Supplemental Information for:

Sex-specific dispersal patterns of threatened Northern River Shark, *Glyphis garricki*

Floriaan Devloo-Delva*, Peter M. Kyne, James R. Marthick, Michael I. Grant, Rasanthi M. Gunasekera, Grant J. Johnson, David L. Morgan, Richard D. Pillans, Thor Saunders, William White, Peter M. Grewe and Pierre Feutry

* *Author produced this Rmarkdown*

Febuary 2025

Contents

1 Background

1.1 Sampling map and metadata

Feutry et al (2020) collected 468 *G. garricki* from northern Australia and Papua New Guinea (PNG). For our study we selected 375 of these samples and an additional six PNG samples for full mitogenome sequencing.

Throughout this supplement these rivers are abbreviated: King Sound (KS), West Cambridge Gulf (WGC), Ord River (OR), Daly River (DR), Adelaide River (AR), Sampan Creek (SC), Wildman River (WR), West Alligator River (WAR), South Alligator River (SAR) and East Alligator River (EAR).

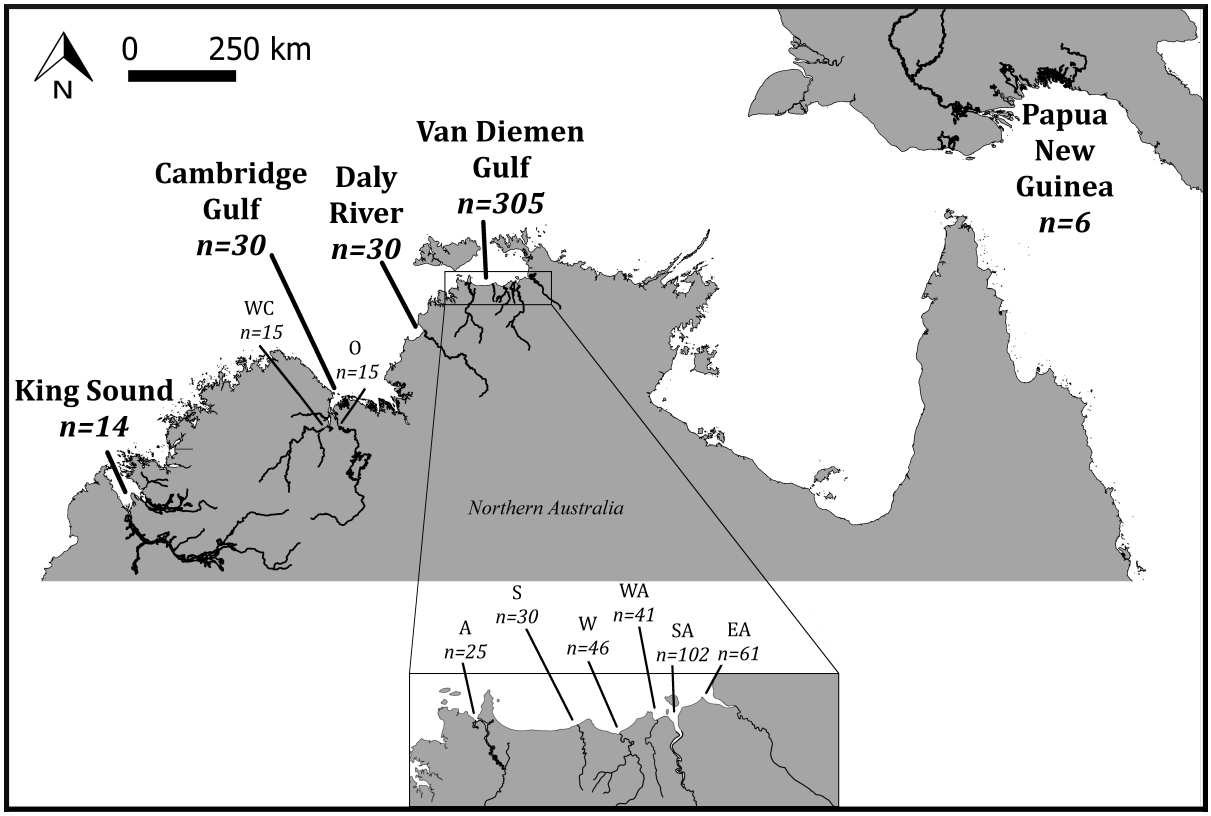


Table 1: Primer sequences used for amplifying the two halves of the mitochondrial genome

Mitogenome fragment	Primer name	Primer sequence	Amplicon length	Ref
A	BSh(1975L_for	5'-AACACAAACTCCGCCTGTTTACCAAAAACATC	9039	This study
A	BSh(10999H_rev	5'-CAACCGGCAATTGGAGCTTCAACGTGGG	9039	This study
B	BSh(9198L_for	5'-GAGCCCATCATAGCTTAATAGAAGGTAAC	9710	This study
B	16s(747-H_rev	5'-GGTGTCTAAAGCTCCATAGGGTCTTCTCGTCT	9710	This study

Table 2: Primer sequences used for amplifying the four quarters of the mitochondrial genome

Mitogenome fragment	Primer name	Primer sequence	Amplicon length (bp)	Ref
A1	BSh(1975L_for	5'-AACACAAACTCCGCCTGTTTACCAAAAACATC	4330	This study
A1	Ggar(6277_rev	5'-CCAAGTAGGCCAATTGCTATTATTGCTCAAA	4330	This study
A2	Ggar(6205_for	5'-GGTTTCGGAATAATCTCACATGTAGTTGCCT	4821	This study
A2	BSh(10999H_rev	5'-CAACCGGCAATTGGAGCTTCAACGTGGG	4821	This study
B1	BSh(9198L_for	5'-GAGCCCATCATAGCTTAATAGAAGGTAAC	5148	This study
B1	Ggar14322_Rev	5'-GAGGGCCATTAGTTCTTGTAGTTGAATAAACAACG	5148	This study
B2	Ggar14277_For	5'-ATTCCTACCTGGACTTTAACCAAGAC	4667	This study
B2	16s(747-H_rev	5'-GGTGTCTAAAGCTCCATAGGGTCTTCTCGTCT	4667	This study

1.2 Sequencing primers

Primers were designed to amplify the mitochondrial genome of *G. garricki*. The mitochondrial genome of most sharks were amplified with two primer pairs, but eight sharks had to be amplified in equal quarters with four primer pairs.

