Supplemental Information for:

Stepping up to genome scan allows stock identification on the worldwide distributed blue shark *Prionace glauca*

Natacha Nikolic§, Floriaan Devloo-Delva§, Diane Bailleul, Ekaterina Noskova, Clément Rougeux, Chrystelle Delord, Philippe Borsa, Cathy Liautard-Haag, Mohamad Hassan, Amandine Marie, Pierre Feutry, Peter Grewe, Campbell Davies, Jessica Farley, Daniel Fernando, Sebastian Biton-Porsmoguer, François Poisson, Denham Parker, Agostino Leone, Jorden Aulich, Matt Lansdell, Francis Marsac, Sophie Arnaud-Haond§

§*Authors produced this Rmarkdown*

April, 2022

Contents

Load packages

```
library(radiator)
library(assigner)
library(strataG)
library(adegenet)
library(stockR)
library(dartR)
library(dplyr)
library(ggplot2)
library(hierfstat)
library(OutFLANK)
library(pcadapt)
library(plyr)
library(qvalue)
library(rnaturalearth)
library(ggspatial)
library(pegas)
library(poppr)
library(diveRsity)
library(grDevices)
library(MASS)
library(wesanderson)
source("Rdata/Filtering funtions.R")

colours8 <- c("#3B9AB2", "#78B7C5", "#046C9A","#EBCC2A","#9986A5",
              "#E6A0C4","#FD6467","#F21A00")
names(colours8) <- c("NATL", "NEATL", "MED", "SAF", "WIO", "NIO", "EIO", "SWPAC")

colours12 <- c("#81D4FA", "#26C6DA","#046C9A", "#0D47A1", "#FFA000", "#FDD835","#9986A5",
              "#E6A0C4","#BA68C8", "#FD6467", "#F21A00", "#B71C1C")
names(colours12) <- c("NATL", "NEATL", "MED1", "MED2", "SAF1", "SAF2", "WIO", "NIO",
                      "EIO", "SWPAC1", "SWPAC2", "SWPAC3")

colours6 <- c("#046C9A","#F21A00","#EBCC2A","#9986A5","#3B9AB2","#FD6467")
shortnames <- c("NATL", "NEATL", "MED1", "MED2", "SAF1", "SAF2", "WIO", "NIO", "EIO",
                "SWPAC1", "SWPAC2", "SWPAC3", "NPAC")
shortnames2 <- c("NATL", "NEATL", "MED", "SAF", "WIO", "NIO", "EIO", "SWPAC")
```

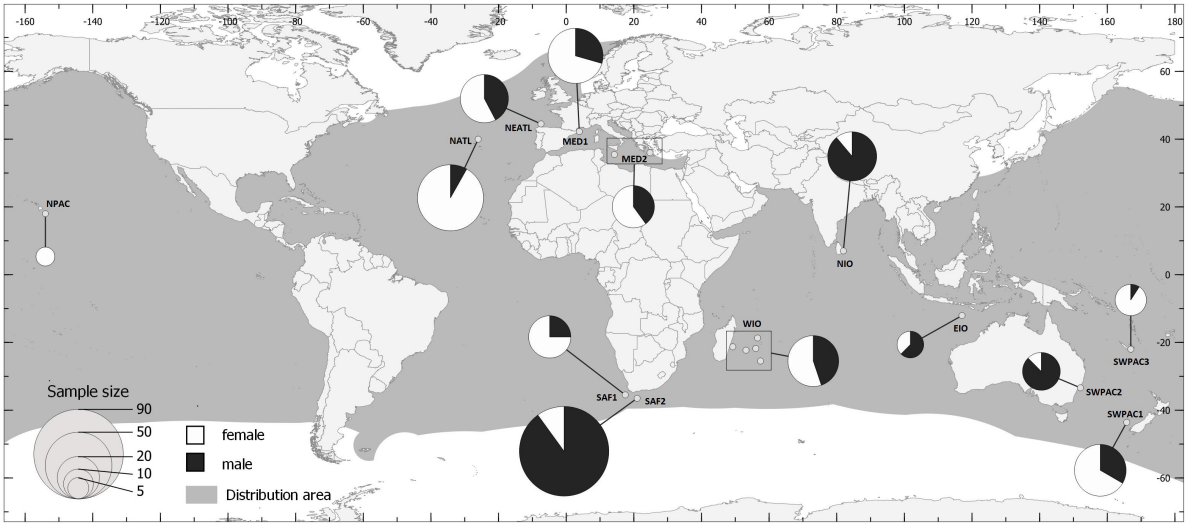
Sample summary

376 blue shark samples were collected between 2009 and 2018; 364 samples passed library construction.

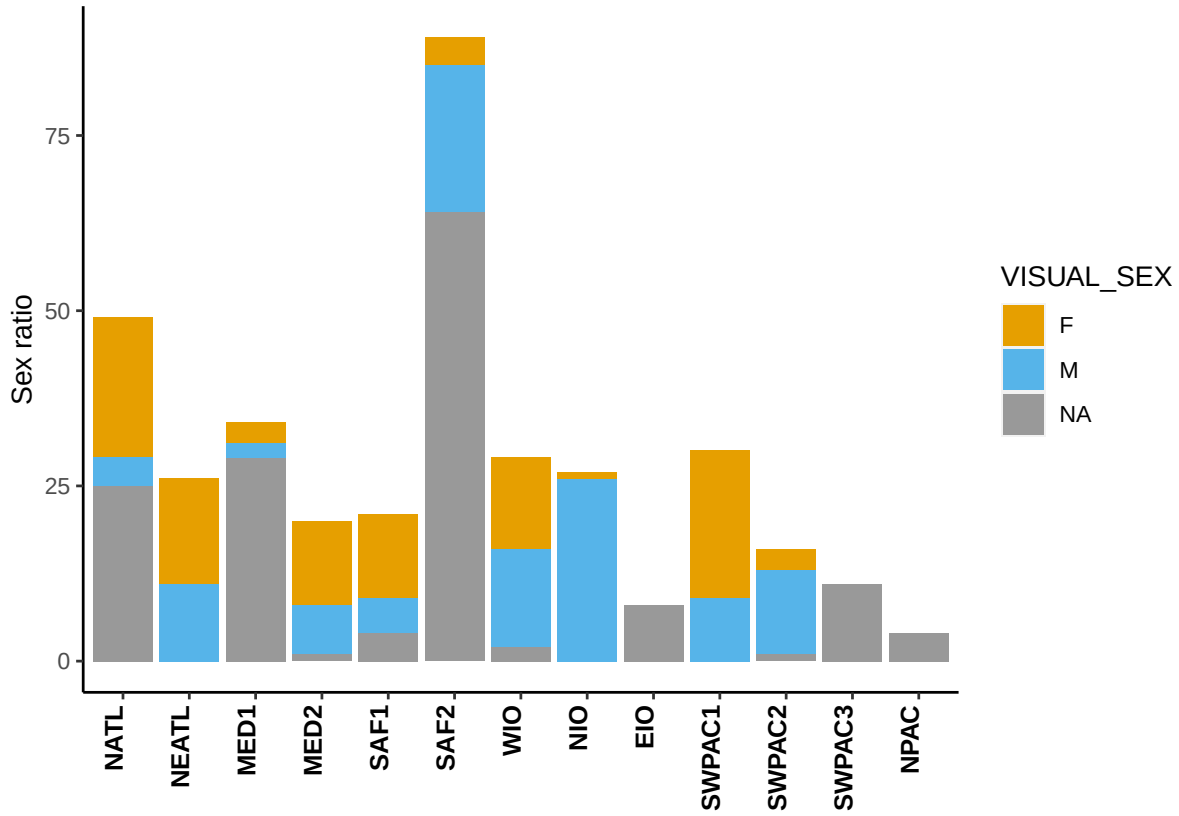
Metadata

```
strata <- "infiles/BLUE_SHARK_META_GENETIC_SEX_20210824_V2.csv"
strata <- readr::read_csv(strata)
pop.levels1 <- c("NATL", "NEATL", "MED1", "MED2", "SAF1", "SAF2", "WIO", "NIO", "EIO",
                 "SWPAC1", "SWPAC2", "SWPAC3", "NPAC")
strata$STRATA <- factor(strata$STRATA, levels = pop.levels1)
strata <- strata[order(strata$STRATA, strata$GENETIC_SEX),]
```

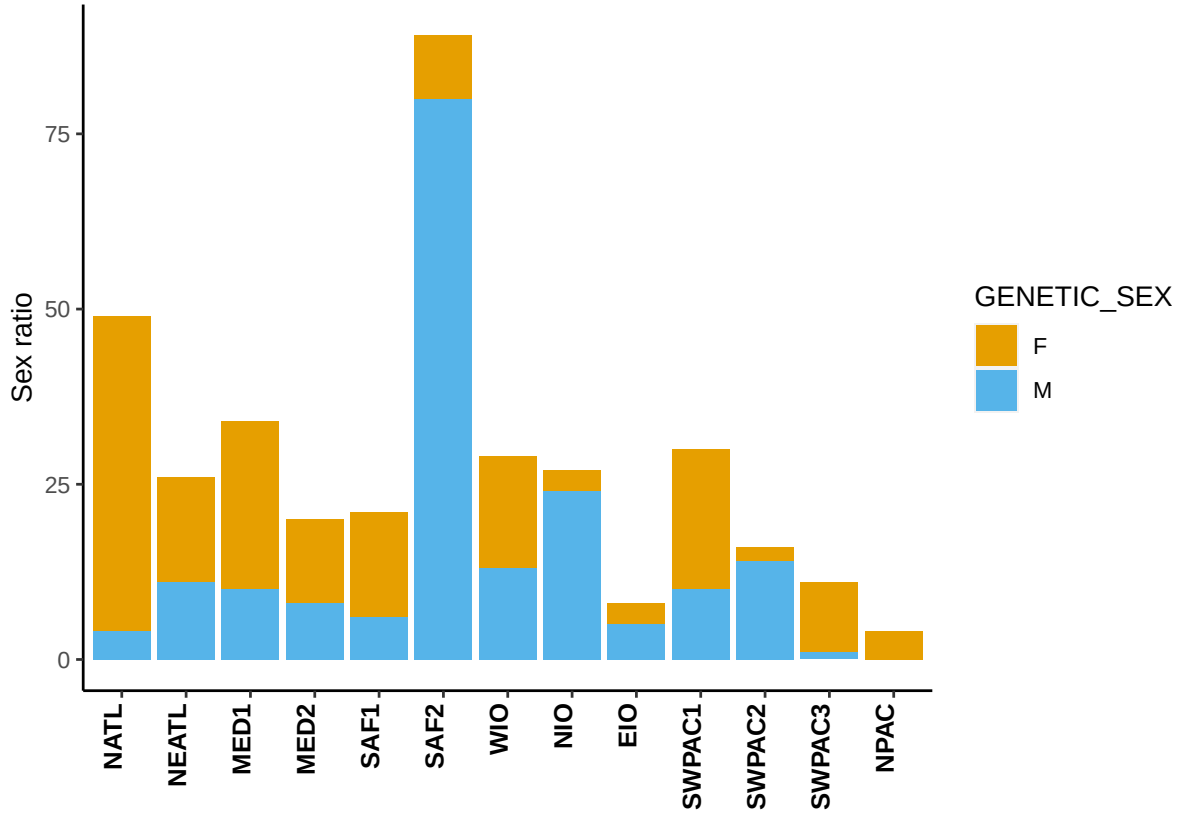
Sample map



By visual sex ID



By genetic sex ID

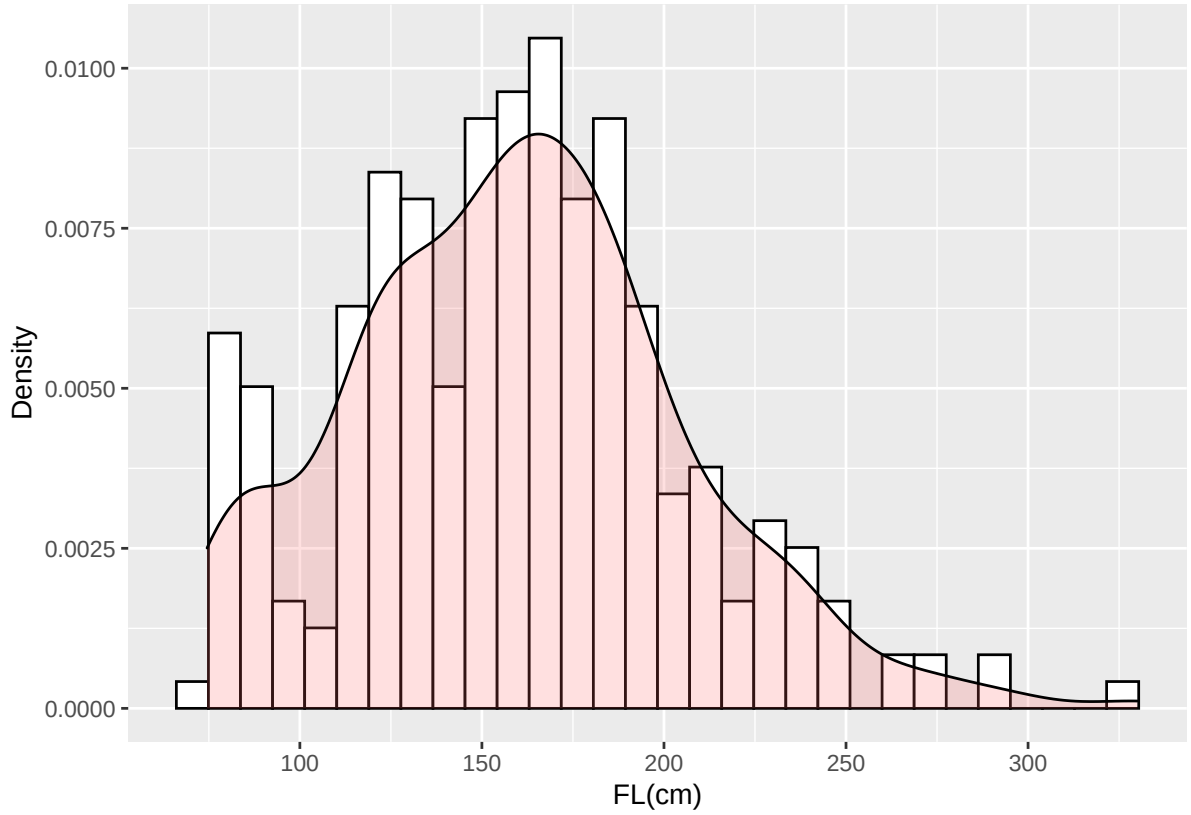


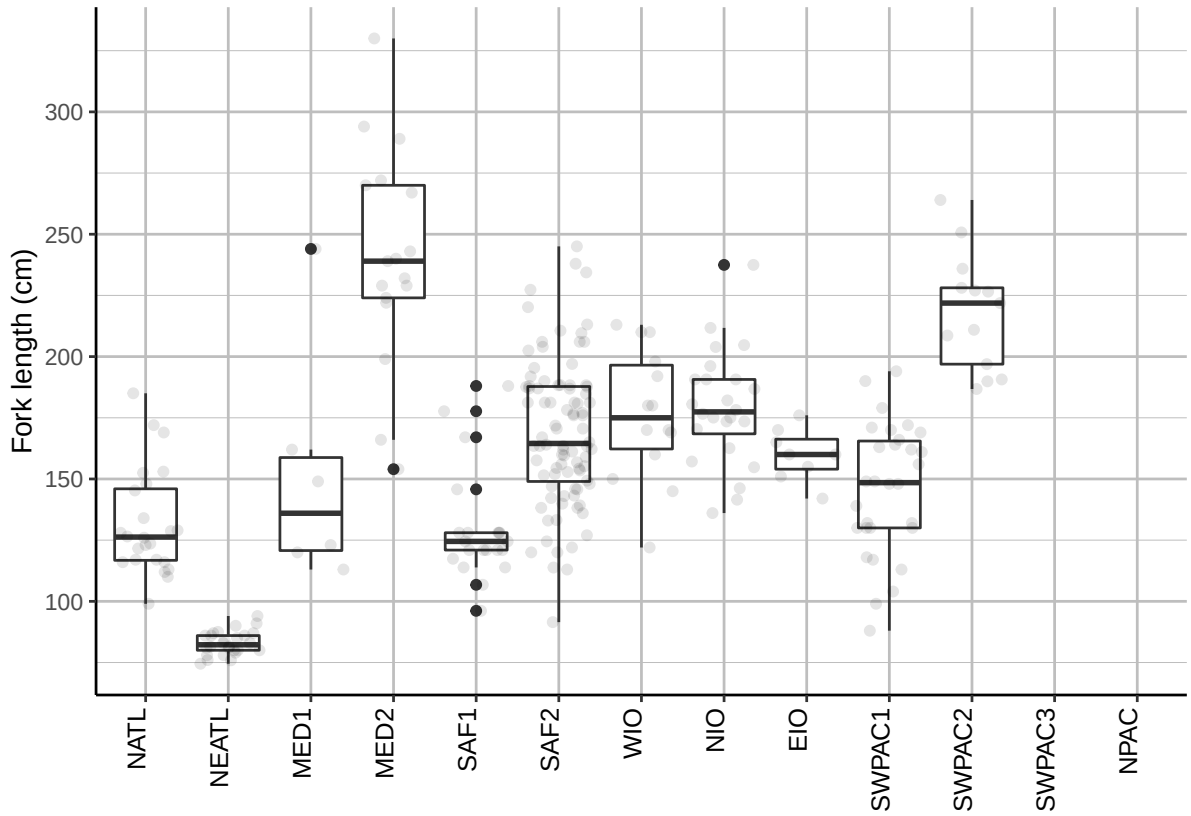
By length

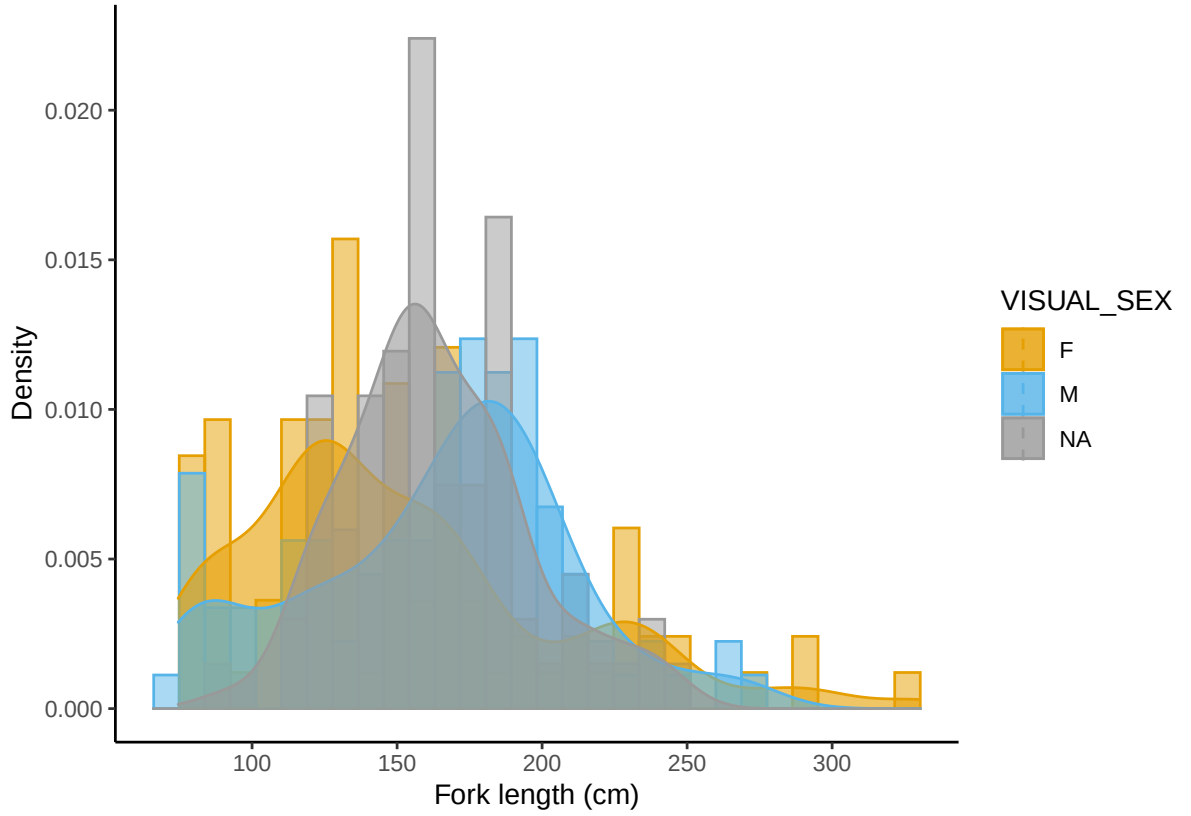
The total length (TL) and interdorsal length (IDS) were converted to fork length (FL) following Cramer et al. (1997).

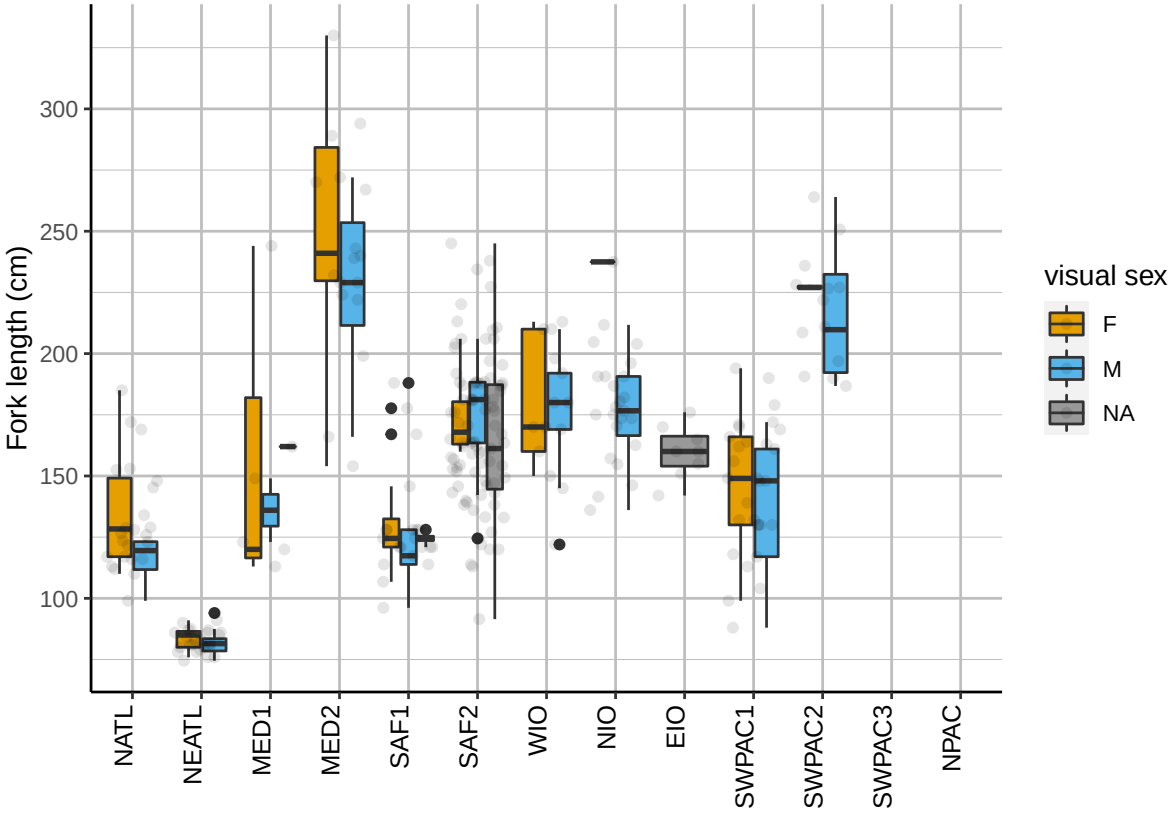
`TL = (IDS + 4.24 )/0.22` `FL = 11.27 + 0.78 TL`

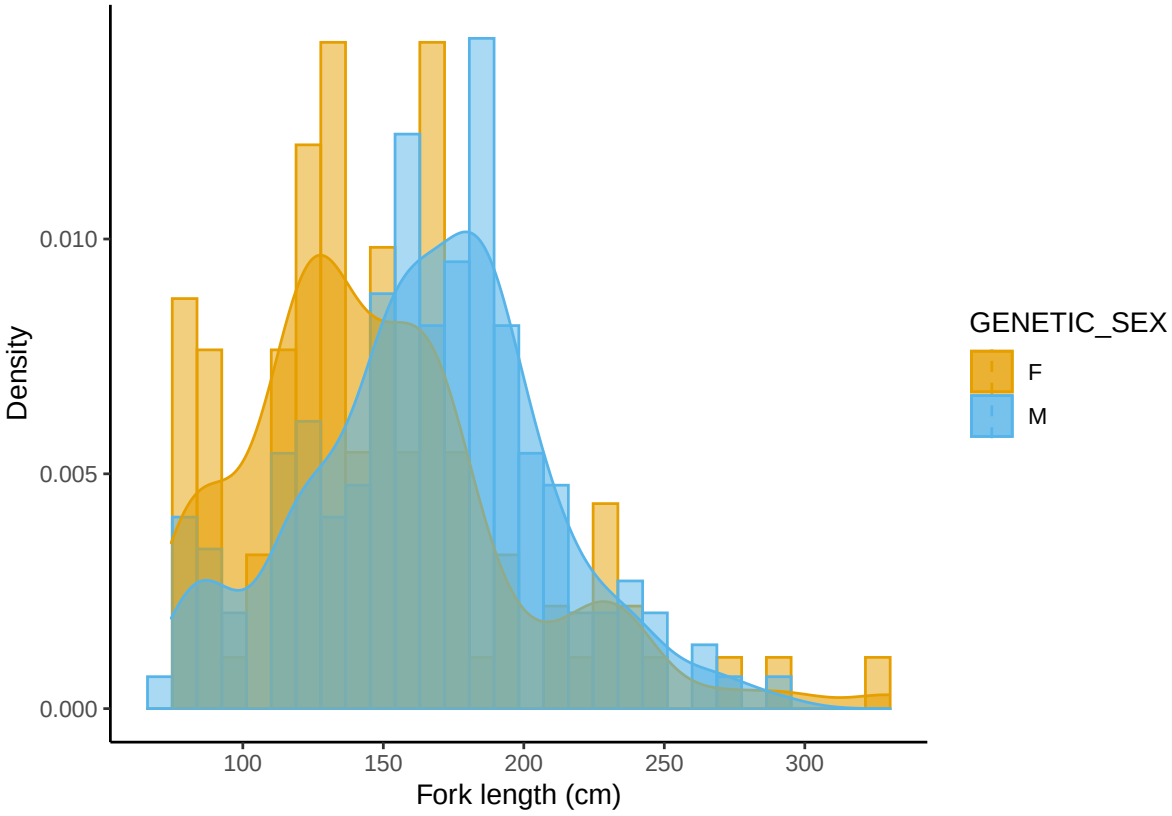
The converted length is only approximate, since growth may vary between geographical location and sex. However, this should suffice for our purposes.

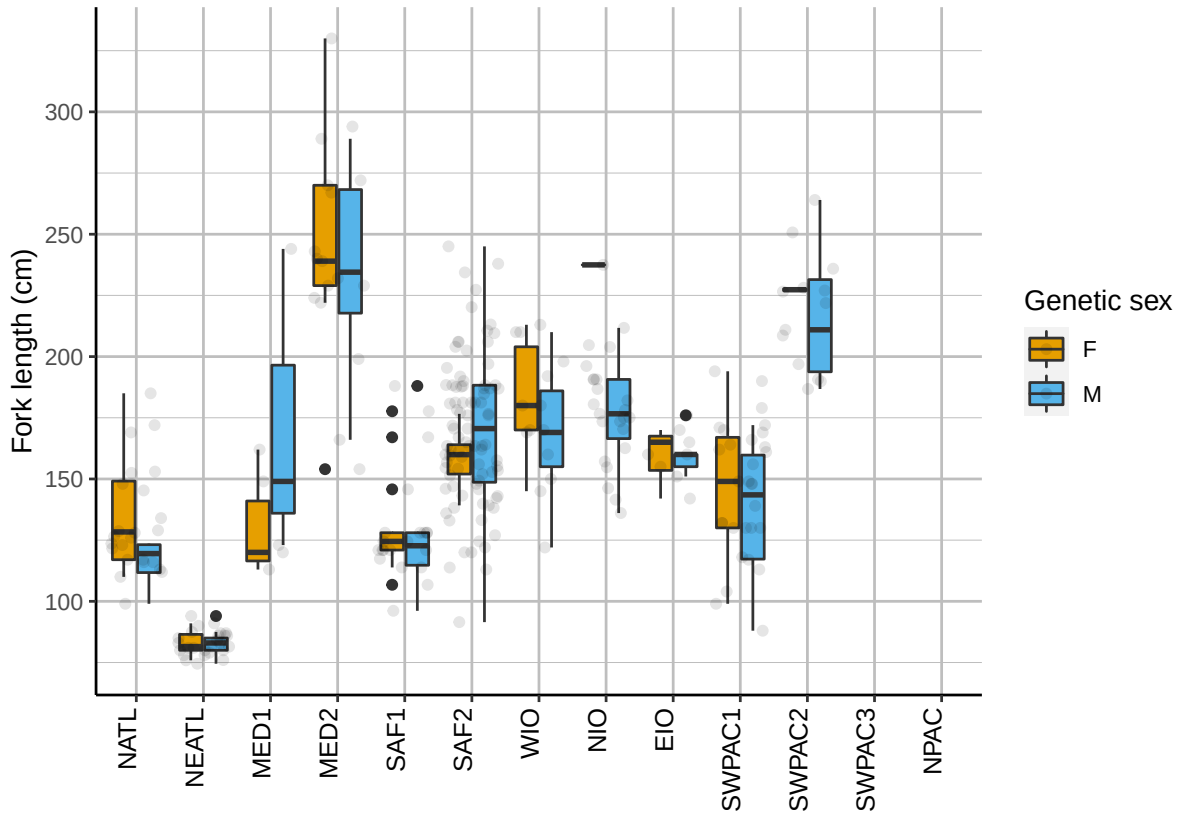



Fork length (cm)
300
250
200
150
100
NATL
NEATL
MED1
MED2
SAF1
SAF2
WIO
NIO
EIO
SWPAC1
SWPAC2
SWPAC3
NPAC


Density
0.020
0.015
0.010
0.005
0.000
100
150
200
250
300
Fork length (cm)
VISUAL_SEX
F
M
NA

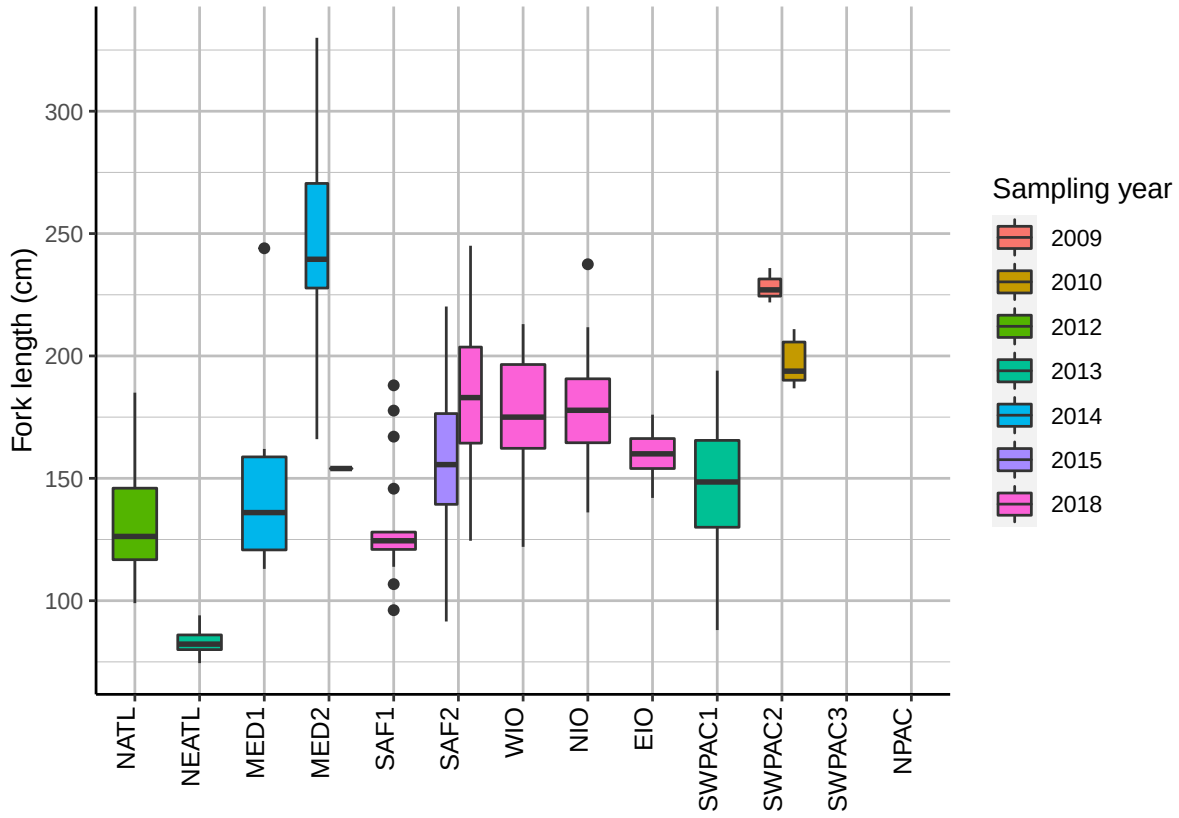

Fork length (cm)
300
250
200
150
100
NATL
NEATL
MED1
MED2
SAF1
SAF2
WIO
NIO
EIO
SWPAC1
SWPAC2
SWPAC3
NPAC
visual sex
F
M
NA


Density
0.010
0.005
0.000
100
150
200
250
300
Fork length (cm)
GENETIC_SEX
F
M



Checking the length between capture years

This is mainly to see if the multi-year sampling in South Africa had an effect.



Data analysis by sample site

Read and filter data

Note: Filtering was performed interactively with the `radiator::filter_rad` function.

Filtering thresholds

- Start: 172384 SNPs, 364 individuals
- Filter reproducibility: 0.96: 155766 SNPs, 364 inds
- Filter monomorphic markers: 155766 SNPs, 364 inds
- Filter common markers: 141869 SNPs, 364 inds
- Filter individual missingness: outliers (0.20): 141869 SNPs,332 inds
- Filter individual heterozygosity: outliers (0.060 / 0.078): 141869 SNPs, 313 inds
- Filter monomorphic markers: 136296 SNPs, 313 inds
- Filter marker minor allele count (MAC): 5: 102764 SNPs, 313 inds
- Filter marker coverage: 10-45: 76714 SNPs, 313 inds
- Filter marker missingness: 0.1: 56361 SNPs, 313 inds
- SNP position on sequence: all: 56361 SNPs, 313 inds
- SNPs per read: 2: 44360 SNPs, 313 inds
- short ld: based on MAC: 35064 SNPs, 313 inds
- Filter individual heterozygosity (mixed genomes): 0.117 - 0.165: 35064 SNPs, 313 inds
- Duplicate genomes: No
- Filter HWE: minimum 3 populations, pvalue < 0.01: 34033 SNPs, 313 inds

```
data <- "infiles/blue_shark_SNP_genotyped.csv"
blue_shark.all <- radiator::filter_rad(
  data = data, strata = strata,
  output = "tidy", interactive.filter = TRUE, parallel.core = 22)

tidy <- blue_shark.all$output$tidy.data
save(tidy, file = "Rdata/blue_shark_filtered_new_strata3.Rdata")
```

Remove sex-linked markers

```
data <- "infiles/blue_shark_SNP_genotyped.csv"
BS.sex <- radiator::sexy_markers(data = data, silicodata = NULL,
                                 strata = strata, parallel.core = 28)

BS.sex.markers <- BS.sex$sexy.summary$CLONE_ID
BS.y <- BS.sex$sexy.summary$CLONE_ID[BS.sex$sexy.summary$MARKER_TYPE ==
                                       "Heterogametic_sex-marker"]
BS.x <- BS.sex$sexy.summary$CLONE_ID[BS.sex$sexy.summary$MARKER_TYPE ==
                                       "Homogametic_sex-marker"]
BS.x.overlap.markers <- BS.x[duplicated(BS.x)] # markers found by 2 methods = reliable
```

```
load("Rdata/blue_shark_filtered_new_strata3.Rdata")
load("Rdata/blue_shark_sex_markers.Rdata")
length(unique(tidy$LOCUS[tidy$LOCUS %in% BS.sex.markers])) #94
length(unique(tidy$LOCUS[tidy$LOCUS %in% BS.x.overlap.markers])) #23
tidy <- tidy[!tidy$LOCUS %in% BS.sex.markers,]
```

Convert to other data formats

```
tidy$GT <- tidy$GT_BIN
BS_all_convert <- radiator::genomic_converter(
  tidy,
  output = c( "genlight","gtypes", "stockr", "structure"))

tidy.all <- BS_all_convert$tidy.data
gl.all <- BS_all_convert$genlight
gt.all <- BS_all_convert$gtypes
stockr.all <- BS_all_convert$stockr
# save(tidy.all, gl.all, gt.all, stockr.all,
# file = "Rdata/blue_shark_filtered_new_strata_converted3.Rdata")

load("Rdata/blue_shark_filtered_new_strata_converted3.Rdata")
```

```
pop.levels <- c("NATL", "NEATL", "MED1", "MED2", "SAF1", "SAF2", "WIO", "NIO", "EIO",
                "SWPAC1", "SWPAC2", "SWPAC3")

tidy.all$STRATA <- factor(tidy.all$STRATA, levels = pop.levels)

gl.all$pop <- factor(gl.all$pop, levels = pop.levels)

attr(stockr.all, "grps") <- factor(attr(stockr.all, "grps"), levels = pop.levels)

meta <- read.csv("infiles/BLUE_SHARK_META_GENETIC_SEX_20210824_V2.csv")
meta$INDIVIDUALS <- stringr::str_replace_all(meta$INDIVIDUALS, "_", "-")
meta$INDIVIDUALS <- stringr::str_replace_all(meta$INDIVIDUALS, " ", "")
meta <- meta[meta$INDIVIDUALS %in% gl.all$ind.names, ]
meta <- meta[order(match(meta$INDIVIDUALS, gl.all$ind.names)), ]

latlong <- data.frame(lat = meta$lat, lon = meta$lon)
rownames(latlong) <- meta$INDIVIDUALS
gl.all$other$ind.metrics <- meta
gl.all$other$loc.metrics <- data.frame(AlleleID = unique(gt.all@data$locus),
                                       CloneID = gl.all$loc.names,
                                       uid = gl.all$loc.names)
gl.all$loc.names <- unique(gt.all@data$locus)
gl.all$other$latlong <- latlong
gt.all@schemes <- meta

radiator::write_vcf(
  tidy.all,
  pop.info = FALSE,
  filename = "BlueShark_313ind_33939SNPs",
  source = NULL,
  empty = FALSE
```

```
)
radiator::write_rad(
  tidy.all,
  path = "./",
  filename = "BlueShark_313ind_33939SNPs.rad",
  tsv = TRUE,
  internal = FALSE,
  append = FALSE,
  col.names = TRUE,
  write.message = "standard",
  verbose = FALSE
)

save(tidy.all,
     gl.all,
     gt.all,
     stockr.all,
     file = "Rdata/Blue_shark_AllData_new_strata4.Rdata")
```

Add correct levels and metadata

Load filtered data without sex-linked markers

```
load("Rdata/Blue_shark_AllData_new_strata4.Rdata")

adegenet::nLoc(gl.all)
```

```
## [1] 33939
```

```
adegenet::nInd(gl.all)
```

```
## [1] 313
```

```
pop.levels2 <- c("NATL", "NEATL", "MED1", "MED2", "SAF1", "SAF2", "WIO", "NIO", "EIO",
                 "SWPAC1", "SWPAC2", "SWPAC3")
gl.all$other$ind.metrics$STRATA <- factor(gl.all$other$ind.metrics$STRATA,
                                          levels = pop.levels2)

knitr::kable(table(gl.all$other$ind.metrics$GENETIC_SEX,
                   gl.all$other$ind.metrics$STRATA),
             col.names = c(shortnames[1:12]),
             caption = "Summary: Number of sharks by location and sex")
```

Table 1: Summary: Number of sharks by location and sex

	NATL	NEATL	MED1	MED2	SAF1	SAF2	WIO	NIO	EIO	SWPAC1	SWPAC2	SWPAC3
F	41	12	21	8	14	8	13	1	3	20	1	8
M	2	9	9	7	5	78	9	15	5	10	13	1

Table 2: Genetic diversity

	NATL	NEATL	MED1	MED2	SAF1	SAF2	WIO	NIO	EIO	SWPAC1	SWPAC2	SWPAC3
ar	1.575	1.554	1.553	1.516	1.520	1.560	1.540	1.527	1.492	1.552	1.509	1.505
size	42.535	20.712	29.665	14.802	18.738	85.066	21.757	15.854	7.925	29.698	13.793	8.877
obs_het	0.149	0.147	0.145	0.142	0.139	0.143	0.144	0.146	0.148	0.146	0.138	0.144
exp_het	0.169	0.167	0.166	0.161	0.161	0.166	0.164	0.162	0.158	0.166	0.159	0.159
uexp_het	0.171	0.171	0.169	0.167	0.165	0.167	0.167	0.167	0.168	0.169	0.165	0.169
fis	0.072	0.076	0.076	0.074	0.087	0.088	0.075	0.058	0.027	0.073	0.087	0.055
hwe_glb	1.000	1.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000
hwe_hom	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
hwe_het	1.000	1.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000
fis_lo	0.060	0.049	0.057	0.020	0.054	0.080	0.050	0.020	-0.089	0.053	0.037	-0.033
fis_hi	0.069	0.067	0.072	0.065	0.079	0.087	0.068	0.050	0.017	0.068	0.078	0.041
ar_lo	1.528	1.523	1.481	1.471	1.461	1.521	1.471	1.477	1.432	1.511	1.458	1.433
ar_hi	1.605	1.591	1.585	1.562	1.561	1.578	1.577	1.574	1.552	1.588	1.561	1.555

Basic data analysis by sample site

1_Diversity Table

```
gl2gpop(gl.all, filename = "Rdata/Genepop_file_new_strata_33939SNP_313ind.txt")

result <- diveRsity::basicStats(
  infile = "Rdata/Genepop_file_new_strata_33939SNP_313ind.txt",
  outfile = "outfile/Result_Diversity_new_strata.txt", fis_ci = TRUE, ar_ci = TRUE,
  fis_boots = 100, ar_boots = 100, mc_reps = 100, rarefaction = FALSE,
  ar_alpha = 0.05, fis_alpha = 0.05)

gendiv <- data.frame(
  "NATL" = result$main_tab$`Pg-AZO-105`$overall,
  "NEATL" = result$main_tab$`Pg-T100`$overall,
  "MED1" = result$main_tab$`Pg-GDL-06`$overall,
  "MED2" = result$main_tab$`Pg-GRE-61`$overall,
  "SAF1" = result$main_tab$`26300`$overall,
  "SAF2" = result$main_tab$`26320`$overall,
  "WIO" = result$main_tab$`33113`$overall,
  "NIO" = result$main_tab$B383$overall,
  "EIO" = result$main_tab$`RITF-Bx1-A08`$overall,
  "SWPAC1" = result$main_tab$`NZ-1.2`$overall,
  "SWPAC2" = result$main_tab$`Pg-AUS-01`$overall,
  "SWPAC3" = result$main_tab$`Pg-VAOIOI-103`$overall
)
rownames(gendiv) <- rownames(result$main_tab$`Pg-AZO-105`)
save(gendiv, file = "Rdata/ALL_Genetic_diversity_new_strata3.Rdata")
readr::write_tsv(gendiv, file = "outfile/Table_Diversity_perSite_new_strata3.txt")
```

```
load("Rdata/ALL_Genetic_diversity_new_strata3.Rdata")
knitr::kable(gendiv, col.names = shortnames[1:12], digits = 3,
             caption = "Genetic diversity") %>%
  kableExtra::kable_styling(full_width = FALSE, font_size = 7)
```

