SeaGen_RDA_GenEnv

Amy Zyck

3/21/2021

Seascape Redundancy Analysis

This code follows that documented by Tom Jenkins.

Prepare genetic data for redundancy analysis.

Notes before execution:

1. Make sure all required R packages are installed.
2. Set working directory to the location of this R script.

```r
# Load packages
library(adegenet)
```

```
## Loading required package: ade4

## Registered S3 method overwritten by 'spdep':
##   method   from
##   plot.mst ape

##
##    /// adegenet 2.1.3 is loaded ////////////
##
##    > overview: '?adegenet'
##    > tutorials/doc/questions: 'adegenetWeb()'
##    > bug reports/feature requests: adegenetIssues()
```

```r
library(poppr)
```

```
## Registered S3 method overwritten by 'pegas':
##   method      from
##   print.amova ade4

## This is poppr version 2.8.5. To get started, type package?poppr
## OMP parallel support: available
```

```r
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
```

```r
library(reshape2)
library(ggplot2)
library(vcfR)
```

```
##
##    *****       ***   vcfR   ***       *****
##    This is vcfR 1.12.0
##      browseVignettes('vcfR') # Documentation
##      citation('vcfR') # Citation
##    *****       *****      *****       *****
```

Outlier SNPs

Making files

Make genind object

`outlierlocihap.recode.vcf` contains outlier SNPs (putatively under selection) for populations PVD, GB, BIS, and NAR called from reads aligned to the Eastern Oyster haplotig masked genome. Steps for generating this VCF file are located in EecSeq_Cvirginica_dDocent.md and EecSeq_Cvirginica_Filtering.md. Population NIN was removed from the VCF file following steps in EecSeq_Cvirginica_OutlierDetection.md.

`strata` contains population, environmental, and library information for each sample - can be accessed here.

```r
my_vcf <- read.vcfR("outlierlocihap.recode.vcf")
```

```
## Scanning file to determine attributes.
## File attributes:
##   meta lines: 77
##   header_line: 78
##   variant count: 841
##   column count: 49
##
Meta line 77 read in.
## All meta lines processed.
## gt matrix initialized.
## Character matrix gt created.
##   Character matrix gt rows: 841
##   Character matrix gt cols: 49
##   skip: 0
##   nrows: 841
##   row_num: 0
##
Processed variant: 841
## All variants processed
```

```r
strata <- read.table("strata", header=TRUE)
```

```r
rad.filt <- vcfR2genind(my_vcf, strata = strata, pop = c(rep("BIS", 10),rep("GB", 10),rep("NAR", 10), r
```

```r
# Explore data
rad.filt
```

```
## /// GENIND OBJECT /////////
##
##  // 40 individuals; 841 loci; 1,681 alleles; size: 809 Kb
##
##  // Basic content
##    @tab:  40 x 1681 matrix of allele counts
##    @loc.n.all: number of alleles per locus (range: 1-2)
##    @loc.fac: locus factor for the 1681 columns of @tab
##    @all.names: list of allele names for each locus
##    @ploidy: ploidy of each individual  (range: 2-2)
##    @type:  codom
##    @call: adegenet::df2genind(X = t(x), sep = sep, pop = ..2, strata = ..1)
##
##  // Optional content
##    @pop: population of each individual (group size range: 10-10)
##    @strata: a data frame with 12 columns ( Individual, Population, Latitude, Longitude, Distance, SE
```

```r
nLoc(rad.filt) # number of loci
```

```
## [1] 841
```

```r
nPop(rad.filt) # number of sites
```

```
## [1] 4
```

```r
nInd(rad.filt) # number of individuals
```

```
## [1] 40
```

```r
summary(rad.filt$pop) # sample size
```

```
## BIS  GB NAR PVD
##  10  10  10  10
```

```r
# Calculate allele frequencies for each site
allele_freqs = data.frame(rraf(rad.filt, by_pop = TRUE, correction = FALSE), check.names = FALSE)
```

```r
# Keep only the first of the two alleles for each SNP (since p=1-q).
allele_freqs = allele_freqs[, seq(1, dim(allele_freqs)[2], 2)]
```

```r
# Export allele frequencies
write.csv(allele_freqs, file = "all_allele_freqs.csv", row.names = TRUE)
```