2 Load packages and functions

```
library(ade4)
library(adegenet)
library(ape)
library(apex)
library(assignPOP)
library(dartRverse)
library(haplotypes)
library(data.table)
library(ggplot2)
library(grid)
library(gridExtra)
library(gtools)
library(kableExtra)
library(knitr)
library(magrittr)
library(mvbutils)
library(OutFLANK)
library(pegas)
library(plyr)
library(qvalue)
library(strataG)
library(vegan)
library(babette)
library(coda)
library(bdskytools)
library(beastio)
library(gplots)
library(ggtree)
library(treeio)
source("Filtering funtions.R")

#set colours
colours.5 <- c("#B294C7","#52AF43", "#fc9403", "#A6CEE3", "#B15928")
names(colours.5) <- c(
  "King Sound", "Cambridge Gulf", "Daly River",
  "Van Diemen Gulf","Papua New Guinea")
colours.11 <- c("#B294C7","#4d6b40","#adcf9f", "#fc9403", "#08306B",
                        "#1764AB","#4B98C9","#96C5DF", "#c5d6e6","#e1eaf2","#B15928")
names(colours.11) <- c(
  "King Sound",
  "West Cambridge Gulf",
  "Ord River",
  "Daly River",
  "Adelaide River",
  "Sampan Creek",
  "Wildman River",
  "West Alligator River",
  "South Alligator River",
  "East Alligator River",
  "Papua New Guinea"
)
colours.6 <- colours.11[5:10]

colours.6.hive <- colours.11[5:10]
names(colours.6.hive) <- c("AR", "SC", "WR", "WAR","SAR", "EAR")
```

```r
shortnames <- c("KS", "WCG", "OR", "DR", "AR", "SC", "WR", "WAR", "SAR", "EAR", "PNG")
```

---

3 Preparing the mitogenome data files

3.1 MultiDNA format

```
f.in <- "../3.Data/6.ggar_Nucleotide_alignment_edited_HTPsites.fasta"
metafile <- "../3.Data/Glyphis garricki_Metadata2_ALL_10-05-2019.csv"
meta <- read.csv(metafile)
ggar.ml <- apex::read.multiFASTA(f.in)
ggar.bin <- ape::read.dna(f.in, format = "fasta", as.matrix = TRUE)

genes <- list(`tRNA1` = ggar.bin[,1:69],
              `12SrRNA` = ggar.bin[,70:1022],
              `tRNA2` = ggar.bin[,1023:1094],
              `16SrRNA` = ggar.bin[,1094:2761],
              `tRNA3` = ggar.bin[,2762:2836],
              `ND1` = ggar.bin[,2837:3811],
              `tRNA4` = ggar.bin[,3812:4023],
              `ND2` = ggar.bin[,4024:5068],
              `tRNA5` = ggar.bin[,5069:5456],
              `COI` = ggar.bin[,5457:7014],
              `tRNA6` = ggar.bin[,7015:7158],
              `CO2` = ggar.bin[,7159:7856],
              `tRNA7` = ggar.bin[,7857:7930],
              `ATP8` = ggar.bin[,7931:8099],
              `ATP6` = ggar.bin[,8100:8773],
              `CO3` = ggar.bin[,8774:9558],
              `tRNA8` = ggar.bin[,9559:9630],
              `ND3` = ggar.bin[,9631:9979],
              `tRNA9` = ggar.bin[,9980:10049],
              `ND4` = ggar.bin[,10050:11720],
              `tRNA10` = ggar.bin[,11721:11928],
              `ND5` = ggar.bin[,11929:13758],
              `ND6` = ggar.bin[,13759:14275],
              `tRNA11` = ggar.bin[,14276:14345],
              `CytB` = ggar.bin[,14346:15492],
              `tRNA12` = ggar.bin[,15493:15635],
              `CR` = ggar.bin[,15636:16704]
              )
ggar.ml <- new("multidna", genes)
meta <- meta[meta$id %in% ggar.ml@labels,]
meta <- meta[order(match(meta$id, ggar.ml@labels)),]
ggar.ml@ind.info <- meta

gene.names <- apex::getLocusNames(ggar.ml)
gene.info <- as.data.frame(t(sapply(ggar.ml@dna, dim)))
ggar.ml@gene.info <- data.frame(gene.names = rownames(gene.info),
                                gene.length = gene.info$V2)
```

3.2 Genind format

```
ggar.gi <- apex::multidna2genind(ggar.ml, mlst = TRUE)
```

3.2.1 Add metadata

```
meta <- read.csv(metafile)
meta <- meta[meta$id %in% indNames(ggar.gi),]
meta <- meta[order(match(meta$id, indNames(ggar.gi))),]
```

```
pop.levels1 <- c("King Sound", "West Cambridge Gulf", "Ord River",
                 "Daly River", "Adelaide River", "Sampan Creek", "Wildman River",
                 "West Alligator River", "South Alligator River",
                 "East Alligator River", "Papua New Guinea")
meta$River <- factor(x = meta$River, levels = pop.levels1)

pop.levels2 <- c("King Sound", "Cambridge Gulf", "Daly River",
                 "Van Diemen Gulf", "Papua New Guinea")
meta$pop <- factor(x = meta$pop, levels = pop.levels2)

ggar.gi$other$ind.meta <- meta
my_strata <- data.frame(rivers = meta$River ,
                        regions = meta$pop)
strata(ggar.gi) <- my_strata
ggar.gi$pop <- as.factor(meta$River)

ggar.gi <- ggar.gi[order(ggar.gi$other$ind.meta$pop, ggar.gi$other$ind.meta$River,
                         indNames(ggar.gi)),]
```