2__Fst

```
gl.all <- gl.all[order(gl.all$pop, gl.all@other$ind.metrics$GENETIC_SEX),]
Fst.site.stampp <- StAMPP::stamppFst(gl.all, nboots = 10000, percent = 95,
                                     nclusters = parallel::detectCores() - 2)

pop.levels2 <- c("NATL", "NEATL", "MED1", "MED2", "SAF1", "SAF2", "WIO", "NIO", "EIO",
                 "SWPAC1", "SWPAC2", "SWPAC3")

Fst.site.stratag <- strataG::pairwiseTest(
  gt.all,
  nrep = 10000,
  by.locus = FALSE,
  hap.locus = 1,
  quietly = FALSE,
  max.cores = parallel::detectCores() - 2,
)

sumres.Fst.neutral <- strataG::pairwiseSummary(Fst.site.stratag, locus = "All")
sumres.Fst.neutral$strata.1 <- factor(sumres.Fst.neutral$strata.1, levels = pop.levels2)
sumres.Fst.neutral$strata.2 <- factor(sumres.Fst.neutral$strata.2, levels = pop.levels2)
sumres.Fst.neutral <- sumres.Fst.neutral[order(sumres.Fst.neutral$strata.1,
                                               sumres.Fst.neutral$strata.2),]
readr::write_csv(sumres.Fst.neutral,
                 file = "outfile/Blue_shark_FST_summary_neutral_by_site3.csv")
m2 <- sumres.Fst.neutral[order(match(sumres.Fst.neutral$strata.1,pop.levels2)),c(2,3,20)]
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.1 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst", type = 'lower')
m2 <- sumres.Fst.neutral[order(match(sumres.Fst.neutral$strata.1,pop.levels2)),c(2,3,21)]
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.2 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst_p.val", type = 'lower')
m2$Fst.p.adj.fdr <- p.adjust(m2$Fst_p.val, "fdr")
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.3 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst.p.adj.fdr", type = 'lower')
m2$Fst.p.adj.bonf <- p.adjust(m2$Fst_p.val, "bonferroni")
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.4 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst.p.adj.bonf", type = 'lower')


sumres.Fst.neutral$Fst.p.adj.fdr <- m2$Fst.p.adj.fdr
sumres.Fst.neutral$Fst.p.adj.bonf <- m2$Fst.p.adj.bonf


write.csv(m2.1, file = "outfile/Blue_shark_FST_neutral_Pairwise_bySites3.csv")
write.csv(m2.2, file = "outfile/Blue_shark_FST_neutral_Pval_Pairwise_bySites3.csv")
write.csv(m2.3,
          file = "outfile/Blue_shark_FST_neutral_Pval_Pairwise_FDR_bySites3.csv")
write.csv(m2.4,
          file = "outfile/Blue_shark_FST_neutral_Pval_Pairwise_bonferroni_bySites3.csv")
write.csv(sumres.Fst.neutral,
          file = "outfile/Blue_shark_FST_bySites3.csv")

save(Fst.site.stampp,
```

Table 3: staMMP: Pairwise Fst with all sampling sites

	NATL	NEATL	MED1	MED2	SAF1	SAF2	WIO	NIO	EIO	SWPAC1	SWPAC2	SWPAC3
NATL	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NEATL	0.0005	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
MED1	0.0004	0.0008	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
MED2	0.0013	0.0012	0.0005	NA	NA	NA	NA	NA	NA	NA	NA	NA
SAF1	0.0014	0.0015	0.0015	0.0028	NA	NA	NA	NA	NA	NA	NA	NA
SAF2	0.0021	0.0025	0.0019	0.0036	1e-04	NA	NA	NA	NA	NA	NA	NA
WIO	0.0022	0.0024	0.0021	0.0036	2e-04	-0.0001	NA	NA	NA	NA	NA	NA
NIO	0.0024	0.0028	0.0025	0.0039	4e-04	0.0003	-1e-04	NA	NA	NA	NA	NA
EIO	0.0025	0.0031	0.0031	0.0049	8e-04	0.0008	-4e-04	0.0007	NA	NA	NA	NA
SWPAC1	0.0028	0.0030	0.0029	0.0044	3e-04	0.0002	0e+00	0.0002	2e-04	NA	NA	NA
SWPAC2	0.0030	0.0032	0.0032	0.0046	-6e-04	0.0007	1e-04	0.0000	-4e-04	0e+00	NA	NA
SWPAC3	0.0034	0.0029	0.0033	0.0051	6e-04	0.0012	-1e-04	0.0012	6e-04	7e-04	-3e-04	NA

Table 4: staMMP: Pairwise Fst P-values with all sampling sites

	NATL	NEATL	MED1	MED2	SAF1	SAF2	WIO	NIO	EIO	SWPAC1	SWPAC2	SWPAC3
NATL	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NEATL	0.0228	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
MED1	0.0129	0.0013	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
MED2	0.0000	0.0008	0.0568	NA	NA	NA	NA	NA	NA	NA	NA	NA
SAF1	0.0000	0.0000	0.0000	0	NA	NA	NA	NA	NA	NA	NA	NA
SAF2	0.0000	0.0000	0.0000	0	0.3721	NA	NA	NA	NA	NA	NA	NA
WIO	0.0000	0.0000	0.0000	0	0.3023	0.6304	NA	NA	NA	NA	NA	NA
NIO	0.0000	0.0000	0.0000	0	0.1222	0.1136	0.5811	NA	NA	NA	NA	NA
EIO	0.0000	0.0000	0.0000	0	0.0850	0.0410	0.7753	0.1445	NA	NA	NA	NA
SWPAC1	0.0000	0.0000	0.0000	0	0.1389	0.0953	0.4704	0.2930	0.3255	NA	NA	NA
SWPAC2	0.0000	0.0000	0.0000	0	0.9382	0.0078	0.3776	0.5497	0.7288	0.4667	NA	NA
SWPAC3	0.0000	0.0000	0.0000	0	0.1643	0.0016	0.5435	0.0172	0.2121	0.0620	0.6856	NA

```
    Fst.site.stratag,
    sumres.Fst.neutral,
    m2,
    file = "Rdata/fst.grouping.ALL_bySite_NEW3.Rdata")
```

```
load("Rdata/fst.grouping.ALL_bySite_NEW3.Rdata")
knitr::kable(
  Fst.site.stampp$Fsts, digits = 4,
  caption = "staMMP: Pairwise Fst with all sampling sites") %>%
  kableExtra::kable_styling(full_width = FALSE, font_size = 7)
```

```
knitr::kable(
  Fst.site.stampp$Pvalues, digits = 4,
  caption = "staMMP: Pairwise Fst P-values with all sampling sites")  %>%
  kableExtra::kable_styling(full_width = FALSE, font_size = 7)
```

```
knitr::kable(
  sumres.Fst.neutral[order(sumres.Fst.neutral$Fst,decreasing = TRUE),c(1,20,21,40,41)],
  digits = 4, caption = "StrataG: Pairwise Fst with all sampling sites") %>%
  kableExtra::kable_styling(full_width = FALSE, font_size = 7)
```

Table 5: StrataG: Pairwise Fst with all sampling sites

	label	Fst	Fst_p.val	Fst.p.adj.fdr	Fst.p.adj.bonf
29	MED2 (15) v. SWPAC3 (9)	0.0025	0.0001	0.0002	0.0066
2	EIO (8) v. MED2 (15)	0.0025	0.0001	0.0002	0.0066
28	MED2 (15) v. SWPAC2 (14)	0.0023	0.0001	0.0002	0.0066
27	MED2 (15) v. SWPAC1 (30)	0.0022	0.0001	0.0002	0.0066
24	MED2 (15) v. NIO (16)	0.0019	0.0001	0.0002	0.0066
30	MED2 (15) v. WIO (22)	0.0018	0.0001	0.0002	0.0066
26	MED2 (15) v. SAF2 (86)	0.0018	0.0001	0.0002	0.0066
4	EIO (8) v. NEATL (21)	0.0017	0.0001	0.0002	0.0066
37	NATL (43) v. SWPAC3 (9)	0.0017	0.0001	0.0002	0.0066
43	NEATL (21) v. SWPAC2 (14)	0.0016	0.0001	0.0002	0.0066
19	MED1 (30) v. SWPAC2 (14)	0.0016	0.0001	0.0002	0.0066
20	MED1 (30) v. SWPAC3 (9)	0.0015	0.0001	0.0002	0.0066
1	EIO (8) v. MED1 (30)	0.0015	0.0001	0.0002	0.0066
36	NATL (43) v. SWPAC2 (14)	0.0015	0.0001	0.0002	0.0066
44	NEATL (21) v. SWPAC3 (9)	0.0015	0.0001	0.0002	0.0066
18	MED1 (30) v. SWPAC1 (30)	0.0015	0.0001	0.0002	0.0066
42	NEATL (21) v. SWPAC1 (30)	0.0015	0.0001	0.0002	0.0066
39	NEATL (21) v. NIO (16)	0.0014	0.0001	0.0002	0.0066
35	NATL (43) v. SWPAC1 (30)	0.0014	0.0001	0.0002	0.0066
25	MED2 (15) v. SAF1 (19)	0.0014	0.0001	0.0002	0.0066
3	EIO (8) v. NATL (43)	0.0013	0.0001	0.0002	0.0066
15	MED1 (30) v. NIO (16)	0.0013	0.0001	0.0002	0.0066
32	NATL (43) v. NIO (16)	0.0013	0.0001	0.0002	0.0066
45	NEATL (21) v. WIO (22)	0.0012	0.0001	0.0002	0.0066
41	NEATL (21) v. SAF2 (86)	0.0011	0.0001	0.0002	0.0066
38	NATL (43) v. WIO (22)	0.0011	0.0001	0.0002	0.0066
21	MED1 (30) v. WIO (22)	0.0011	0.0001	0.0002	0.0066
34	NATL (43) v. SAF2 (86)	0.0010	0.0001	0.0002	0.0066
17	MED1 (30) v. SAF2 (86)	0.0010	0.0001	0.0002	0.0066
40	NEATL (21) v. SAF1 (19)	0.0008	0.0001	0.0002	0.0066
16	MED1 (30) v. SAF1 (19)	0.0008	0.0001	0.0002	0.0066
33	NATL (43) v. SAF1 (19)	0.0007	0.0001	0.0002	0.0066
22	MED2 (15) v. NATL (43)	0.0007	0.0002	0.0004	0.0132
23	MED2 (15) v. NEATL (21)	0.0006	0.0034	0.0066	0.2244
50	NIO (16) v. SWPAC3 (9)	0.0005	0.0532	0.0878	1.0000
6	EIO (8) v. SAF1 (19)	0.0004	0.0963	0.1486	1.0000
59	SAF2 (86) v. SWPAC3 (9)	0.0004	0.0305	0.0544	1.0000
14	MED1 (30) v. NEATL (21)	0.0004	0.0035	0.0066	0.2310
7	EIO (8) v. SAF2 (86)	0.0003	0.0747	0.1202	1.0000
10	EIO (8) v. SWPAC3 (9)	0.0003	0.2247	0.2966	1.0000
5	EIO (8) v. NIO (16)	0.0003	0.1903	0.2563	1.0000
12	MED1 (30) v. MED2 (15)	0.0003	0.1081	0.1622	1.0000
58	SAF2 (86) v. SWPAC2 (14)	0.0002	0.0376	0.0636	1.0000
62	SWPAC1 (30) v. SWPAC3 (9)	0.0002	0.1757	0.2416	1.0000
46	NIO (16) v. SAF1 (19)	0.0002	0.1290	0.1851	1.0000
13	MED1 (30) v. NATL (43)	0.0002	0.0125	0.0229	0.8250
31	NATL (43) v. NEATL (21)	0.0002	0.0374	0.0636	1.0000
55	SAF1 (19) v. SWPAC3 (9)	0.0002	0.2550	0.3242	1.0000
47	NIO (16) v. SAF2 (86)	0.0002	0.0968	0.1486	1.0000
53	SAF1 (19) v. SWPAC1 (30)	0.0002	0.1427	0.2004	1.0000
8	EIO (8) v. SWPAC1 (30)	0.0001	0.3438	0.4202	1.0000
48	NIO (16) v. SWPAC1 (30)	0.0001	0.2554	0.3242	1.0000
56	SAF1 (19) v. WIO (22)	0.0001	0.3076	0.3830	1.0000
57	SAF2 (86) v. SWPAC1 (30)	0.0001	0.1282	0.1851	1.0000
65	SWPAC2 (14) v. WIO (22)	0.0000	0.4304	0.5165	1.0000
52	SAF1 (19) v. SAF2 (86)	0.0000	0.4394	0.5179	1.0000
63	SWPAC1 (30) v. WIO (22)	0.0000	0.4769	0.5522	1.0000
51	NIO (16) v. WIO (22)	0.0000	0.5460	0.6211	1.0000
61	SWPAC1 (30) v. SWPAC2 (14)	0.0000	0.5552	0.6211	1.0000
60	SAF2 (86) v. WIO (22)	0.0000	0.6688	0.7119	1.0000
49	NIO (16) v. SWPAC2 (14)	0.0000	0.5948	0.6543	1.0000
66	SWPAC3 (9) v. WIO (22)	-0.0001	0.6935	0.7265	1.0000
9	EIO (8) v. SWPAC2 (14)	-0.0002	0.6656	0.7119	1.0000
64	SWPAC2 (14) v. SWPAC3 (9)	-0.0002	0.7337	0.7566	1.0000
11	EIO (8) v. WIO (22)	-0.0002	0.7620	0.7737	1.0000
54	SAF1 (19) v. SWPAC2 (14)	-0.0003	0.9416	0.9416	1.0000

3_AMOVA

```
dist <- dist(tab(gl.all)) #euclidean distance
pop <- adegenet::pop(gl.all)
amova.result <- pegas::amova(dist ~ pop,
                             nperm = 1000)
print(amova.result)
```

```
##
##  Analysis of Molecular Variance
##
## Call: pegas::amova(formula = dist ~ pop, nperm = 1000)
##
##              SSD      MSD  df
## pop     75405.88 6855.080  11
## Error 1691619.54 6506.229 260
## Total 1767025.42 6520.389 271
##
## Variance components:
##         sigma2 P.value
## pop     16.365       0
## Error 6506.229
##
## Phi-statistics:
## pop.in.GLOBAL
##   0.002508974
##
## Variance coefficients:
##        a
## 21.31684
```

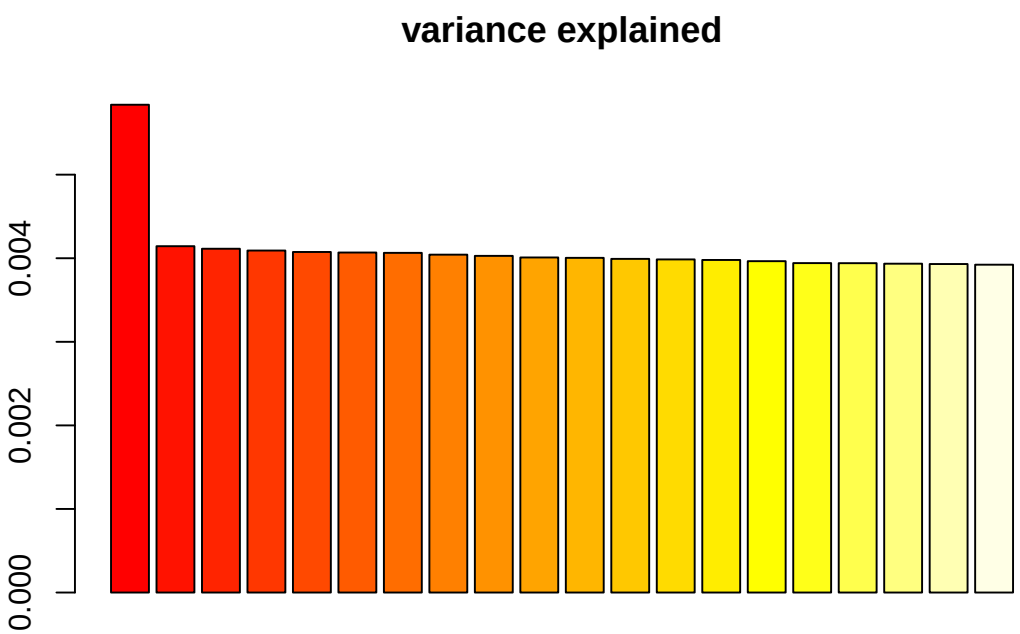
5_PCA

```
pca.all <- adegenet::glPca(gl.all, nf = 4,
                           parallel = TRUE,
                           n.cores = parallel::detectCores())

save(pca.all,
     file = "Rdata/PCA_ALL_objects_New_strata3.Rdata")
```

```
load("Rdata/PCA_ALL_objects_New_strata3.Rdata")
var_frac <- pca.all$eig/sum(pca.all$eig)
print(paste0("Total variance explained: ",signif(sum(var_frac[1:4]) * 100, 3), "%"))
```

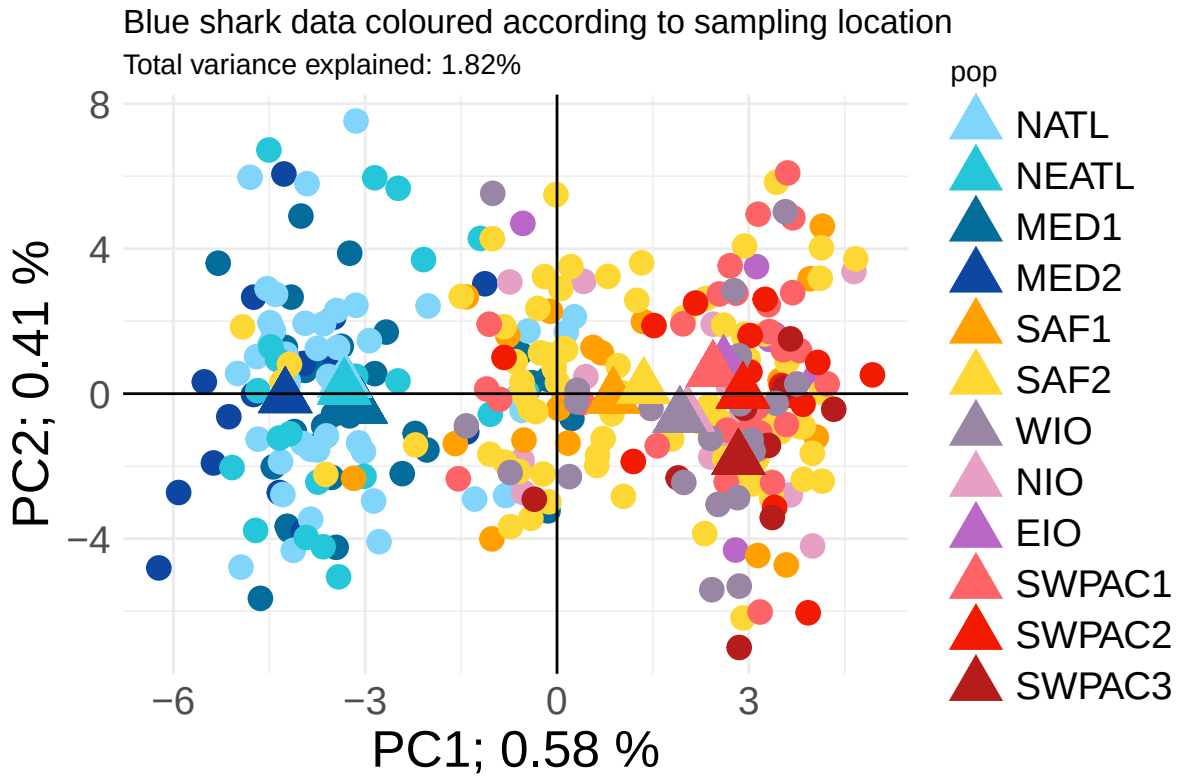
```
## [1] "Total variance explained: 1.82%"
```

```
barplot(var_frac[1:20], main = "variance explained",
        col = heat.colors(length(var_frac[1:20])))
```




```
pop <- gl.all$pop

data <- data.frame(pca.all$scores, pop = pop)
data <- merge(data, aggregate(cbind(mean.x = PC1, mean.y = PC2) ~
                              pop, data, mean), by = "pop")

pca.plot <- ggplot2::ggplot(data, ggplot2::aes(PC1, PC2, color = pop)) +
  ggplot2::geom_point(size = 4) +
  ggplot2::geom_point(ggplot2::aes(x = mean.x,
                                   y = mean.y, color = pop),
```

```
                      size = 7,
                      shape = 17) +
  ggplot2::scale_colour_manual(values = colours12) +
  ggplot2::scale_fill_manual(values = colours12) +
  ggplot2::geom_hline(yintercept = 0) +
  ggplot2::geom_vline(xintercept = 0) +
  ggplot2::labs(
    subtitle = paste0("Total variance explained: ",
                      signif(sum(var_frac[1:4]) * 100, 3), "%"),
    y = paste0("PC2; ", signif(var_frac[2] * 100, digits = 2), " %"),
    x = paste0("PC1; ", signif(var_frac[1] * 100, digits = 2), " %"),
    title = "Blue shark data coloured according to sampling location",
    caption = "") +
  ggplot2::theme_minimal() +
  ggplot2::theme(
    axis.text = ggplot2::element_text(size = 15),
    axis.title.x = ggplot2::element_text(size = 20),
    axis.title.y = ggplot2::element_text(size = 20),
    legend.text = ggplot2::element_text(size = 15)
  )

print(pca.plot)
```



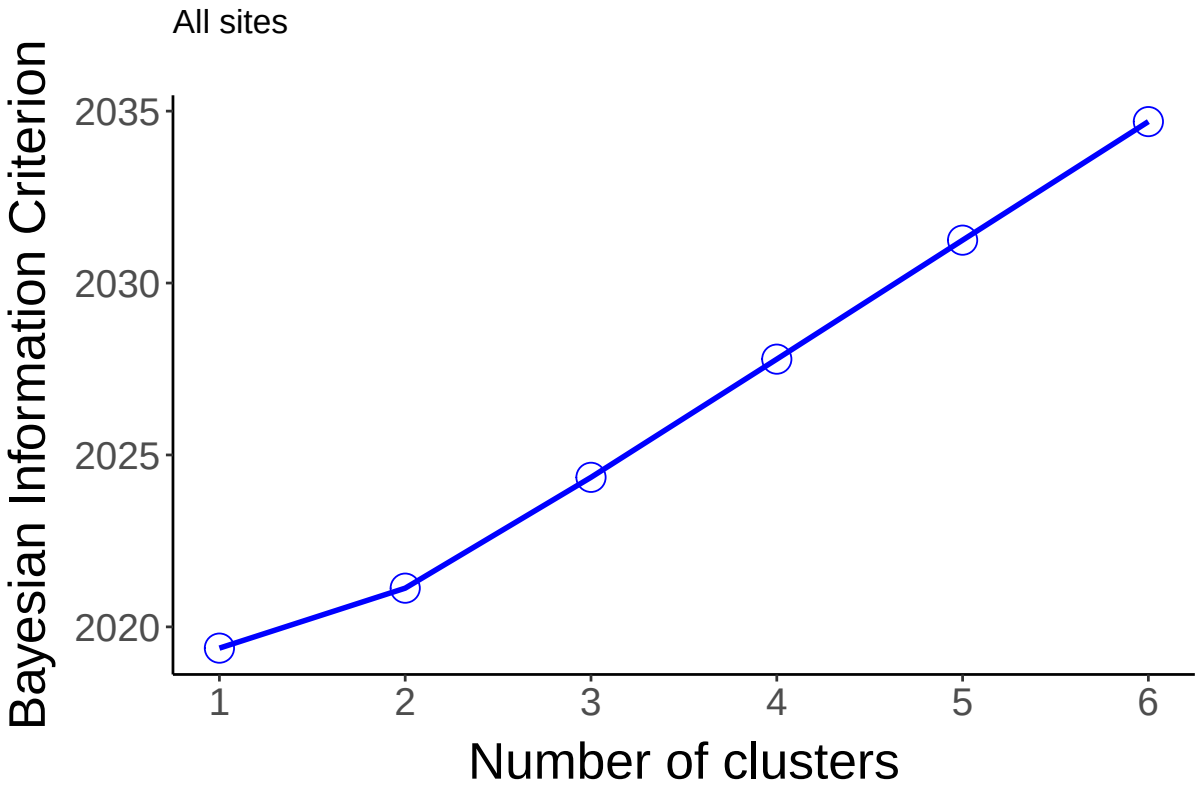

```
ggplot2::ggsave(pca.plot,
                filename = "figures/2.ALL_PCA_New_strata_33939SNPs2.png",
                width = 30, height = 15, units = "cm")
```

6_ DAPC

```
set.seed(124)
grp1 <-
  adegenet::find.clusters(
    gl.all,
    max.n.clust = 6,
    n.pca = adegenet::nInd(gl.all) / 3,
    stat = "BIC",
    parallel = TRUE,
    n.cores = parallel::detectCores() - 2
  )

save(grp1, file = "Rdata/BSH_kgrp_New_strata3.rdata")
```

```
load("Rdata/BSH_kgrp_New_strata3.rdata")

y <- as.numeric(grp1$Kstat)
x <- 1:6
data <- data.frame(x,y,stringsAsFactors = F)
plot <- ggplot2::ggplot(data, ggplot2::aes(x,y)) +
  ggplot2::geom_point(size = 5, shape = 1, color = "blue") +
  ggplot2::geom_line(size = 1,color = "blue") +
  ggplot2::scale_x_continuous(name = waiver(),
                              breaks = seq(from = 0,
                                           to = nrow(gl.all) - 1, by = 1)) +
  ggplot2::labs(subtitle = "",
                y = "Bayesian Information Criterion",
                x = "Number of clusters",
                title = "All sites",
                caption = "") +
  ggplot2::theme_classic() +
  ggplot2::theme(
    axis.text = ggplot2::element_text(size = 15),
    axis.title.x = ggplot2::element_text(size = 20,vjust = 0, hjust = 0.5),
    axis.title.y = ggplot2::element_text(size = 20,vjust = 2, hjust = 0.5))
print(plot)
```

Optimal number of clusters with neutral markers



Cross-validation to identify the optimal number of clusters: number of PC axis to account for Uses a training - holdout set of individuals to check how reliable individuals can be assigned.

```
set.seed(124)
x <- gl.all

png(file="figures/3.BSH_neutral_refined_n_pca_new_strata3.png")
xval <-
  adegenet::xvalDapc(
    tab(x, NA.method = "mean"),
    adegenet::pop(x),
    n.da = 3,
    n.pca = seq(1,101, 5),
    n.rep = 100,
    parallel = "snow",
    ncpus = parallel::detectCores() - 2,
    xval.plot = TRUE
  )
dev.off()
save(xval, file = "Rdata/xval_ALL_refined_new_strata3.rdata")
```

```
load("Rdata/xval_ALL_refined_new_strata3.rdata")
print(xval[2:6])
```

```
## $`Median and Confidence Interval for Random Chance`
##       2.5%        50%      97.5%
## 0.05603751 0.08246071 0.11709560
##
## $`Mean Successful Assignment by Number of PCs of PCA`
```

```
##         1         6        11        16        21        26
## 0.1459962 0.1929823 0.1826540 0.1944924 0.1855467 0.1846338
##        31        36        41        46        51        56
## 0.1637134 0.1723523 0.1642487 0.1616907 0.1848169 0.1795492
##        61        66        71        76        81        86
## 0.1755543 0.1630871 0.1636263 0.1662386 0.1724066 0.1665530
##        91        96       101
## 0.1671086 0.1710783 0.1699293
##
## $`Number of PCs Achieving Highest Mean Success`
## [1] "16"
##
## $`Root Mean Squared Error by Number of PCs of PCA`
##         1         6        11        16        21        26
## 0.8541363 0.8079890 0.8182548 0.8069857 0.8158237 0.8165255
##        31        36        41        46        51        56
## 0.8373070 0.8286457 0.8369540 0.8393598 0.8165885 0.8215388
##        61        66        71        76        81        86
## 0.8258765 0.8381415 0.8380738 0.8349780 0.8287768 0.8345831
##        91        96       101
## 0.8339885 0.8305782 0.8317423
##
## $`Number of PCs Achieving Lowest MSE`
## [1] "16"
```

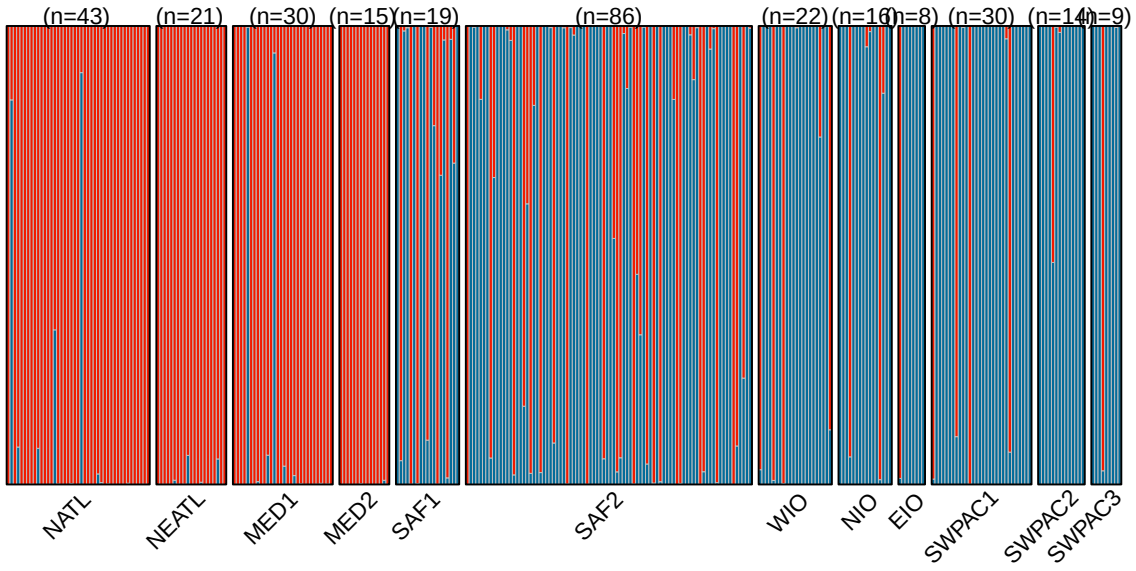
```
PC1 <- as.numeric(xval$`Number of PCs Achieving Lowest MSE`)
```

```
dapc.all.object.names <- c()
for (K in 2:6) {
  set.seed(124)
  grp <- adegenet::find.clusters(gl.all, n.clust = K, n.pca = PC1)
  set.seed(124)
  dapc.all <- adegenet::dapc(gl.all, grp$grp, n.da = K - 1,
                                    n.pca = PC1)
  assign(paste0("dapc.all", K),value = dapc.all)
  dapc.all.object.names <- c(dapc.all.object.names,
                                    paste0("dapc.all", K))
}
save(list=dapc.all.object.names, dapc.all.object.names,
     file = "Rdata/DAPC.ALL_new_strata3.Rdata")
```

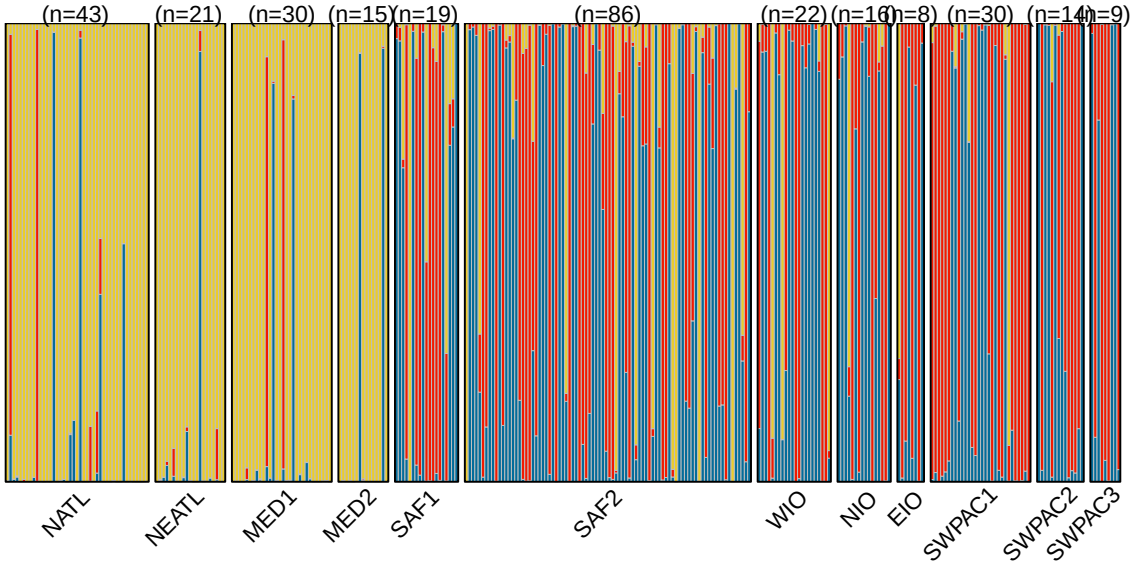
DAPC barplot Group individuals according to DAPC posterior membership.

```
load("Rdata/DAPC.ALL_new_strata3.Rdata")
for (K in 2:6) {
  colour <- colours6[1:K]
  dapc <- get(dapc.all.object.names[K - 1])
  post <- as.matrix(dapc$posterior)
  colnames(post) <- paste0("Group", 1:nlevels(dapc$grp))
  locations <- gl.all$pop

  plot.dapc.FDD(x = post,
                locations = locations,
                colour = colour,
                region.lwd = 1,
                plotTitle = paste0("DAPC: all blue sharks - data\n for K=",
                                   K, " & PC=", PC1, sep = ""))
```

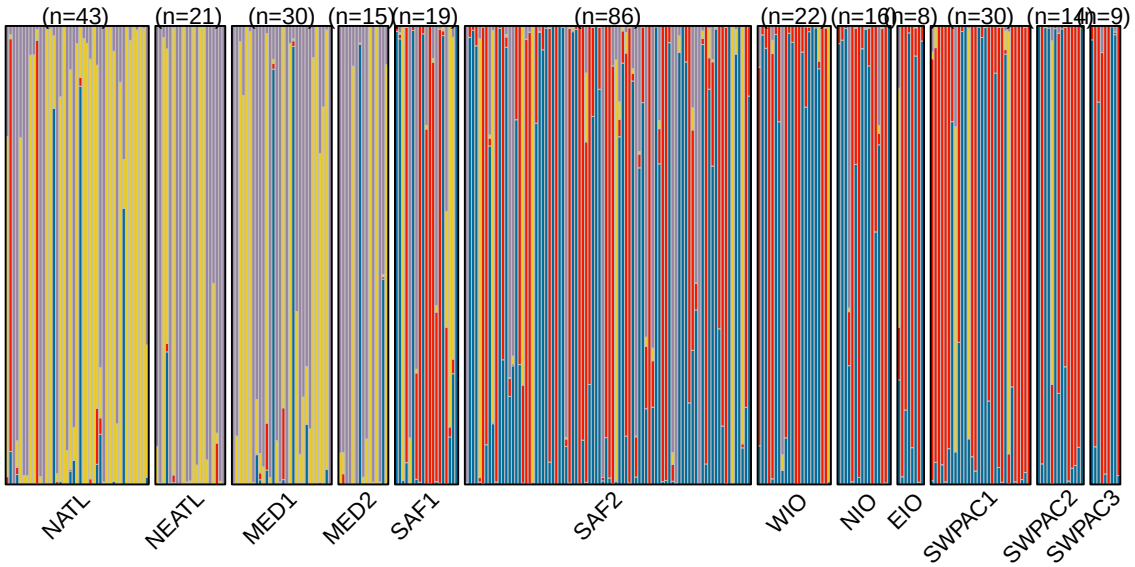
```
  dev.print(
    device = png,
    file = paste0("figures/3.DAPC_barplot_new_strata_33939SNPs_K", K, ".png"),
    res = 300,
    width = 30,
    height = 15,
    units = "cm")
}
```



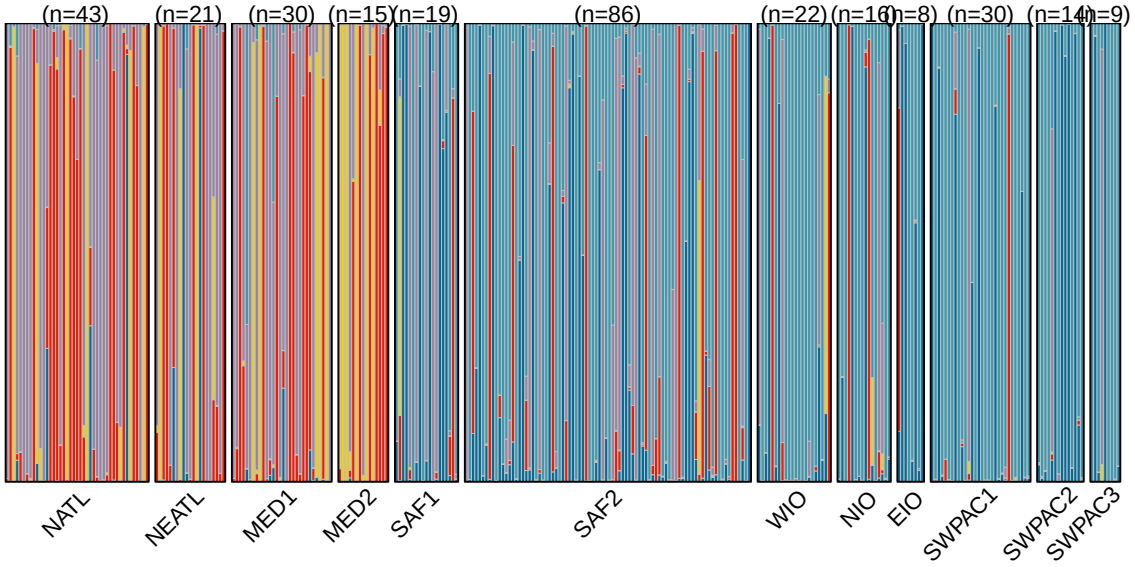
DAPC: all blue sharks − data for K=3 & PC=16



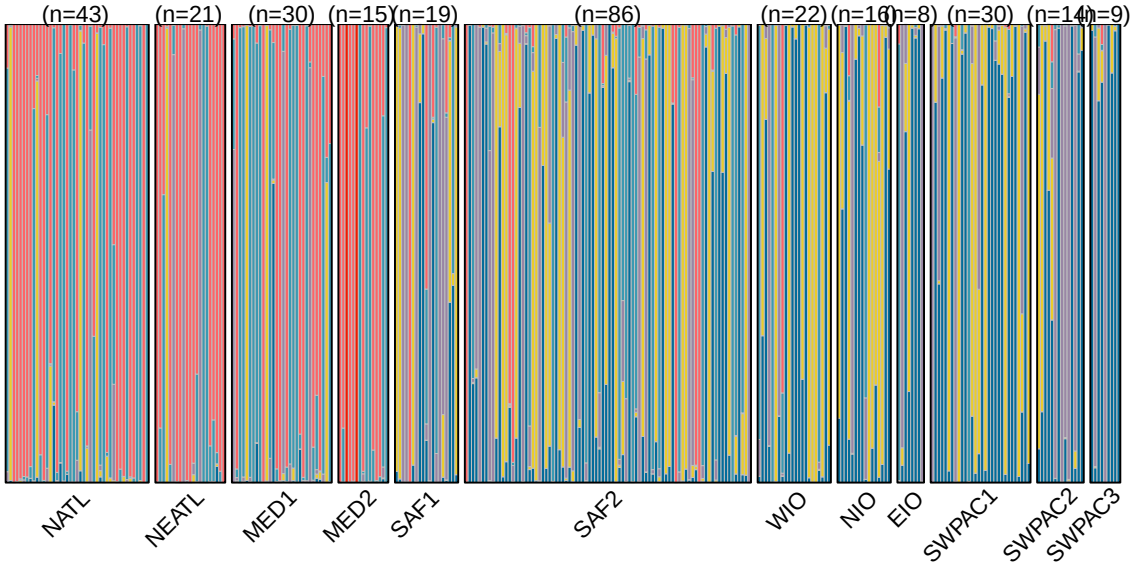
DAPC: all blue sharks − data for K=4 & PC=16



DAPC: all blue sharks − data for K=5 & PC=16



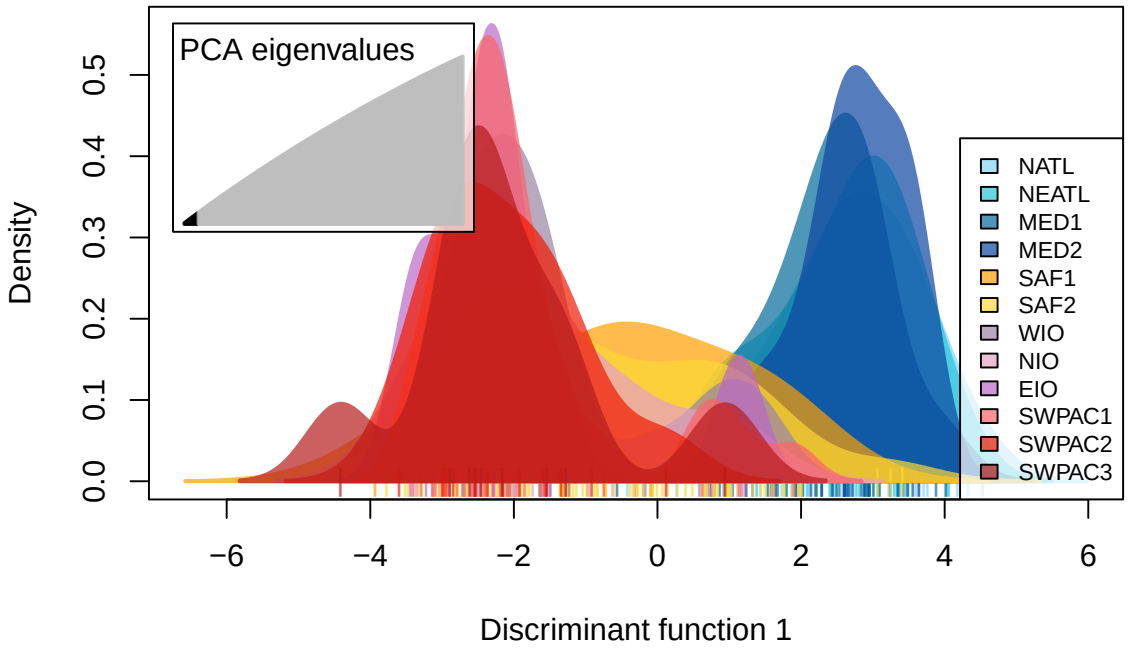
DAPC: all blue sharks − data for K=6 & PC=16