Calculate minor allele frequencies

```r
# Separate genind object by site
site_list = seppop(rad.filt)
names(site_list)
```

```
## [1] "BIS" "GB"  "NAR" "PVD"
```

```r
# Calculate the minor allele frequency for each site
maf_list = lapply(site_list, FUN = minorAllele)
```

```r
# Convert list to dataframe
maf = as.data.frame(maf_list) %>% t() %>% as.data.frame()
```

```r
# Export minor allele frequencies
write.csv(maf, file = "minor_allele_freqs.csv", row.names = TRUE)
```

Visualise allele frequencies

```r
# Add site labels
allele_freqs$site = rownames(allele_freqs)
```

```r
# Function to add regional labels to dataframe
addregion = function(x){
  # If pop label is present function will output the region
  if(x=="BIS") y = " Bissel Cove "
  if(x=="GB") y = " Greenwich Bay "
  if(x=="NAR") y = " Narrow River "
  if(x=="PVD") y = " Bold Point Park "
  return(y)
}
```

```r
# Add regional labels
allele_freqs$region = sapply(rownames(allele_freqs), addregion)
```

```r
# Convert dataframe to long format
allele_freqs.long = melt(allele_freqs, id.vars=c("site","region"))
```

```r
# Define order of facets using the levels argument in factor
unique(allele_freqs.long$site)
```

```
## [1] "BIS" "GB"  "NAR" "PVD"
```

```r
site_order =  c("BIS","GB","NAR","PVD")
allele_freqs.long$site_ord = factor(allele_freqs.long$site, levels = site_order)
```

```r
# Define region order
region_order = c(" Bissel Cove "," Greenwich Bay "," Narrow River ", " Bold Point Park ")
allele_freqs.long$region = factor(allele_freqs.long$region, levels = region_order)
```

```r
# Create colour scheme
# blue=#377EB8, green=#7FC97F, orange=#FDB462, red=#E31A1C
col_scheme = c("#7FC97F","#377EB8","#FDB462","#E31A1C")
```