3.3 Gtypes format

```
river <- ggar.gi$other$ind.meta$River[order(match(indNames(ggar.gi), ggar.ml@labels))]
pop <- ggar.gi$other$ind.meta$pop[order(match(indNames(ggar.gi), ggar.ml@labels))]

strata.schemes <- data.frame(pop, river)
row.names(strata.schemes) <- ggar.ml@labels

ggar.gt <- strataG::sequence2gtypes(ggar.ml, strata = river, seq.names = NULL,
                                    schemes = strata.schemes,
                                    description = NULL, other = NULL)
```

3.4 Haplotype format

3.4.1 Haplotypes

26 HTs

```
ggar.dna <- haplotypes::read.fas(f.in)
ggar.hap <- haplotypes::haplotype(ggar.dna,indels = "sic")

river <- ggar.gi$other$ind.meta$River[order(match(indNames(ggar.gi), ggar.dna@seqnames))]
pop <- ggar.gi$other$ind.meta$pop[order(match(indNames(ggar.gi), ggar.dna@seqnames))]

ggar.hapgroup <- haplotypes::grouping(ggar.hap,river)
```

3.4.2 Pegas

26 HTs

```
ggar.bin <- ape::read.dna(f.in, format = "fasta", as.matrix = TRUE)

HTs <- paste0("HT", sprintf('%0.2d', 1:26))
ggar.h <- pegas::haplotype(ggar.bin, labels = HTs)
```

3.5 Save data

```
save(ggar.ml,ggar.bin, ggar.dna, ggar.gi, ggar.gt, ggar.hap, ggar.h, ggar.SNP.gi,
     ggar.SNP.gl,ggar.snp.gt, file = "6.ggar.Rdata")
```

3.6 Load data

```
print(load("6.ggar.Rdata")) #
```

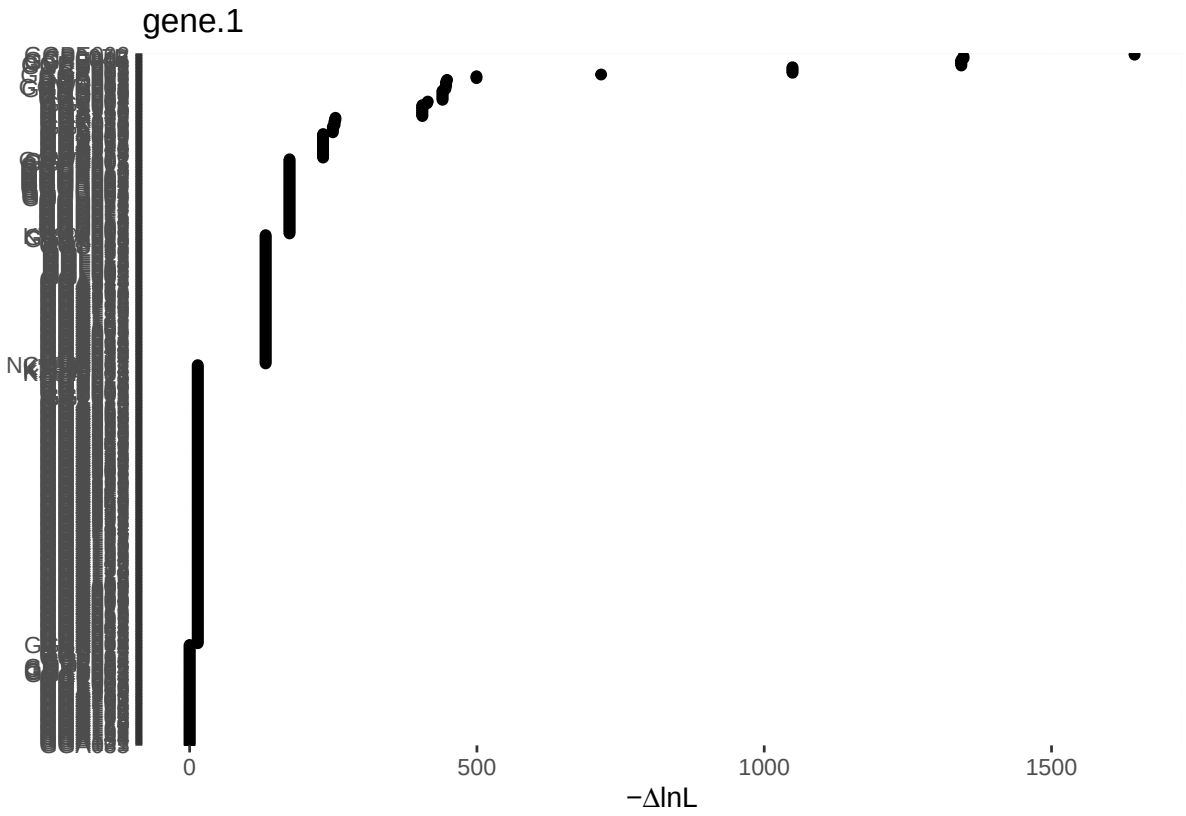
```
## [1] "ggar.ml" "ggar.bin" "ggar.dna" "ggar.gi" "ggar.gt"
## [6] "ggar.hap" "ggar.h" "ggar.SNP.gi" "ggar.SNP.gl"
"ggar.snp.gt"
```