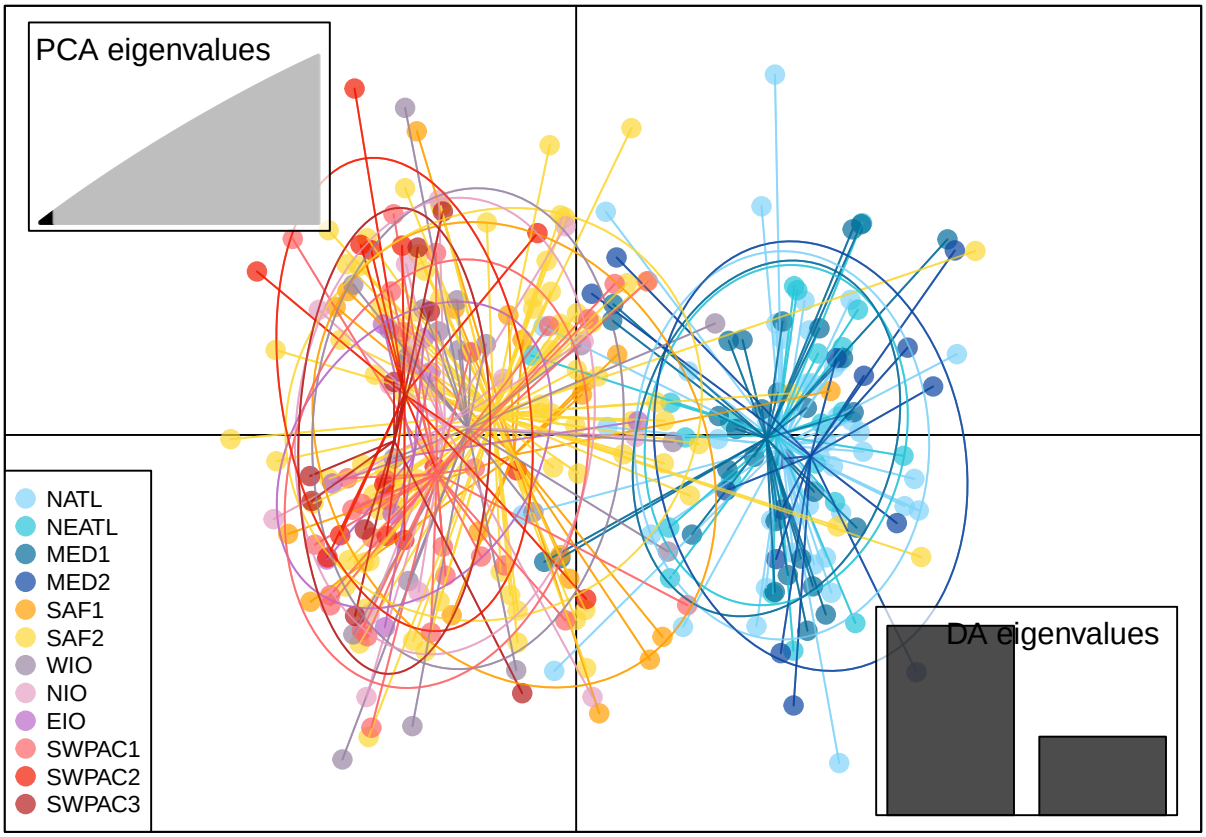


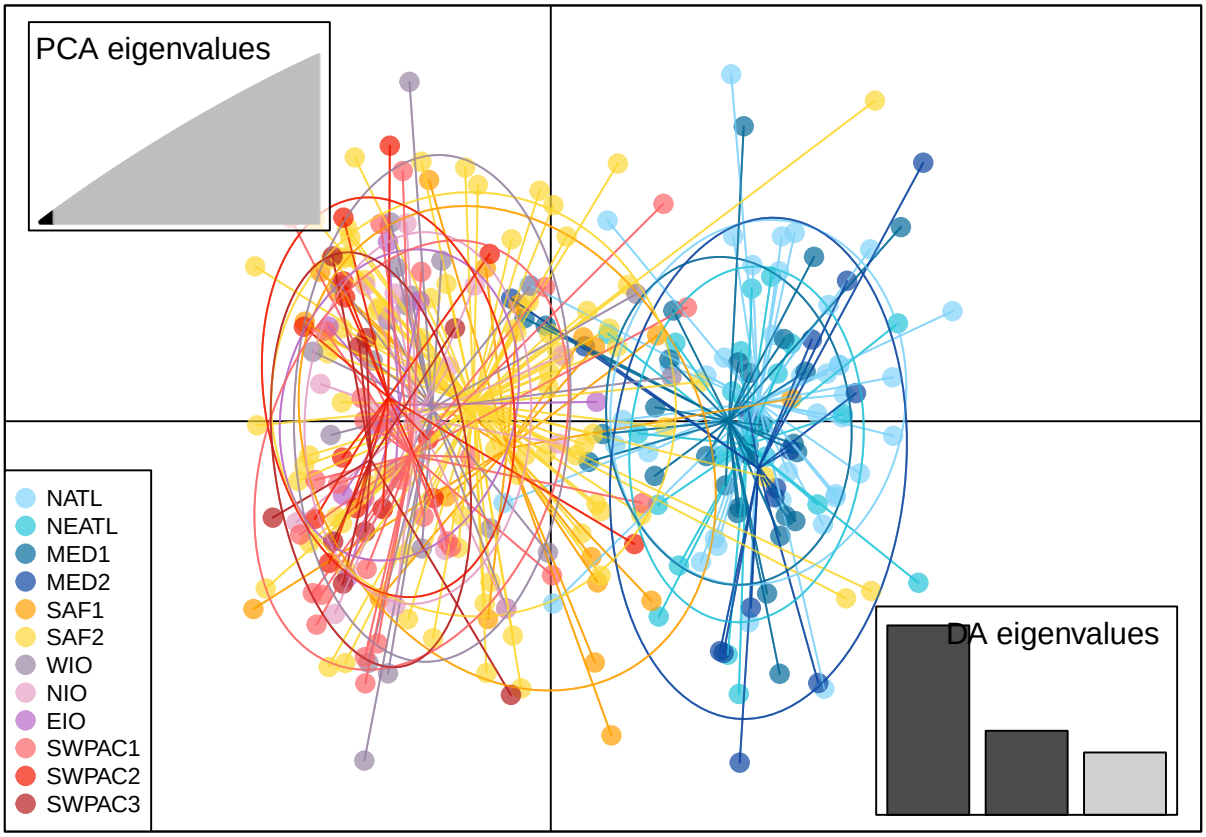

```
load("Rdata/DAPC.ALL_new_strata3.Rdata")
```

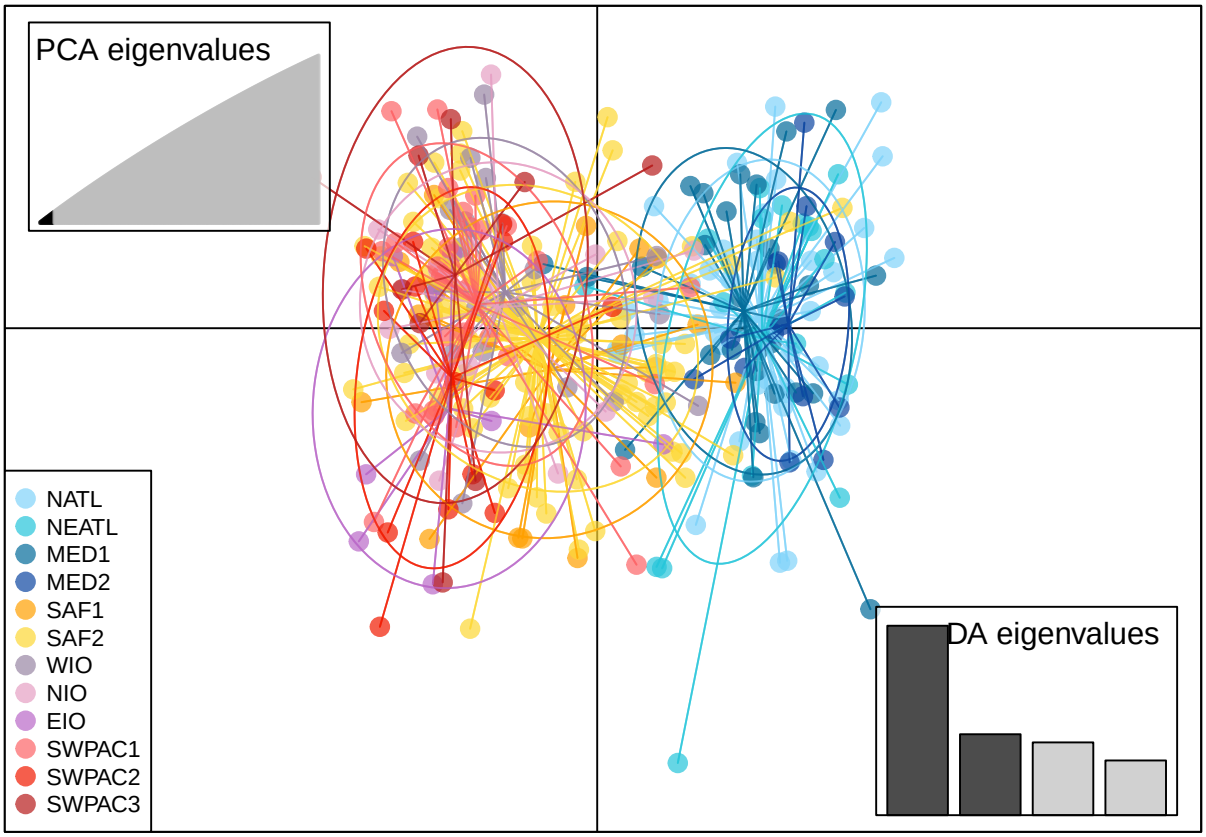
```
for (K in 2:6) {
  dapc.neutral <- get(dapc.all.object.names[K - 1])
  if (K == 2) {
    posi.leg <- "bottomright"
  } else {posi.leg <- "bottomleft"}
  dapc.neutral.plot <- ade4::scatter(dapc.neutral, grp = pop(gl.all), cex = 2,
                                     legend = TRUE, col = colours12,
                                     clabel = FALSE, posi.leg = posi.leg,
                                     scree.pca = TRUE, posi.pca = "topleft",
                                     cleg = 0.75, xax = 1, yax = 2,
                                     inset.solid = 0.70)
  dev.print(
    device = png,
    file = paste0("figures/3.DAPC_Scatter_new_strata_33939SNPs_K", K, ".png"),
    width = 30,
    height = 15,
    units = "cm",
    res = 300
  )
}
```

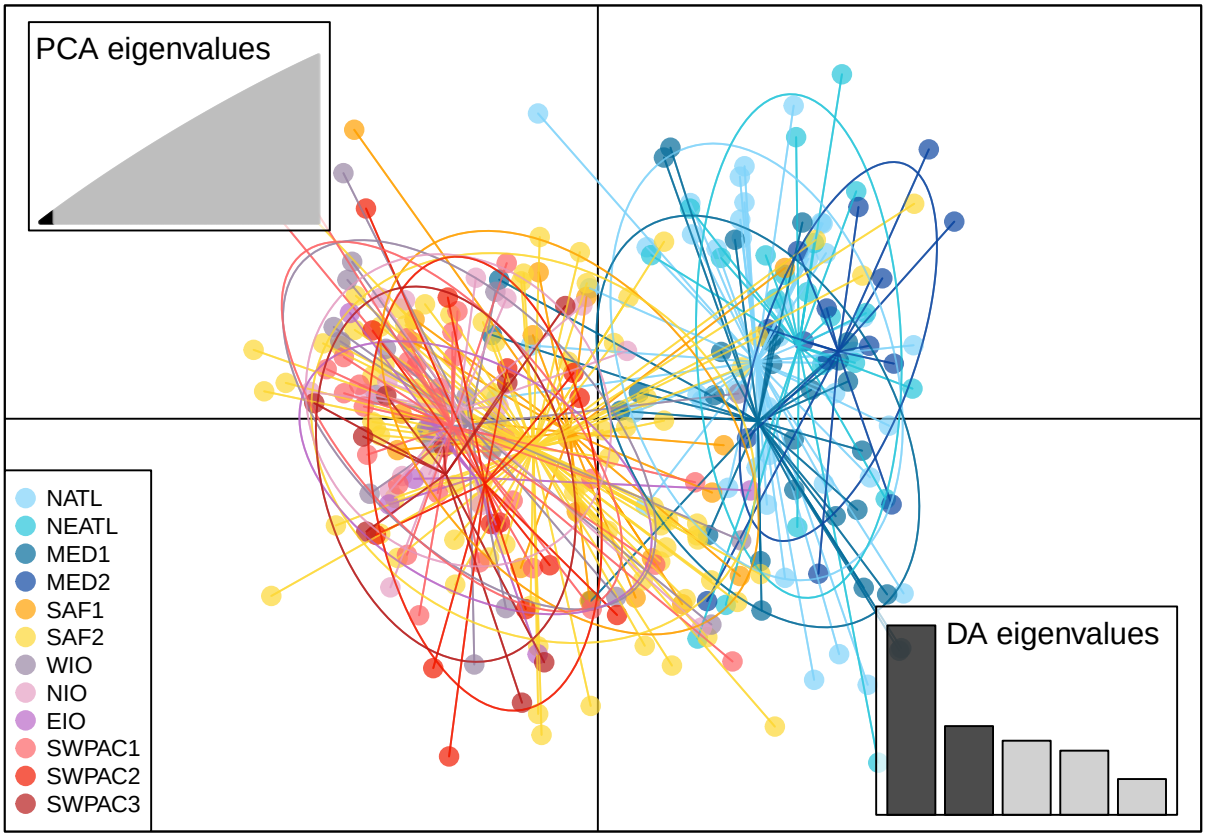
DAPC scatterplot




PCA eigenvalues
NATL
NEATL
MED1
MED2
SAF1
SAF2
WIO
NIO
EIO
SWPAC1
SWPAC2
SWPAC3
DA eigenvalues

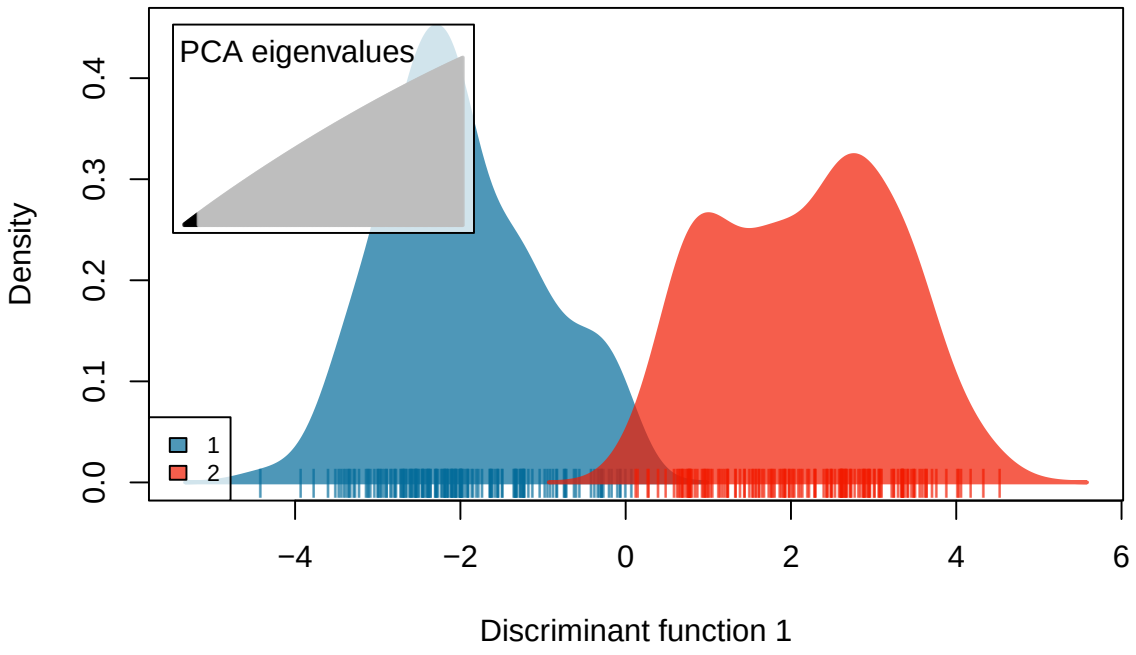

PCA eigenvalues
NATL
NEATL
MED1
MED2
SAF1
SAF2
WIO
NIO
EIO
SWPAC1
SWPAC2
SWPAC3
DA eigenvalues

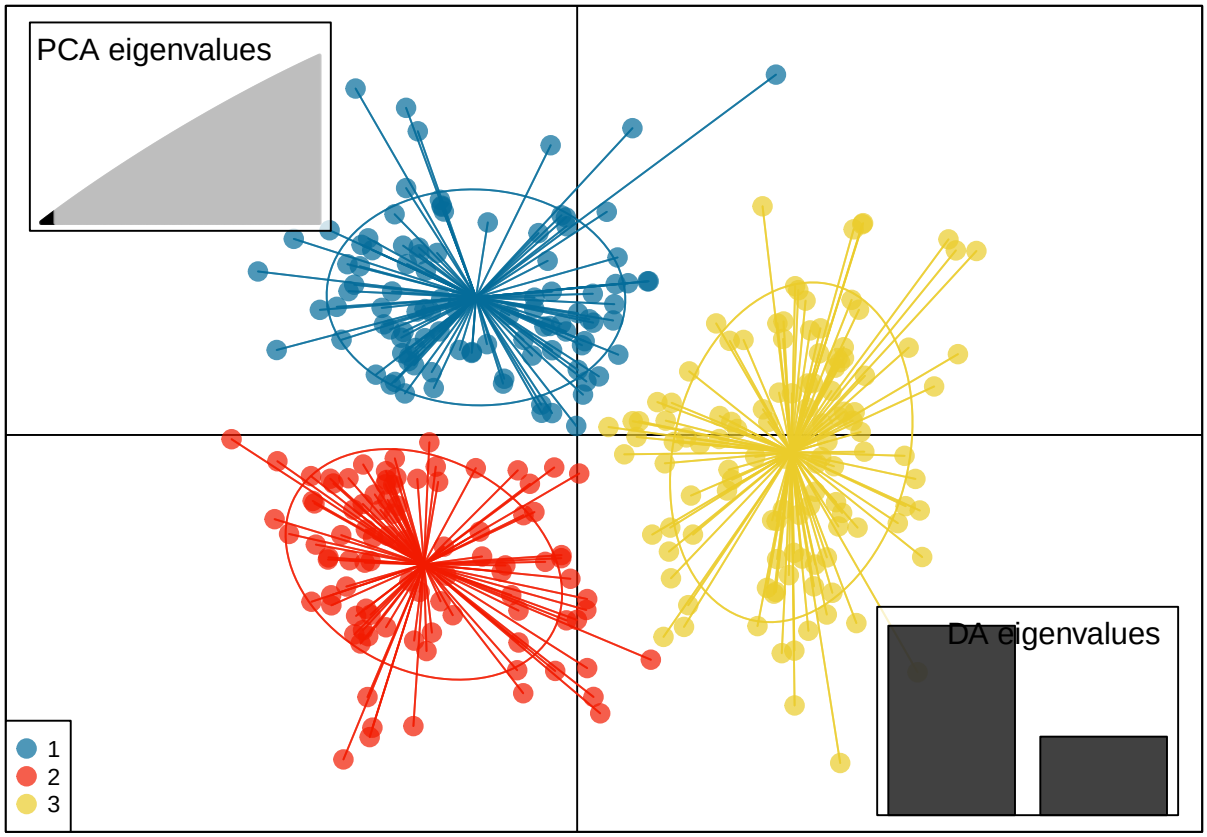


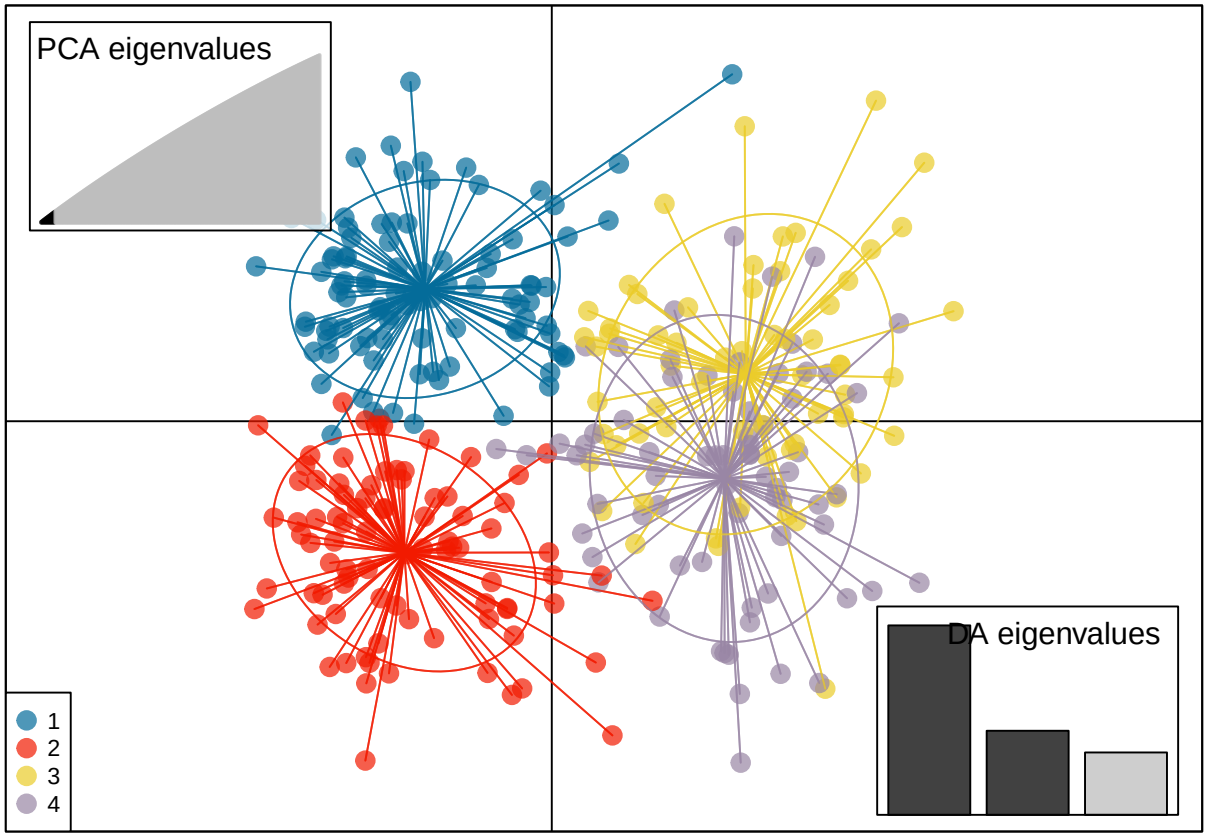


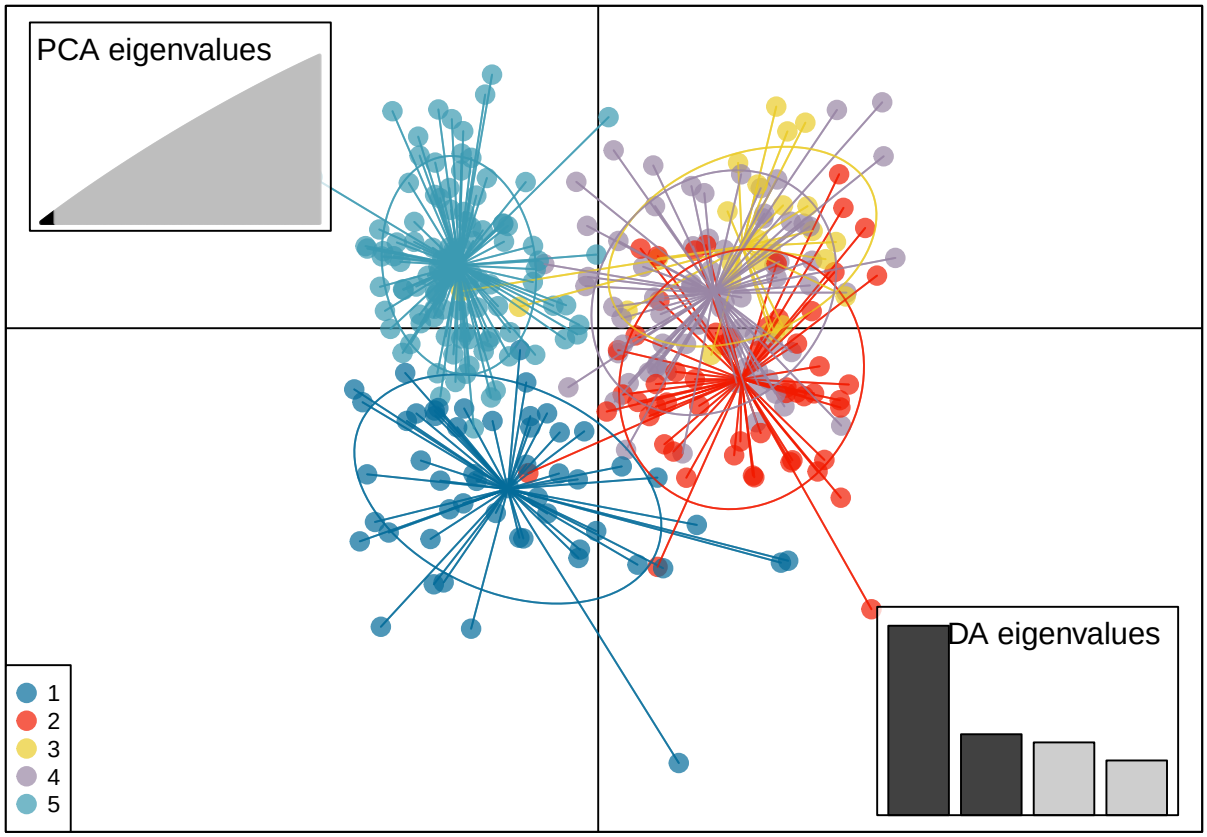

```
for (K in 2:6) {
  colour <- colours6[1:K]
  dapc.neutral <- get(dapc.all.object.names[K - 1])
```

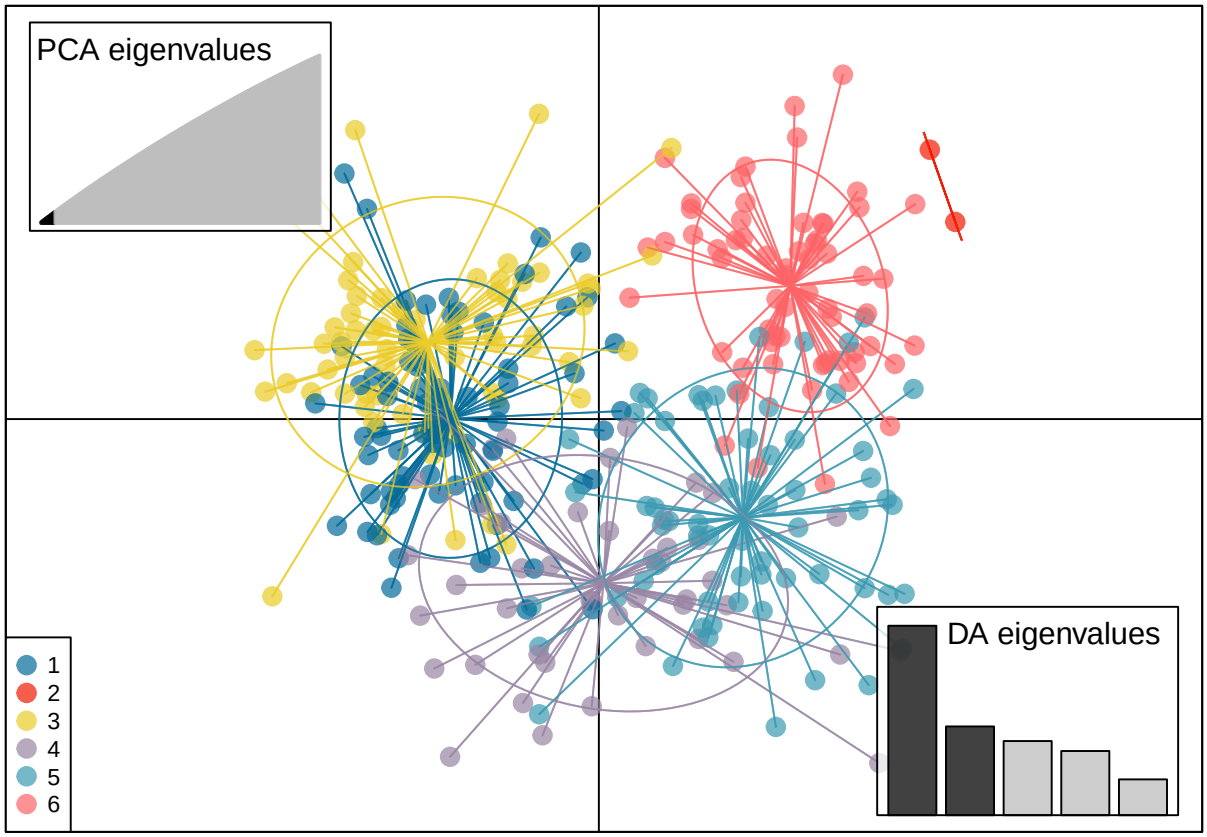
```
  dapc.neutral.plot <- ade4::scatter(dapc.neutral, cex = 2, legend = TRUE,
                                     col = colour, clabel = FALSE,
                                     posi.leg = "bottomleft", scree.pca = TRUE,
                                     posi.pca = "topleft", cleg = 0.75,
                                     xax = 1, yax = 2, inset.solid = 0.75)
  dev.print(
    device = png,
    file = paste0("figures/3.DAPC_Scatter_new_strata_33939SNPs_pergroups_K",
                  K,".png"),
    width = 30,
    height = 15,
    units = "cm",
    res = 300
  )
}
```




PCA eigenvalues
DA eigenvalues
1
2
3


PCA eigenvalues
DA eigenvalues
1
2
3
4






```
dapc <- get(dapc.all.object.names[1])
```

```
df <- data.frame(gl.all$other$ind.metrics, DAPC_GROUP = dapc$assign)
write.csv(df, file = "new_metadata_with_DAPC_grouping_new_strata3.csv",
          row.names = FALSE)
```

Identify South African groups for K=2

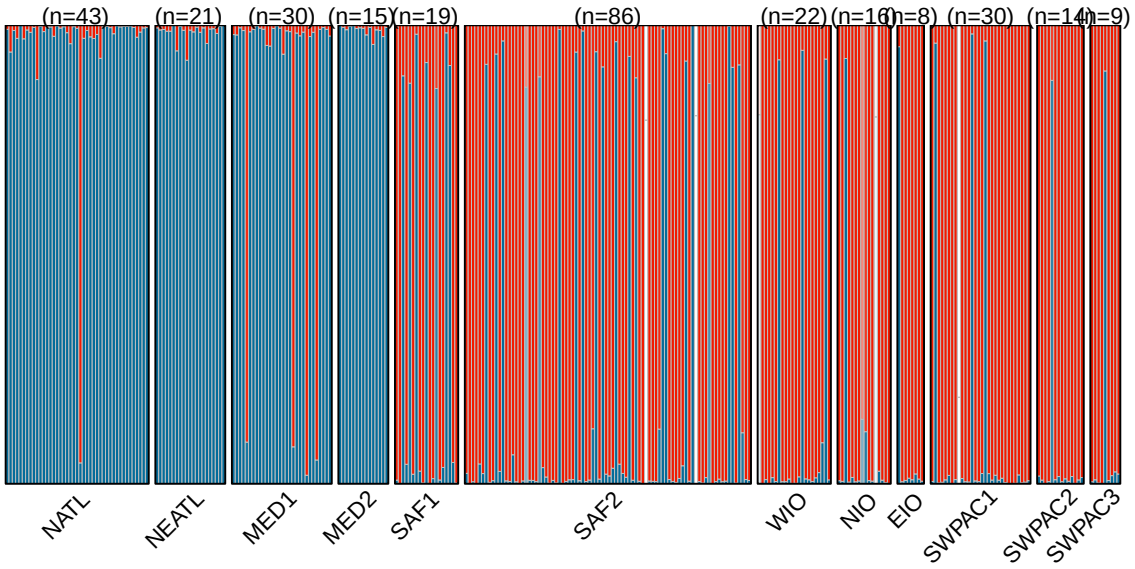
7_stockR

```
sample.grps <- attr(stockr.all, "sample.grps")
stock.all.object.names <- c()
for (K in 2:6) {
  stock.all <- stockR::stockSTRUCTURE(
    SNPdata = stockr.all,
    K = K,
  )

  stockBOOT.all <- stockR::stockBOOT(stock.all, B = 100,
                                     mc.cores = parallel::detectCores() - 2)
  assign(paste0("stockBOOT.all", K),value = stockBOOT.all)
  stock.all.object.names <- c(stock.all.object.names,
                                      paste0("stockBOOT.all", K))
}
save(list=stock.all.object.names , stock.all.object.names,
     file = "Rdata/stockR.ALL_new_strata3.Rdata")
```

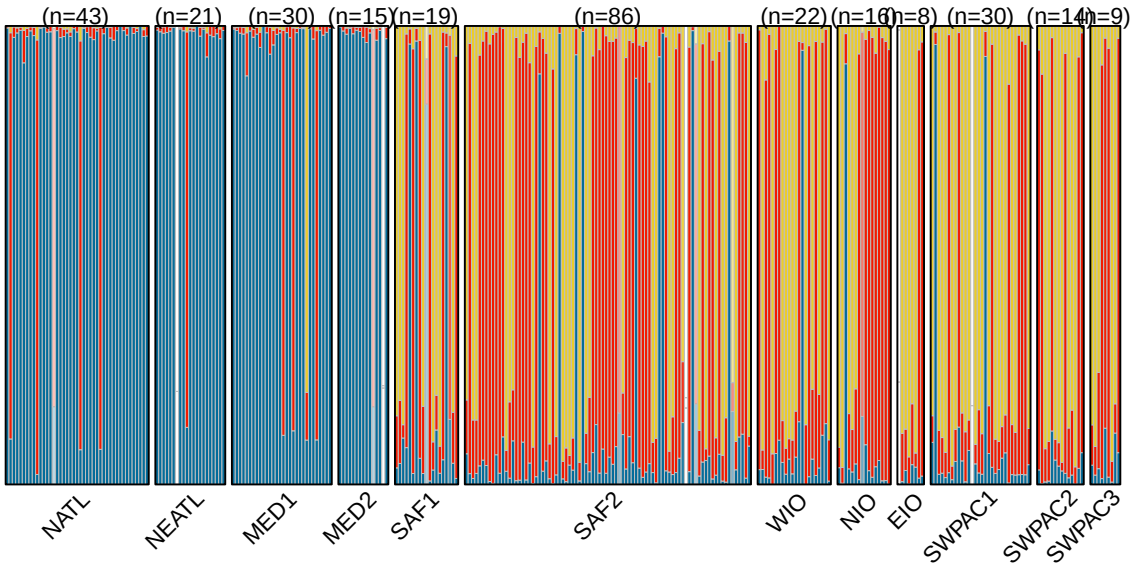
```
load("Rdata/stockR.ALL_new_strata3.Rdata")
grps <- attr(stockr.all, "grps")

for (K in 2:6) {
  colour <- colours6[1:K]
  plot.stockBOOT.object(
    get(stock.all.object.names[K - 1]),
    locations = data.frame(grps,
    order = as.integer(grps), las = 2),
    region.lwd = 1,
    CI.width = 0.70,
    colour = colour,
    plotTitle = paste0("stockR: Blue shark data for K=", K)
  )
  dev.print(
    device = png,
    file = paste0("figures/4.StockR_barplot_new_strata_33939SNPs_K",K,".png"),
    width = 30,
    height = 15,
    units = "cm",
    res = 300
  )
}
```

stockR barplot

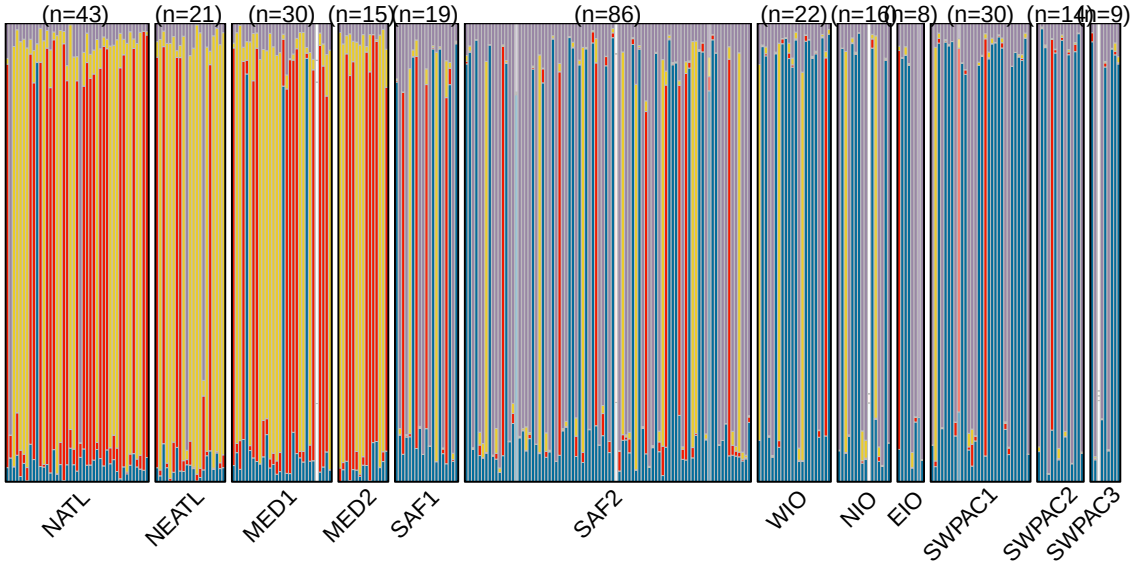
stockR: Blue shark data for K=2



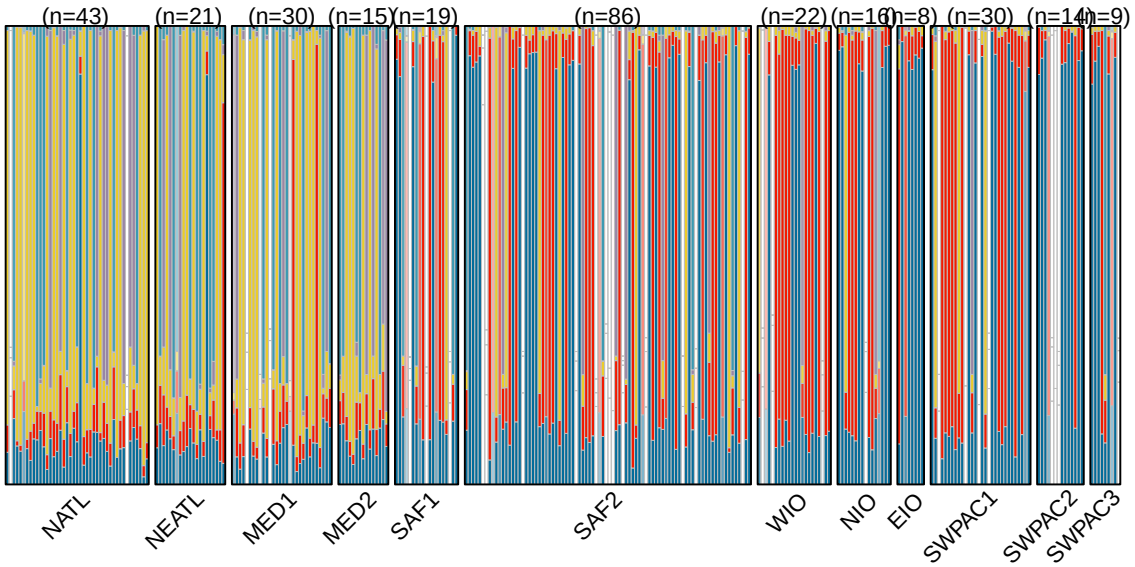
stockR: Blue shark data for K=3



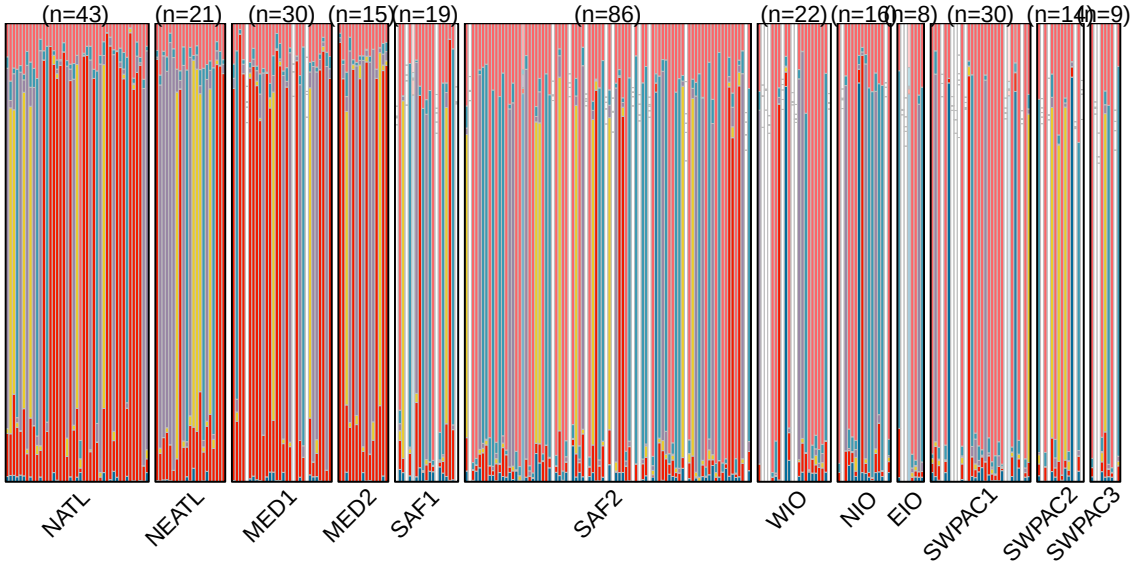
stockR: Blue shark data for K=4



stockR: Blue shark data for K=5



stockR: Blue shark data for K=6



8_ADMIXTURE

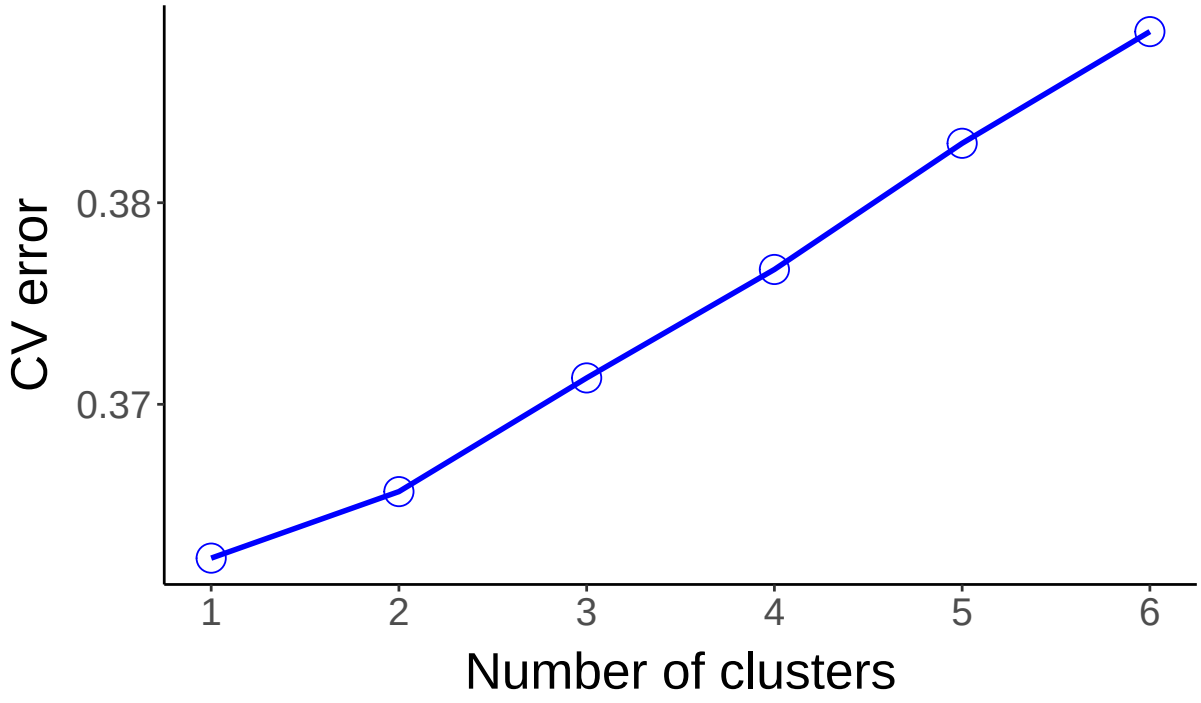
```
gl2Adm(gl.all, filename = "outfile/BS_all_ADMIXTURE_NEW3")
# gl2Adm(gl.all[gl.all$pop %in%  c("SAF1", "SAF2"),],
#        filename = "outfile/BS_SAF_ADMIXTURE_NEW3")
# gl2Adm(gl.all[gl.all$pop %in% c("NATL","NEATL", "MED1", "MED2", "SAF1", "SAF2"),],
#        filename = "outfile/BS_ATL_ADMIXTURE_NEW3")
# gl2Adm(gl.all[!gl.all$pop %in%
#                 c("SAF1", "SAF2", "WIO", "NIO", "EIO", "SWPAC1", "SWPAC2", "SWPAC3"),],
#        filename = "outfile/BS_IO_ADMIXTURE_NEW3")
```

ADMIXTURE with all markers per site The ADMIXTURE software was run in a Linux environment with the following command line:

```
for K in 1 2 3 4 5 6; \
do ./admixture -B10000 --cv=100 BS_all_ADMIXTURE.ped $K | tee log${K}.out; done
```

```
CV <- c()
for (K in 1:6) {
  log <- data.table::fread(input =  paste0("./Rdata/ADMIXTURE_new_strata3/log",K,".out",
                                           sep = ""), sep = "", header = FALSE,
                           blank.lines.skip = TRUE)
  log <- log$V1
  CV1 <- subset(log, grepl(pattern = "CV error", x = log, perl = TRUE))
  CV2 <- unlist(strsplit(CV1, " "))
  CV[K] <- as.numeric(CV2[4])
}

y <- CV
x <- 1:6
cvdata <- data.frame(x,y,stringsAsFactors = F)
plot <- ggplot2::ggplot(cvdata, ggplot2::aes(x,y)) +
  ggplot2::geom_point(size = 5, shape = 1, color = "blue") +
  ggplot2::geom_line(size = 1,color = "blue") +
  ggplot2::scale_x_continuous(name = waiver(),
                              breaks = seq(from = 0, to = 8, by = 1)) +
  ggplot2::labs(subtitle = "",
                y = "CV error",
                x = "Number of clusters",
                title = "",
                caption = "") +
  ggplot2::theme_classic() +
  ggplot2::theme(
    axis.text = ggplot2::element_text(size = 15),
    axis.title.x = ggplot2::element_text(size = 20,vjust = 0, hjust = 0.5),
    axis.title.y = ggplot2::element_text(size = 20,vjust = 2, hjust = 0.5))
print(plot)
```

