of sets
x <- list(  AU1 = AU1$x,  AU2 = AU2$x,  NZ1 = NZ1$x,  NZ2 = NZ2$x,  ALL = ALL$x)

# Create the Venn diagram with custom fill colors
venn.plot <- venn.diagram(  x,
  category.names = c("AU1", "AU2", "NZ1", "NZ2", "ALL"),
  fill = fill_colors,
  filename = NULL
)

pdf("Sv10_baypass_venn_ALL.pdf", width=8.5, height=8)
# Plot the Venn diagram
grid.draw(venn.plot)
dev.off()
```





GENES

```
cd /nesi/nobackup/uoa02613/kstuart_projects/At4_MynaStarling/data/resources/genomes
sed 's/starling/SV_vAU_seq/g' Svulgaris.all.renamed.func.protdom.gff > Svulgaris_vAU_1.0.gff

cd /nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/analysis/popgen/kat_reanalysis/selection

cat  Outliers_NativeAU1.txt  Outliers_NativeAU2.txt  Outliers_NativeNZ1.txt  Outliers_NativeNZ2.txt | sort | uniq -d > Outliers_Nativecombined.txt
cat  Outliers_NativeALL.txt  | sort | uniq -d > Outliers_Nativecombined.txt
sed 's/\./\t/g' Outliers_NativeALL.txt | awk '{print $1"\t"$2-10000"\t"$2+10000}' > Outliers_NativeALL.bed

SNP=/nesi/nobackup/uoa02613/kstuart_projects/Sv10_NZstarlings/data/processing_rawdata/BCFtools/final_filtering/starling_noduprel_qual_miss_filt.recode.vcf

module load VCFtools/0.1.15-GCC-9.2.0-Perl-5.30.1

vcftools --vcf $SNP --snps  Outliers_NativeALL.txt   --recode --recode-INFO-all  --out Outliers_Nativecombined

module load BEDTools/2.30.0-GCC-11.3.0

GFF=/nesi/nobackup/uoa02613/kstuart_projects/At4_MynaStarling/data/resources/genomes/Svulgaris_vAU_1.0.gff

awk '$3=="gene"' $GFF > Svulgaris_vAU_1.0_GENES.gff

bedtools intersect -b Outliers_Nativecombined.recode.vcf -a  Svulgaris_vAU_1.0_GENES.gff > overlapped_genes.txt
bedtools intersect -b  Outliers_NativeALL.bed -a  Svulgaris_vAU_1.0_GENES.gff > overlapped_genes.txt
bedtools intersect -b $SNP -a  Svulgaris_vAU_1.0_GENES.gff > overlapped_genes_ALL.txt

sed -nr 's/.*Similar to +([^ ]+) .*/\1/p' overlapped_genes.txt | sed 's|[:,]||g' > overlapped_genelist.txt
sed -nr 's/.*Similar to +([^ ]+) .*/\1/p' overlapped_genes_ALL.txt | sed 's|[:,]||g' > overlapped_ALLgenelist.txt

sed 's/Ontology_term=/\t/g' overlapped_genes.txt | cut -f 10 | sed -e 's/;/\n/g' -e 's/,/\n/g' | awk '/^GO/'  > overlapped_genelist_GO.txt


module purge
module load R-bundle-Bioconductor/3.17-gimkl-2022a-R-4.3.1
R
setwd("/nesi/nobackup/uoa02613/kstuart_projects/At4_MynaStarling/data/resources/genomes")
library("topGO")
```