---

4 Genetic diversity

4.1 Overall diversity

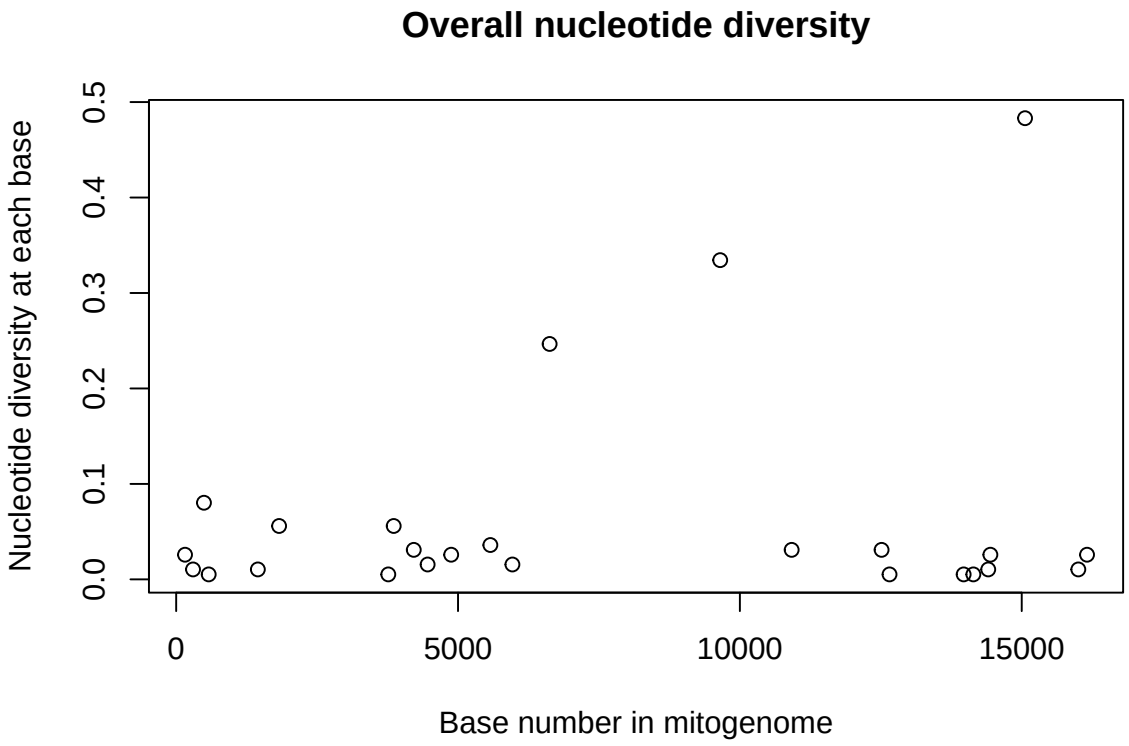
```
seqLL <- strataG::sequenceLikelihoods(ggar.bin, model = "N",
                                      pairwise.deletion = FALSE, n = NULL,
                                      plot = TRUE, simplify = TRUE)
```




```
bases <- haplotypes::basecomp(ggar.dna)
{print(paste0("Mitogenome base composition:"))
print(colMeans(bases))}
```

```
## [1] "Mitogenome base composition:"
##         A         T         G         C
## 0.3152367 0.2625469 0.1293226 0.2928937
```

```
{pi.gt <- strataG::nucleotideDiversity(ggar.gt, bases = c("a", "c", "g", "t"),
                                       simplify = TRUE)
plot(names(pi.gt[pi.gt != 0]),as.numeric(pi.gt[pi.gt != 0]), xlab = "Base number in mitogenome",
     ylab = "Nucleotide diversity at each base", main = "Overall nucleotide diversity")}
```




```
print(paste0("Nucleotide diversity according to strataG = ",mean(pi.gt)))
```

```
## [1] "Nucleotide diversity according to strataG =
9.50237058915988e-05"
```

```
{print(paste0("Nucleotide diversity and variance according to pegas:"))
pegas::nuc.div(ggar.bin, variance = TRUE, pairwise.deletion = FALSE)}
```

```
## [1] "Nucleotide diversity and variance according to
pegas:"
```

```
## [1] 9.503508e-05 3.923969e-09
```

```
{print(paste0("Haplotype diversity and variance according to pegas:"))
pegas::hap.div(ggar.bin, variance = TRUE, method = "Nei")}
```

```
## [1] "Haplotype diversity and variance according to
pegas:"
```

```
## [1] 0.7803097617 0.0002403597
```

```
varsites <- strataG::variableSites(ggar.gt, bases = c("a", "c", "g", "t", "-"),
                                   simplify = TRUE)
knitr::kable(
  t(varsites$site.freqs),
  format = "latex",
  caption = "Variable sites in mitogenome for all individuals"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE,
latex_options = "scale_down")
```

Table 3: Variable sites in mitogenome for all individuals

	a	c	g	t	-
157	0	5	0	378	0
299	0	2	0	381	0
493	16	0	367	0	0
577	0	382	0	1	0
1449	2	0	381	0	0
1826	11	0	372	0	0
3763	382	0	1	0	0
3859	372	0	11	0	0
4215	0	6	0	377	0
4461	0	380	0	3	0
4879	5	0	378	0	0
5401	0	0	0	4	379
5574	0	376	0	7	0
5965	0	3	0	380	0
6625	0	328	0	55	0
9651	0	302	0	81	0
10919	0	6	0	377	0
12514	0	6	0	377	0
12655	0	382	0	1	0
13970	0	1	0	382	0
14142	0	382	0	1	0
14406	2	0	381	0	0
14444	378	0	5	0	0
15059	0	155	0	228	0
16004	381	0	2	0	0
16159	0	378	0	5	0
16561	24	0	0	0	359