CVplot

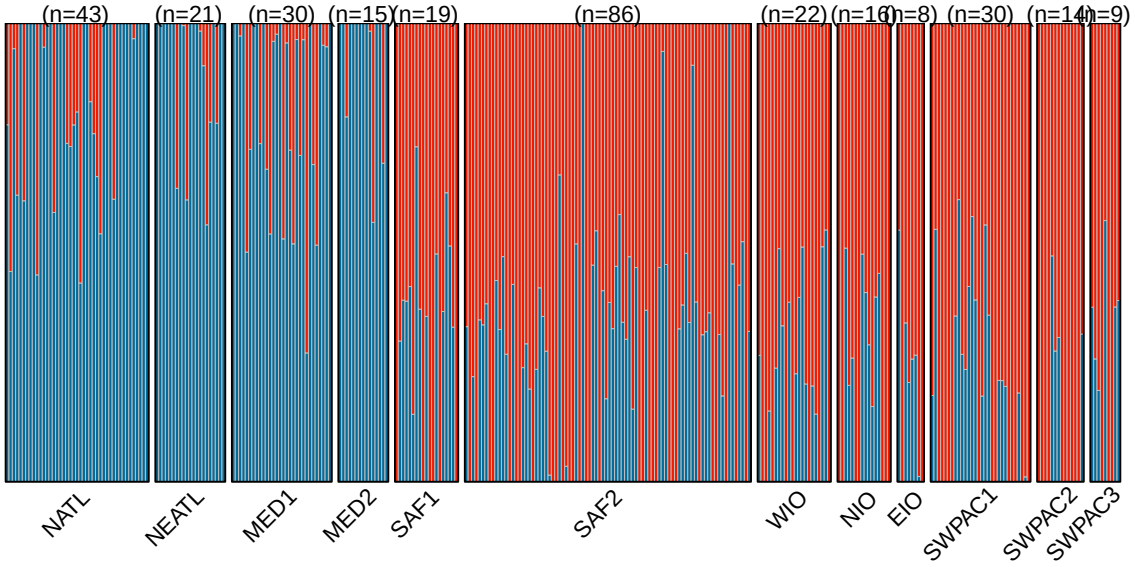



```
ggsave(filename = "./figures/9.ADMIXTURE_CVlot_new_strata_33939SNPs.png",
       plot = plot, width = 15, height = 15, units = "cm")
```

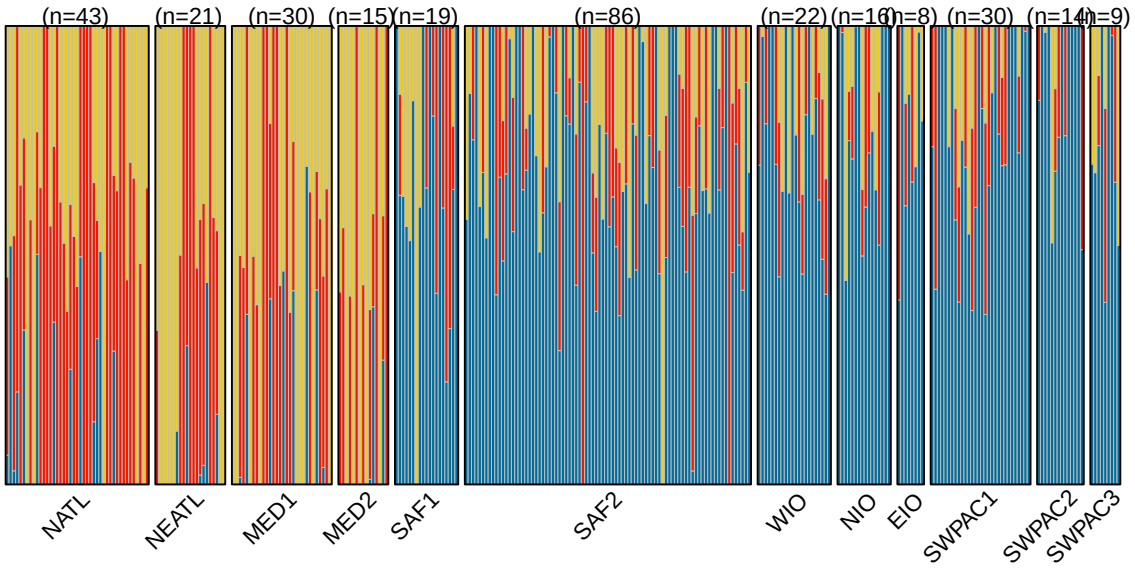
```
Ind.Names <- read.table("./Rdata/ADMIXTURE_new_strata3/BS_all_ADMIXTURE_NEW3.ped")

for (K in 2:6) {
  colour <- colours6[1:K]
  tbl <- read.table(paste0("./Rdata/ADMIXTURE_new_strata3/BS_all_ADMIXTURE_NEW3.",K,".Q",
                           sep = ""))
  row.names(tbl) <- Ind.Names$V2
  colnames(tbl) <- paste0("Group", 1:ncol(tbl))
  tbl <- as.matrix(tbl)
  locations <- gl.all$pop[order(match(gl.all$ind.names,Ind.Names$V2))]
  plot.admixture.FDD(x = tbl, locations = locations,
                     colour = colour, region.lwd = 1,
                     plotTitle = paste0("ADMIXTURE: K=",K))
  dev.print(
    device = png,
    file = paste0("./figures/9.ADMIXTURE_barplot_new_strata_33939SNPs_K",
                  K,".png"),
    width = 30,
    height = 15,
    units = "cm",
    res = 300
  )
}
```

Barplot

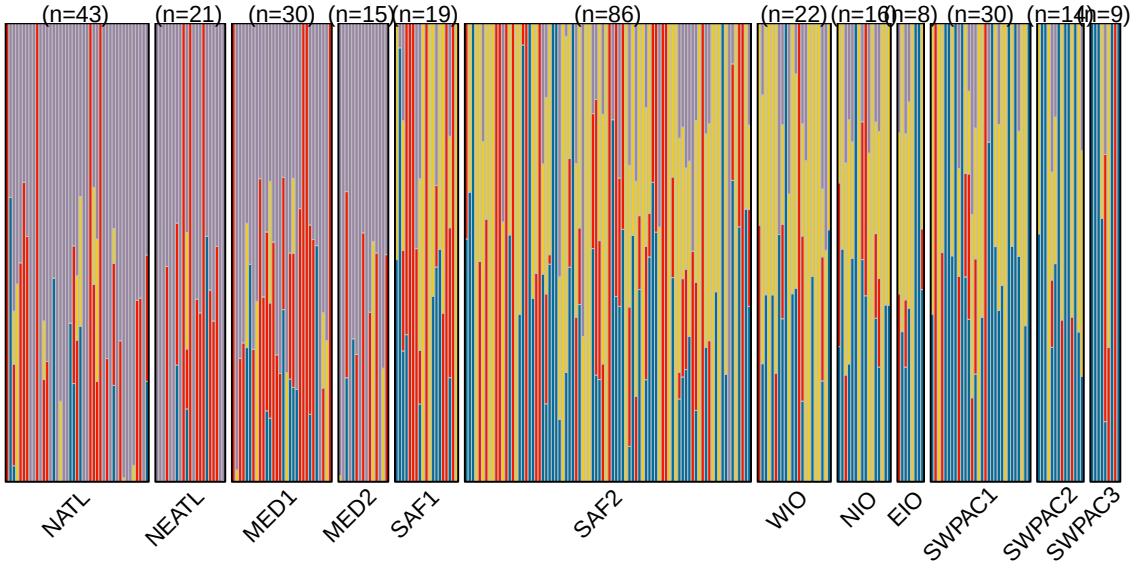
ADMIXTURE: K=2



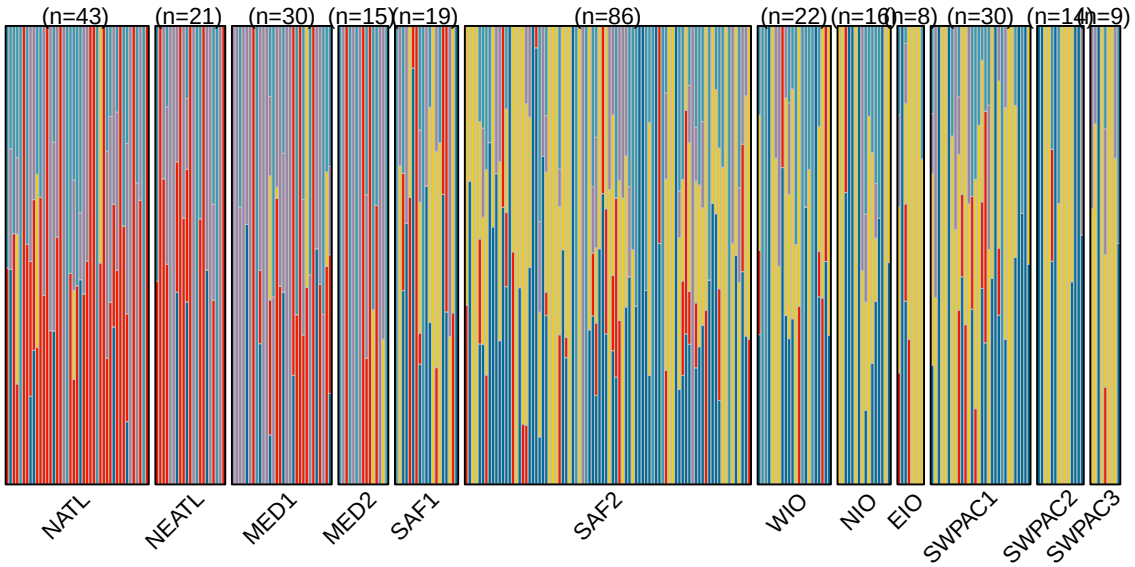
ADMIXTURE: K=3



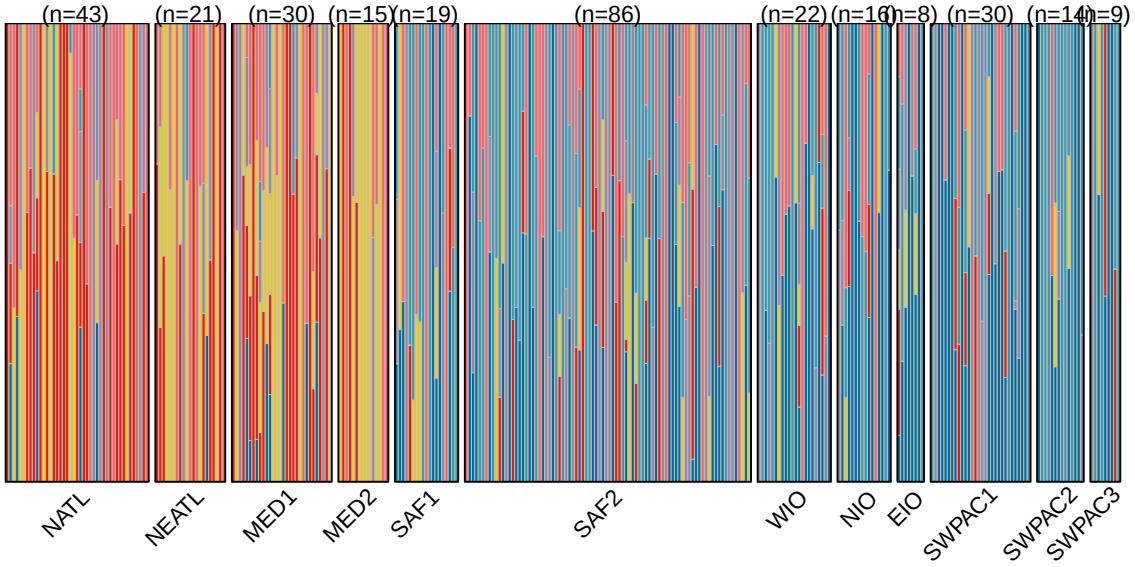
ADMIXTURE: K=4



ADMIXTURE: K=5



ADMIXTURE: K=6



Data analysis by geographic region

Result: samples from MED1/MED2, SAF1/SAF2 and SWPAC1/SWPAC2/SWPAC3 could be grouped, after correcting for multiple pairwise tests.

Read and filter data

Note: Filtering was performed interactively with the `radiator::filter_rad` function.

Filtering thresholds

- Start: 172384 SNPs, 364 individuals
- Filter reproducibility: 0.959: 156195 SNPs, 364 inds
- Filter monomorphic markers: 156195 SNPs, 364 inds
- Filter common markers: 142272 SNPs, 364 inds
- Filter individual missingness: outliers: 142272 SNPs,332 inds
- Filter individual heterozygosity: outliers (0.060 / 0.078): 142272 SNPs, 312 inds
- Filter monomorphic markers: 142272-5624 SNPs, 312 inds
- Filter marker minor allele count (MAC): 4: 110261 SNPs, 312 inds
- Filter marker coverage: 7-200: 95216 SNPs, 312 inds
- Filter marker missingness: 0.1: 68083 SNPs, 312 inds
- SNP position on sequence: outliers: 1SNP per 8bp: all: 68083 SNPs, 312 inds
- SNPs per read: 4: 66837 SNPs, 312 inds
- short ld: based on MAC: 45889 SNPs, 312 inds
- Filter individual heterozygosity (mixed genomes): 0.117 - 0.15: 45889 SNPs, 312 inds
- Duplicate genomes: No
- Filter HWE: minimum 3 populations, pvalue < 0.0001: 45810 SNPs, 312 inds

```
data <- "infiles/blue_shark_SNP_genotyped.csv"
blue_shark.all <- radiator::filter_rad(
  data = data, strata = strata,
  output = "tidy", interactive.filter = TRUE, parallel.core = 22)

tidy <- blue_shark.all$tidy
save(tidy, file = "Rdata/blue_shark_filtered.Rdata")
```

Remove sex-linked markers

```
data <- "infiles/blue_shark_SNP_genotyped.csv"
BS.sex <- radiator::sexy_markers(data = data, silicodata = NULL,
                                 strata = strata, parallel.core = 28)

BS.sex.markers <- BS.sex$sexy.summary$CLONE_ID
BS.y <- BS.sex$sexy.summary$CLONE_ID[BS.sex$sexy.summary$MARKER_TYPE ==
```

```
                                        "Heterogametic_sex-marker"]
BS.x <- BS.sex$sexy.summary$CLONE_ID[BS.sex$sexy.summary$MARKER_TYPE ==
                                        "Homogametic_sex-marker"]
BS.x.overlap.markers <- BS.x[duplicated(BS.x)] # markers found by 2 methods = reliable
```

```
load("Rdata/blue_shark_filtered.Rdata")
load("Rdata/blue_shark_sex_markers.Rdata")
length(unique(tidy$LOCUS[tidy$LOCUS %in% BS.sex.markers])) #143
length(unique(tidy$LOCUS[tidy$LOCUS %in% BS.x.overlap.markers])) #36
tidy <- tidy[!tidy$LOCUS %in% BS.sex.markers,]
```

Convert to other data formats

```
BS_all_convert <- radiator::genomic_converter(
  tidy,
  output = c("genlight", "gtypes", "pcadapt", "stockr", "structure",
             "faststructure"))

tidy.all <- BS_all_convert$tidy.all.data
gl.all <- BS_all_convert$genlight
gt.all <- BS_all_convert$gtypes
stockr.all <- BS_all_convert$stockr
pcadapt.all <- BS_all_convert$pcadapt
```

```
pop.levels <- c("Atlantic-N","Atlantic-NE", "Mediterranean","Atlantic-SE",
                "Indian_Ocean-SW", "Indian_Ocean-N","Indian_Ocean-EC",
                "Pacific-SW")

tidy.all$POP_ID <- factor(tidy.all$POP_ID, levels = pop.levels)

gl.all$pop <- factor(gl.all$pop, levels = pop.levels)

attr(stockr.all, "grps") <- factor(attr(stockr.all, "grps"), levels = pop.levels)

pcadapt.all$pop.string <- factor(pcadapt.all$pop.string, levels = pop.levels)

meta <- read.csv("infiles/BLUE_SHARK_META_GENETIC_SEX.csv")
meta$INDIVIDUALS <- stringr::str_replace_all(meta$INDIVIDUALS, "_", "-")
meta$INDIVIDUALS <- stringr::str_replace_all(meta$INDIVIDUALS, " ", "")
meta <- meta[meta$INDIVIDUALS %in% gl.all$ind.names, ]
meta <- meta[order(match(meta$INDIVIDUALS, gl.all$ind.names)), ]

latlong <- data.frame(lat = meta$lat, lon = meta$lon)
rownames(latlong) <- meta$INDIVIDUALS
gl.all$other$ind.metrics <- meta
gl.all$other$loc.metrics <- data.frame(AlleleID = unique(gt.all@data$locus),
                                       CloneID = gl.all$loc.names,
                                       uid = gl.all$loc.names)
gl.all$loc.names <- unique(gt.all@data$locus)
gl.all$other$latlong <- latlong
gt.all@schemes <- meta
```

```
save(tidy.all,
     gl.all,
     gt.all,
     stockr.all,
     pcadapt.all,
     structure.all,
     faststructure.all,
     file = "Rdata/Blue_shark_AllData.Rdata")
```

Add correct levels and metadata

Identify outliers loci

OutFLANK

OutFLANk identifies 67 outliers

```
load("Rdata/Blue_shark_AllData.Rdata")

gl <- gl.all
SNPmat <- as.matrix(gl)
colnames(SNPmat) <- NULL
row.names(SNPmat) <- NULL
SNPmat[is.na(SNPmat)] <- 9
FstDataFrame <- OutFLANK::MakeDiploidFSTMat(SNPmat, gl@loc.names,
                                            as.character(gl@pop))
plot(FstDataFrame$FST, FstDataFrame$FSTNoCorr, xlim = c(-0.01,0.3),
     ylim = c(-0.01, 0.3), pch = 20) + abline(0, 1)

Outliers <- OutFLANK::OutFLANK(FstDataFrame, NumberOfSamples = length(levels(gl@pop)),
                               qthreshold = 0.01, LeftTrimFraction = 0.01,
                               RightTrimFraction = 0.1,
                               Hmin = 0.0001)
OutFLANK::OutFLANKResultsPlotter(Outliers, withOutliers = TRUE, NoCorr = TRUE,
                                 Hmin = 0.01, binwidth = 0.005, Zoom = FALSE,
                                 RightZoomFraction = 0.05,
                                 titletext = NULL)

Outliers$numberLowFstOutliers
Outliers$numberHighFstOutliers
OutflankNames <- Outliers$results$LocusName[Outliers$results$OutlierFlag ==
                                              TRUE &
                                              !is.na(Outliers$results$OutlierFlag)]
print(OutflankNames)
```

PCadapt

PCadapt identifies 8196 outliers

```
poplist <- pcadapt.all$pop.string
data <- pcadapt.all$genotype.matrix
data <- pcadapt::read.pcadapt(data)
K <- 25
x <- pcadapt::pcadapt(data, K = K, min.maf = 0.005)
x$singular.values
```

```
plot(x, option = "screeplot") #K = 2-3

plot(x, option = "scores", i = 1, j = 2, pop = poplist)
plot(x, option = "scores", i = 1, j = 3, pop = poplist)
plot(x, option = "scores", i = 2, j = 3, pop = poplist)
plot(x, option = "scores", i = 3, j = 4, pop = poplist)

K <- 2
x <- pcadapt::pcadapt(data, K = K, min.maf = 0.005)
x$singular.values
plot(x, option = "manhattan")
plot(x, option = "qqplot", threshold = 0.05)
hist(x$pvalues, xlab = "p-values", main = NULL, breaks = 50)
plot(x, option = "stat.distribution")
qval <- qvalue::qvalue(x$pvalues)$qvalues
alpha <- 0.01
pcadapt.outliers <- which(qval < alpha)
length(pcadapt.outliers)
PCadaptkNames <- gl.all$loc.names[pcadapt.outliers]
print(PCadaptkNames)
```

Subset outliers commmon between PCAdapt and OUTFlank

We found 9 common outliers

```
sum(PCadaptkNames %in% OutflankNames)
sum(OutflankNames %in% PCadaptkNames)

outliers_both <-intersect(OutflankNames, PCadaptkNames)
outliers_both


dim(tidy.all)#14248104 16
tidy.all.neutral <- tidy.all[!tidy.all$MARKERS %in% outliers_both,]
dim(tidy.all.neutral)#14245296 16
tidy.all.outlier <- tidy.all[tidy.all$MARKERS %in% outliers_both,]
dim(tidy.all.outlier)#2808 16


removed <- gl.all$loc.names %in% outliers_both

gl.all.neutral <- gl.all[, !removed]
gl.all.neutral$other$loc.metrics <- gl.all.neutral$other$loc.metrics[!removed, ]

gl.all.outlier <- gl.all[, removed]
gl.all.outlier$other$loc.metrics <- gl.all.outlier$other$loc.metrics[removed, ]

gt.all.neutral <- gt.all[, !removed]
gt.all.outlier <- gt.all[, removed]
```

Supplemental monomorphic and MAF filtering

After careful filtering many markers remained and several were monomorphic. Consequently, we opted to suplement out filtering with additional monomorphic and minor allele frequency filtering steps.

- We found 2,832 markers that were monomorphic

- We found 5,171 markers with a frequency below 0.01

```
gl.all.neutral  <- dartR::gl.filter.monomorphs(gl.all.neutral, v = 0)
gl.all.neutral <- dartR::gl.filter.maf(gl.all.neutral,threshold = 0.01, v = 0)
gl.all.neutral  <- dartR::gl.recalc.metrics(gl.all.neutral, v = 0)

tidy.all.neutral <- tidy.all.neutral[tidy.all.neutral$MARKERS %in%
                                       gl.all.neutral$loc.names,]
gt.all.neutral <- gt.all.neutral[,gl.all.neutral$loc.names]

save(tidy.all.neutral,
     tidy.all.outlier,
     gl.all.neutral,
     gl.all.outlier,
     gt.all.neutral,
     gt.all.outlier,
     file = "Rdata/ALL_neutral_outlier_data.Rdata")
```

Convert to other data formats without outliers and sex markers

```
load("Rdata/ALL_neutral_outlier_data.Rdata")
BS_all_convert_neutral <- radiator::genomic_converter(
  tidy.all.neutral,
  output = c("stockr", "structure", "faststructure", "snprelate","vcf"))

stockr.all.neutral <- BS_all_convert_neutral$stockr
snprelate.all.neutral <- BS_all_convert_neutral$snprelate


BS_all_convert_outlier <- radiator::genomic_converter(
  tidy.all.outlier,
  output = c("stockr", "structure", "faststructure", "snprelate","vcf"))

stockr.all.outlier <- BS_all_convert_outlier$stockr
snprelate.all.outlier <- BS_all_convert_outlier$snprelate
```

```
strata2 <- dplyr::filter(strata, TARGET_ID %in%
                           gl.all.neutral$other$ind.metrics$TARGET_ID)
strata2 <- strata2[order(match(strata2$TARGET_ID,
                               gl.all.neutral$other$ind.metrics$TARGET_ID)),]
strata2$INDIINDIVIDUALS <- stringr::str_replace_all(string = INDIVIDUALS2,
                                                    pattern = "_ |_", replacement = "-")
pop.levels2 <- c("NATL", "NEATL", "MED1", "MED2", "SAF1", "SAF2", "WIO",
                 "NIO", "EIO", "SWPAC1", "SWPAC2", "SWPAC3")
strata2$STRATA <- factor(strata2$STRATA, levels = pop.levels2)
gl.all.neutral$other$ind.metrics <- strata2
gi.all.neutral$other$ind.metrics <- strata2
gt.all.neutral@schemes <- strata2

gl.all.outlier$other$ind.metrics <- strata2
gi.all.outlier$other$ind.metrics <- strata2
gt.all.outlier@schemes <- strata2
```

```
strata2 %<>% dplyr::rename(INDIVIDUALS2 = INDIVIDUALS) %>%
  dplyr::mutate(INDIVIDUALS = stringr::str_replace_all(string = INDIVIDUALS2,
                                                       pattern = "_ |_",
                                                       replacement = "-") )
tidy.all.neutral2 <- dplyr::left_join(tidy.all.neutral, strata2, by = "INDIVIDUALS")
tidy.all.outlier2 <- dplyr::left_join(tidy.all.outlier, strata2, by = "INDIVIDUALS")

tidy.all.neutral2$POP_ID <- NULL
convert <- radiator::genomic_converter(tidy.all.neutral2, strata = strata2,
                                       output = "gtypes")
gt.all.neutral <- convert$gtypes

tidy.all.outlier2$POP_ID <- NULL
convert <- radiator::genomic_converter(tidy.all.outlier2, strata = strata2,
                                       output = "gtypes")
gt.all.outlier <- convert$gtypes

gl.all.neutral$pop <- gl.all.neutral$other$ind.metrics$STRATA
gi.all.neutral@pop <- gi.all.neutral$other$ind.metrics$STRATA
gl.all.outlier$pop <- gl.all.neutral$other$ind.metrics$STRATA
gi.all.outlier@pop <- gi.all.neutral$other$ind.metrics$STRATA

strat <- as.character(gt.all.neutral@schemes$STRATA)
names(strat) <- stringr::str_replace_all(string = gt.all.neutral@schemes$INDIVIDUALS,
                                         pattern = "_ |_", replacement = "-")
setStrata(gt.all.neutral) <- strat

unique(gt.all.neutral@data$id) %in%
  stringr::str_replace_all(string = gt.all.neutral@schemes$INDIVIDUALS,
                           pattern = "_ |_", replacement = "-")

radiator::write_vcf(
  tidy.all.neutral,
  pop.info = FALSE,
  filename = "BlueShark_312ind_37655_neutral_SNPs",
  source = NULL,
  empty = FALSE
)
radiator::write_rad(
  tidy.all.neutral,
  path = "./",
  filename = "BlueShark_312ind_37655_neutral_SNPs",
  tsv = TRUE,
  internal = FALSE,
  append = FALSE,
  col.names = TRUE,
  write.message = "standard",
  verbose = FALSE
)

radiator::write_vcf(
  tidy.all.outlier,
  pop.info = FALSE,
  filename = "BlueShark_312ind_9_outlier_SNPs",
  source = NULL,
  empty = FALSE
)
```

```
radiator::write_rad(
  tidy.all.outlier,
  path = "./",
  filename = "BlueShark_312ind_9_outlier_SNPs.rad",
  tsv = TRUE,
  internal = FALSE,
  append = FALSE,
  col.names = TRUE,
  write.message = "standard",
  verbose = FALSE
)

save(tidy.all.neutral,
     gl.all.neutral,
     gi.all.neutral,
     gt.all.neutral,
     stockr.all.neutral,
     snprelate.all.neutral,
     tidy.all.outlier,
     gl.all.outlier,
     gi.all.outlier,
     gt.all.outlier,
     stockr.all.outlier,
     snprelate.all.outlier,
     file = "Rdata/Blue_shark_Neutral_and_OutlierData_Corrected2.Rdata")
```

Add correct levels and metadata

Load filtered data without sex-linked markers

```
load("Rdata/Blue_shark_Neutral_and_OutlierData_Corrected2.Rdata")

adegenet::nLoc(gl.all.neutral)
```

```
## [1] 37655
```

```
adegenet::nLoc(gl.all.outlier)
```

```
## [1] 9
```

```
adegenet::nInd(gl.all.neutral)
```

```
## [1] 312
```

```
pop.levels <- c("NATL", "NEATL", "MED", "SAF", "WIO", "NIO", "EIO", "SWPAC")
gl.all.neutral$other$ind.metrics$STRATA2 <- factor(gl.all.neutral$other$ind.metrics$STRATA2, levels
gl.all.neutral$pop <- gl.all.neutral$other$ind.metrics$STRATA2

gl.all.outlier$other$ind.metrics$STRATA2 <- factor(gl.all.outlier$other$ind.metrics$STRATA2, levels
gl.all.outlier$pop <- gl.all.outlier$other$ind.metrics$STRATA2

knitr::kable(table(gl.all.neutral$other$ind.metrics$GENETIC_SEX,
                   gl.all.neutral$other$ind.metrics$STRATA2),
             col.names = c(shortnames2),
             caption = "Summary: Number of sharks by location and sex")
```

Table 6: Summary: Number of sharks by location and sex

	NATL	NEATL	MED	SAF	WIO	NIO	EIO	SWPAC
F	40	12	29	22	13	1	3	29
M	2	9	16	83	9	15	5	24

Table 7: Genetic diversity for neutral loci

	NATL	NEATL	MED	SAF	WIO	NIO	EIO	SWPAC
ar	1.594	1.568	1.583	1.583	1.562	1.547	1.506	1.581
size	41.400	20.519	44.369	103.498	21.687	15.837	7.917	52.123
obs_het	0.145	0.142	0.142	0.140	0.141	0.143	0.145	0.139
exp_het	0.168	0.166	0.166	0.166	0.163	0.161	0.157	0.166
uexp_het	0.170	0.170	0.168	0.166	0.167	0.166	0.167	0.167
fis	0.091	0.102	0.099	0.115	0.088	0.067	0.031	0.113
hwe_glb	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000
hwe_hom	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
hwe_het	1.000	1.000	0.000	0.000	1.000	1.000	1.000	0.000
fis_lo	0.074	0.069	0.083	0.103	0.059	0.028	-0.067	0.096
fis_hi	0.088	0.094	0.096	0.113	0.081	0.057	0.021	0.110
ar_lo	1.539	1.496	1.532	1.531	1.488	1.485	1.440	1.530
ar_hi	1.622	1.612	1.608	1.602	1.605	1.601	1.577	1.606

Basic data analysis by geographic region

1_Diversity Table

```
gl2gpop(gl.all.neutral, filename = "Rdata/Genepop_file_37655SNP_Neutral_312ind.txt")

result <- diveRsity::basicStats(
  infile = "Rdata/Genepop_file_37655SNP_Neutral_312ind.txt",
  outfile = "outfile/Result_Diversity.txt", fis_ci = TRUE, ar_ci = TRUE,
  fis_boots = 1000, ar_boots = 1000, mc_reps = 1000, rarefaction = FALSE,
  ar_alpha = 0.05, fis_alpha = 0.05)

gendiv.neutral <- data.frame(
  "Atlantic-N" = result$main_tab$`Pg-AZO-105`$overall,
  "Atlantic-NE" = result$main_tab$`Pg-T100`$overall,
  "Mediterranean" = result$main_tab$`Pg-GDL-06`$overall,
  "Atlantic-SE" = result$main_tab$`26300`$overall,
  "IndianOcean-SW" = result$main_tab$`33113`$overall,
  "IndianOcean-N" = result$main_tab$B383$overall,
  "IndianOcean-E" = result$main_tab$`RITF-Bx1-A08`$overall,
  "Pacific-SW" = result$main_tab$`NZ-1.2`$overall
)
rownames(gendiv.neutral) <- rownames(result$main_tab$`Pg-AZO-105`)
save(gendiv.neutral, file = "Rdata/ALL_Genetic_diversity.Rdata")
readr::write_tsv(gendiv.neutral, path = "outfile/Table_Diversity_neutral_perPop.txt")
```

```
load("Rdata/ALL_Genetic_diversity.Rdata")
knitr::kable(gendiv.neutral, col.names = shortnames2, digits = 3,
             caption = "Genetic diversity for neutral loci")
```

Neutral data

Table 8: Genetic diversity for outlier loci

	NATL	NEATL	MED	SAF	WIO	NIO	EIO	SWPAC
ar	1.512	1.379	1.463	1.665	1.551	1.757	1.701	1.566
size	40.778	19.556	42.889	101.889	21.333	15.556	7.556	50.222
obs_het	0.140	0.108	0.106	0.192	0.177	0.207	0.157	0.164
exp_het	0.151	0.127	0.141	0.233	0.212	0.269	0.277	0.231
uexp_het	0.152	0.131	0.143	0.234	0.217	0.278	0.296	0.233
fis	0.040	0.177	0.217	0.113	0.102	0.271	0.381	0.214
hwe_glb	0.513	0.736	0.043	0.033	0.973	0.639	0.828	0.026
hwe_hom	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
hwe_het	0.725	0.843	0.049	0.004	0.882	0.327	0.417	0.007
fis_lo	-0.053	-0.143	0.013	0.051	-0.024	-0.057	0.058	0.118
fis_hi	0.123	0.390	0.404	0.202	0.235	0.419	0.519	0.370
ar_lo	1.222	1.111	1.111	1.553	1.444	1.556	1.556	1.444
ar_hi	1.778	1.556	1.667	1.778	1.667	1.889	1.778	1.667

```
gl2gpop(gl.all.outlier, filename = "Rdata/Genepop_file_9SNP_Outlier_312ind.txt")

result <- diveRsity::basicStats(
  infile = "Rdata/Genepop_file_9SNP_Outlier_312ind.txt",
  outfile = "outfile/Result_Diversity.txt", fis_ci = TRUE, ar_ci = TRUE,
  fis_boots = 1000, ar_boots = 1000, mc_reps = 1000, rarefaction = FALSE,
  ar_alpha = 0.05, fis_alpha = 0.05)

gendiv.outlier <- data.frame(
  "Atlantic-N" = result$main_tab$`Pg-AZO-105`$overall,
  "Atlantic-NE" = result$main_tab$`Pg-T100`$overall,
  "Mediterranean" = result$main_tab$`Pg-GDL-06`$overall,
  "Atlantic-SE" = result$main_tab$`26300`$overall,
  "IndianOcean-SW" = result$main_tab$`33113`$overall,
  "IndianOcean-N" = result$main_tab$B383$overall,
  "IndianOcean-E" = result$main_tab$`RITF-Bx1-A08`$overall,
  "Pacific-SW" = result$main_tab$`NZ-1.2`$overall
)
rownames(gendiv.outlier) <- rownames(result$main_tab$`Pg-AZO-105`)
save(gendiv.neutral,gendiv.outlier, file = "Rdata/ALL_Genetic_diversity.Rdata")
readr::write_tsv(gendiv.outlier, path = "outfile/Table_Diversity_Outlier_perPop.txt")
```

```
load("Rdata/ALL_Genetic_diversity.Rdata")
knitr::kable(gendiv.outlier, col.names = shortnames2, digits = 3,
             caption = "Genetic diversity for outlier loci")
```

Outlier data

2__Fst

```
pop.levels <- c("NATL", "NEATL", "MED", "SAF", "WIO", "NIO", "EIO", "SWPAC")
gl.all.neutral <- gl.all.neutral[order(gl.all.neutral$pop,
                                       gl.all.neutral@other$ind.metrics$GENETIC_SEX),]
Fst.region.stampp.neutral <- StAMPP::stamppFst(gl.all.neutral, nboots = 10000, percent = 95,
                                       nclusters = parallel::detectCores() - 2)

strat <- as.character(gl.all.neutral$pop)
names(strat) <- gl.all.neutral$ind.names
strataG::setStrata(gt.all.neutral) <- strat

Fst.region.stratag.neutral <- strataG::pairwiseTest(
  gt.all.neutral,
  nrep = 10000,
  by.locus = FALSE,
  hap.locus = 1,
  quietly = FALSE,
  max.cores = parallel::detectCores() - 2,
)


gl.all.outlier <- gl.all.outlier[order(gl.all.outlier$pop,
                                       gl.all.outlier@other$ind.metrics$GENETIC_SEX),]
gt.all.outlier@data$stratum
Fst.region.stampp.outlier <- StAMPP::stamppFst(gl.all.outlier, nboots = 10000, percent = 95,
                                       nclusters = parallel::detectCores() - 2)

strat <- as.character(gl.all.outlier$pop)
names(strat) <- gl.all.outlier$ind.names
strataG::setStrata(gt.all.outlier) <- strat

Fst.region.stratag.outlier <- strataG::pairwiseTest(
  gt.all.outlier,
  nrep = 10000,
  by.locus = FALSE,
  hap.locus = 1,
  quietly = FALSE,
  max.cores = parallel::detectCores() - 2,
)



sumres.Fst.neutral <- strataG::pairwiseSummary(Fst.region.stratag.neutral, locus = "All")
sumres.Fst.neutral$strata.1 <- factor(sumres.Fst.neutral$strata.1, levels = pop.levels)
sumres.Fst.neutral$strata.2 <- factor(sumres.Fst.neutral$strata.2, levels = pop.levels)
sumres.Fst.neutral <- sumres.Fst.neutral[order(sumres.Fst.neutral$strata.1,
                                               sumres.Fst.neutral$strata.2),]
readr::write_csv(sumres.Fst.neutral,
                 file = "outfile/Blue_shark_FST_summary_neutral_by_region2.csv")
m2 <- sumres.Fst.neutral[order(match(sumres.Fst.neutral$strata.1,pop.levels)),c(2,3,20)]
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.1 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst", type = 'lower')
m2 <- sumres.Fst.neutral[order(match(sumres.Fst.neutral$strata.1,pop.levels)),c(2,3,21)]
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.2 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
```

```
                                    attr = "Fst_p.val", type = 'lower')
m2$Fst.p.adj.fdr <- p.adjust(m2$Fst_p.val, "fdr")
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.3 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst.p.adj.fdr", type = 'lower')
m2$Fst.p.adj.bonf <- p.adjust(m2$Fst_p.val, "bonferroni")
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.4 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst.p.adj.bonf", type = 'lower')


sumres.Fst.neutral$Fst.p.adj.fdr <- m2$Fst.p.adj.fdr
sumres.Fst.neutral$Fst.p.adj.bonf <- m2$Fst.p.adj.bonf


write.csv(m2.1, file = "outfile/Blue_shark_FST_neutral_Pairwise_byRegion2.csv")
write.csv(m2.2, file = "outfile/Blue_shark_FST_neutral_Pval_Pairwise_byRegion2.csv")
write.csv(m2.3,
          file = "outfile/Blue_shark_FST_neutral_Pval_Pairwise_FDR_byRegion2.csv")
write.csv(m2.4,
          file = "outfile/Blue_shark_FST_neutral_Pval_Pairwise_bonferroni_byRegion2.csv")
write.csv(sumres.Fst.neutral,
          file = "outfile/Blue_shark_FST_neutral_byRegion2.csv")



sumres.Fst.outlier <- strataG::pairwiseSummary(Fst.region.stratag.outlier, locus = "All")
sumres.Fst.outlier$strata.1 <- factor(sumres.Fst.outlier$strata.1, levels = pop.levels)
sumres.Fst.outlier$strata.2 <- factor(sumres.Fst.outlier$strata.2, levels = pop.levels)
sumres.Fst.outlier <- sumres.Fst.outlier[order(sumres.Fst.outlier$strata.1,
                                               sumres.Fst.outlier$strata.2),]
readr::write_csv(sumres.Fst.outlier,
                 file = "outfile/Blue_shark_FST_summary_outlier_by_Region2.csv")
m2 <- sumres.Fst.outlier[order(match(sumres.Fst.outlier$strata.1,pop.levels)),c(2,3,20)]
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.1 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst", type = 'lower')
m2 <- sumres.Fst.outlier[order(match(sumres.Fst.outlier$strata.1,pop.levels)),c(2,3,21)]
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.2 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst_p.val", type = 'lower')
m2$Fst.p.adj.fdr <- p.adjust(m2$Fst_p.val, "fdr")
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.3 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst.p.adj.fdr", type = 'lower')
m2$Fst.p.adj.bonf <- p.adjust(m2$Fst_p.val, "bonferroni")
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.4 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst.p.adj.bonf", type = 'lower')


sumres.Fst.outlier$Fst.p.adj.fdr <- m2$Fst.p.adj.fdr
sumres.Fst.outlier$Fst.p.adj.bonf <- m2$Fst.p.adj.bonf


write.csv(m2.1, file = "outfile/Blue_shark_FST_outlier_Pairwise_byRegion2.csv")
write.csv(m2.2, file = "outfile/Blue_shark_FST_outlier_Pval_Pairwise_byRegion2.csv")
```

Table 9: staMMP: Pairwise Fst for neutral loci by regions

	NATL	NEATL	MED	SAF	WIO	NIO	EIO	SWPAC
NATL	NA	NA	NA	NA	NA	NA	NA	NA
NEATL	0.0005	NA	NA	NA	NA	NA	NA	NA
MED	0.0007	0.0010	NA	NA	NA	NA	NA	NA
SAF	0.0012	0.0017	0.0015	NA	NA	NA	NA	NA
WIO	0.0016	0.0015	0.0017	0e+00	NA	NA	NA	NA
NIO	0.0011	0.0015	0.0017	-1e-04	-4e-04	NA	NA	NA
EIO	0.0011	0.0015	0.0023	0e+00	-3e-04	4e-04	NA	NA
SWPAC	0.0017	0.0020	0.0022	1e-04	-1e-04	-3e-04	-3e-04	NA

Table 10: staMMP: Pairwise Fst P-values Fst for neutral loci by regions

	NATL	NEATL	MED	SAF	WIO	NIO	EIO	SWPAC
NATL	NA	NA	NA	NA	NA	NA	NA	NA
NEATL	0.0062	NA	NA	NA	NA	NA	NA	NA
MED	0.0000	0.0000	NA	NA	NA	NA	NA	NA
SAF	0.0000	0.0000	0	NA	NA	NA	NA	NA
WIO	0.0000	0.0000	0	0.4447	NA	NA	NA	NA
NIO	0.0011	0.0003	0	0.7368	0.8879	NA	NA	NA
EIO	0.0178	0.0069	0	0.4819	0.7177	0.2221	NA	NA
SWPAC	0.0000	0.0000	0	0.2427	0.7522	0.9026	0.7337	NA

```
write.csv(m2.3,
          file = "outfile/Blue_shark_FST_outlier_Pval_Pairwise_FDR_byRegion2.csv")
write.csv(m2.4,
          file = "outfile/Blue_shark_FST_outlier_Pval_Pairwise_bonferroni_byRegion2.csv")
write.csv(sumres.Fst.outlier,
          file = "outfile/Blue_shark_FST_outlier_byRegion2.csv")

save(Fst.region.stampp.neutral,
     Fst.region.stratag.neutral,
     Fst.region.stampp.outlier, Fst.region.stratag.outlier,
     sumres.Fst.neutral, sumres.Fst.outlier,
     file = "Rdata/fst.ALL.neutral.outlier2.Rdata")
```

```
load("Rdata/fst.ALL.neutral.outlier2.Rdata")

knitr::kable(
  Fst.region.stampp.neutral$Fsts, digits = 4,
  caption = "staMMP: Pairwise  Fst for neutral loci by regions") %>%
  kableExtra::kable_styling(full_width = FALSE, font_size = 7)
```

```
knitr::kable(
  Fst.region.stampp.neutral$Pvalues, digits = 4,
  caption = "staMMP: Pairwise Fst P-values Fst for neutral loci by regions")  %>%
  kableExtra::kable_styling(full_width = FALSE, font_size = 7)
```

```
knitr::kable(
  sumres.Fst.neutral[order(sumres.Fst.neutral$Fst,
                                      decreasing = TRUE),
                                  c(1,20,21,40,41)], digits = 4,
  caption = "StrataG: Pairwise Fst for neutral loci by regions")
```

```
knitr::kable(
  Fst.region.stampp.outlier$Fsts, digits = 4,
```

Table 11: StrataG: Pairwise Fst for neutral loci by regions

	label	Fst	Fst_p.val	Fst.p.adj.fdr	Fst.p.adj.bonf
1	EIO (8) v. MED (45)	0.0012	0.0015	0.0025	0.0420
12	MED (45) v. SWPAC (53)	0.0011	0.0001	0.0002	0.0028
3	EIO (8) v. NEATL (21)	0.0010	0.0005	0.0009	0.0140
21	NEATL (21) v. SWPAC (53)	0.0009	0.0001	0.0002	0.0028
10	MED (45) v. NIO (16)	0.0009	0.0001	0.0002	0.0028
17	NATL (42) v. SWPAC (53)	0.0009	0.0001	0.0002	0.0028
13	MED (45) v. WIO (22)	0.0008	0.0001	0.0002	0.0028
18	NATL (42) v. WIO (22)	0.0008	0.0001	0.0002	0.0028
19	NEATL (21) v. NIO (16)	0.0008	0.0001	0.0002	0.0028
20	NEATL (21) v. SAF (105)	0.0008	0.0001	0.0002	0.0028
11	MED (45) v. SAF (105)	0.0008	0.0001	0.0002	0.0028
22	NEATL (21) v. WIO (22)	0.0007	0.0001	0.0002	0.0028
2	EIO (8) v. NATL (42)	0.0007	0.0013	0.0023	0.0364
15	NATL (42) v. NIO (16)	0.0006	0.0001	0.0002	0.0028
16	NATL (42) v. SAF (105)	0.0006	0.0001	0.0002	0.0028
9	MED (45) v. NEATL (21)	0.0004	0.0001	0.0002	0.0028
8	MED (45) v. NATL (42)	0.0004	0.0001	0.0002	0.0028
14	NATL (42) v. NEATL (21)	0.0002	0.0189	0.0294	0.5292
4	EIO (8) v. NIO (16)	0.0002	0.2461	0.3627	1.0000
5	EIO (8) v. SAF (105)	0.0001	0.3771	0.5028	1.0000
26	SAF (105) v. SWPAC (53)	0.0000	0.3120	0.4368	1.0000
6	EIO (8) v. SWPAC (53)	0.0000	0.4836	0.5887	1.0000
27	SAF (105) v. WIO (22)	0.0000	0.4741	0.5887	1.0000
23	NIO (16) v. SAF (105)	0.0000	0.5639	0.6579	1.0000
28	SWPAC (53) v. WIO (22)	0.0000	0.6694	0.7209	1.0000
24	NIO (16) v. SWPAC (53)	-0.0001	0.7228	0.7496	1.0000
7	EIO (8) v. WIO (22)	-0.0001	0.6364	0.7128	1.0000
25	NIO (16) v. WIO (22)	-0.0002	0.8549	0.8549	1.0000