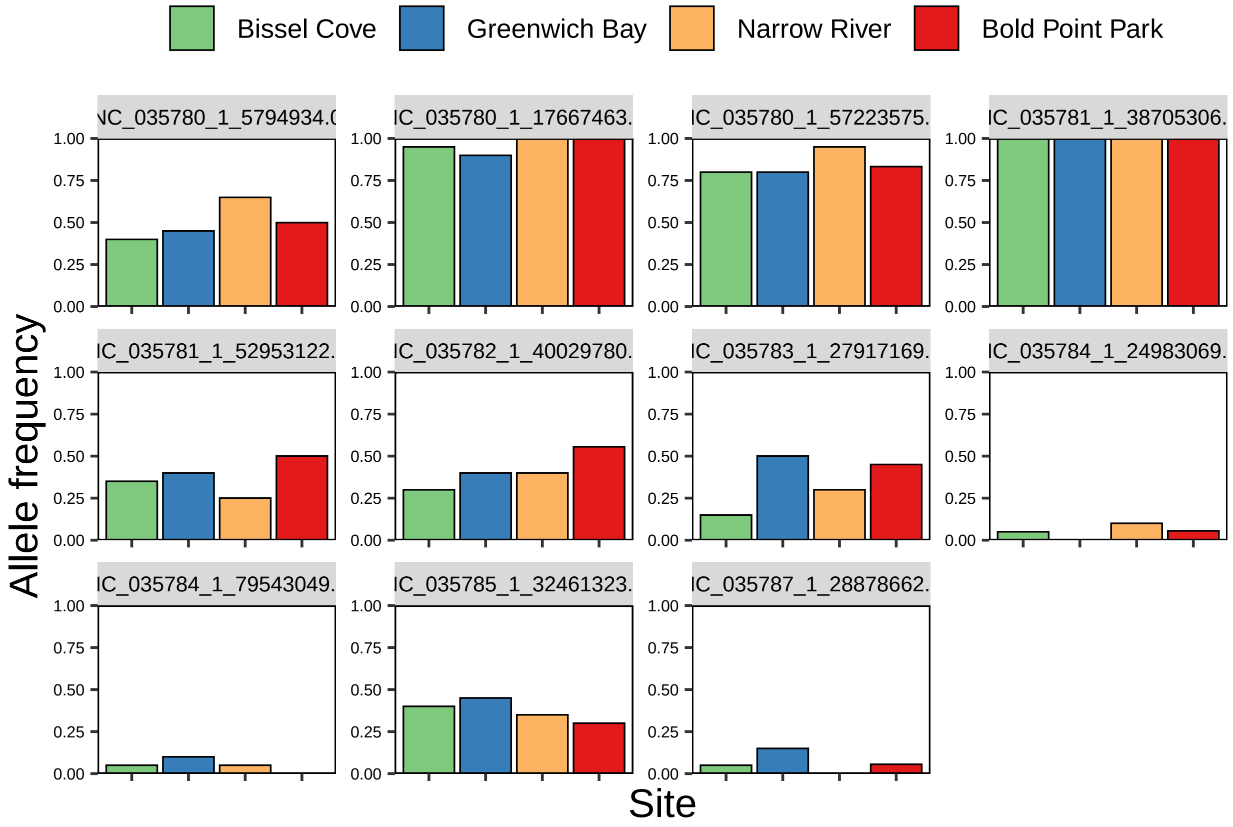
A subset of the putatively outlier loci was selected, spanning all 4 outlier detection programs.

```r
# Vector of outlier SNP loci to subset
desired_loci = c("NC_035780.1_5794934","NC_035780.1_17667463","NC_035780.1_57223575","NC_035781.1_38705
desired_loci_ID = sapply(paste(desired_loci, "..", sep = ""),
                         grep,
                         levels(allele_freqs.long$variable),
                         value = TRUE) %>% as.vector()
```

```r
# Subset dataset to plot desired SNP loci
allele_freqs.sub = allele_freqs.long %>% filter(variable %in% desired_loci_ID)
```

```r
# ggplot2 theme
ggtheme = theme(
  axis.text.x = element_blank(),
  axis.text.y = element_text(colour="black", size=6),
  axis.title = element_text(colour="black", size=15),
  panel.background = element_rect(fill="white"),
  panel.grid.minor = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_rect(colour="black", fill=NA, size=0.5),
  plot.title = element_text(hjust = 0.5, size=18),
  legend.title = element_blank(),
  legend.text = element_text(size=10),
  legend.position = "top",
  legend.justification = "centre",
  # facet labels
  strip.text = element_text(colour="black", size=8)
)
```

```r
# Plot barplot
ggplot(data = allele_freqs.sub, aes(x = site_ord, y = value, fill = region))+
  geom_bar(stat = "identity", colour = "black", size = 0.3)+
  facet_wrap(~variable, scales = "free")+
  scale_y_continuous(limits = c(0,1), expand = c(0,0))+
  scale_fill_manual(values = col_scheme)+
  ylab("Allele frequency")+
  xlab("Site")+
  ggtheme
```



```
ggsave("allele_freq.png", width=10, height=8, dpi=300)
ggsave("allele_freq.pdf", width=10, height=8)
```

Prepare environmental data for redundancy analysis.

Environmental variables:

- Distance from sewage effluent source (km)
- Sewage Effluent (PW stats)
- Mean temperature (deg C)
- Mean Salinity (psu)
- Mean pH
- Mean Chlorophyll-a (ug/L)
- Mean Dissolved Oxygen (mg/L)

```
# All environmental data was previously saved in strata file
strata_pop <- read.table("strata_pop", header=TRUE)
strata_pop
```

```
##   Population Latitude Longitude    Sewage Temperature Salinity  pH Chlorophylla
## 1        BIS   41.545   -71.431  8.824636          23       30 7.9          4.9
## 2         GB   41.654   -71.445 14.596049          24       28 7.4         18.8
## 3        NAR   41.505   -71.453  2.027484          25       18 7.6          4.6
## 4        PVD   41.816   -71.391 59.860038          23       25 7.4          8.1
##    DO
## 1 8.2
## 2 5.7
## 3 7.0
## 4 4.9
```

```
# Export data as a csv file
write.csv(strata_pop, file="environmental_data.csv", row.names = FALSE)
```

I also prepared spatial data for the redundancy analysis which is documented here.

Allele frequency, environmental, and spatial csv files are saved to your working directory and must be imported into the Rscript to run the redundancy analysis. Documentation of the RDA can be accessed here.