4.2 Diversity per river

```
strata <- as.character(ggar.gt@schemes$river)
names(strata) <- as.character(ggar.gt@schemes$id)
strataG::setStrata(ggar.gt) <- strata

divers.river <- data.frame(River = character(), pi.strataG = numeric(),
                           pi.pegas = numeric(), pi.pegas.var = numeric(),
                           HT.richness.strataG = numeric(), h.pegas = numeric(),
                           h.pegas.var = numeric())
for (river in levels(ggar.gt@schemes$river)) {
  print(river)
  ggar.gt.sub <- ggar.gt[,,river, drop = TRUE]
  ggar.bin.sub <-  ggar.bin[dimnames(ggar.bin)[[1]] %in% ggar.gt.sub@data$id, , drop = TRUE]
  ggar.h.sub <- strataG::labelHaplotypes(ggar.bin.sub, prefix = "HT",
                                         use.indels = TRUE)
  ggar.dna.sub <- as.dna(ggar.dna[names(ggar.dna) %in% ggar.gt.sub@data$id, ,
                                  drop = TRUE])
  bases <- haplotypes::basecomp(ggar.dna.sub)
  print(paste0("Mitogenome base composition:"))
  print(colMeans(bases))

  pi.gt <-
    strataG::nucleotideDiversity(ggar.gt.sub,bases = c("a", "c", "g", "t"),
    simplify = TRUE)
  pi.peg <- pegas::nuc.div(ggar.bin.sub, variance = TRUE,
                           pairwise.deletion = FALSE)
  h.peg <- pegas::hap.div(ggar.bin.sub, variance = TRUE, method = "Nei")
  res <- data.frame(river, mean(pi.gt, na.rm = TRUE), pi.peg[1], pi.peg[2],
                    length(unique(ggar.h.sub$haps)),
                    h.peg[1], h.peg[2])
  colnames(res) <- c("River", "pi.strataG", "pi.pegas", "pi.pegas.var",
                     "HT.richness.strataG", "h.pegas", "h.pegas.var")
  divers.river <- rbind(divers.river, res)

  varsites <- strataG::variableSites(ggar.gt.sub,
                                     bases = c("a", "c", "g", "t", "-"),
                                     simplify = TRUE)
  print(paste0("Number of variable sites for ",river))
  print(varsites$site.freqs)
}
dA.gt.river <- strataG::nucleotideDivergence(ggar.gt,
                                       probs = c(0, 0.025, 0.5, 0.975, 1),
                                       model = "RAW")
save(divers.river, dA.gt.river, file = "Genetic_diversity.Rdata")
```

```
load("Genetic_diversity.Rdata")
strata <- as.character(ggar.gt@schemes$river)
names(strata) <- as.character(ggar.gt@schemes$id)
strataG::setStrata(ggar.gt) <- strata

knitr::kable(
  divers.river,
  caption = "Nucleotide and haplotype diversity per river",
  format = "latex",
  digits = 6
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

Table 4: Nucleotide and haplotype diversity per river

	River	pi.strataG	pi.pegas	pi.pegas.var	HT.richness.strataG	h.pegas	h.pegas.var
1	King Sound	0.000022	0.000022	0	2	0.362637	0.016935
11	West Cambridge Gulf	0.000194	0.000194	0	4	0.666667	0.009284
12	Ord River	0.000016	0.000016	0	2	0.133333	0.012982
13	Daly River	0.000036	0.000036	0	6	0.735632	0.002768
14	Adelaide River	0.000044	0.000044	0	6	0.696667	0.006729
15	Sampan Creek	0.000082	0.000082	0	7	0.777011	0.002681
16	Wildman River	0.000058	0.000058	0	6	0.492929	0.006055
17	West Alligator River	0.000017	0.000017	0	2	0.139024	0.004893
18	South Alligator River	0.000060	0.000060	0	5	0.593465	0.000816
19	East Alligator River	0.000064	0.000064	0	7	0.736612	0.001252
110	Papua New Guinea	0.000040	0.000040	0	3	0.600000	0.045370

Table 5: Nucleotide divergence within each river

	stratum	mean	q.0	q.0.025	q.0.5	q.0.975	q.1
10	West Cambridge Gulf	0.000194	0	0	6e-05	0.000419	0.000479
7	Sampan Creek	0.000082	0	0	6e-05	0.000180	0.000239
3	East Alligator River	0.000064	0	0	6e-05	0.000180	0.000180
8	South Alligator River	0.000060	0	0	6e-05	0.000120	0.000120
11	Wildman River	0.000058	0	0	0e+00	0.000180	0.000239
1	Adelaide River	0.000044	0	0	6e-05	0.000120	0.000180
6	Papua New Guinea	0.000040	0	0	6e-05	0.000099	0.000120
2	Daly River	0.000036	0	0	6e-05	0.000120	0.000120
4	King Sound	0.000022	0	0	0e+00	0.000060	0.000060
9	West Alligator River	0.000017	0	0	0e+00	0.000120	0.000120
5	Ord River	0.000016	0	0	0e+00	0.000120	0.000120

```
knitr::kable(
  dA.gt.river$within[order(dA.gt.river$within$mean, decreasing = TRUE), -1],
  format = "latex",
  caption = "Nucleotide divergence within each river",
  digits = 6
  ) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")
```

```
kableExtra::kbl(dA.gt.river$between[order(dA.gt.river$between$mean, decreasing = TRUE), -1],
                    digits = 6,
                    caption = "Nucleotide divergence between each river",
                    longtable = TRUE) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