Table 12: staMMP: Pairwise Fst for outlier loci by regions

	NATL	NEATL	MED	SAF	WIO	NIO	EIO	SWPAC
NATL	NA	NA	NA	NA	NA	NA	NA	NA
NEATL	-0.0061	NA	NA	NA	NA	NA	NA	NA
MED	-0.0119	-0.0003	NA	NA	NA	NA	NA	NA
SAF	0.2671	0.2933	0.2664	NA	NA	NA	NA	NA
WIO	0.4617	0.4919	0.4707	0.0460	NA	NA	NA	NA
NIO	0.3691	0.3922	0.3745	0.0070	-0.0099	NA	NA	NA
EIO	0.5148	0.5405	0.5287	0.1246	0.0264	0.0361	NA	NA
SWPAC	0.3923	0.4132	0.3988	0.0266	-0.0079	-0.0040	0.0368	NA

Table 13: staMMP: Pairwise Fst P-values Fst for outlier loci by regions

	NATL	NEATL	MED	SAF	WIO	NIO	EIO	SWPAC
NATL	NA	NA	NA	NA	NA	NA	NA	NA
NEATL	0.7021	NA	NA	NA	NA	NA	NA	NA
MED	0.9997	0.5046	NA	NA	NA	NA	NA	NA
SAF	0.0000	0.0000	0	NA	NA	NA	NA	NA
WIO	0.0000	0.0000	0	0.0039	NA	NA	NA	NA
NIO	0.0000	0.0000	0	0.3190	0.8006	NA	NA	NA
EIO	0.0000	0.0000	0	0.0015	0.1679	0.1470	NA	NA
SWPAC	0.0000	0.0000	0	0.0088	0.8443	0.6449	0.0561	NA

```
  caption = "staMMP: Pairwise  Fst for outlier loci by regions") %>%
  kableExtra::kable_styling(full_width = FALSE, font_size = 7)
```

```
knitr::kable(
  Fst.region.stampp.outlier$Pvalues, digits = 4,
  caption = "staMMP: Pairwise Fst P-values Fst for outlier loci by regions")  %>%
  kableExtra::kable_styling(full_width = FALSE, font_size = 7)
```

```
knitr::kable(
  sumres.Fst.outlier[order(sumres.Fst.outlier$Fst,
                                       decreasing = TRUE),
                                   c(1,20,21,40,41)], digits = 4,
  caption = "StrataG: Pairwise Fst for outlier loci by regions")
```

Table 14: StrataG: Pairwise Fst for outlier loci by regions

	label	Fst	Fst_p.val	Fst.p.adj.fdr	Fst.p.adj.bonf
22	NEATL (21) v. WIO (22)	0.3243	0.0001	0.0002	0.0028
3	EIO (8) v. NEATL (21)	0.3237	0.0001	0.0002	0.0028
1	EIO (8) v. MED (45)	0.2902	0.0001	0.0002	0.0028
13	MED (45) v. WIO (22)	0.2883	0.0001	0.0002	0.0028
2	EIO (8) v. NATL (42)	0.2865	0.0001	0.0002	0.0028
18	NATL (42) v. WIO (22)	0.2858	0.0001	0.0002	0.0028
21	NEATL (21) v. SWPAC (53)	0.2813	0.0001	0.0002	0.0028
12	MED (45) v. SWPAC (53)	0.2493	0.0001	0.0002	0.0028
17	NATL (42) v. SWPAC (53)	0.2465	0.0001	0.0002	0.0028
19	NEATL (21) v. NIO (16)	0.2324	0.0001	0.0002	0.0028
10	MED (45) v. NIO (16)	0.2000	0.0001	0.0002	0.0028
15	NATL (42) v. NIO (16)	0.1999	0.0001	0.0002	0.0028
20	NEATL (21) v. SAF (105)	0.1985	0.0001	0.0002	0.0028
16	NATL (42) v. SAF (105)	0.1659	0.0001	0.0002	0.0028
11	MED (45) v. SAF (105)	0.1658	0.0001	0.0002	0.0028
5	EIO (8) v. SAF (105)	0.0553	0.0049	0.0076	0.1372
27	SAF (105) v. WIO (22)	0.0242	0.0028	0.0046	0.0784
4	EIO (8) v. NIO (16)	0.0171	0.1668	0.2458	1.0000
26	SAF (105) v. SWPAC (53)	0.0134	0.0021	0.0037	0.0588
6	EIO (8) v. SWPAC (53)	0.0133	0.1781	0.2493	1.0000
7	EIO (8) v. WIO (22)	0.0085	0.2635	0.3513	1.0000
23	NIO (16) v. SAF (105)	0.0022	0.2885	0.3672	1.0000
9	MED (45) v. NEATL (21)	0.0002	0.3653	0.4447	1.0000
24	NIO (16) v. SWPAC (53)	-0.0024	0.5445	0.6307	1.0000
14	NATL (42) v. NEATL (21)	-0.0030	0.5631	0.6307	1.0000
28	SWPAC (53) v. WIO (22)	-0.0037	0.6490	0.6730	1.0000
25	NIO (16) v. WIO (22)	-0.0052	0.6417	0.6730	1.0000
8	MED (45) v. NATL (42)	-0.0059	0.9629	0.9629	1.0000

3_AMOVA

```
pop.levels <- c("NATL", "NEATL", "MED", "SAF", "WIO", "NIO", "EIO", "SWPAC")
gl.all.neutral$pop <- factor(gl.all.neutral$other$ind.metrics$STRATA2,
                             levels = pop.levels)

dist <- dist(tab(gl.all.neutral)) #euclidean distance
amova.result <- pegas::amova(dist ~ pop, data = adegenet::strata(gl.all.neutral),
                             nperm = 1000)
print(amova.result)
```

```
##
##  Analysis of Molecular Variance
##
## Call: pegas::amova(formula = dist ~ pop, data = adegenet::strata(gl.all.neutral),
##     nperm = 1000)
##
##              SSD      MSD  df
## pop     53110.54 7587.219   7
## Error 2179066.70 7167.983 304
## Total 2232177.24 7177.419 311
##
## Variance components:
##         sigma2 P.value
## pop     11.667       0
## Error 7167.983
##
## Phi-statistics:
## pop.in.GLOBAL
##    0.00162507
##
## Variance coefficients:
##        a
## 35.93223
```

```
dist <- dist(tab(gl.all.outlier)) #euclidean distance
amova.result <- pegas::amova(dist ~ pop, data = adegenet::strata(gl.all.outlier),
                             nperm = 1000)
print(amova.result)
```

```
##
##  Analysis of Molecular Variance
##
## Call: pegas::amova(formula = dist ~ pop, data = adegenet::strata(gl.all.outlier),
##     nperm = 1000)
##
##             SSD       MSD  df
## pop   273.7149 39.102133   7
## Error 673.9552  2.216958 304
## Total 947.6702  3.047171 311
##
## Variance components:
##       sigma2 P.value
## pop   1.0265       0
## Error 2.2170
##
```

```
## Phi-statistics:
## pop.in.GLOBAL
##     0.3164875
##
## Variance coefficients:
##        a
## 35.93223
```

4_Isolation-by-distance

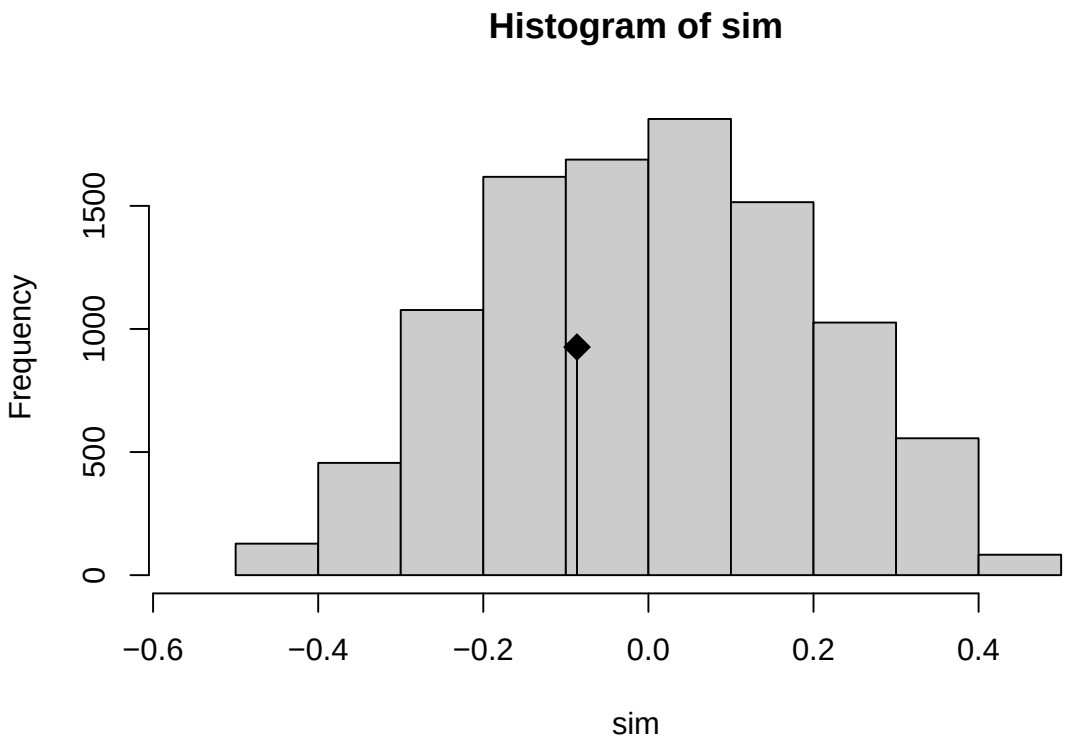
Test the correlation between the genetic and geographic distance between the sampling sites. The genetic distance is calculated based on Fst and Euclidean distance.

```
pop.levels <- c("NATL", "NEATL", "MED", "SAF", "WIO", "NIO", "EIO", "SWPAC")
gl.all.neutral$pop <- factor(gl.all.neutral$other$ind.metrics$STRATA2,
                             levels = pop.levels)
gi.all.neutral <- gl2gi_mvb(gl.all.neutral)
# geographic file that contain coordinates of the 8 sampling locations
popm <- read.csv("infiles/Geo8pop.csv", header = TRUE, sep = ";")
```

```
D <- hierfstat::pairwise.fst(gi.all.neutral)
save(D, file = "./Rdata/ibd_all_neutral.Rdata")
```

```
load("./Rdata/ibd_all_neutral.Rdata")
Dgeo <- dist(popm[,3:4])
geo.dist <- geodist::geodist(popm[,3:4], measure = "geodesic")/1000
Dgeo <- as.dist(geo.dist)

fst.test <- ade4::mantel.randtest(D, Dgeo, nrepet = 10000)
plot(fst.test)
```

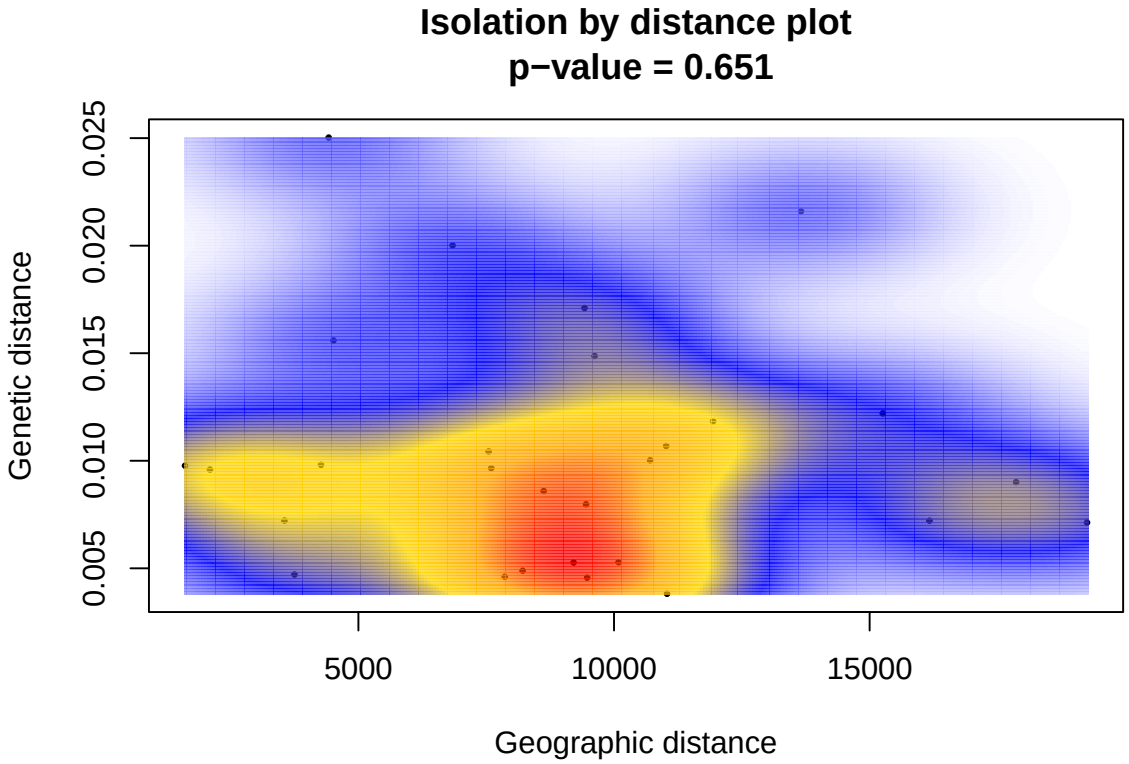
Based on Fst distance



```
fst.test
```

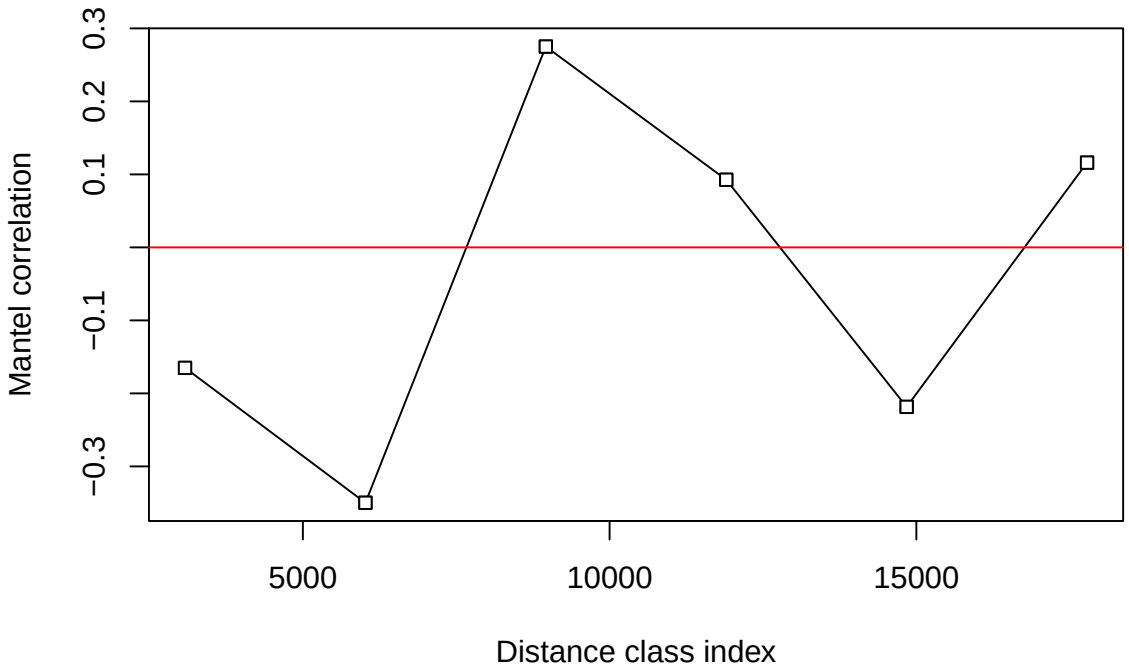
```
## Monte-Carlo test
## Call: ade4::mantel.randtest(m1 = D, m2 = Dgeo, nrepet = 10000)
##
## Observation: -0.08648858
##
## Based on 10000 replicates
## Simulated p-value: 0.6506349
## Alternative hypothesis: greater
##
##       Std.Obs   Expectation      Variance
## -0.4455972554 -0.0006587453  0.0371015081
```

```
#FST with Kernel
#n=Number of grid points in each direction. Can be scalar or a length-2 integer vector.
dens <- MASS::kde2d(Dgeo,D, n = 312)
myPal <- colorRampPalette(c("white","blue","gold", "orange", "red"))
plot(Dgeo, D, pch = 20,cex = 0.5,
     xlab = "Geographic distance",
     ylab = "Genetic distance",
     main = paste0("Isolation by distance plot\n p-value = ",
                   round(fst.test$pvalue,3)))
image(dens, col = transp(myPal(300),.7), add = TRUE)
```




```
# abline(lm(D~Dgeo),col = "red",lty = 2)


corr.test <- vegan::mantel.correlog(
```

```
  D, D.geo = Dgeo, XY = NULL, n.class = 0, break.pts = NULL,
  cutoff = FALSE, r.type = "pearson", nperm = 999, mult = "holm", progressive = TRUE)
plot(corr.test)
```




```
gpop <- adegenet::genind2genpop(gi.all.neutral)
```

Based on Euclidean distance

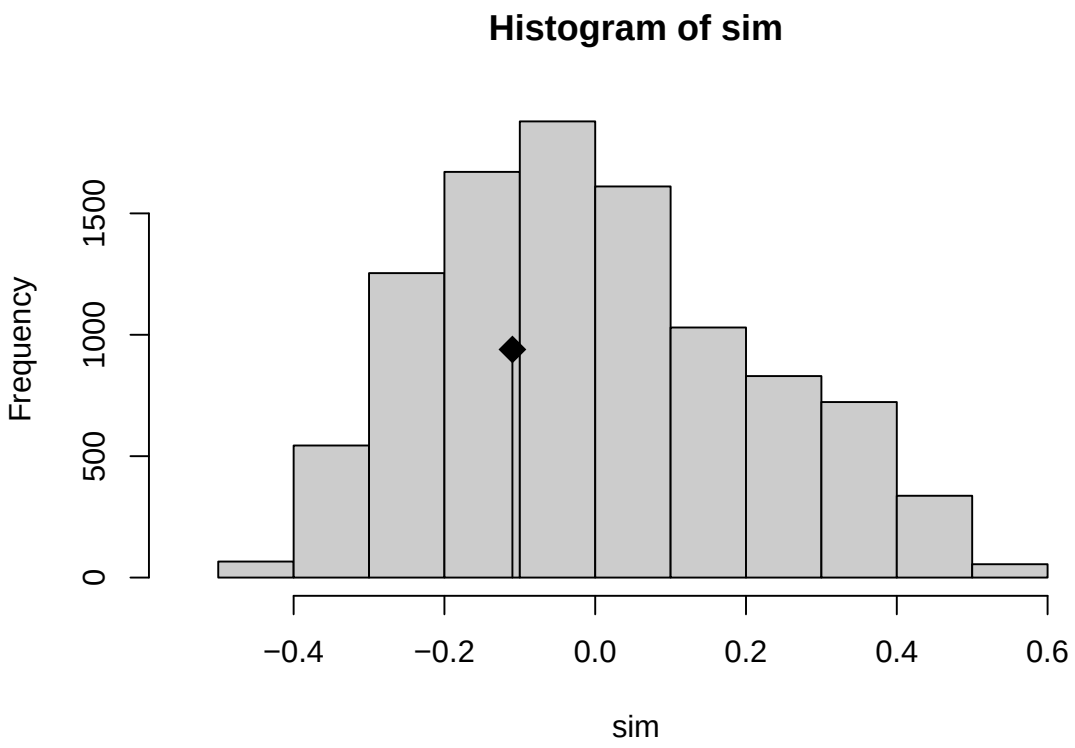
```
##
##  Converting data from a genind to a genpop object...
##
## ...done.
```

```
Dgen <- adegenet::dist.genpop(gpop, method = 2) #method 2 Euclidean = Angular distance
names(Dgen) <- NULL

Dgeo <- dist(popm[,3:4],method = "euclidean")
eucl.test <- ade4::mantel.randtest(Dgen, Dgeo, nrepet = 10000)
eucl.test
```

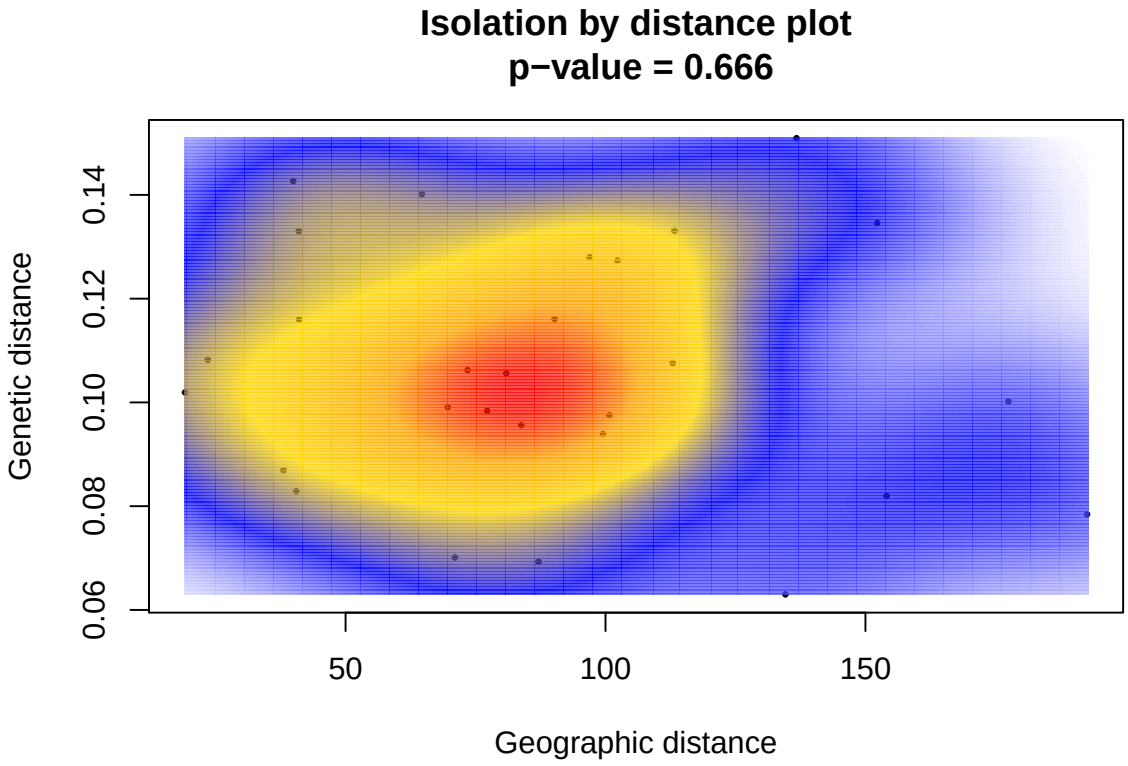
```
## Monte-Carlo test
## Call: ade4::mantel.randtest(m1 = Dgen, m2 = Dgeo, nrepet = 10000)
##
## Observation: -0.1098316
##
## Based on 10000 replicates
```

```
## Simulated p-value: 0.6659334
## Alternative hypothesis: greater
##
##       Std.Obs   Expectation      Variance
## -5.208625e-01 -5.298856e-05  4.442113e-02
```

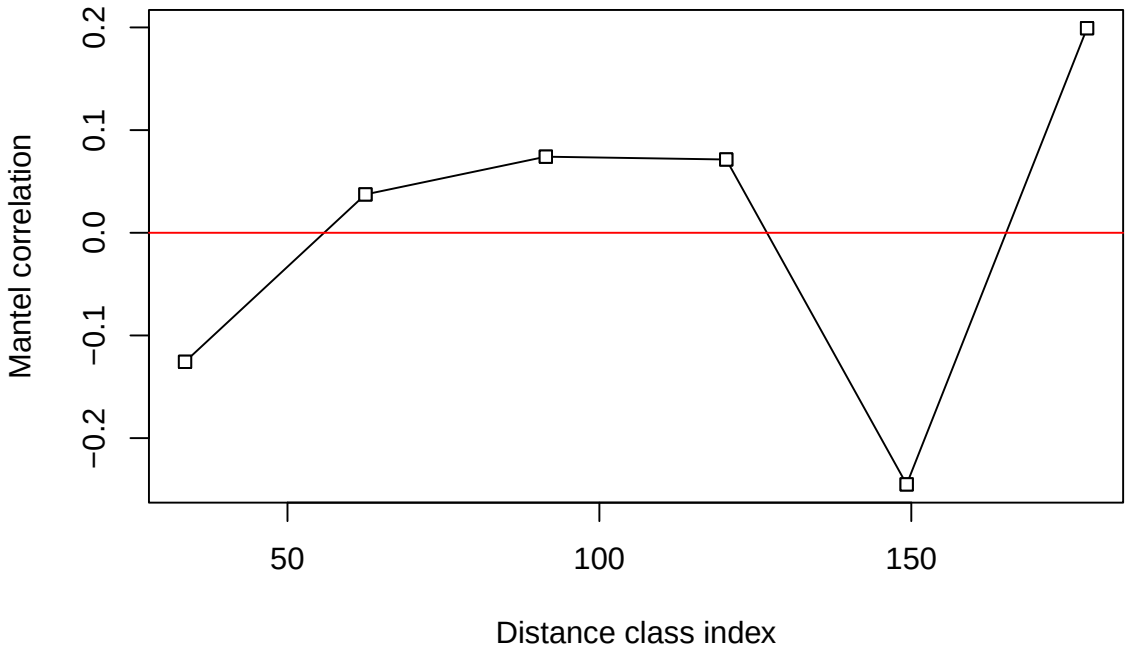
```
plot(eucl.test)
```




```
#IBD with KERNEL
dens <- MASS::kde2d(Dgeo,Dgen, n = 312)
myPal <- colorRampPalette(c("white","blue","gold", "orange", "red"))
plot(Dgeo, Dgen, pch = 20, cex = 0.5,
     xlab = "Geographic distance",
     ylab = "Genetic distance",
     main = paste0("Isolation by distance plot\n p-value = ",
                   round(eucl.test$pvalue,3)))
image(dens, col = transp(myPal(300),.7), add = TRUE)
```




```
# abline(lm(D~Dgeo),col = "red",lty = 2)


corr.test <- vegan::mantel.correlog(Dgen, D.geo = Dgeo, XY = NULL, n.class = 0,
                                    break.pts = NULL,
                                    cutoff = FALSE, r.type = "pearson", nperm = 999,
                                    mult = "holm", progressive = TRUE)
plot(corr.test)
```

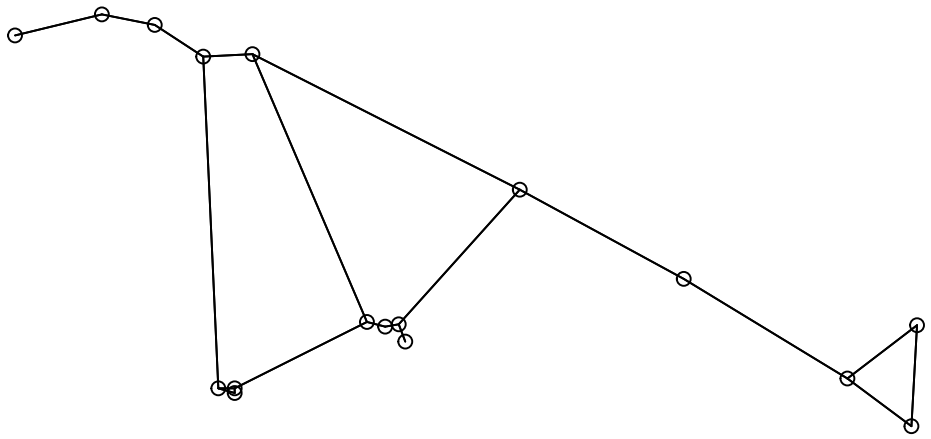



```
popmInd <- read.csv("infiles/Geo.csv", header = TRUE, sep = ";")
head(popmInd)
```

```
##   popid    popname  long   lat
## 1 60104 Pacific-SW 165.8 -43.6
## 2 60105 Pacific-SW 165.8 -43.6
## 3 60117 Pacific-SW 165.8 -43.6
## 4 60118 Pacific-SW 165.8 -43.6
## 5 60119 Pacific-SW 165.8 -43.6
## 6 60120 Pacific-SW 165.8 -43.6
```

```
#Boundaries
gab <- adegenet::chooseCN(unique(popmInd[,3:4]),ask = FALSE,type = 2)
```

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



5_PCA

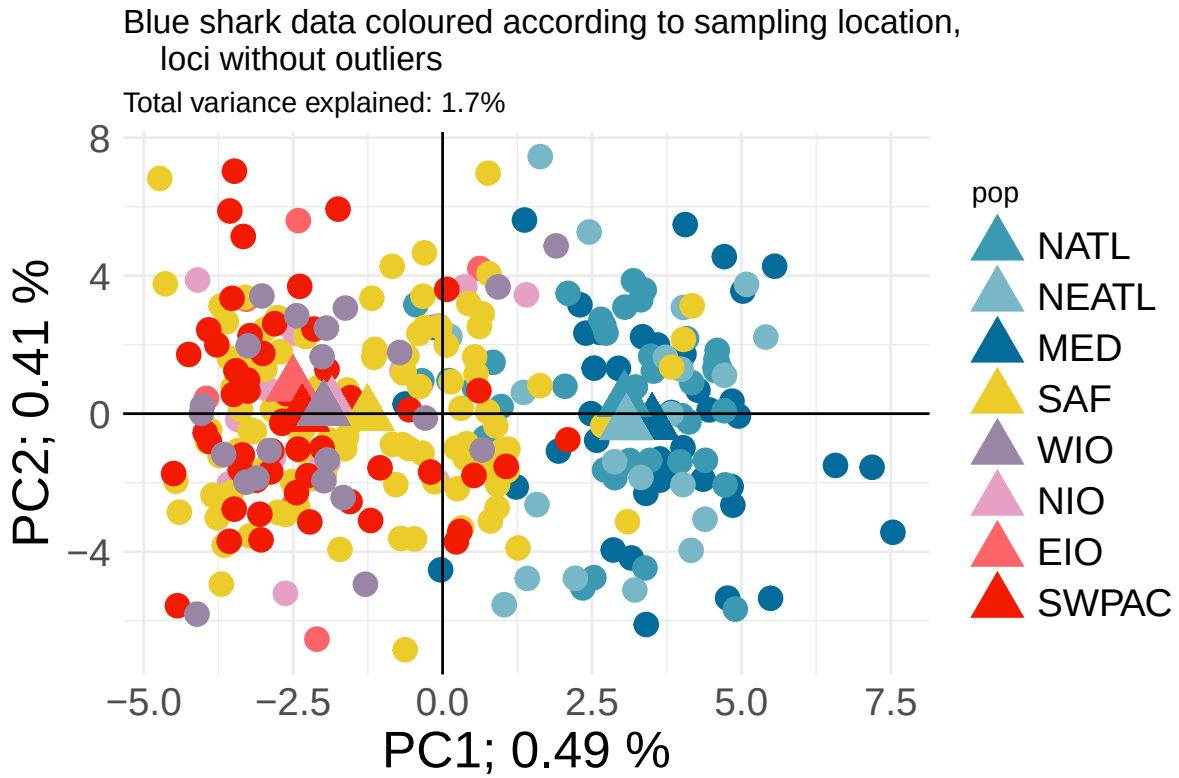
```
pca.all.neutral <- adegenet::glPca(gl.all.neutral, nf = 4,
                                   parallel = TRUE,
                                   n.cores = parallel::detectCores())

pca.all.outlier <- adegenet::glPca(gl.all.outlier, nf = 4,
                                   parallel = TRUE,
                                   n.cores = parallel::detectCores())

save(pca.all.neutral, pca.all.outlier,
     file = "Rdata/PCA_ALL_objects.Rdata")
```

```
load("Rdata/PCA_ALL_objects.Rdata")
var_frac <- pca.all.neutral$eig/sum(pca.all.neutral$eig)
print(paste0("Total variance explained: ",signif(sum(var_frac[1:4]) * 100, 3), "%"))
```

PCA with neutral markers

```
## [1] "Total variance explained: 1.7%"
```

```
pop <- gl.all.neutral$pop

data <- data.frame(pca.all.neutral$scores, pop = pop)
data <- merge(data, aggregate(cbind(mean.x = PC1, mean.y = PC2) ~
                                pop, data, mean), by = "pop")

pca.neutral.plot <- ggplot2::ggplot(data, ggplot2::aes(PC1, PC2, color = pop)) +
  ggplot2::geom_point(size = 4) +
  ggplot2::geom_point(ggplot2::aes(x = mean.x,
                                   y = mean.y, color = pop),
                      size = 7,
                      shape = 17) +
  ggplot2::scale_colour_manual(values = colours8) +
  ggplot2::scale_fill_manual(values = colours8) +
  ggplot2::geom_hline(yintercept = 0) +
  ggplot2::geom_vline(xintercept = 0) +
  ggplot2::labs(
    subtitle = paste0("Total variance explained: ",
                      signif(sum(var_frac[1:4]) * 100, 3), "%"),
    y = paste0("PC2; ", signif(var_frac[2] * 100, digits = 2), " %"),
    x = paste0("PC1; ", signif(var_frac[1] * 100, digits = 2), " %"),
    title = "Blue shark data coloured according to sampling location,
    loci without outliers",
    caption = ""
  ) + ggplot2::theme_minimal() +
  ggplot2::theme(
    axis.text = ggplot2::element_text(size = 15),
    axis.title.x = ggplot2::element_text(size = 20),
    axis.title.y = ggplot2::element_text(size = 20),
    legend.text = ggplot2::element_text(size = 15)
  )

print(pca.neutral.plot)
```




```
ggplot2::ggsave(pca.neutral.plot,
                filename = "figures/2.ALL_PCA_no_outliers_37655SNPs.png",
                width = 30, height = 15, units = "cm")
```

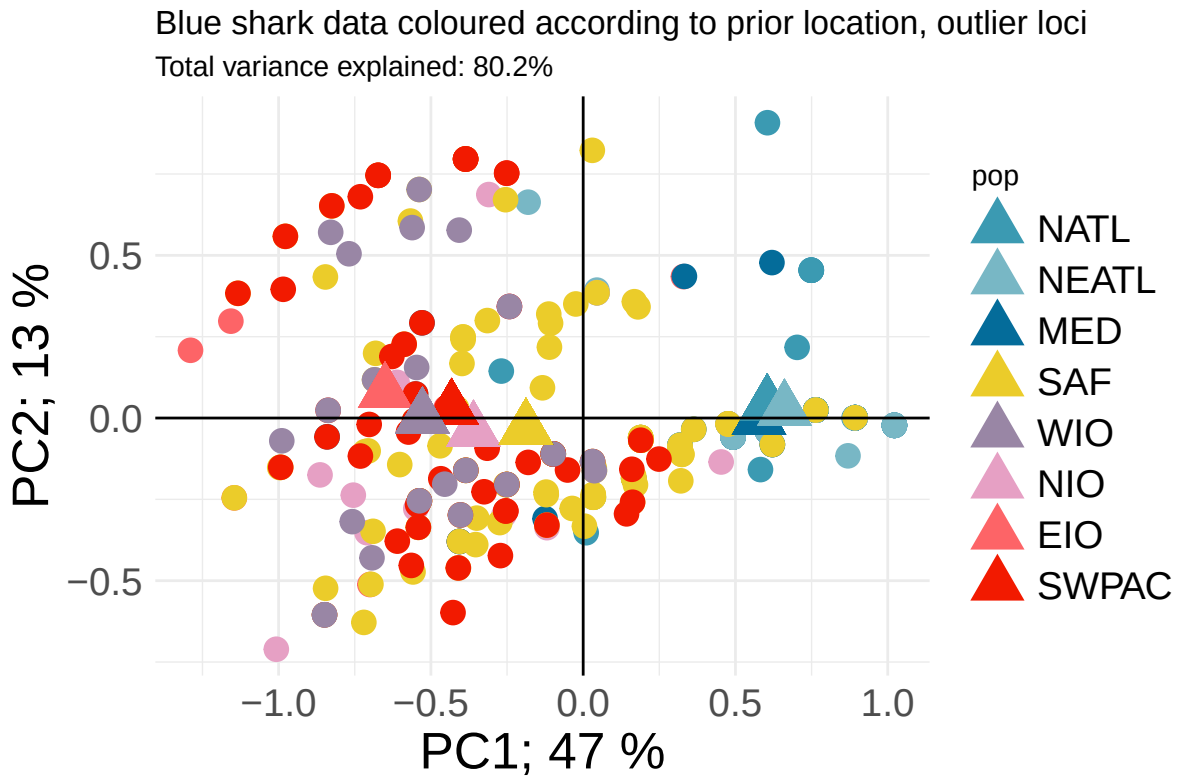
```
var_frac <- pca.all.outlier$eig/sum(pca.all.outlier$eig)
print(paste0("Total variance explained: ",signif(sum(var_frac[1:4]) * 100, 3), "%"))
```

PCA with outlier markers

```
## [1] "Total variance explained: 80.2%"
```

```
pop.levels <- c("NATL", "NEATL", "MED", "SAF", "WIO", "NIO", "EIO", "SWPAC")
gl.all.outlier$pop <- factor(gl.all.neutral$other$ind.metrics$STRATA2,
                             levels = pop.levels)
pop <- gl.all.outlier$pop

data <- data.frame(pca.all.outlier$scores, pop = pop)
data <- merge(data, aggregate(cbind(mean.x = PC1, mean.y = PC2) ~
                                pop, data, mean), by = "pop")

pca.outlier.plot <- ggplot2::ggplot(data, ggplot2::aes(PC1, PC2, color = pop)) +
  ggplot2::geom_point(size = 4) +
  ggplot2::geom_point(ggplot2::aes(x = mean.x,
                                   y = mean.y, color = pop),
                      size = 7,
                      shape = 17
  ) +
```

```
  ggplot2::scale_colour_manual(values = colours8) +
  ggplot2::scale_fill_manual(values = colours8) +
  ggplot2::geom_hline(yintercept = 0) +
  ggplot2::geom_vline(xintercept = 0) +
  ggplot2::labs(
    subtitle = paste0("Total variance explained: ",
                      signif(sum(var_frac[1:4]) * 100, 3), "%"),
    y = paste0("PC2; ", signif(var_frac[2] * 100, digits = 2), " %"),
    x = paste0("PC1; ", signif(var_frac[1] * 100, digits = 2), " %"),
    title = "Blue shark data coloured according to prior location, outlier loci",
    caption = ""
  ) + ggplot2::theme_minimal() +
  ggplot2::theme(
    axis.text = ggplot2::element_text(size = 15),
    axis.title.x = ggplot2::element_text(size = 20),
    axis.title.y = ggplot2::element_text(size = 20),
    legend.text = ggplot2::element_text(size = 15)
  )

print(pca.outlier.plot)
```



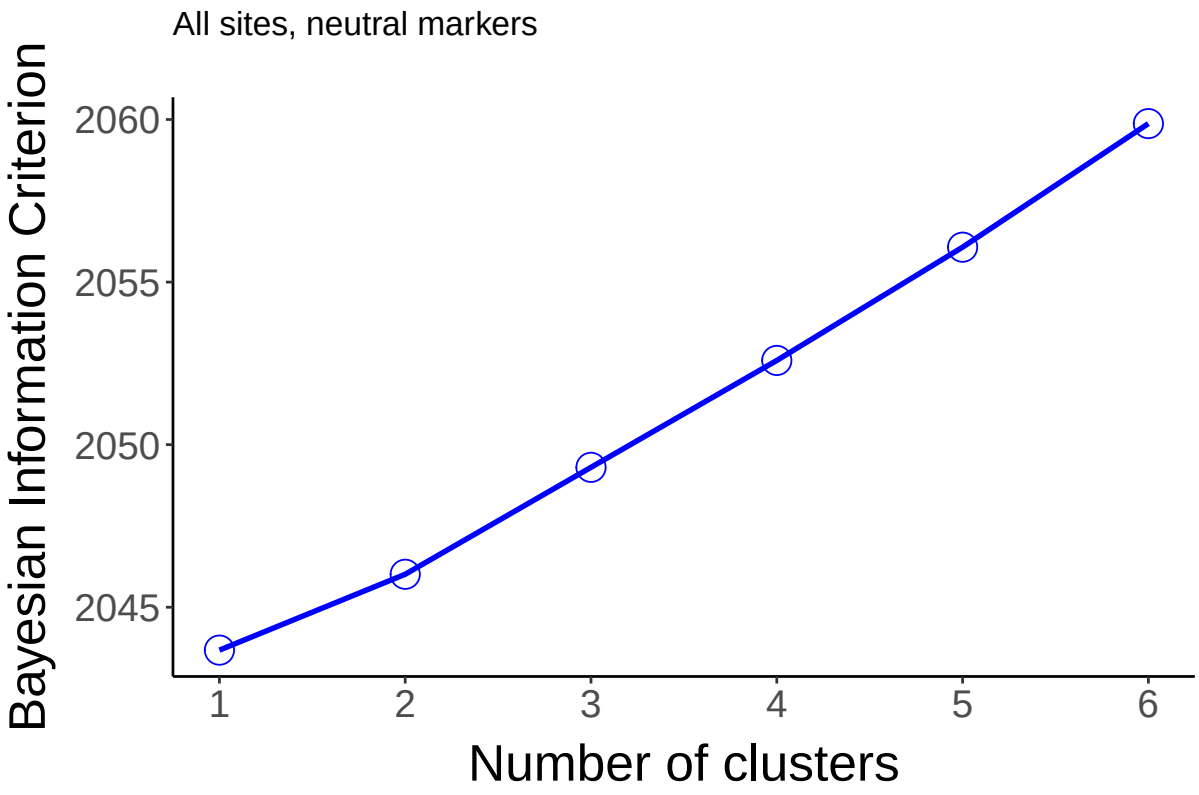

```
ggplot2::ggsave(pca.outlier.plot, filename = "figures/2.PCA_only_outliers_SNPs.png",
                width = 30, height = 15, units = "cm")
```

6_ DAPC

```
set.seed(124)
grpneut <-
  adegenet::find.clusters(
    gl.all.neutral,
    max.n.clust = 6,
    n.pca = adegenet::nInd(gl.all.neutral) / 3,
    stat = "BIC",
    parallel = TRUE,
    n.cores = parallel::detectCores() - 2
  )

save(grpneut, file = "Rdata/BSH_kgrp_no_outlier.rdata")
```

```
load("Rdata/BSH_kgrp_no_outlier.rdata")

y <- as.numeric(grpneut$Kstat)
x <- 1:6
data <- data.frame(x,y,stringsAsFactors = F)
plot <- ggplot2::ggplot(data, ggplot2::aes(x,y)) +
  ggplot2::geom_point(size = 5, shape = 1, color = "blue") +
  ggplot2::geom_line(size = 1,color = "blue") +
  ggplot2::scale_x_continuous(name = waiver(),
                              breaks = seq(from = 0,
                                           to = nrow(gl.all.neutral) - 1,
                                           by = 1)) +
  ggplot2::labs(subtitle = "",
                y = "Bayesian Information Criterion",
                x = "Number of clusters",
                title = "All sites, neutral markers",
                caption = "") +
  ggplot2::theme_classic() +
  ggplot2::theme(
    axis.text = ggplot2::element_text(size = 15),
    axis.title.x = ggplot2::element_text(size = 20,vjust = 0, hjust = 0.5),
    axis.title.y = ggplot2::element_text(size = 20,vjust = 2, hjust = 0.5))
print(plot)
```