Table 6: Nucleotide divergence between each river

	strata.1	strata.2	dA	mean	q.0	q.0.025	q.0.5	q.0.975	q.1
14	Daly River	Papua New Guinea	0.000302	0.000339	0.000299	0.000299	0.000359	0.000419	0.000419
44	Papua New Guinea	West Cambridge Gulf	0.000214	0.000331	0.000299	0.000299	0.000299	0.000419	0.000419
35	Ord River	Papua New Guinea	0.000299	0.000327	0.000299	0.000299	0.000299	0.000419	0.000479
43	Papua New Guinea	West Alligator River	0.000291	0.000319	0.000299	0.000299	0.000299	0.000359	0.000359
45	Papua New Guinea	Wildman River	0.000270	0.000319	0.000239	0.000239	0.000299	0.000359	0.000419
42	Papua New Guinea	South Alligator River	0.000265	0.000315	0.000239	0.000239	0.000299	0.000359	0.000359
41	Papua New Guinea	Sampan Creek	0.000252	0.000313	0.000239	0.000239	0.000299	0.000419	0.000419
5	Adelaide River	Papua New Guinea	0.000270	0.000312	0.000239	0.000239	0.000299	0.000359	0.000419
22	East Alligator River	Papua New Guinea	0.000247	0.000299	0.000239	0.000239	0.000299	0.000359	0.000419
29	King Sound	Papua New Guinea	0.000241	0.000272	0.000239	0.000239	0.000239	0.000359	0.000359
53	West Alligator River	West Cambridge Gulf	0.000106	0.000212	0.000120	0.000120	0.000120	0.000419	0.000419
55	West Cambridge Gulf	Wildman River	0.000086	0.000212	0.000060	0.000120	0.000120	0.000419	0.000479
51	South Alligator River	West Cambridge Gulf	0.000081	0.000207	0.000060	0.000060	0.000120	0.000419	0.000419
48	Sampan Creek	West Cambridge Gulf	0.000068	0.000206	0.000060	0.000060	0.000180	0.000419	0.000479
9	Adelaide River	West Cambridge Gulf	0.000085	0.000204	0.000060	0.000060	0.000120	0.000419	0.000479
26	East Alligator River	West Cambridge Gulf	0.000062	0.000191	0.000060	0.000060	0.000120	0.000419	0.000479
33	King Sound	West Cambridge Gulf	0.000057	0.000165	0.000060	0.000060	0.000120	0.000359	0.000419
18	Daly River	West Cambridge Gulf	0.000037	0.000152	0.000000	0.000000	0.000060	0.000419	0.000479
17	Daly River	West Alligator River	0.000114	0.000140	0.000120	0.000120	0.000120	0.000180	0.000180
19	Daly River	Wildman River	0.000093	0.000140	0.000060	0.000060	0.000120	0.000180	0.000239
39	Ord River	West Cambridge Gulf	0.000034	0.000139	0.000000	0.000000	0.000060	0.000419	0.000539
16	Daly River	South Alligator River	0.000088	0.000136	0.000060	0.000060	0.000120	0.000180	0.000180
15	Daly River	Sampan Creek	0.000075	0.000134	0.000060	0.000060	0.000120	0.000239	0.000239
1	Adelaide River	Daly River	0.000093	0.000132	0.000060	0.000060	0.000120	0.000180	0.000239
38	Ord River	West Alligator River	0.000111	0.000128	0.000120	0.000120	0.000120	0.000239	0.000239
40	Ord River	Wildman River	0.000091	0.000128	0.000060	0.000060	0.000120	0.000239	0.000299
37	Ord River	South Alligator River	0.000086	0.000124	0.000060	0.000060	0.000120	0.000239	0.000239
36	Ord River	Sampan Creek	0.000073	0.000122	0.000060	0.000060	0.000120	0.000239	0.000299
4	Adelaide River	Ord River	0.000090	0.000120	0.000060	0.000060	0.000120	0.000239	0.000299
11	Daly River	East Alligator River	0.000069	0.000119	0.000060	0.000060	0.000120	0.000180	0.000239
21	East Alligator River	Ord River	0.000067	0.000107	0.000060	0.000060	0.000120	0.000239	0.000299
10	Adelaide River	Wildman River	0.000051	0.000102	0.000000	0.000000	0.000120	0.000180	0.000239
8	Adelaide River	West Alligator River	0.000067	0.000097	0.000000	0.000000	0.000120	0.000180	0.000180

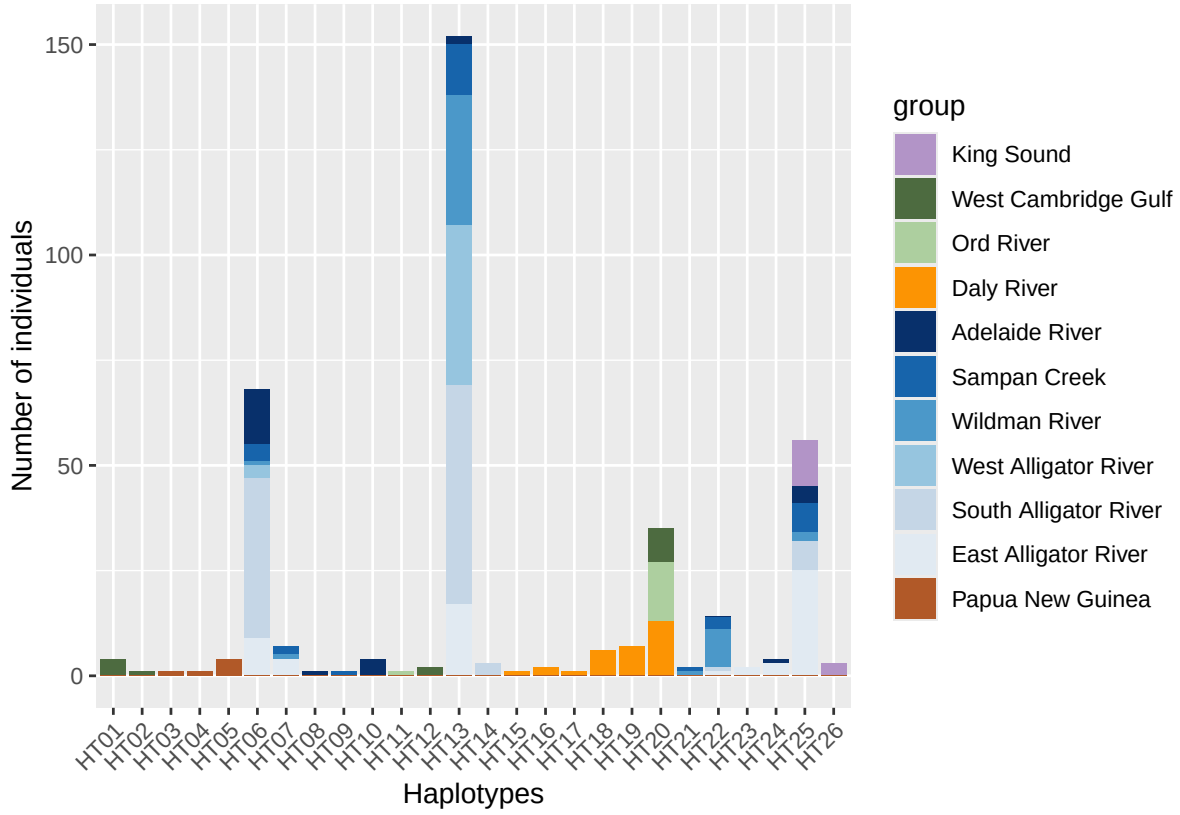
12	Daly River	King Sound	0.000064	0.000093	0.000060	0.000060	0.000060	0.000180	0.000180
6	Adelaide River	Sampan Creek	0.000020	0.000083	0.000000	0.000000	0.000060	0.000180	0.000239
28	King Sound	Ord River	0.000062	0.000081	0.000060	0.000060	0.000060	0.000180	0.000239
49	Sampan Creek	Wildman River	0.000006	0.000076	0.000000	0.000000	0.000060	0.000180	0.000239
27	East Alligator River	Wildman River	0.000013	0.000074	0.000000	0.000000	0.000060	0.000180	0.000239
23	East Alligator River	Sampan Creek	0.000000	0.000073	0.000000	0.000000	0.000060	0.000180	0.000239
46	Sampan Creek	South Alligator River	0.000002	0.000073	0.000000	0.000000	0.000060	0.000180	0.000180
32	King Sound	West Alligator River	0.000054	0.000073	0.000060	0.000060	0.000060	0.000120	0.000120
34	King Sound	Wildman River	0.000033	0.000073	0.000000	0.000000	0.000060	0.000120	0.000180
7	Adelaide River	South Alligator River	0.000019	0.000071	0.000000	0.000000	0.000060	0.000120	0.000180
2	Adelaide River	East Alligator River	0.000017	0.000071	0.000000	0.000000	0.000060	0.000120	0.000180
31	King Sound	South Alligator River	0.000028	0.000069	0.000000	0.000000	0.000060	0.000120	0.000120
52	South Alligator River	Wildman River	0.000010	0.000068	0.000000	0.000000	0.000060	0.000180	0.000180
47	Sampan Creek	West Alligator River	0.000018	0.000067	0.000000	0.000000	0.000060	0.000180	0.000180
24	East Alligator River	South Alligator River	0.000005	0.000067	0.000000	0.000000	0.000060	0.000180	0.000180
30	King Sound	Sampan Creek	0.000015	0.000067	0.000000	0.000000	0.000060	0.000180	0.000180
25	East Alligator River	West Alligator River	0.000026	0.000066	0.000000	0.000000	0.000060	0.000180	0.000180
3	Adelaide River	King Sound	0.000033	0.000066	0.000000	0.000000	0.000060	0.000120	0.000180
20	East Alligator River	King Sound	0.000009	0.000052	0.000000	0.000000	0.000060	0.000120	0.000180
50	South Alligator River	West Alligator River	0.000014	0.000052	0.000000	0.000000	0.000000	0.000120	0.000120
54	West Alligator River	Wildman River	0.000006	0.000043	0.000000	0.000000	0.000000	0.000180	0.000180
13	Daly River	Ord River	0.000002	0.000028	0.000000	0.000000	0.000000	0.000120	0.000180

```
hapcount <- pegas::haploFreq(ggar.dna@sequence, fac = ggar.gt@schemes$river,
                             split = NULL, what = 1, haplo = NULL)
row.names(hapcount) <- paste0("HT", sprintf('%0.2d', 1:ggar.hap@nhap))
knitr::kable(
  hapcount,
  caption = "Haplotype counts per river according to pegas",
  format = "latex",
  col.names = shortnames
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
group.levels <- levels(ggar.gi$other$ind.meta$River)
group.names <- ggar.gi$other$ind.meta$River

HTs <- paste0("HT", sprintf('%0.2d', 1:ggar.hap@nhap))
df <- data.frame(group = NULL,nind = NULL, hap = NULL, nhap = NULL,
                 hapfreq = NULL, stringsAsFactors = FALSE)
for (k in 1:length(group.levels)) {
  group.inds <- adegenet::indNames(ggar.gi)[group.names == group.levels[k]]
  for (i in 1:ggar.hap@nhap) {
    group <- group.levels[k]
    nind <- length(group.inds)
    hap <- HTs[i]
    nhap <- sum(group.inds %in% ggar.hap@haplist[[i]])
    hapfreq <- nhap/nind
    df <- plyr::rbind.fill(df, data.frame(group = group, nind = nind, hap = hap,
                                          nhap = nhap, hapfreq = hapfreq,
                                          stringsAsFactors = FALSE))
  }
}

df$group <- factor(df$group, levels = group.levels)
ggplot2::ggplot(data = df) +
  ggplot2::geom_bar(ggplot2::aes(x = hap, y = nhap, fill = group),
                    stat = "identity",position = "stack") +
  ggplot2::labs(x = "Haplotypes", y = "Number of individuals") +
  ggplot2::scale_fill_manual(values = colours.11) +
  ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 45, hjust = 1))
```