Optimal number of clusters with neutral markers



Cross-validation to identify the optimal number of clusters: number of PC axis to account for Uses a training - holdout set of individuals to check how reliable individuals can be assigned.

```
set.seed(124)
x <- gl.all.neutral

png(file="figures/3.BSH_neutral_refined_n_pca.png")
xval <-
  adegenet::xvalDapc(
    tab(x, NA.method = "mean"),
    adegenet::pop(x),
    n.da = 3,
    n.pca = 15:40,
    n.pca = NULL,
    n.rep = 1000,
    parallel = "multicore",
    ncpus = parallel::detectCores() - 2,
    xval.plot = TRUE
  )
dev.off()
save(xval, file = "Rdata/xval_ALL_neutral_refined.rdata")
```

```
load("Rdata/xval_ALL_neutral_refined.rdata")
print(xval[2:6])
```

```
## $`Median and Confidence Interval for Random Chance`
##       2.5%        50%      97.5%
## 0.08816016 0.12135381 0.15891224
##
```

```
## $`Mean Successful Assignment by Number of PCs of PCA`
##        15        16        17        18        19        20
## 0.2667440 0.2693393 0.2738494 0.2742578 0.2848539 0.2820980
##        21        22        23        24        25        26
## 0.2891410 0.2826085 0.2858906 0.2844277 0.2862953 0.3057143
##        27        28        29        30        31        32
## 0.2895559 0.2896827 0.2995394 0.3038764 0.3003777 0.2939277
##        33        34        35        36        37        38
## 0.3024423 0.3034057 0.2936328 0.2807839 0.2927189 0.2898741
##        39        40
## 0.2855994 0.2856081
##
## $`Number of PCs Achieving Highest Mean Success`
## [1] "26"
##
## $`Root Mean Squared Error by Number of PCs of PCA`
##        15        16        17        18        19        20
## 0.7349155 0.7322835 0.7278107 0.7276381 0.7170129 0.7202626
##        21        22        23        24        25        26
## 0.7125025 0.7193452 0.7161227 0.7179875 0.7153159 0.6967440
##        27        28        29        30        31        32
## 0.7127842 0.7123653 0.7032346 0.6986533 0.7028884 0.7082008
##        33        34        35        36        37        38
## 0.7004829 0.6994426 0.7094142 0.7217085 0.7094749 0.7123026
##        39        40
## 0.7170472 0.7169054
##
## $`Number of PCs Achieving Lowest MSE`
## [1] "26"
```

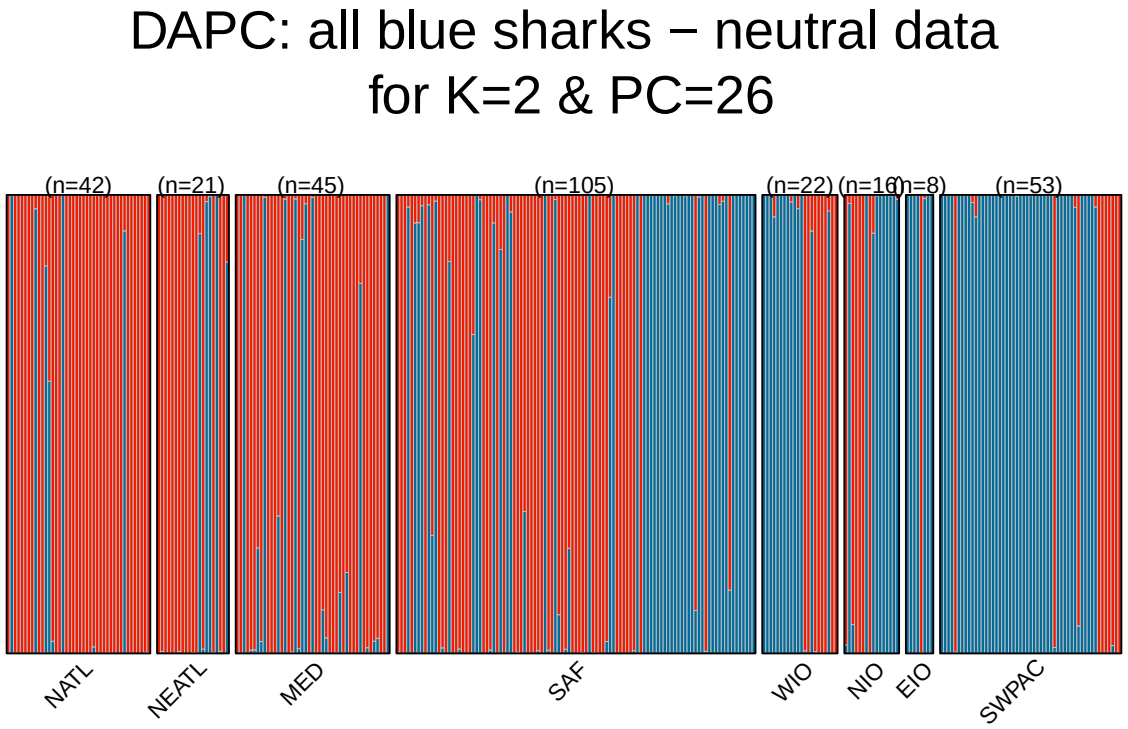
```
PCneut <- as.numeric(xval$`Number of PCs Achieving Lowest MSE`)
```

```
dapc.all.neutral.object.names <- c()
for (K in 2:6) {
  set.seed(124)
  grp <- adegenet::find.clusters(gl.all.neutral, n.clust = K, n.pca = PCneut)
  set.seed(124)
  dapc.all.neutral <- adegenet::dapc(gl.all.neutral, grp$grp, n.da = K - 1,
                                     n.pca = PCneut)
  assign(paste0("dapc.all.neutral", K),value = dapc.all.neutral)
  dapc.all.neutral.object.names <- c(dapc.all.neutral.object.names,
                                     paste0("dapc.all.neutral", K))
}
save(list=dapc.all.neutral.object.names, dapc.all.neutral.object.names,
     file = "Rdata/DAPC.ALL_neutral.Rdata")
```

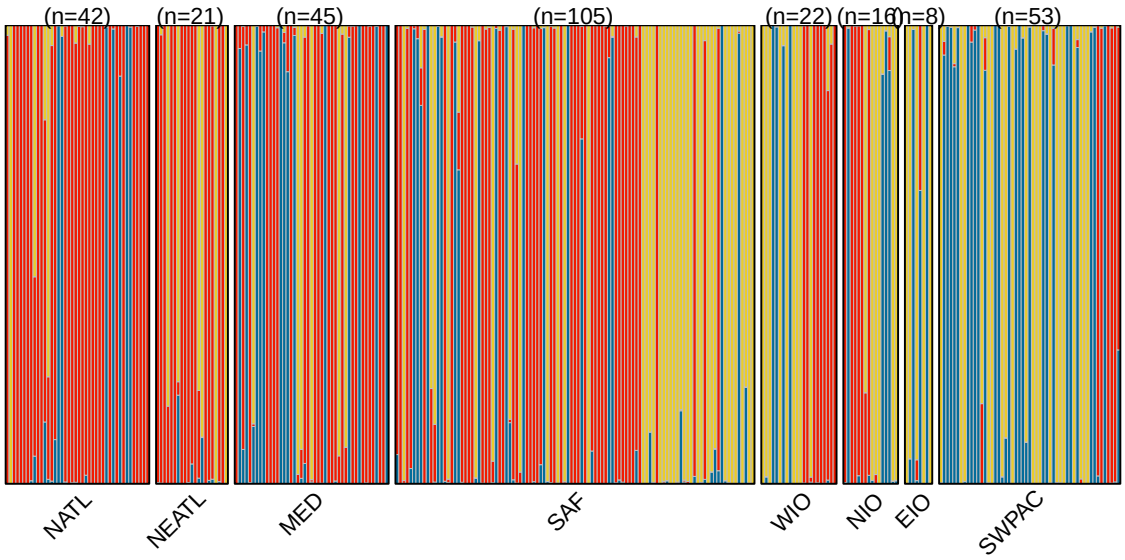
DAPC barplot with neutral markers Group individuals according to DAPC posterior membership.

```
load("Rdata/DAPC.ALL_neutral.Rdata")
for (K in 2:6) {
  colour <- colours6[1:K]
  dapc <- get(dapc.all.neutral.object.names[K - 1])
  post <- as.matrix(dapc$posterior)
  colnames(post) <- paste0("Group", 1:nlevels(dapc$grp))
  locations <- gl.all.neutral$pop

  plot.dapc.FDD(x = post,
```

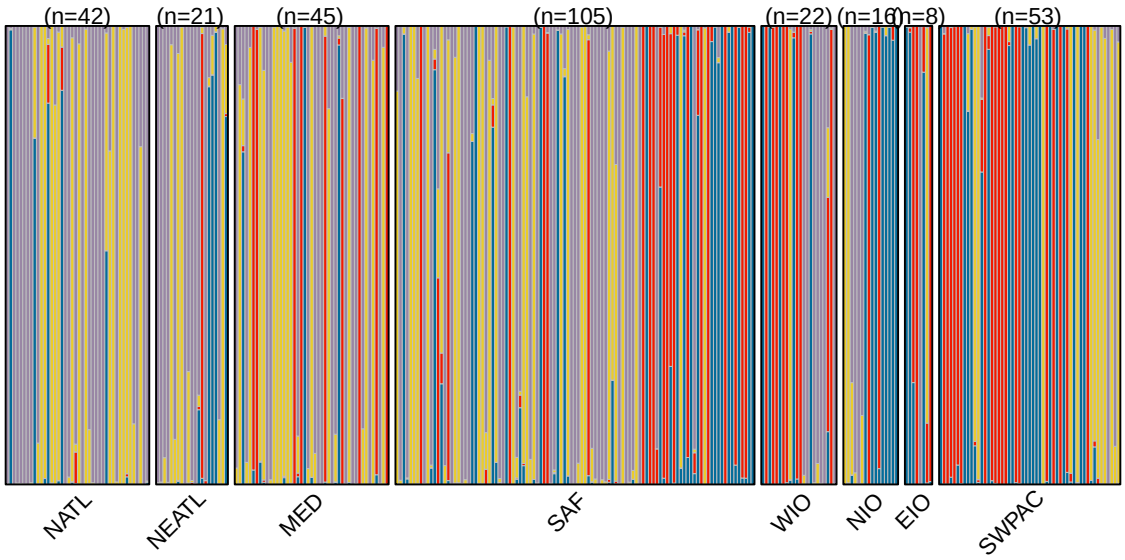
```
                locations = locations,
                colour = colour,
                region.lwd = 1,
                plotTitle = paste0("DAPC: all blue sharks - neutral data\n for K=",
                                   K, " & PC=", PCneut, sep = ""))
  dev.print(
    device = png,
    file = paste0("figures/3.DAPC_barplot_neutral_37655SNPs_K", K, ".png"),
    res = 300,
    width = 30,
    height = 15,
    units = "cm")
}
```



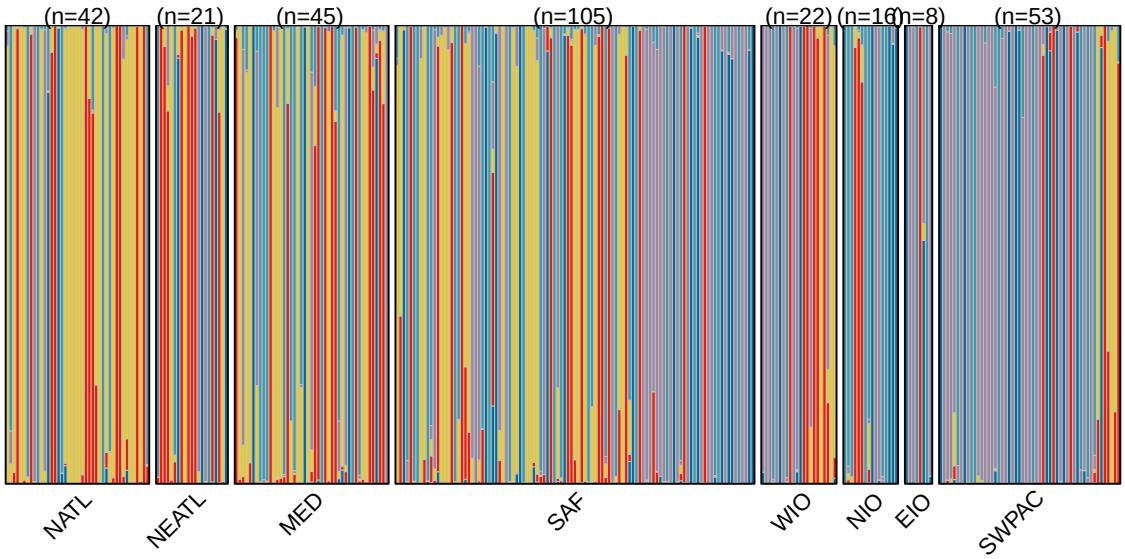
DAPC: all blue sharks − neutral data for K=3 & PC=26



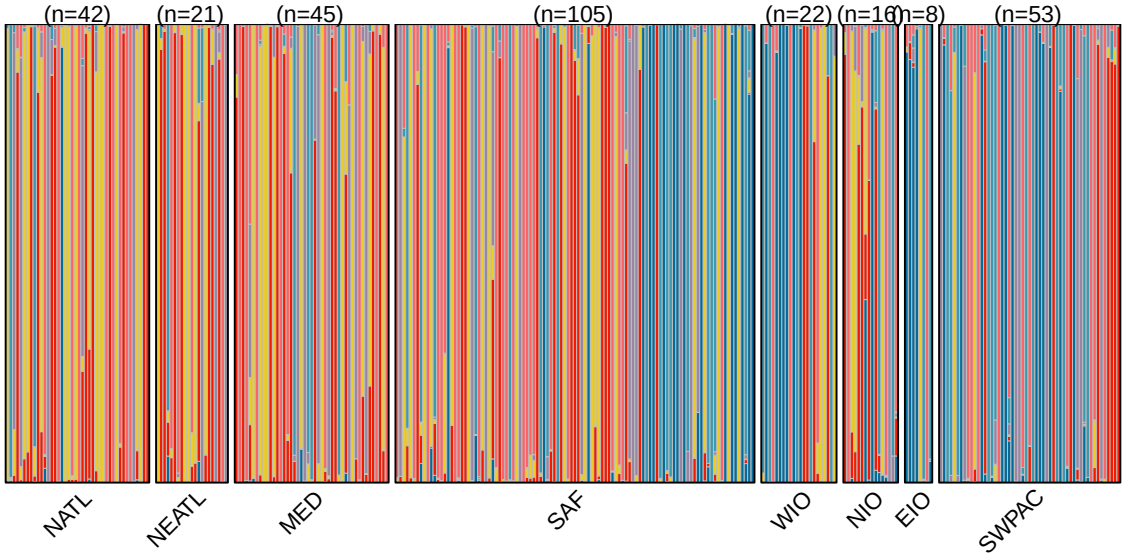
DAPC: all blue sharks − neutral data for K=4 & PC=26



DAPC: all blue sharks − neutral data for K=5 & PC=26



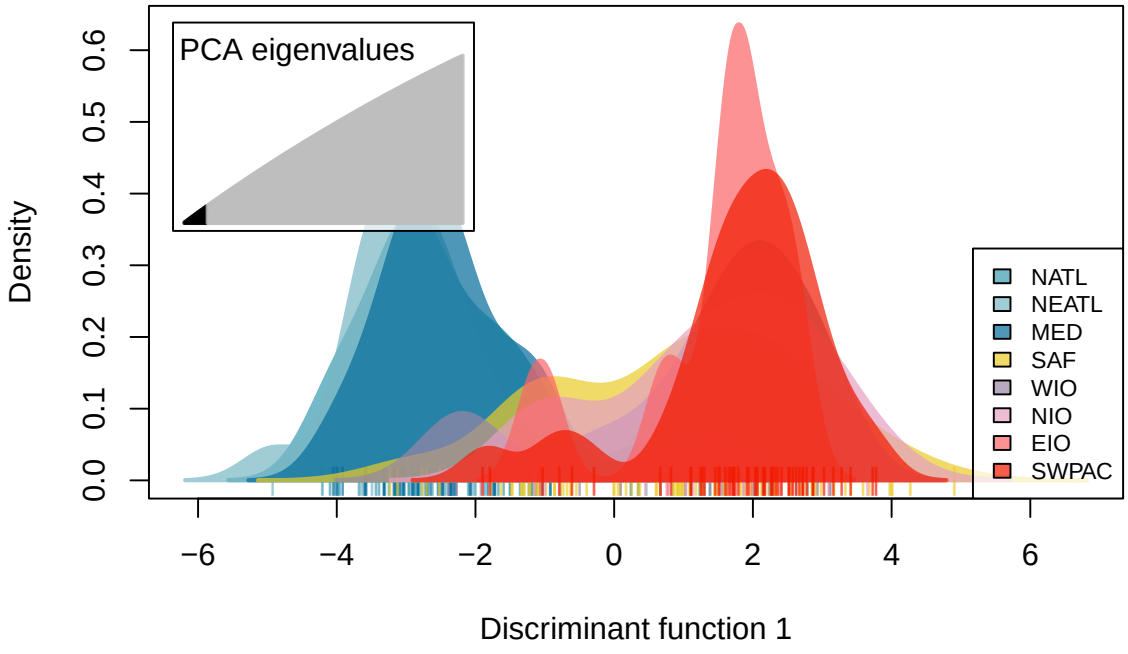
DAPC: all blue sharks − neutral data for K=6 & PC=26

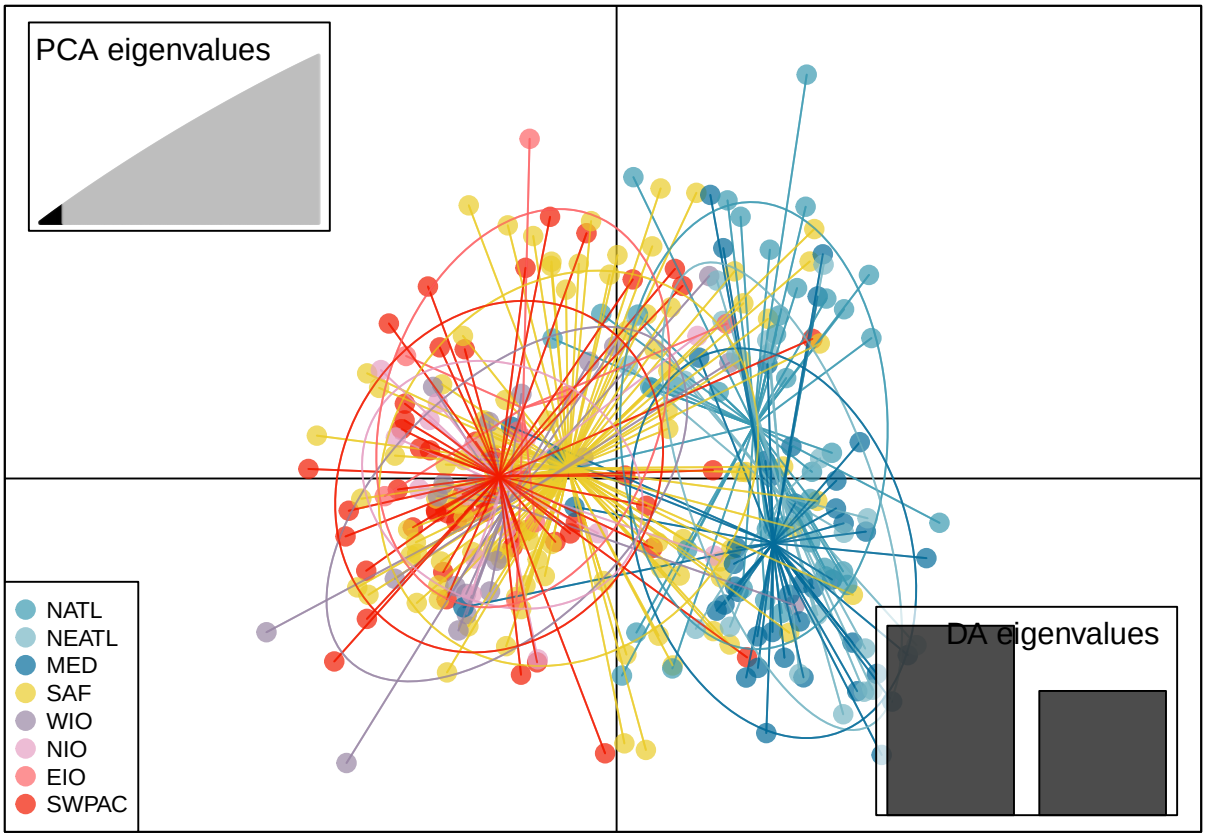


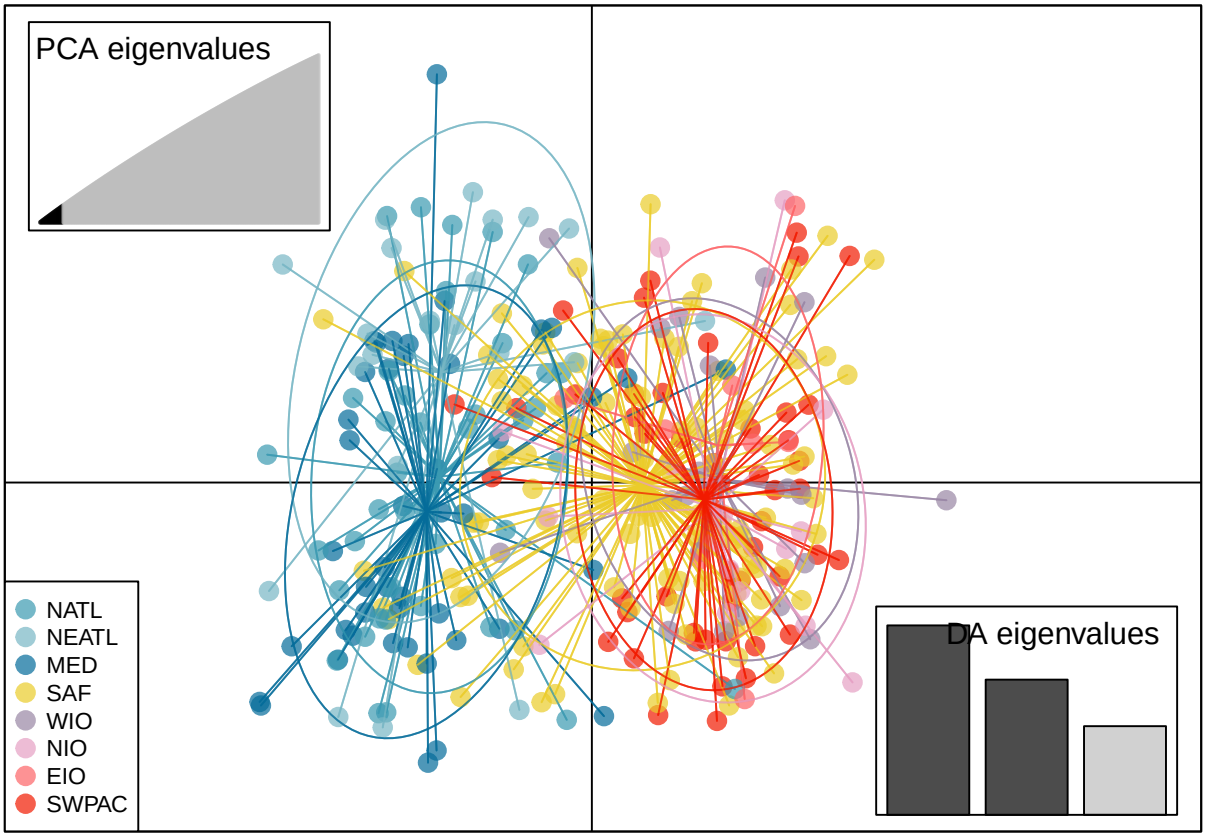

```
load("Rdata/DAPC.ALL_neutral.Rdata")
```

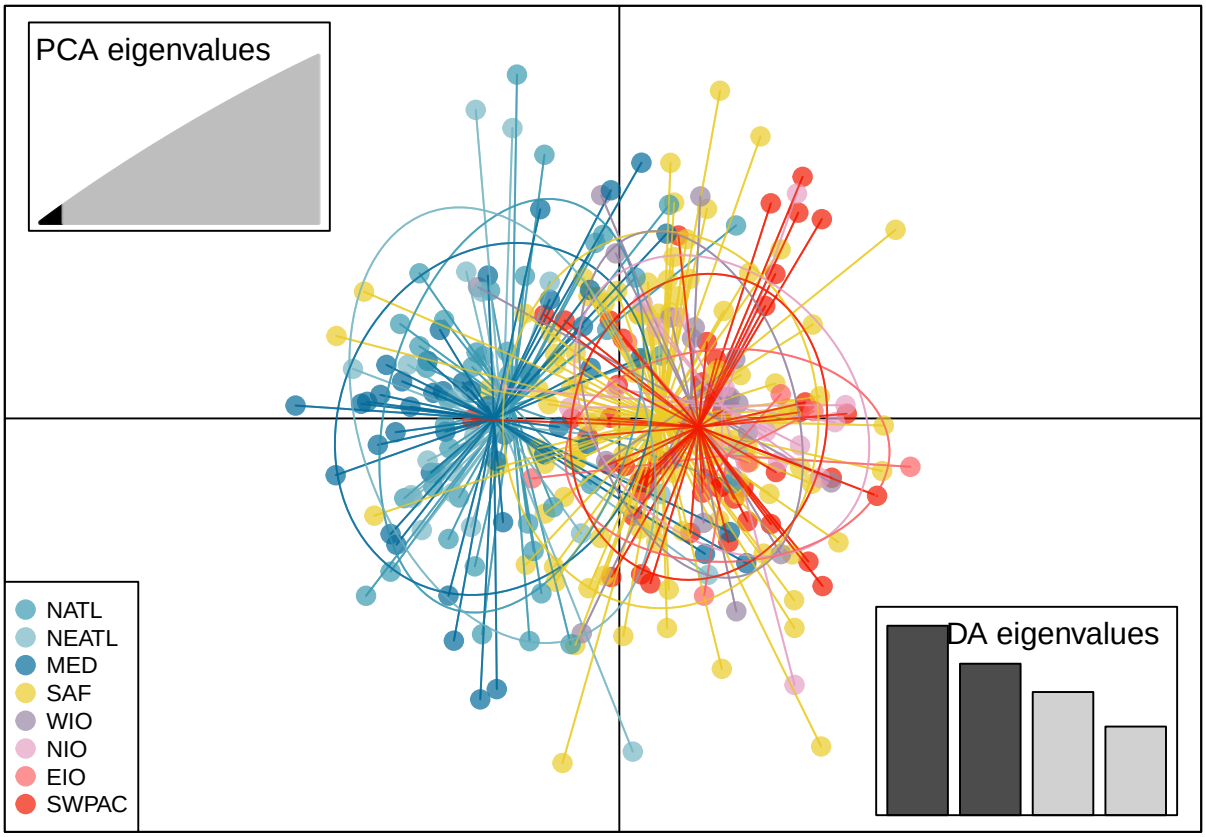
```
for (K in 2:6) {
  dapc.neutral <- get(dapc.all.neutral.object.names[K - 1])
  if (K == 2) {
    posi.leg <- "bottomright"
  } else {posi.leg <- "bottomleft"}
  dapc.neutral.plot <- ade4::scatter(dapc.neutral, grp = pop(gl.all.neutral), cex = 2,
                                     legend = TRUE, col = colours8, clabel = FALSE,
                                     posi.leg = posi.leg, scree.pca = TRUE,
                                     posi.pca = "topleft", cleg = 0.75, xax = 1,
                                     yax = 2, inset.solid = 0.70)
  dev.print(
    device = png,
    file = paste0("figures/3.DAPC_Scatter_neutral_37655SNPs_K", K, ".png"),
    width = 30,
    height = 15,
    units = "cm",
    res = 300
  )
}
```

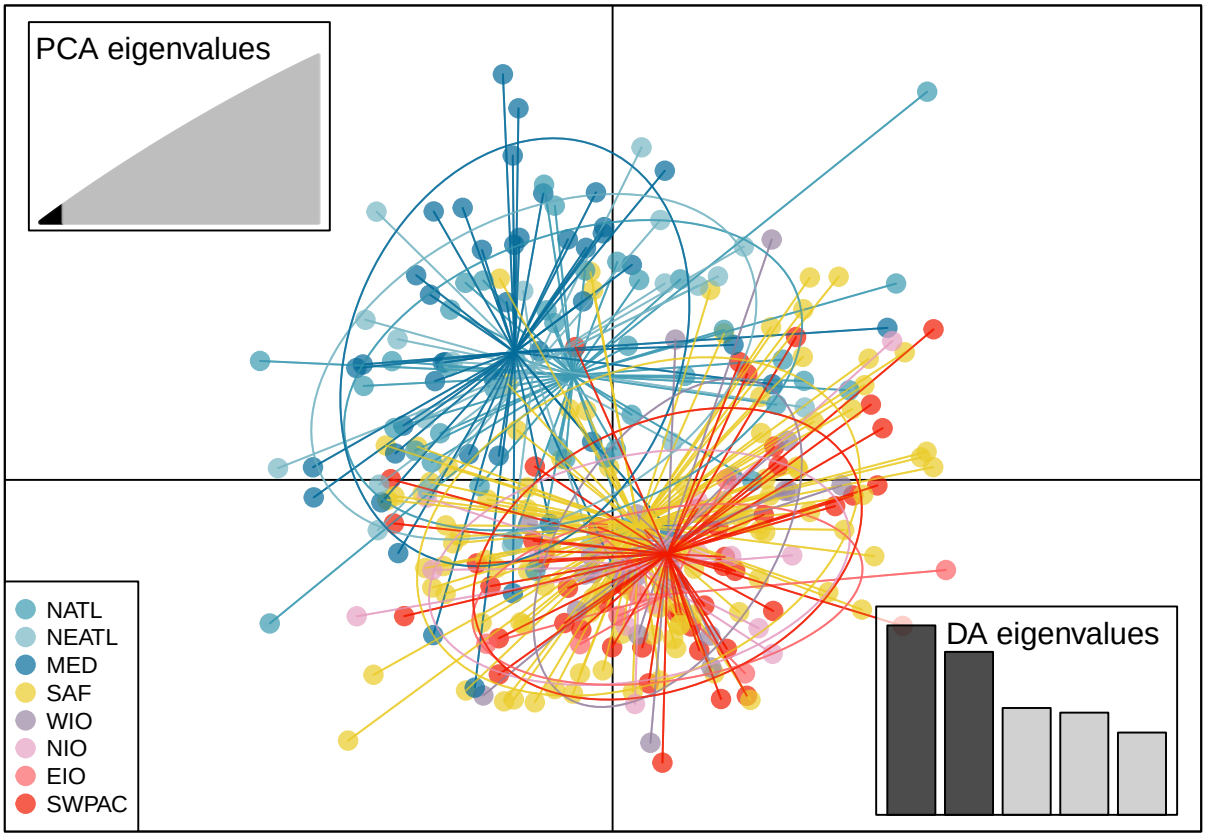
DAPC scatterplot with neutral markers




PCA eigenvalues
NATL
NEATL
MED
SAF
WIO
NIO
EIO
SWPAC
DA eigenvalues

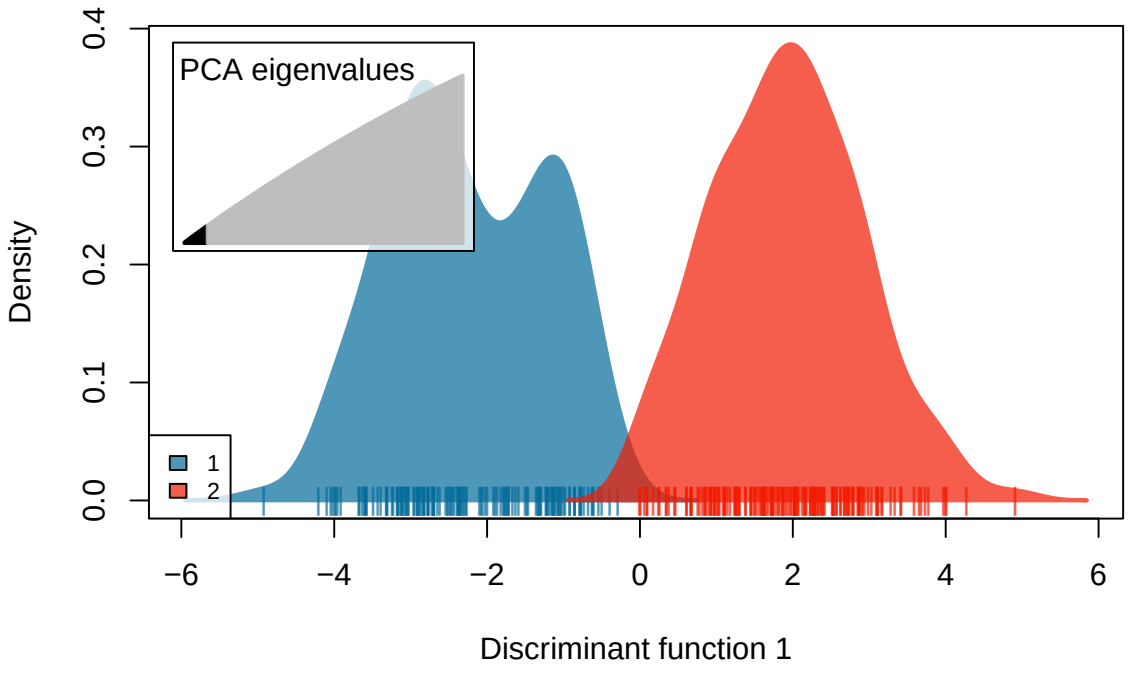

PCA eigenvalues
NATL
NEATL
MED
SAF
WIO
NIO
EIO
SWPAC
DA eigenvalues

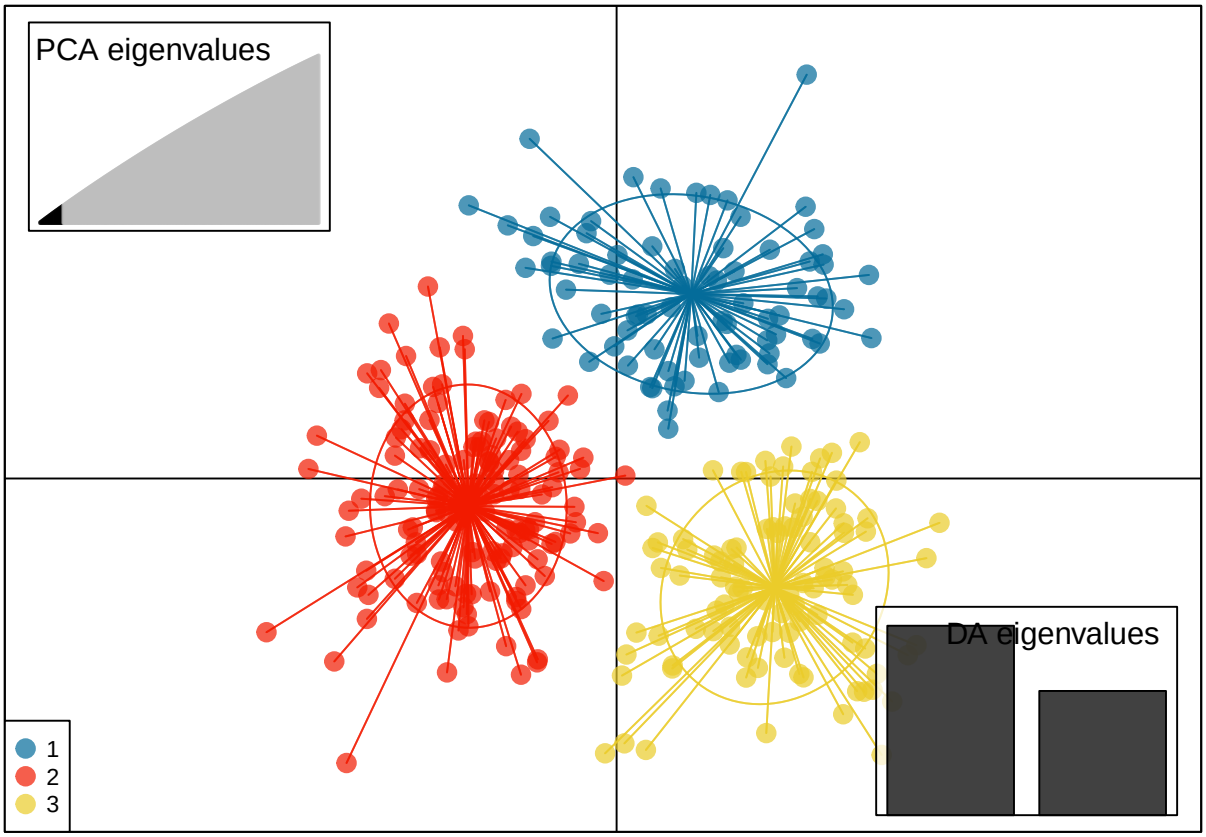


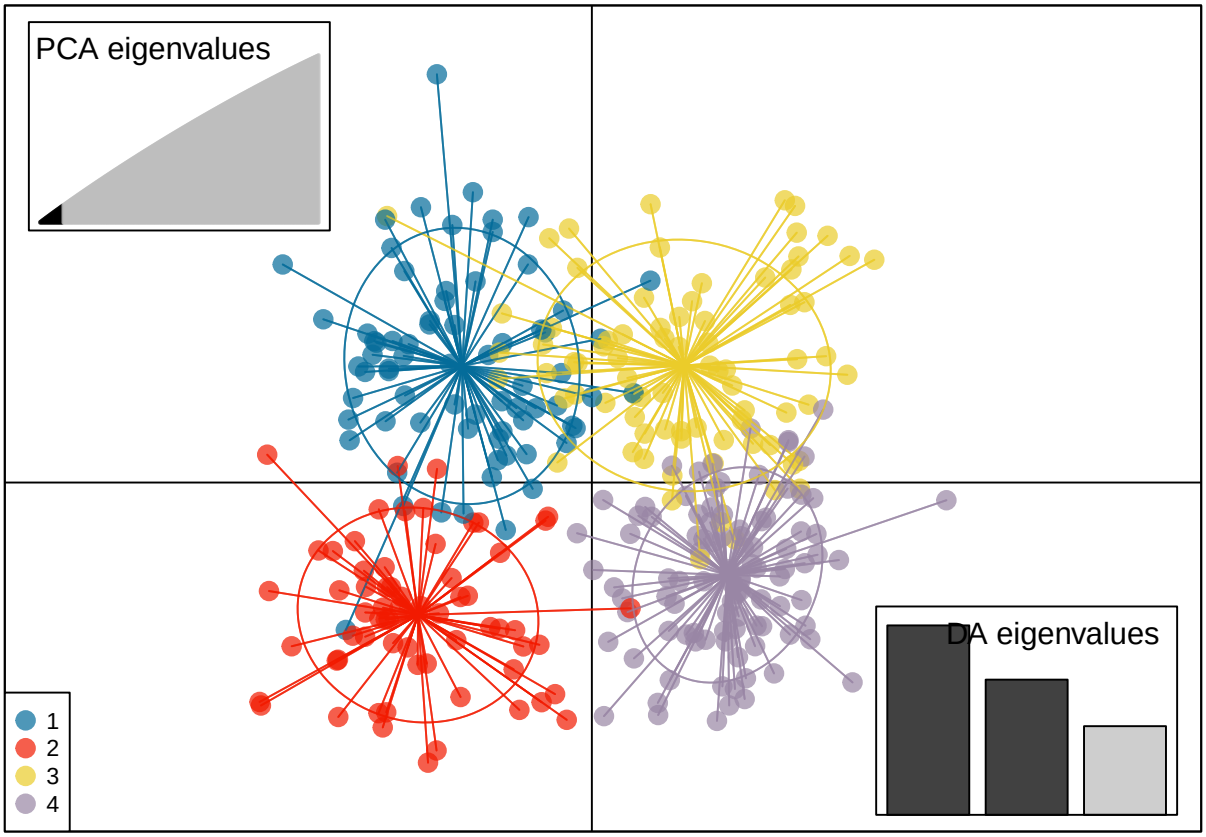


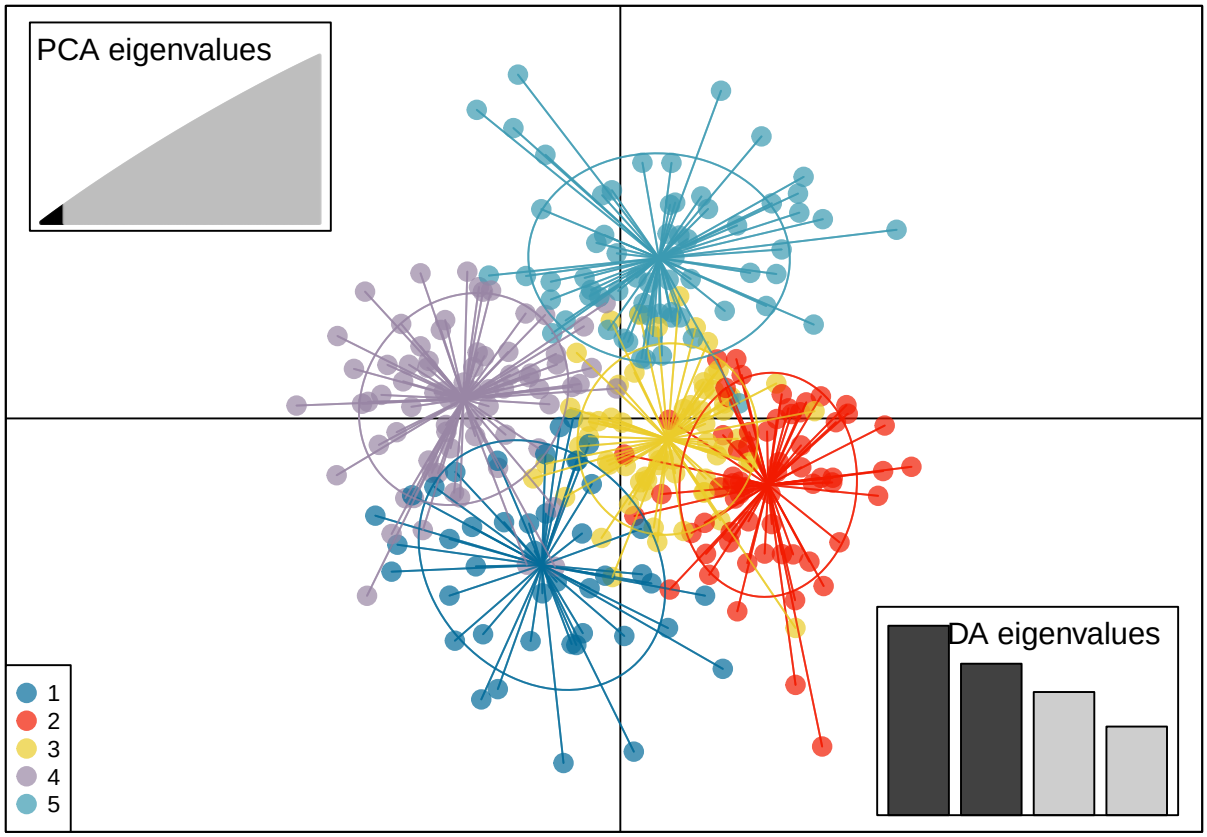

```
for (K in 2:6){
  colour <- colours6[1:K]
  dapc.neutral <- get(dapc.all.neutral.object.names[K - 1])
```

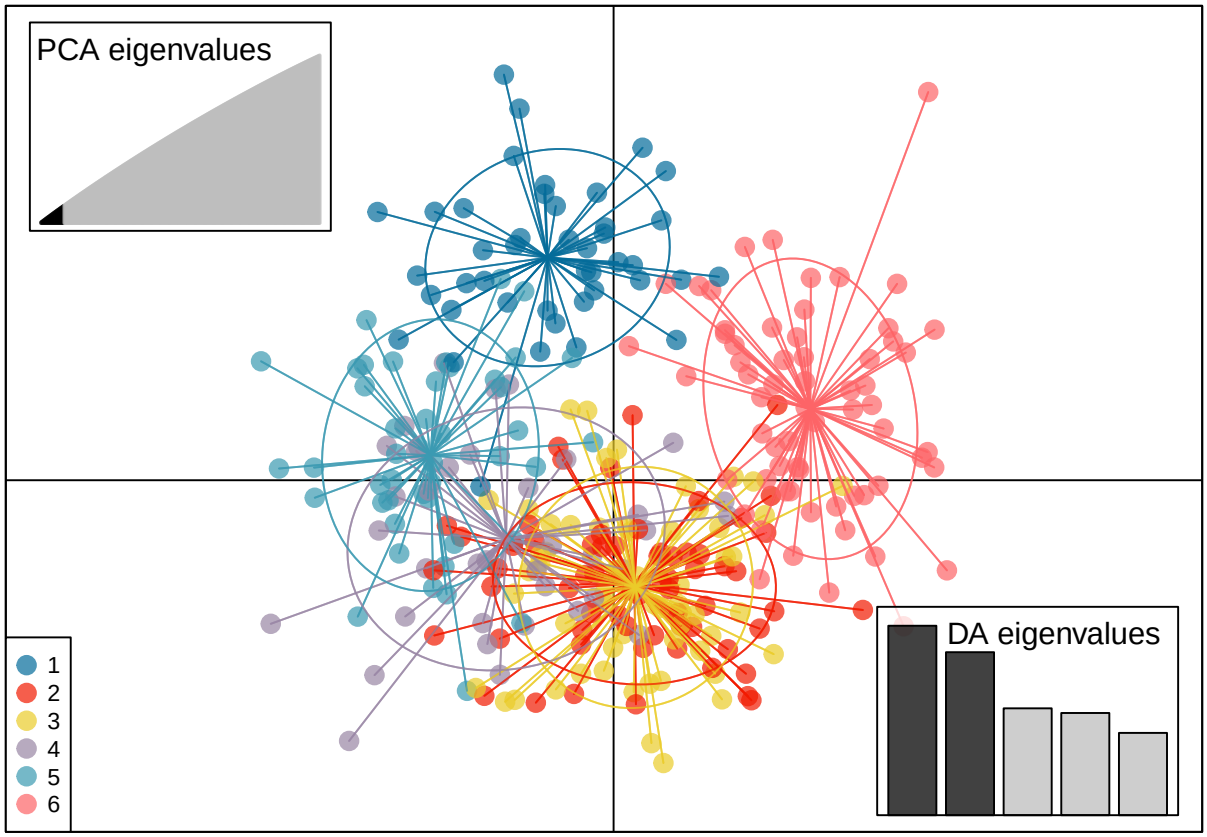
```
  dapc.neutral.plot <- ade4::scatter(dapc.neutral, cex = 2, legend = TRUE,
                                     col = colour, clabel = FALSE,
                                     posi.leg = "bottomleft", scree.pca = TRUE,
                                     posi.pca = "topleft", cleg = 0.75, xax = 1,
                                     yax = 2, inset.solid = 0.75)
  dev.print(
    device = png,
    file = paste0("figures/3.DAPC_Scatter_neutral_37655SNPs_pergroups_K",
                  K,".png"),
    width = 30,
    height = 15,
    units = "cm",
    res = 300
  )
}
```




PCA eigenvalues
DA eigenvalues
1
2
3


PCA eigenvalues
DA eigenvalues
1
2
3
4






```
dapc <- get(dapc.all.neutral.object.names[1])
```

```
df <- data.frame(gl.all.neutral$other$ind.metrics, DAPC_GROUP = dapc$assign)
write.csv(df, file = "outfile/new_metadata_with_DAPC_grouping.csv",
          row.names = FALSE)
```

Identify South African groups for K=2

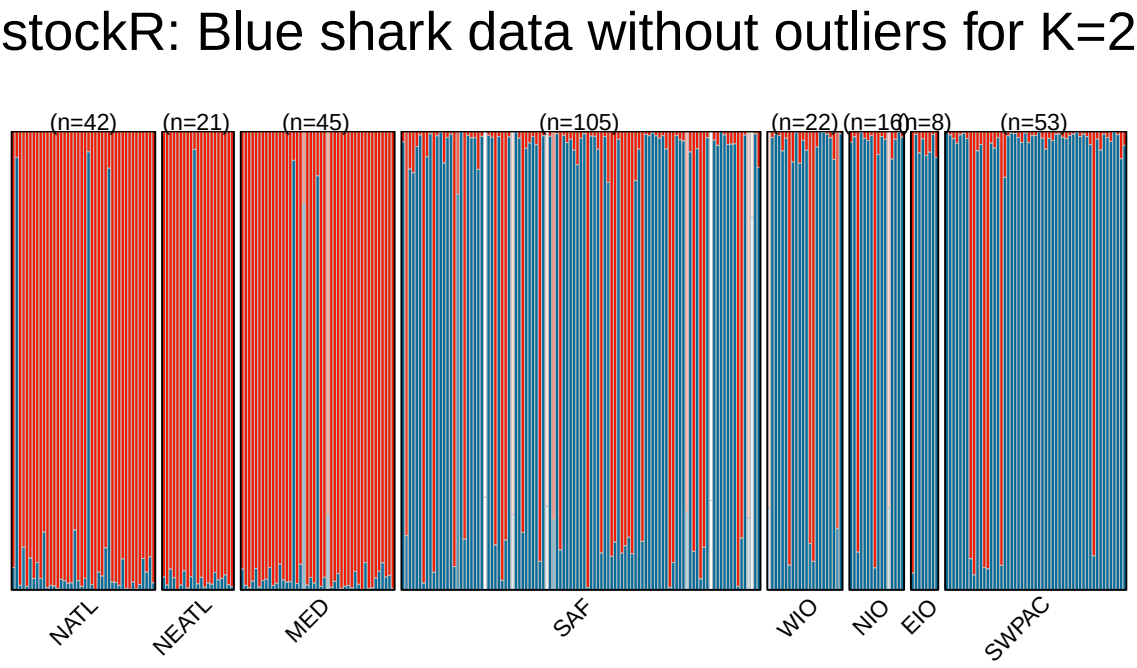
7_stockR

```
################only non outlier
sample.grps <- attr(stockr.all.neutral, "sample.grps")
stock.all.neutral.object.names <- c()
for (K in 2:6) {
  stock.all.neutral <- stockR::stockSTRUCTURE(
    SNPdata = stockr.all.neutral,
    K = K,
  )

  stockBOOT.all.neutral <- stockR::stockBOOT(stock.all.neutral, B = 100,
                                             mc.cores = parallel::detectCores() - 2)
  assign(paste0("stockBOOT.all.neutral", K),value = stockBOOT.all.neutral)
  stock.all.neutral.object.names <- c(stock.all.neutral.object.names,
                                      paste0("stockBOOT.all.neutral", K))
}
save(list=stock.all.neutral.object.names , stock.all.neutral.object.names,
     file = "Rdata/stockR.ALL.neutral.Rdata")


####################only outlier
sample.grps <- attr(stockr.all.outlier, "sample.grps")
stock.all.outlier.object.names <- c()
for (K in 2:6) {
  stock.all.outlier <- stockR::stockSTRUCTURE(
    SNPdata = stockr.all.outlier,
    K = K,
  )

  stockBOOT.all.outlier <- stockR::stockBOOT(stock.all.outlier, B = 100,
                                             mc.cores = parallel::detectCores() - 2)
  assign(paste0("stockBOOT.all.outlier", K),value = stockBOOT.all.outlier)
  stock.all.outlier.object.names <- c(stock.all.outlier.object.names,
                                      paste0("stockBOOT.all.outlier", K))
}
save(list=stock.all.outlier.object.names , stock.all.outlier.object.names,
     file = "Rdata/stockR.ALL.outlier.Rdata")
```

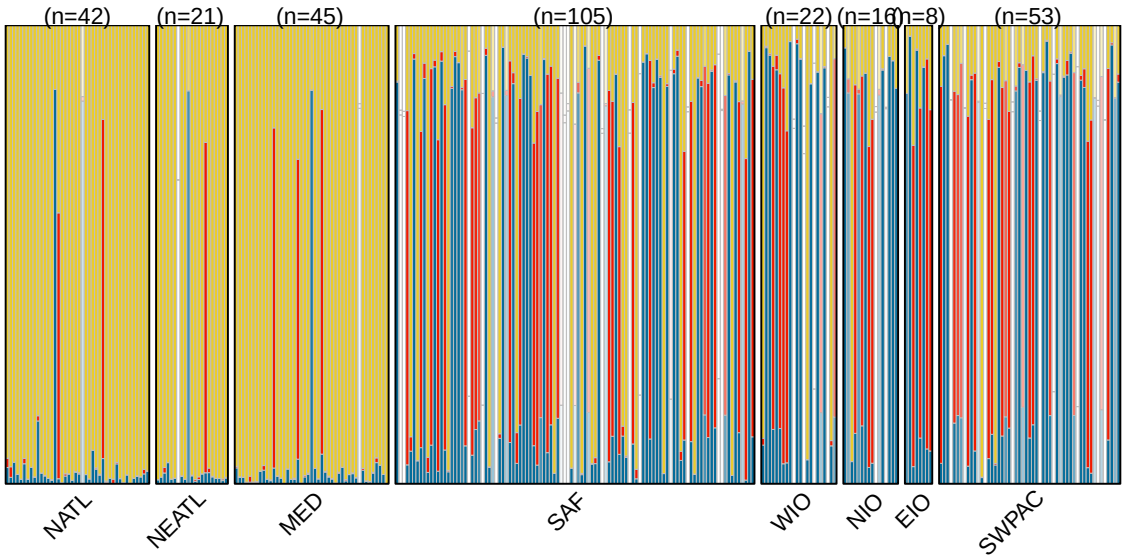
```
load("Rdata/stockR.ALL.neutral.Rdata")
grps <- attr(stockr.all.neutral, "grps")
levels(grps) <- c("NATL", "NEATL", "MED", "SAF", "WIO", "NIO", "EIO", "SWPAC")
for (K in 2:6) {
  colour <- colours6[1:K]
  plot.stockBOOT.object(
    get(stock.all.neutral.object.names[K - 1]),
    locations = data.frame(grps,
    order = as.integer(grps), las = 2),
    region.lwd = 1,
    CI.width = 0.70,
    colour = colour,
    plotTitle = paste0("stockR: Blue shark data without outliers for K=", K)
  )
  dev.print(
```

```
    device = png,
    file = paste0("figures/4.StockR_barplot_neutral_37655SNPs_K",K,".png"),
    width = 30,
    height = 15,
    units = "cm",
    res = 300
  )
}
```

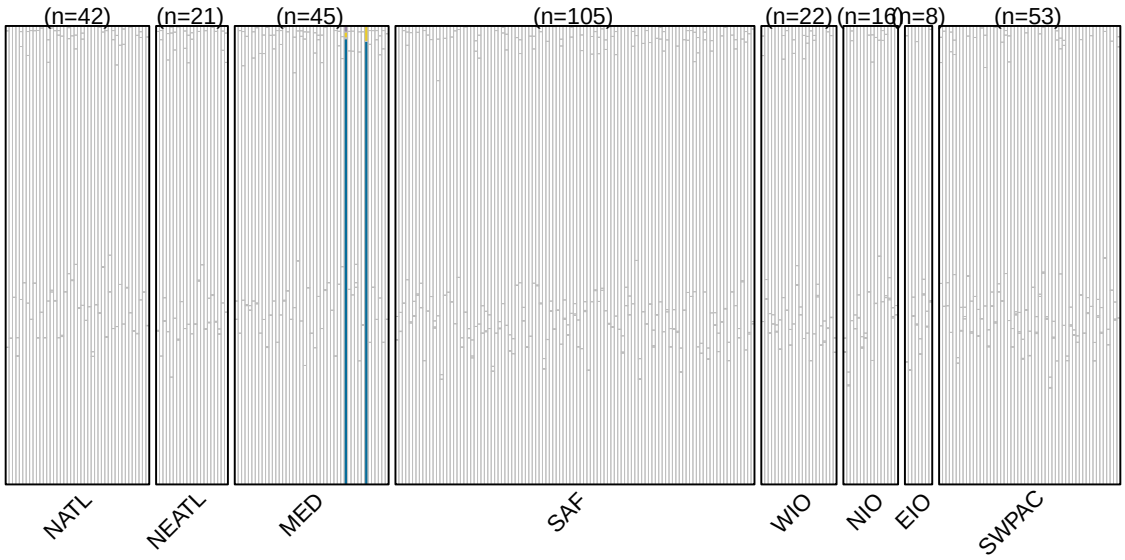
stockR barplot with neutral markers



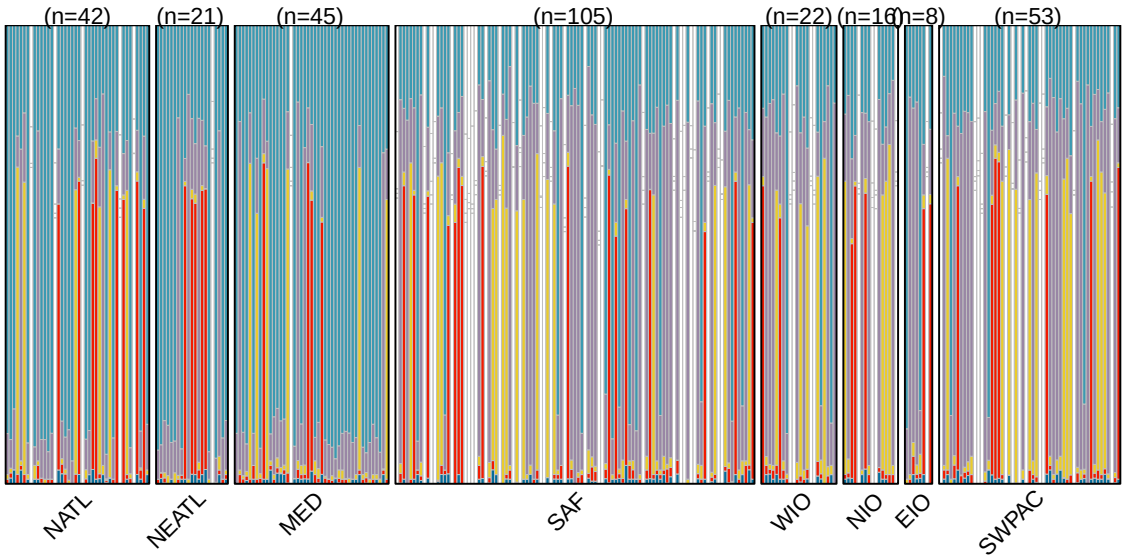
stockR: Blue shark data without outliers for K=3



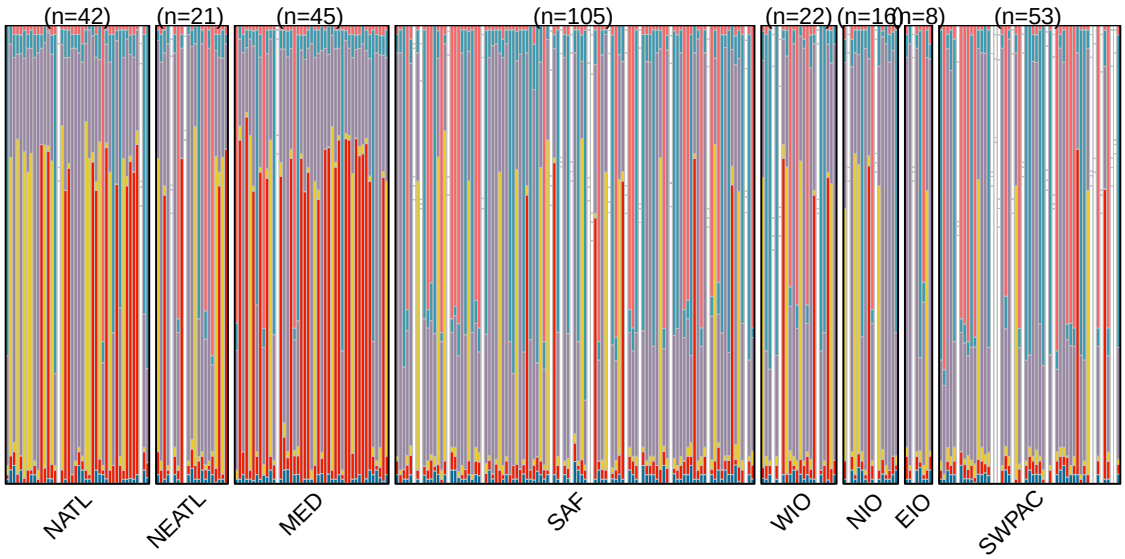
stockR: Blue shark data without outliers for K=4



stockR: Blue shark data without outliers for K=5



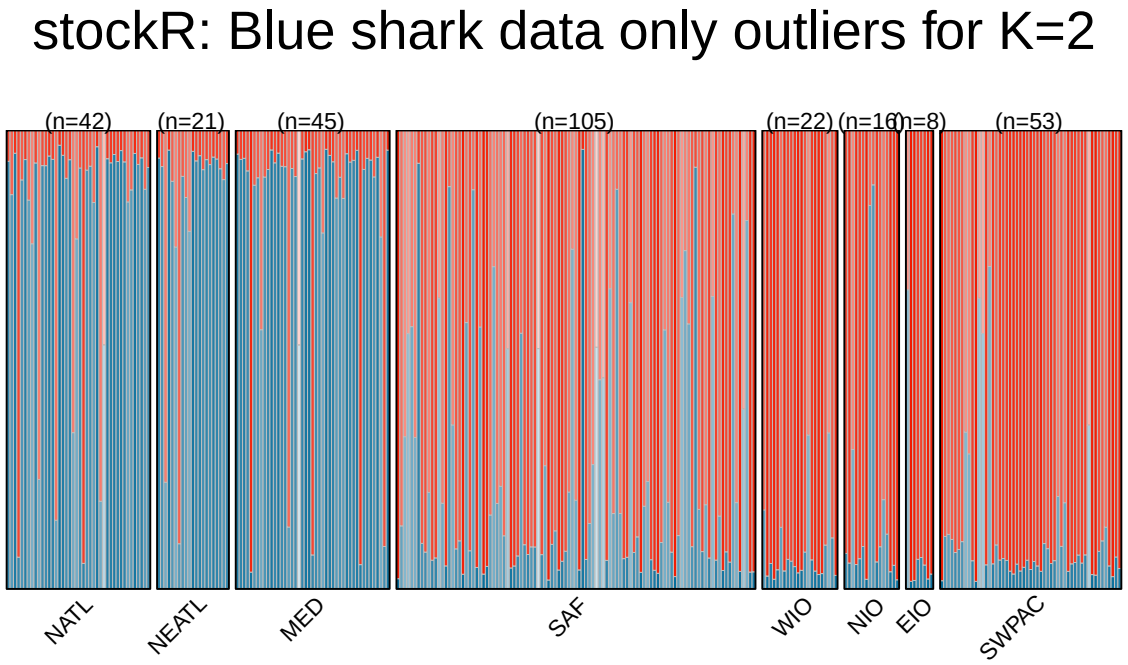
stockR: Blue shark data without outliers for K=6



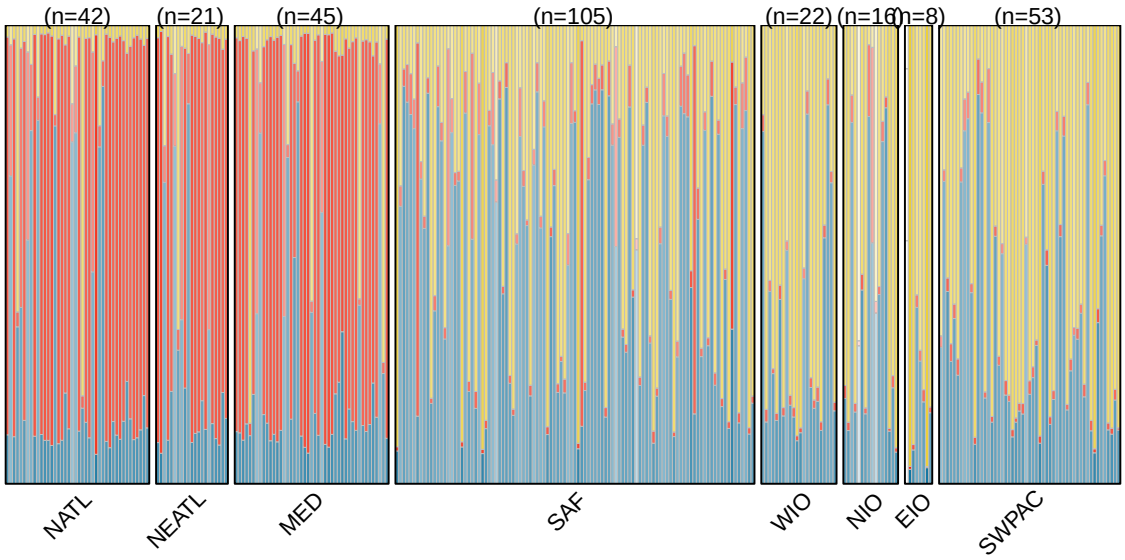

```
load("Rdata/stockR.ALL.outlier.Rdata")
```

```
grps <- attr(stockr.all.outlier, "grps")
levels(grps) <- c("NATL", "NEATL", "MED", "SAF", "WIO", "NIO", "EIO", "SWPAC")
for (K in 2:6) {
  colour <- colours6[1:K]
  plot.stockBOOT.object(
    get(stock.all.outlier.object.names[K - 1]),
    locations = data.frame(grps,
                           order = as.integer(grps, las = 2)),
    region.lwd = 1,
    CI.width = 0.70,
    colour = colour,
    plotTitle = paste0("stockR: Blue shark data only outliers for K=", K)
  )
  dev.print(
    device = png,
    file = paste0("figures/4.StockR_barplot_outliers_9SNPs_K", K, ".png"),
    width = 30,
    height = 15,
    units = "cm",
    res = 300
  )
}
```

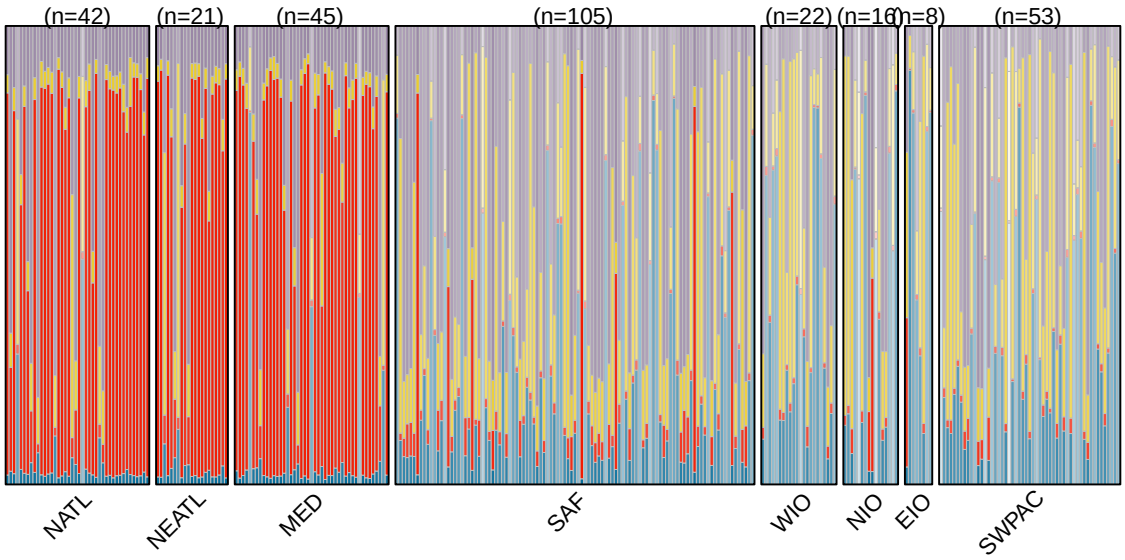
stockR barplot with outlier markers



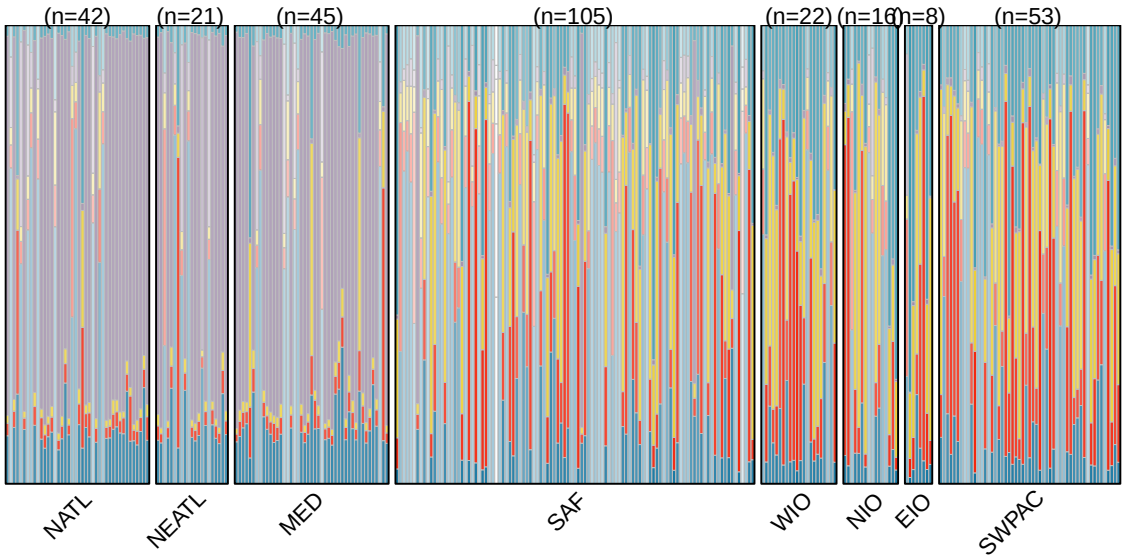
stockR: Blue shark data only outliers for K=3



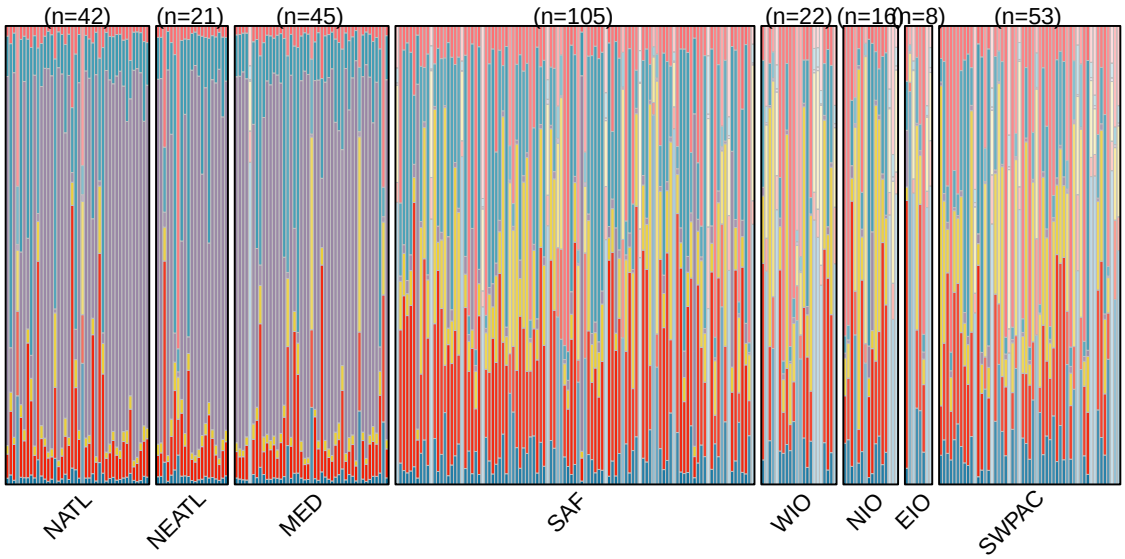
stockR: Blue shark data only outliers for K=4



stockR: Blue shark data only outliers for K=5



stockR: Blue shark data only outliers for K=6



8_ADMIXTURE

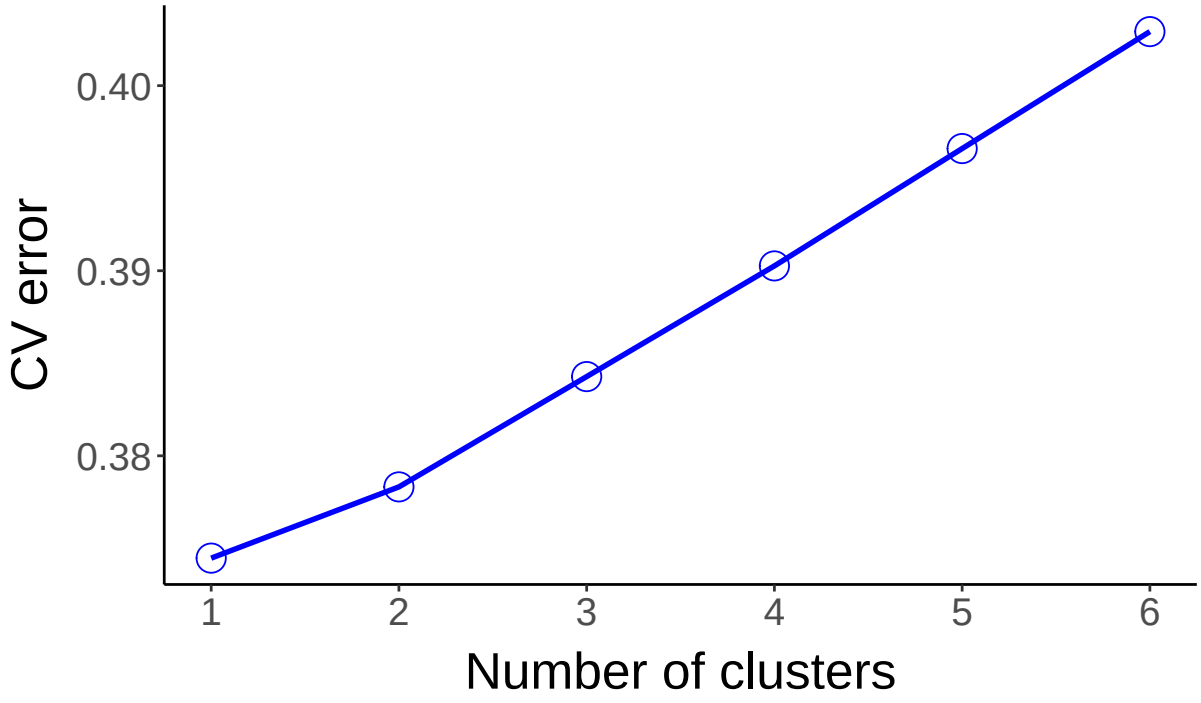
```
gl2Adm(gl.all.neutral, filename = "BS_all_ADMIXTURE")

gl2Adm(gl.all.neutral[gl.all.neutral$pop == "Atlantic-SE"],
       filename = "BS_SAF_ADMIXTURE")
gl2Adm(gl.all.neutral[gl.all.neutral$pop %in% c("Atlantic-SE","Atlantic-N",
                                                "Atlantic-NE", "Mediterranean")],
       filename = "BS_ATL_ADMIXTURE")
gl2Adm(gl.all.neutral[!gl.all.neutral$pop %in% c("Atlantic-N",
                                                 "Atlantic-NE", "Mediterranean")],
       filename = "BS_IO_ADMIXTURE")
```

ADMIXTURE with neutral markers The ADMIXTURE software was run in a Linux environment with the following command line:

```
for K in 1 2 3 4 5 6; \
do ./admixture -B10000 --cv=100 BS_all_ADMIXTURE.ped $K | tee log${K}.out; done
```

```
CV <- c()
for (K in 1:6) {
  log <- data.table::fread(input =  paste0("./Rdata/ADMIXTURE/BSlog",K,".out",
                                           sep = ""), sep = "", header = FALSE,
                           blank.lines.skip = TRUE)
  log <- log$V1
  CV1 <- subset(log, grepl(pattern = "CV error", x = log, perl = TRUE))
  CV2 <- unlist(strsplit(CV1, " "))
  CV[K] <- as.numeric(CV2[4])
}

y <- CV
x <- 1:6
cvdata <- data.frame(x,y,stringsAsFactors = F)
plot <- ggplot2::ggplot(cvdata, ggplot2::aes(x,y)) +
  ggplot2::geom_point(size = 5, shape = 1, color = "blue") +
  ggplot2::geom_line(size = 1,color = "blue") +
  ggplot2::scale_x_continuous(name = waiver(),
                              breaks = seq(from = 0, to = 8, by = 1)) +
  ggplot2::labs(subtitle = "",
                y = "CV error",
                x = "Number of clusters",
                title = "",
                caption = "") +
  ggplot2::theme_classic() +
  ggplot2::theme(
    axis.text = ggplot2::element_text(size = 15),
    axis.title.x = ggplot2::element_text(size = 20,vjust = 0, hjust = 0.5),
    axis.title.y = ggplot2::element_text(size = 20,vjust = 2, hjust = 0.5))
print(plot)
```

CVplot

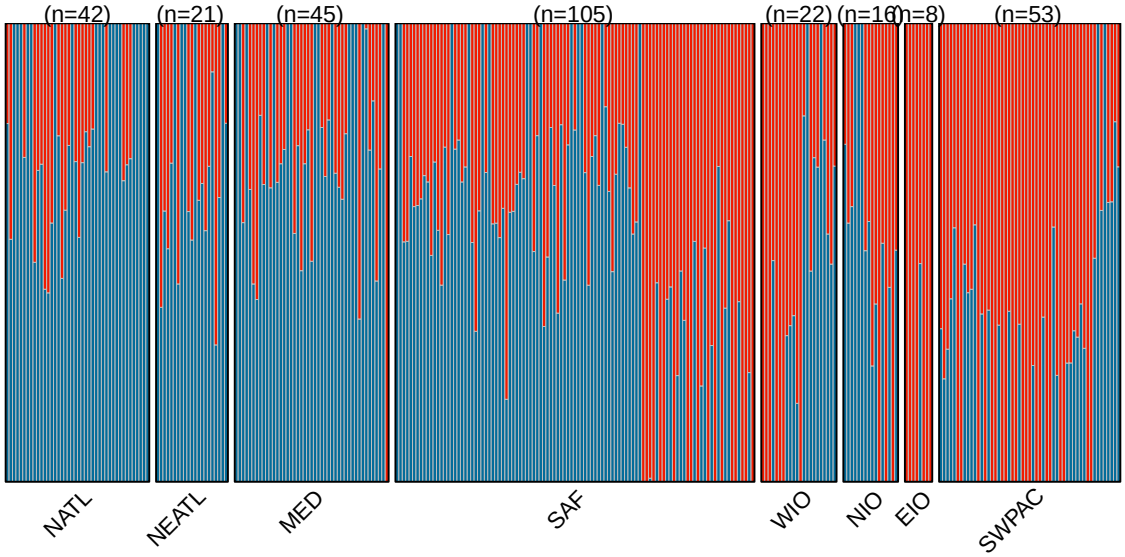



```
ggsave(filename = "./figures/9.ADMIXTURE_CVlot_neutral_37655SNPs.png",
       plot = plot, width = 15, height = 15, units = "cm")
```

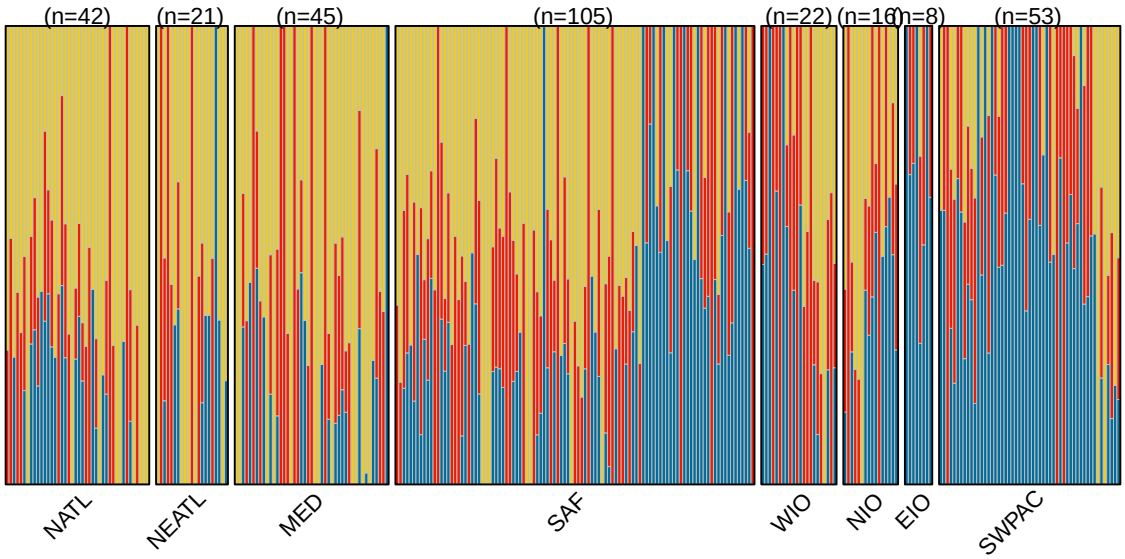
```
Ind.Names <- read.table("./Rdata/ADMIXTURE/BS_all_ADMIXTURE.ped")

for (K in 2:6) {
  colour <- colours6[1:K]
  tbl <- read.table(paste0("./Rdata/ADMIXTURE/BS_all_ADMIXTURE.",K,".Q",
                           sep = ""))
  row.names(tbl) <- Ind.Names$V2
  colnames(tbl) <- paste0("Group", 1:ncol(tbl))
  tbl <- as.matrix(tbl)
  locations <- gl.all.neutral$pop[order(match(gl.all.neutral$ind.names,Ind.Names$V2))]
  plot.admixture.FDD(x = tbl, locations = locations,
                     colour = colour, region.lwd = 1,
                     plotTitle = paste0("ADMIXTURE: K=",K))
  dev.print(
    device = png,
    file = paste0("./figures/9.ADMIXTURE_barplot_neutral_37655SNPs_K",
                  K,".png"),
    width = 30,
    height = 15,
    units = "cm",
    res = 300
  )
}
```

Barplot

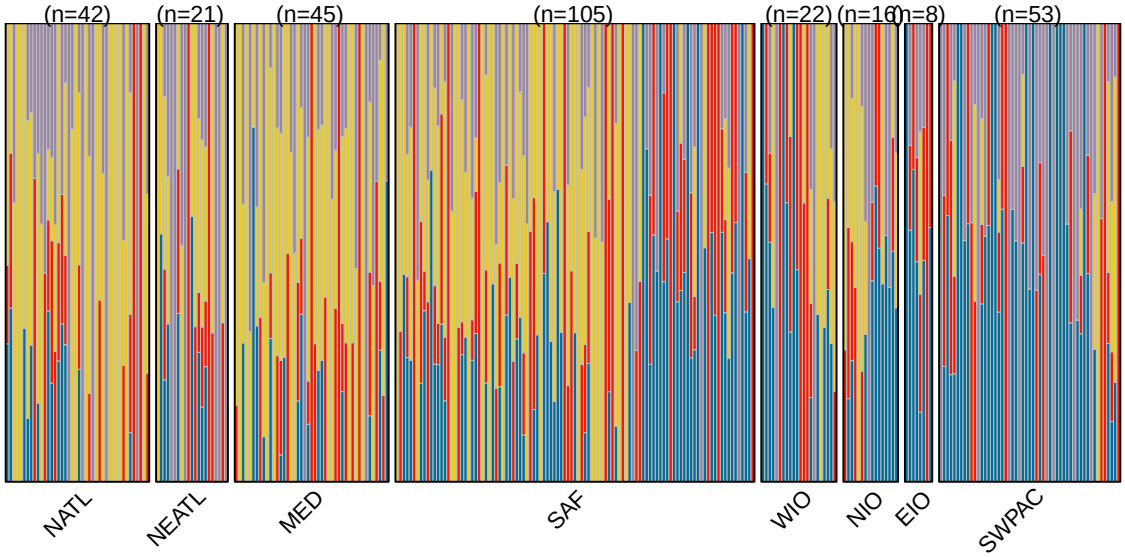
ADMIXTURE: K=2



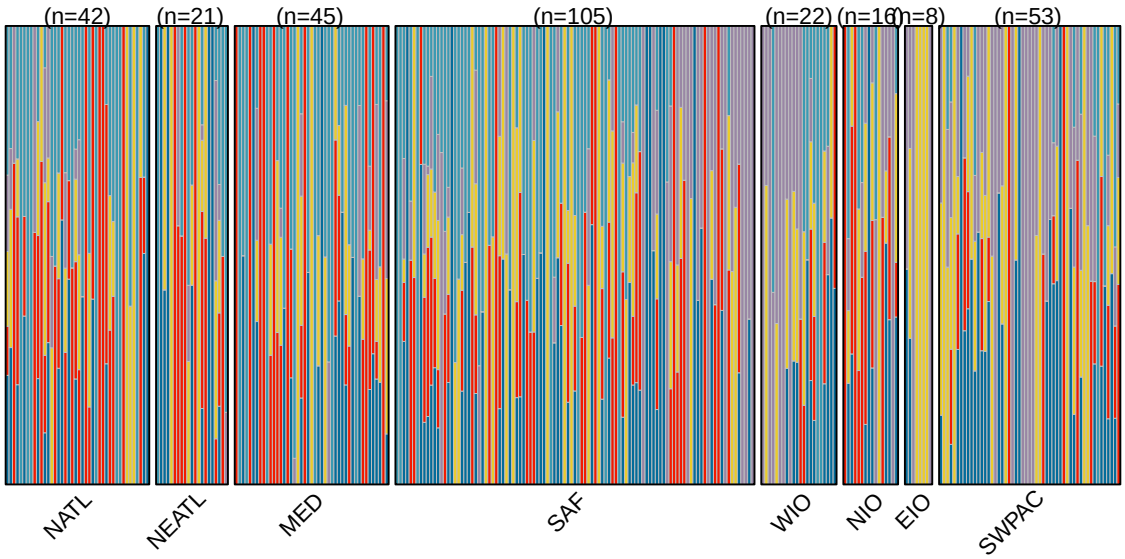
ADMIXTURE: K=3

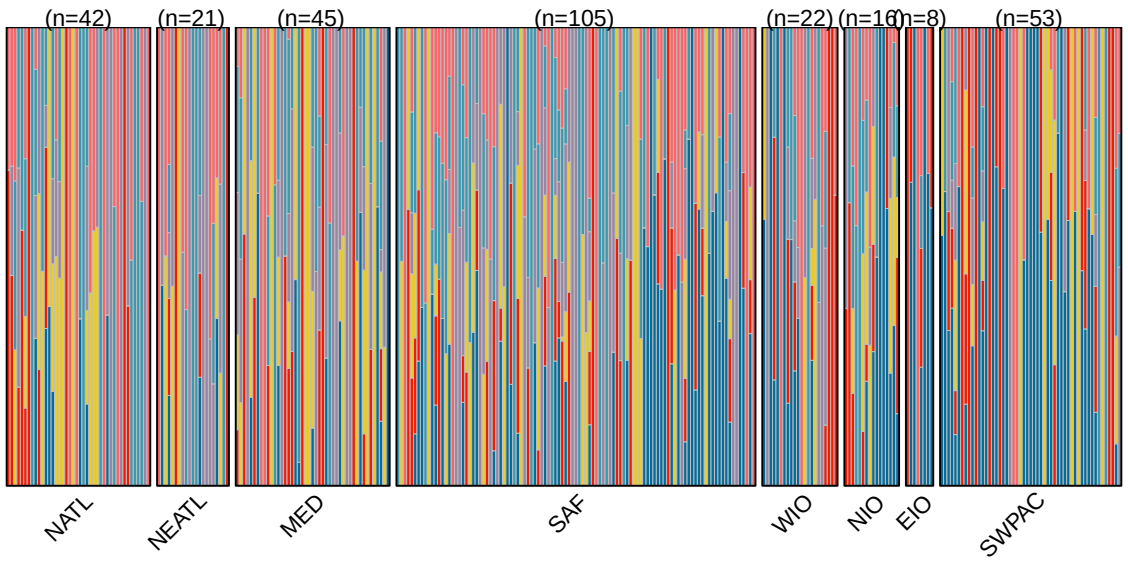


ADMIXTURE: K=4



ADMIXTURE: K=5






```
gl2Adm(gl.all.outlier, filename = "BS_all_outlier_ADMIXTURE")
```

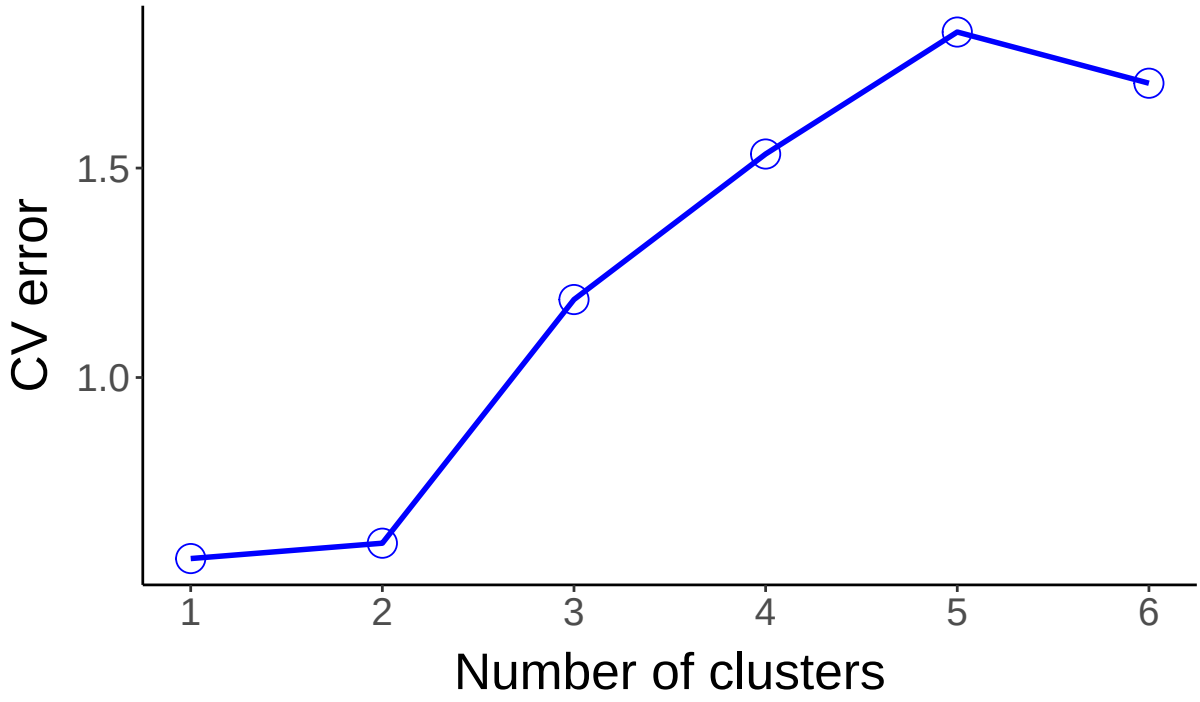
ADMIXTURE with outlier markers The ADMIXTURE software was run in a Linux environment with the following command line:

```
for K in 1 2 3 4 5 6; \
do ./admixture -B10000 --cv=100 BS_all_outlier_ADMIXTURE.ped $K | tee log${K}.out;
done
```

```
CV <- c()
for (K in 1:6) {
  log <- data.table::fread(input =  paste0("./Rdata/ADMIXTURE_outliers/log",K,".out",
                                           sep = ""),
                           sep = "", header = FALSE, blank.lines.skip = TRUE)
  log <- log$V1
  CV1 <- subset(log, grepl(pattern = "CV error", x = log, perl = TRUE))
  CV2 <- unlist(strsplit(CV1, " "))
  CV[K] <- as.numeric(CV2[4])
}

y <- CV
x <- 1:6
cvdata <- data.frame(x,y,stringsAsFactors = F)
plot <- ggplot2::ggplot(cvdata, ggplot2::aes(x,y)) +
  ggplot2::geom_point(size = 5, shape = 1, color = "blue") +
```

```
  ggplot2::geom_line(size = 1,color = "blue") +
  ggplot2::scale_x_continuous(name = ggplot2::waiver(),
                              breaks = seq(from = 0, to = 8, by = 1)) +
  ggplot2::labs(subtitle = "",
                y = "CV error",
                x = "Number of clusters",
                title = "",
                caption = "") +
  ggplot2::theme_classic() +
  ggplot2::theme(
    axis.text = ggplot2::element_text(size = 15),
    axis.title.x = ggplot2::element_text(size = 20,vjust = 0, hjust = 0.5),
    axis.title.y = ggplot2::element_text(size = 20,vjust = 2, hjust = 0.5))
print(plot)
```

CVplot

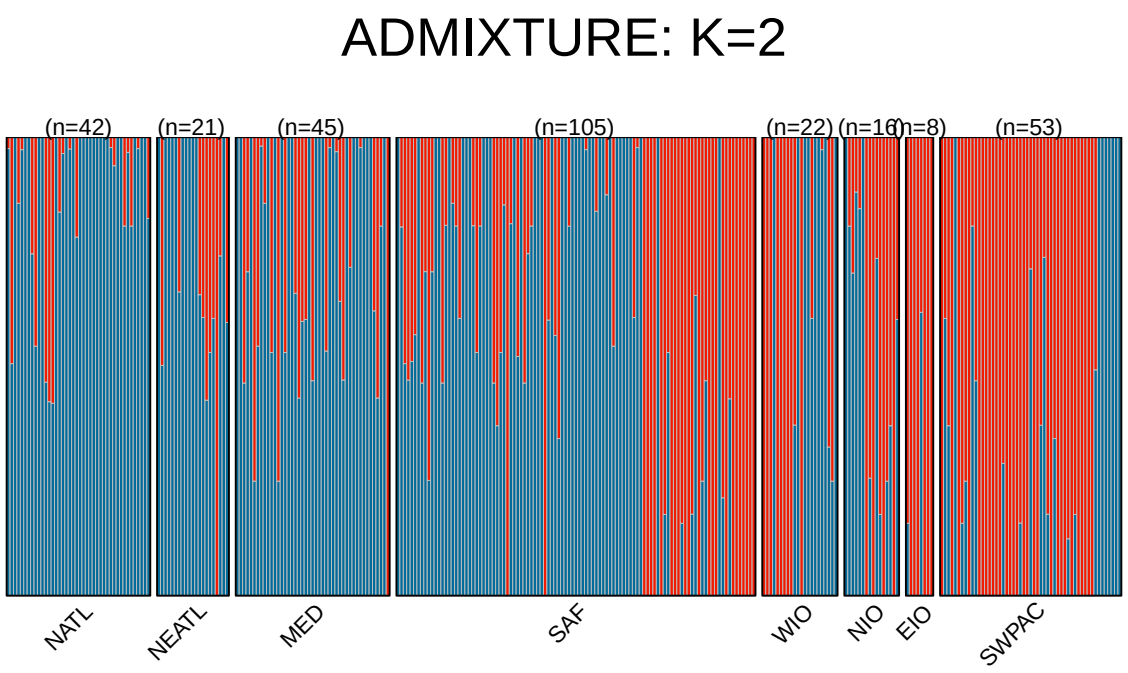



```
ggsave(filename = "./figures/9.ADMIXTURE_CVlot_outliers_9SNPs.png", plot = plot,
       width = 15, height = 15, units = "cm")
```

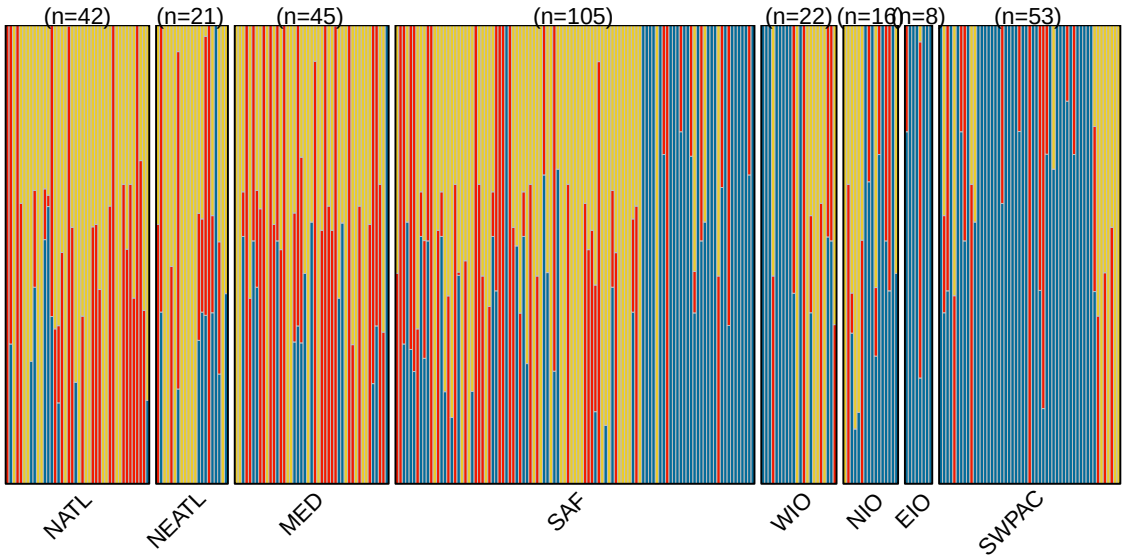
```
Ind.Names <- read.table("./Rdata/ADMIXTURE_outliers/BS_all_outlier_ADMIXTURE.ped")
for (K in 2:6) {
  colour <- colours6[1:K]
  tbl <- read.table(paste0("./Rdata/ADMIXTURE_outliers/BS_all_outlier_ADMIXTURE.",
                           K,".Q", sep = ""))
  row.names(tbl) <- Ind.Names$V2
  colnames(tbl) <- paste0("Group", 1:ncol(tbl))
```

```
  tbl <- as.matrix(tbl)
  locations <- gl.all.neutral$pop[order(match(gl.all.neutral$ind.names,Ind.Names$V2))]
  plot.admixture.FDD(x = tbl, locations = locations,
                     colour = colour, region.lwd = 1,
                     plotTitle = paste0("ADMIXTURE: K=",K))
  dev.print(
    device = png,
    file = paste0("./figures/9.ADMIXTURE_barplot_outliers_9SNPs_K",
                  K,".png"),
    width = 30,
    height = 15,
    units = "cm",
    res = 300
  )
}
```

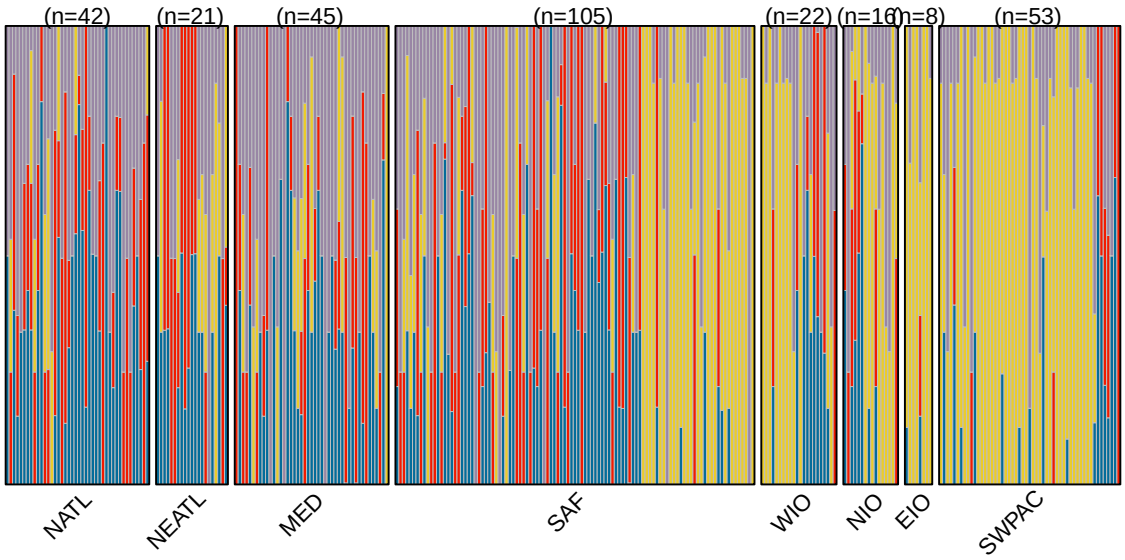
Barplot



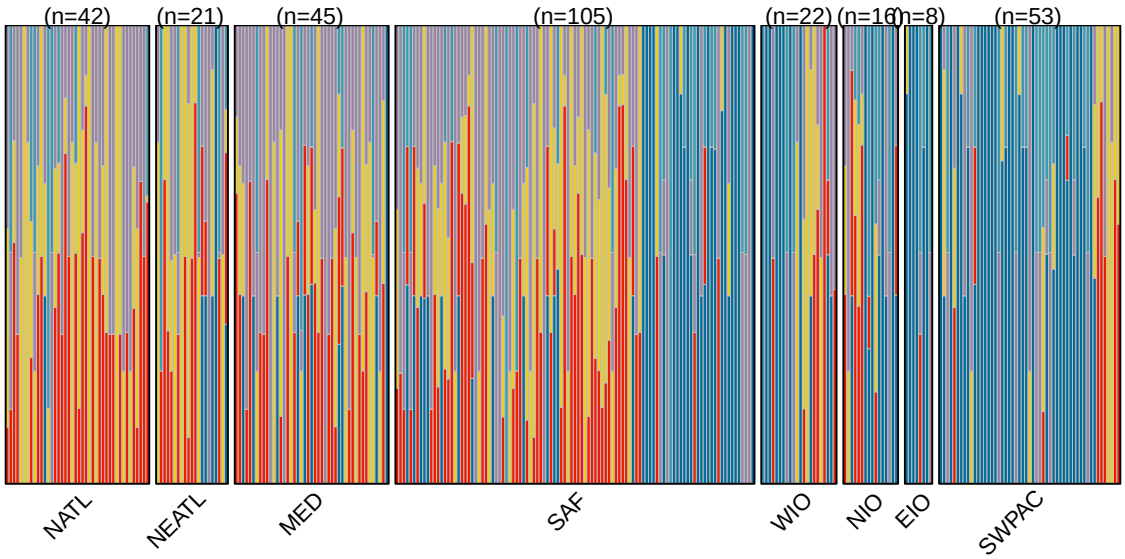
ADMIXTURE: K=3



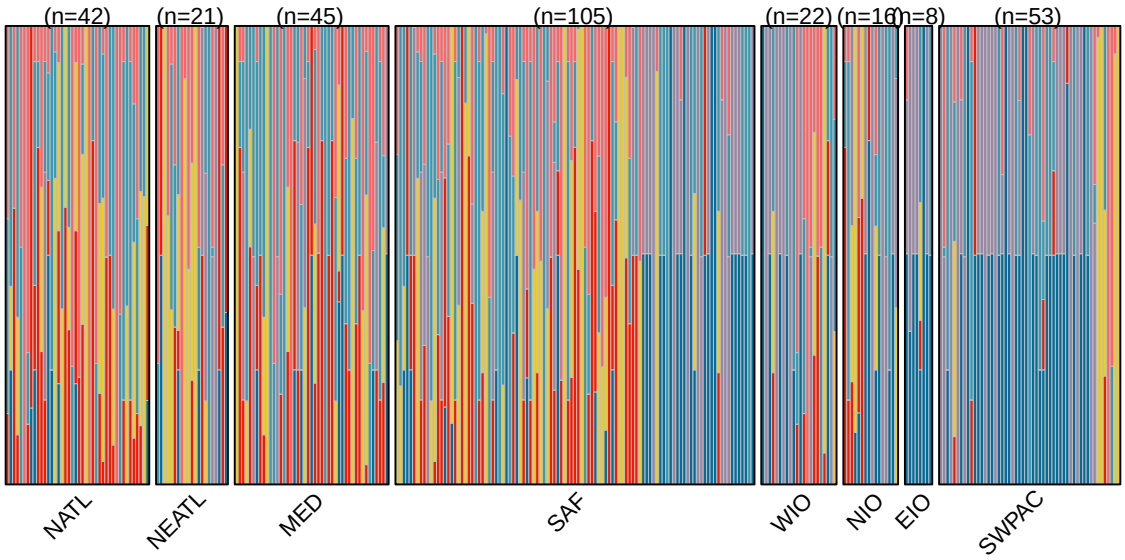
ADMIXTURE: K=4



ADMIXTURE: K=5



ADMIXTURE: K=6



Extra analyses

For these extra analyses I assume the population structure is known, based on the above analyses. This will increase the sample size and power for demograhic analyses.

```
# load("Rdata/Blue_shark_Neutral_and_OutlierData_Corrected2.Rdata")
gl.all.neutral <- gl.all.neutral[order(gl.all.neutral$pop,gl.all.neutral$ind.names),]

strat <- as.character(gl.all.neutral$pop)
names(strat) <- gl.all.neutral$ind.names
# gt.all.neutral@schemes$id <- gt.all.neutral@schemes$INDIVIDUALS
strataG::setStrata(gt.all.neutral) <- strat
```

1_Sex-specific connectivity

```


load("Rdata/Blue_shark_sex_data.Rdata")
## Using all X-linked marker
adegenet::nInd(gl.x.F.filtered)
```

Sex-linked markers

```
## [1] 143
```

```
adegenet::nLoc(gl.x.F.filtered)
```

```
## [1] 298
```

```
levels(gl.x.F.filtered$pop) <-  c("NATL", "NATL", "MED", "IO_PAC", "IO_PAC",
                                  "IO_PAC", "IO_PAC", "IO_PAC")

gl.x.F.filtered <- gl.x.F.filtered[order(gl.x.F.filtered$pop),]
strat <- as.character(gl.x.F.filtered$pop)
names(strat) <- gl.x.F.filtered$ind.names
strataG::setStrata(gt.x.F.filtered) <- strat


## Using X-linked marker that were found by both heterozygosity and coverage method
adegenet::nInd(gl.x.F.strict.filtered)
```

```
## [1] 143
```

```
adegenet::nLoc(gl.x.F.strict.filtered)
```

```
## [1] 69
```

```
levels(gl.x.F.strict.filtered$pop) <- c("NATL", "NATL", "MED", "IO_PAC", "IO_PAC",
                                        "IO_PAC", "IO_PAC", "IO_PAC")
gl.x.F.strict.filtered <- gl.x.F.strict.filtered[order(gl.x.F.strict.filtered$pop),]
strat <- as.character(gl.x.F.strict.filtered$pop)
names(strat) <- gl.x.F.strict.filtered$ind.names
strataG::setStrata(gt.x.F.strict.filtered) <- strat
```

```
gl2gpop(gl.all.neutral,
        filename = "Rdata/Genepop_file_45667SNP_Neutral_312ind_perPOP.txt")
result <- diveRsity::basicStats(
  infile = "Rdata/Genepop_file_45667SNP_Neutral_312ind_perPOP.txt",
  outfile = "outfile/Result_Diversity.txt", fis_ci = TRUE, ar_ci = TRUE,
  fis_boots = 1000, ar_boots = 1000, mc_reps = 1000, rarefaction = FALSE,
  ar_alpha = 0.05, fis_alpha = 0.05)

gendiv.neutral.pop <- data.frame(
  "Atlantic_N" = result$main_tab$`Pg-AZO-105`$overall,
  "Mediterranean" = result$main_tab$`Pg-GDL-06`$overall,
  "Indian_Pacific" = result$main_tab$`NZ-1.2`$overall
)
rownames(gendiv.neutral.pop) <- rownames(result$main_tab$`Pg-AZO-105`)
readr::write_tsv(gendiv.neutral.pop,
                 path = "outfile/Table_Diversity_neutral_perPop.txt")



gl2gpop(gl.x.F.filtered,
        filename = "Rdata/Genepop_file_298SNP_X-linked_143ind.txt")
result <- diveRsity::basicStats(
  infile = "Rdata/Genepop_file_298SNP_X-linked_143ind.txt",
  outfile = "outfile/Result_Diversity.txt", fis_ci = TRUE, ar_ci = TRUE,
  fis_boots = 1000, ar_boots = 1000, mc_reps = 1000, rarefaction = FALSE,
  ar_alpha = 0.05, fis_alpha = 0.05)

gendiv.x.F.filtered <- data.frame(
  "Atlantic_N" = result$main_tab$`67616`$overall,
  "Mediterranean" = result$main_tab$`68274`$overall,
  "Indian_Pacific" = result$main_tab$`60112`$overall
)
rownames(gendiv.x.F.filtered) <- rownames(result$main_tab$`67616`)
readr::write_tsv(gendiv.x.F.filtered,
                 path = "outfile/Table_Diversity_X-linked1_perPop.txt")



gl2gpop(gl.x.F.strict.filtered,
        filename = "Rdata/Genepop_file_69SNP_X-linked_143ind.txt")
result <- diveRsity::basicStats(
  infile = "Rdata/Genepop_file_69SNP_X-linked_143ind.txt",
  outfile = "outfile/Result_Diversity.txt", fis_ci = TRUE, ar_ci = TRUE,
  fis_boots = 1000, ar_boots = 1000, mc_reps = 1000, rarefaction = FALSE,
  ar_alpha = 0.05, fis_alpha = 0.05)

gendiv.x.F.strict.filtered <- data.frame(
  "Atlantic_N" = result$main_tab$`67616`$overall,
  "Mediterranean" = result$main_tab$`68274`$overall,
  "Indian_Pacific" = result$main_tab$`60112`$overall
)
rownames(gendiv.x.F.strict.filtered) <- rownames(result$main_tab$`67616`)
readr::write_tsv(gendiv.x.F.strict.filtered,
                 path = "outfile/Table_Diversity_X-linked2_perPop.txt")
```

```
save(gendiv.neutral.pop,gendiv.x.F.filtered,gendiv.x.F.strict.filtered,
     file = "Rdata/ALL_Genetic_diversity_perPOP.Rdata")
```

```
load("Rdata/ALL_Genetic_diversity_perPOP.Rdata")
knitr::kable(gendiv.neutral.pop, col.names = c("NATL", "MED", "IO_PAC"),
             caption = "Genetic diversity for all neutral loci")
```

Genetic diversity

Table 15: Genetic diversity for all neutral loci

	NATL	MED	IO_PAC
ar	1.794	1.751	1.803
size	61.887	44.354	201.001
obs_het	0.121	0.119	0.117
exp_het	0.142	0.138	0.139
uexp_het	0.143	0.140	0.139
fis	0.098	0.093	0.111
hwe_glb	0.000	1.000	0.000
hwe_hom	1.000	1.000	1.000
hwe_het	0.000	1.000	0.000
fis_lo	0.085	0.079	0.102
fis_hi	0.096	0.091	0.108
ar_lo	1.770	1.724	1.787
ar_hi	1.815	1.776	1.817

```
knitr::kable(gendiv.x.F.filtered, col.names = c("NATL", "MED", "IO_PAC"),
             caption = "Genetic diversity for all X-linked loci: Only females")
```

Table 16: Genetic diversity for all X-linked loci: Only females

	NATL	MED	IO_PAC
ar	1.555	1.479	1.563
size	53.997	25.232	59.376
obs_het	0.089	0.091	0.094
exp_het	0.111	0.107	0.109
uexp_het	0.112	0.109	0.110
fis	0.119	0.120	0.114
hwe_glb	0.934	1.000	0.888
hwe_hom	1.000	1.000	1.000
hwe_het	0.608	1.000	0.672
fis_lo	0.090	0.063	0.087
fis_hi	0.136	0.137	0.126
ar_lo	1.507	1.440	1.517
ar_hi	1.597	1.513	1.607

```
knitr::kable(gendiv.x.F.strict.filtered, col.names = c("NATL", "MED", "IO_PAC"),
             caption = "Genetic diversity for X-linked loci (detected by 2 methods):
             Only females")
```

Table 17: Genetic diversity for X-linked loci (detected by 2 methods): Only females

	NATL	MED	IO_PAC
ar	1.744	1.732	1.785
size	52.551	24.609	57.362
obs_het	0.197	0.214	0.215
exp_het	0.233	0.226	0.246
uexp_het	0.235	0.231	0.249
fis	0.182	0.087	0.225
hwe_glb	0.001	0.991	0.000
hwe_hom	1.000	1.000	1.000
hwe_het	0.000	0.996	0.000
fis_lo	0.109	-0.027	0.143
fis_hi	0.207	0.114	0.239
ar_lo	1.696	1.681	1.725
ar_hi	1.797	1.768	1.841

```
Fst.all.pop <-
  strataG::popStructTest(
    gt.all.neutral,
    nrep = 100,
    stats = c("fst", "fis"),
    type = "pairwise",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores() - 2,
    write.output = FALSE
  )


Fst.x.F.filtered <-
  strataG::popStructTest(
    gt.x.F.filtered,
    nrep = 100,
    stats = c("fst", "fis"),
    type = "pairwise",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores() - 2,
    write.output = FALSE
  )


Fst.x.F.strict.filtered <-
  strataG::popStructTest(
    gt.x.F.strict.filtered,
    nrep = 100,
    stats = c("fst", "fis"),
    type = "pairwise",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores() - 2,
    write.output = FALSE
  )
```

Table 18: Pairwise Fst for autosomal, X-linked markers, and X-linked markers under strict filtering

	Fst	Fst.p.val	Fst.1	Fst.p.val.1	Fst.2	Fst.p.val.2
NATL(56) v. IO_PAC (61)	0.0014225	0.009901	-0.0001363	0.5742574	-0.0023956	0.8613861
NATL(56) v. MED (26)	0.0005994	0.009901	-0.0034242	0.9207921	-0.0019781	0.6237624
IO_PAC (61) v. MED (26)	0.0017585	0.009901	0.0009555	0.3168317	0.0017521	0.2871287

Table 19: Pairwise Fis for autosomal and X-linked markers

	Fis	Fis.p.val	Fis.1	Fis.p.val.1	Fis.2	Fis.p.val.2
NATL(56) v. IO_PAC (61)	-0.2859330	0.0693069	-0.0965236	0.0396040	0.0258977	0.0891089
NATL(56) v. MED (26)	-0.4081348	1.0000000	-1.0323827	0.8514851	-0.3258592	0.9405941
IO_PAC (61) v. MED (26)	-0.2945052	0.8613861	-0.4039360	0.4356436	-0.0192009	0.9504950

```
save(Fst.all.pop, Fst.x.F.filtered, Fst.x.F.strict.filtered,
     file = "Rdata/Fst_sex-markers.Rdata")
```

```
load("Rdata/Fst_sex-markers.Rdata")

pwFst <- data.frame(Fst.all.pop$pairwise$result[,-c(1:4, 7:8)],
                    Fst.x.F.filtered$pairwise$result[,c(5:6)],
                    Fst.x.F.strict.filtered$pairwise$result[,c(5:6)])
knitr::kable(pwFst,
             caption = "Pairwise Fst for autosomal, X-linked markers, and
             X-linked markers under strict filtering") %>%
  kableExtra::kable_styling(full_width = FALSE, font_size = 9)
```

nuFst - xFst

```
pwFis <- data.frame(Fst.all.pop$pairwise$result[,-c(1:6)],
                    Fst.x.F.filtered$pairwise$result[,c(7:8)],
                    Fst.x.F.strict.filtered$pairwise$result[,c(7:8)])
knitr::kable(pwFis, caption = "Pairwise Fis for autosomal and X-linked markers") %>%
  kableExtra::kable_styling(full_width = FALSE, font_size = 9)
```

nuFis - xFis

Individual-based tests

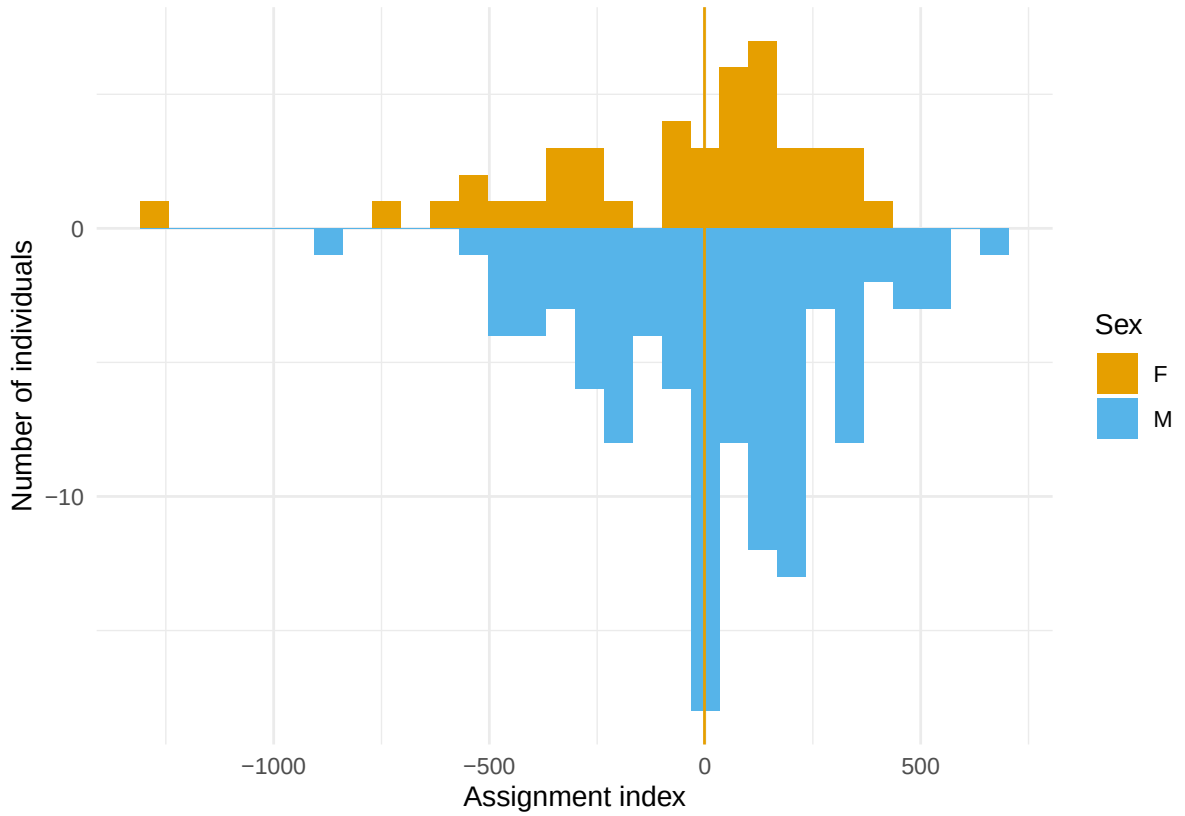
Assignment test I first need the correct length data to keep only adults.

```
gl <- gl.all.neutral
gl <- gl[gl$other$ind.metrics$LENGTH_CM2 > 140 &
           !is.na(gl$other$ind.metrics$LENGTH_CM2),]
dat <- as.data.frame(gl)
dat <- data.frame(Pop = gl$pop[gl$other$ind.metrics$LENGTH_CM2 > 140], dat)
sex <- gl$other$ind.metrics$GENETIC_SEX
mAIc <- hierfstat::sexbias.test(dat,sex,nperm=100,test="mAIc",alternative="two.sided")
vAIc <- hierfstat::sexbias.test(dat,sex,nperm=100,test="vAIc",alternative="two.sided")
```

```
aic <- hierfstat::AIc(dat)
plot.data <- data.frame(sex, aic)

save(mAIc, vAIc, aic, plot.data, file = "Rdata/Aic_by_sex2.rdata")
```

```
load("Rdata/Aic_by_sex2.rdata")
sex <- gl.all.neutral$other$ind.metrics$GENETIC_SEX[
  gl.all.neutral$other$ind.metrics$LENGTH_CM2 > 140 &
  !is.na(gl.all.neutral$other$ind.metrics$LENGTH_CM2)]
aic.F <- aic[sex=="F"]
aic.M <- aic[sex=="M"]

maic.F <- mean(aic.F)
maic.M <- mean(aic.M)
vaic.F <- var(aic.F)
vaic.M <- var(aic.M)

aic.plot <- ggplot2::ggplot() +
  ggplot2::geom_histogram(data = subset(plot.data, sex=="F"),
                          ggplot2::aes(aic, fill="F", y= ..count..))+
  ggplot2::geom_histogram(data = subset(plot.data, sex=="M"),
                          ggplot2::aes(aic, fill="M", y= -..count..))+
  ggplot2::geom_vline(ggplot2::aes(xintercept = maic.F), colour = "#E69F00") +
  ggplot2::geom_vline(ggplot2::aes(xintercept = maic.M), colour = "#56B4E9") +
  ggplot2::scale_fill_manual(values = c("#E69F00", "#56B4E9")) +
  ggplot2::labs(x = "Assignment index",
                y = "Number of individuals",
                fill = "Sex") +
  ggplot2::theme_minimal()
print(aic.plot)
```

```
## Warning: Removed 1 rows containing missing values
## (geom_vline).
```




```
ggsave(plot = aic.plot, filename = "figures/10.Assignment_index_by_sex2.png",
       width = 45, height = 15, units = "cm")
```

```
## Warning: Removed 1 rows containing missing values
## (geom_vline).
```

```
gt.all.neutral.F <- gt.all.neutral[gt.all.neutral@schemes$GENETIC_SEX == "F" &
                                     gt.all.neutral@schemes$LENGTH_CM2 > 140 &
                                     !is.na(gt.all.neutral@schemes$LENGTH_CM),]
gt.all.neutral.M <- gt.all.neutral[gt.all.neutral@schemes$GENETIC_SEX == "M" &
                                     gt.all.neutral@schemes$ LENGTH_CM2 > 140 &
                                     !is.na(gt.all.neutral@schemes$LENGTH_CM),]

Fst.all.pop.F <-
  strataG::popStructTest(
    gt.all.neutral.F,
    nrep = 100,
    stats = c("fst", "fis"),
    type = "pairwise",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores() - 2,
    write.output = FALSE
  )

Fst.all.pop.M <-
  strataG::popStructTest(
```

Table 20: Pairwise Fst between females amd males for autosomal markers

	Fst	Fst.p.val	Fst.1	Fst.p.val.1
NATL (32) v. IO_PAC (91)	0.0015	0.0099	0.0016	0.0099
NATL (32) v. MED (26)	0.0008	0.0099	0.0004	0.1287
IO_PAC (91) v. MED (26)	0.0018	0.0099	0.0020	0.0099

Table 21: Pairwise Fis between females amd males for autosomal markers

	Fis	Fis.p.val	Fis.1	Fis.p.val.1
NATL (32) v. IO_PAC (91)	-0.3966	0.0099	-0.3457	0.5446
NATL (32) v. MED (26)	-0.8266	1.0000	-0.8976	0.9802
IO_PAC (91) v. MED (26)	-0.4580	0.7723	-0.3870	0.8911

```
    gt.all.neutral.M,
    nrep = 100,
    stats = c("fst", "fis"),
    type = "pairwise",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores() - 2,
    write.output = FALSE
  )
save(Fst.all.pop.F, Fst.all.pop.M, file = "Rdata/Fst_by_sex.Rdata")
```

```
load("Rdata/Fst_by_sex.Rdata")

pwFst.FM <- data.frame(Fst.all.pop.F$pairwise$result[,-c(1:4, 7:8)],
                       Fst.all.pop.M$pairwise$result[,-c(1:4, 7:8)])
knitr::kable(pwFst.FM, digits = 4,
             caption = "Pairwise Fst between females amd males for autosomal markers")
```

Fst test

```
pwFis.FM <- data.frame(Fst.all.pop.F$pairwise$result[,-c(1:6)],
                       Fst.all.pop.M$pairwise$result[,c(7:8)])
knitr::kable(pwFis.FM, digits = 4,
             caption = "Pairwise Fis between females amd males for autosomal markers")
```

Fis test

Citations for packages

```
## [1] "Yihui Xie (2021). knitr: A General-Purpose Package"
## [2] "   for Dynamic Report Generation in R. R package"
## [3] "   version 1.37."
## character(0)
## [1] "Jombart, T. (2008) adegenet: a R package for the"
## [2] "   multivariate analysis of genetic markers."
## [3] "   Bioinformatics 24: 1403-1405. doi:"
## [4] "   10.1093/bioinformatics/btn129"
## [1] "Xiuwen Zheng, David Levine, Jess Shen, Stephanie M."
## [2] "   Gogarten, Cathy Laurie, Bruce S. Weir. A"
## [3] "   High-performance Computing Toolset for Relatedness"
## [4] "   and Principal Component Analysis of SNP Data."
## [5] "   Bioinformatics 2012; doi:"
## [6] "   10.1093/bioinformatics/bts606"
## [1] "Gosselin, T, Anderson, E. C., Bradbury, I. (2020)."
## [2] "   assigner: Assignment Analysis with GBS/RAD Data"
## [3] "   using R. R package version 0.5.8."
## [4] "   http://thierrygosselin.github.io/assigner/. doi :"
## [5] "   10.5281/zenodo.592677"
## [1] "Gosselin, T. (2020). radiator: RADseq Data"
## [2] "   Exploration, Manipulation and Visualization using"
## [3] "   R. R package version 1.1.9"
## [4] "   https://thierrygosselin.github.io/radiator/. doi :"
## [5] "   10.5281/zenodo.3687060"
## [1] "Scott D. Foster (2020). stockR: Identifying Stocks in"
## [2] "   Genetic Data. R package version 1.0.74."
## [3] "   https://CRAN.R-project.org/package=stockR"
## [1] "Archer, F. I., Adams, P. E. and Schneiders, B. B."
## [2] "   (2016) strataG: An R package for manipulating,"
## [3] "   summarizing and analysing population genetic data."
## [4] "   Mol Ecol Resour. doi:10.1111/1755-0998.12559"
## [1] "Paradis E. 2010. pegas: an R package for population"
## [2] "   genetics with an integrated-modular approach."
## [3] "   Bioinformatics 26: 419-420."
## [1] "Keenan, K., McGinnity, P., Cross, T.F., Crozier,"
## [2] "   W.W., & Prodöhl, P.A., (2013), diveRsity: An R"
## [3] "   package for the estimation of population genetics"
## [4] "   parameters and their associated errors, Methods in"
## [5] "   Ecology and Evolution, doi:"
## [6] "   10.1111/2041-210X.12067"
## [1] "Hadley Wickham, Romain François, Lionel Henry and"
## [2] "   Kirill Müller (2021). dplyr: A Grammar of Data"
## [3] "   Manipulation. R package version 1.0.7."
## [4] "   https://CRAN.R-project.org/package=dplyr"
## [1] "Hadley Wickham (2011). The Split-Apply-Combine"
## [2] "   Strategy for Data Analysis. Journal of Statistical"
## [3] "   Software, 40(1), 1-29. URL"
## [4] "   http://www.jstatsoft.org/v40/i01/."
## [1] "Kamvar ZN, Tabima JF, Grünwald NJ. (2014) Poppr: an R"
## [2] "   package for genetic analysis of populations with"
## [3] "   clonal, partially clonal, and/or sexual"
## [4] "   reproduction. PeerJ 2:e281. doi: 10.7717/peerj.281"
## [1] "Jerome Goudet and Thibaut Jombart (2021). hierfstat:"
## [2] "   Estimation and Tests of Hierarchical F-Statistics."
## [3] "   R package version 0.5-10."
## [4] "   https://CRAN.R-project.org/package=hierfstat"
```

```
## [1] "Michael C. Whitlock and Katie Lotterhos (2014)."
## [2] "   OutFLANK: Fst outliers with trimming. R package"
## [3] "   version 0.2."
## [1] "John D. Storey, Andrew J. Bass, Alan Dabney and David"
## [2] "   Robinson (2021). qvalue: Q-value estimation for"
## [3] "   false discovery rate control. R package version"
## [4] "   2.26.0. http://github.com/jdstorey/qvalue"
## [1] "H. Wickham. ggplot2: Elegant Graphics for Data"
## [2] "   Analysis. Springer-Verlag New York, 2016."
## [1] "Dewey Dunnington (2021). ggspatial: Spatial Data"
## [2] "   Framework for ggplot2. R package version 1.1.5."
## [3] "   https://CRAN.R-project.org/package=ggspatial"
## [1] "Andy South (2017). rnaturalearth: World Map Data from"
## [2] "   Natural Earth. R package version 0.1.0."
## [3] "   https://CRAN.R-project.org/package=rnaturalearth"
## [1] "Venables, W. N. & Ripley, B. D. (2002) Modern Applied"
## [2] "   Statistics with S. Fourth Edition. Springer, New"
## [3] "   York. ISBN 0-387-95457-0"
## [1] "JJ Allaire and Yihui Xie and Jonathan McPherson and"
## [2] "   Javier Luraschi and Kevin Ushey and Aron Atkins"
## [3] "   and Hadley Wickham and Joe Cheng and Winston Chang"
## [4] "   and Richard Iannone (2021). rmarkdown: Dynamic"
## [5] "   Documents for R. R package version 2.11. URL"
## [6] "   https://rmarkdown.rstudio.com."
```

Session info

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Australia.1252
## [2] LC_CTYPE=English_Australia.1252
## [3] LC_MONETARY=English_Australia.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_Australia.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets
## [6] methods   base
##
## other attached packages:
##  [1] wesanderson_0.3.6   MASS_7.3-54
##  [3] diveRsity_1.9.90    poppr_2.9.3
##  [5] pegas_1.1           ape_5.6
##  [7] ggspatial_1.1.5     rnaturalearth_0.1.0
##  [9] plyr_1.8.6          pcadapt_4.3.3
## [11] OutFLANK_0.2        qvalue_2.26.0
## [13] hierfstat_0.5-10    dplyr_1.0.7
## [15] dartR_1.9.9.1       ggplot2_3.3.5
## [17] stockR_1.0.74       adegenet_2.1.5
## [19] ade4_1.7-18         strataG_2.5.01
## [21] assigner_0.5.8      radiator_1.2.0
## [23] formatR_1.11        knitr_1.37
##
## loaded via a namespace (and not attached):
##   [1] utf8_1.2.2          R.utils_2.11.0
##   [3] tidyselect_1.1.1    htmlwidgets_1.5.4
##   [5] grid_4.1.0          combinat_0.0-8
##   [7] StAMPP_1.6.3        devtools_2.4.3
##   [9] munsell_0.5.0       codetools_0.2-18
##  [11] units_0.7-2         withr_2.4.3
##  [13] gdsfmt_1.30.0       colorspace_2.0-2
##  [15] highr_0.9           rstudioapi_0.13
##  [17] stats4_4.1.0        wk_0.5.0
##  [19] robustbase_0.93-9   labeling_0.4.2
##  [21] RgoogleMaps_1.4.5.3 mnormt_2.0.2
##  [23] farver_2.1.0        rprojroot_2.0.2
##  [25] vctrs_0.3.8         generics_0.1.1
##  [27] xfun_0.29           R6_2.5.1
##  [29] doParallel_1.0.16   cachem_1.0.6
##  [31] reshape_0.8.8       assertthat_0.2.1
##  [33] promises_1.2.0.1    scales_1.1.1
##  [35] nnet_7.3-16         gtable_0.3.0
##  [37] processx_3.5.2      phangorn_2.8.1
##  [39] rlang_0.4.12        systemfonts_1.0.3
##  [41] calibrate_1.7.7     splines_4.1.0
##  [43] rgdal_1.5-28        checkmate_2.0.0
##  [45] s2_1.0.7            abind_1.4-5
##  [47] yaml_2.2.1          reshape2_1.4.4
```

```
##  [49] backports_1.4.1     httpuv_1.6.4
##  [51] Hmisc_4.6-0         tools_4.1.0
##  [53] usethis_2.1.5       lavaan_0.6-9
##  [55] psych_2.1.9         spData_2.0.1
##  [57] kableExtra_1.3.4    ellipsis_0.3.2
##  [59] raster_3.5-11       RColorBrewer_1.1-2
##  [61] proxy_0.4-26        sessioninfo_1.2.2
##  [63] Rcpp_1.0.7          base64enc_0.1-3
##  [65] classInt_0.4-3      purrr_0.3.4
##  [67] ps_1.6.0            prettyunits_1.1.1
##  [69] deldir_1.0-6        rpart_4.1-15
##  [71] pbapply_1.5-0       qgraph_1.9
##  [73] cluster_2.1.2       fs_1.5.2
##  [75] magrittr_2.0.1      data.table_1.14.2
##  [77] genetics_1.3.8.1.3  tmvnsim_1.0-2
##  [79] mvtnorm_1.1-3       pkgload_1.2.4
##  [81] mime_0.12           evaluate_0.14
##  [83] xtable_1.8-4        jpeg_0.1-9
##  [85] gridExtra_2.3       testthat_3.1.1
##  [87] compiler_4.1.0      tibble_3.1.6
##  [89] KernSmooth_2.23-20  crayon_1.4.2
##  [91] apex_1.0.4          gdistance_1.3-6
##  [93] R.oo_1.24.0         htmltools_0.5.2
##  [95] spdep_1.2-1         mgcv_1.8-38
##  [97] corpcor_1.6.10      later_1.3.0
##  [99] Formula_1.2-4       tidyr_1.1.4
## [101] DBI_1.1.2           PopGenReport_3.0.4
## [103] sf_1.0-5            boot_1.3-28
## [105] Matrix_1.4-0        permute_0.9-5
## [107] cli_3.1.0           quadprog_1.5-8
## [109] R.methodsS3_1.8.1   gdata_2.18.0
## [111] parallel_4.1.0      igraph_1.2.10
## [113] pkgconfig_2.0.3     foreign_0.8-81
## [115] sp_1.4-6            terra_1.4-22
## [117] xml2_1.3.3          foreach_1.5.1
## [119] svglite_2.0.0       pbivnorm_0.6.0
## [121] SNPRelate_1.28.0    webshot_0.5.2
## [123] rvest_1.0.2         stringr_1.4.0
## [125] callr_3.7.0         digest_0.6.29
## [127] vegan_2.5-7         polysat_1.7-6
## [129] rmarkdown_2.11      fastmatch_1.1-3
## [131] htmlTable_2.4.0     gap_1.2.3-1
## [133] shiny_1.7.1         gtools_3.9.2
## [135] glasso_1.11         lifecycle_1.0.1
## [137] nlme_3.1-153        dismo_1.3-5
## [139] seqinr_4.2-8        viridisLite_0.4.0
## [141] desc_1.4.0          fansi_0.5.0
## [143] pillar_1.6.4        lattice_0.20-44
## [145] GGally_2.1.2        httr_1.4.2
## [147] fastmap_1.1.0       DEoptimR_1.0-9
## [149] pkgbuild_1.3.1      survival_3.2-13
## [151] glue_1.6.0          remotes_2.4.2
## [153] fdrtool_1.2.17      mmod_1.3.3
## [155] png_0.1-7           iterators_1.0.13
## [157] class_7.3-19        stringi_1.7.6
## [159] latticeExtra_0.6-29 memoise_2.0.1
## [161] e1071_1.7-9
```