Table 7: Haplotype counts per river according to pegas

	KS	WCG	OR	DR	AR	SC	WR	WAR	SAR	EAR	PNG
HT01	0	4	0	0	0	0	0	0	0	0	0
HT02	0	1	0	0	0	0	0	0	0	0	0
HT03	0	0	0	0	0	0	0	0	0	0	1
HT04	0	0	0	0	0	0	0	0	0	0	1
HT05	0	0	0	0	0	0	0	0	0	0	4
HT06	0	0	0	0	13	4	1	3	38	9	0
HT07	0	0	0	0	0	2	1	0	0	4	0
HT08	0	0	0	0	1	0	0	0	0	0	0
HT09	0	0	0	0	0	1	0	0	0	0	0
HT10	0	0	0	0	4	0	0	0	0	0	0
HT11	0	0	1	0	0	0	0	0	0	0	0
HT12	0	2	0	0	0	0	0	0	0	0	0
HT13	0	0	0	0	2	12	31	38	52	17	0
HT14	0	0	0	0	0	0	0	0	3	0	0
HT15	0	0	0	1	0	0	0	0	0	0	0
HT16	0	0	0	2	0	0	0	0	0	0	0
HT17	0	0	0	1	0	0	0	0	0	0	0
HT18	0	0	0	6	0	0	0	0	0	0	0
HT19	0	0	0	7	0	0	0	0	0	0	0
HT20	0	8	14	13	0	0	0	0	0	0	0
HT21	0	0	0	0	0	1	1	0	0	0	0
HT22	0	0	0	0	0	3	9	0	1	1	0
HT23	0	0	0	0	0	0	0	0	0	2	0
HT24	0	0	0	0	1	0	0	0	0	3	0
HT25	11	0	0	0	4	7	2	0	7	25	0
HT26	3	0	0	0	0	0	0	0	0	0	0



4.3 Diversity per region

```
strata <- as.character(ggar.gt@schemes$pop)
names(strata) <- as.character(ggar.gt@schemes$id)
strataG::setStrata(ggar.gt) <- strata

divers.region <- data.frame(Region = character(), pi.strataG = numeric(),
                            pi.pegas = numeric(), pi.pegas.var = numeric(),
                            HT.richness.strataG = numeric(), h.pegas = numeric(),
                            h.pegas.var = numeric())

for (pop in levels(ggar.gt@schemes$pop)) {
  print(pop)
  ggar.gt.sub <- ggar.gt[,,pop, drop = TRUE]
  ggar.bin.sub <-  ggar.bin[dimnames(ggar.bin)[[1]] %in% ggar.gt.sub@data$id, , drop = TRUE]
  ggar.h.sub <- strataG::labelHaplotypes(ggar.bin.sub, prefix = "HT",
                                         use.indels = TRUE)
  ggar.dna.sub <- as.dna(ggar.dna[names(ggar.dna) %in% ggar.gt.sub@data$id, ,
                                  drop = TRUE])

  bases <- haplotypes::basecomp(ggar.dna.sub)
  print(paste0("Mitogenome base composition:"))
  print(colMeans(bases))

  pi.gt <- strataG::nucleotideDiversity(ggar.gt.sub,
                                        bases = c("a", "c", "g", "t"),
                                        simplify = TRUE)
  pi.peg <- pegas::nuc.div(ggar.bin.sub, variance = TRUE,
                           pairwise.deletion = FALSE)
  h.peg <- pegas::hap.div(ggar.bin.sub, variance = TRUE, method = "Nei")
```

Table 8: Nucleotide and haplotype diversity per region

	Region	pi.strataG	pi.pegas	pi.pegas.var	HT.richness.strataG	h.pegas	h.pegas.var
1	King Sound	NaN	0.000022	0	2	0.362637	0.016935
11	Cambridge Gulf	NaN	0.000122	0	5	0.452874	0.011085
12	Daly River	NaN	0.000036	0	6	0.735632	0.002768
13	Van Diemen Gulf	6.6e-05	0.000066	0	12	0.674928	0.000449
14	Papua New Guinea	NaN	0.000040	0	3	0.600000	0.045370

Table 9: Nucleotide divergence within each region

	stratum	mean	q.0	q.0.025	q.0.5	q.0.975	q.1
1	Cambridge Gulf	0.0001224	0	0	0.00e+00	0.0004191	0.0005389
5	Van Diemen Gulf	0.0000662	0	0	5.99e-05	0.0001796	0.0002395
4	Papua New Guinea	0.0000399	0	0	5.99e-05	0.0000988	0.0001197
2	Daly River	0.0000355	0	0	5.99e-05	0.0001197	0.0001197
3	King Sound	0.0000217	0	0	0.00e+00	0.0000599	0.0000599

```
  res <- data.frame(pop, mean(pi.gt), pi.peg[1], pi.peg[2],
                    length(unique(ggar.h.sub$haps)), h.peg[1], h.peg[2])
  colnames(res) <- c("Region", "pi.strataG", "pi.pegas", "pi.pegas.var",
                     "HT.richness.strataG", "h.pegas", "h.pegas.var")
  divers.region <- rbind(divers.region, res)

  varsites <- strataG::variableSites(ggar.gt.sub,
                                     bases = c("a", "c", "g", "t", "-"),
                                     simplify = TRUE)
  print(paste0("Number of variable sites for ",pop))
  print(varsites$site.freqs)
}

dA.gt.pop <- strataG::nucleotideDivergence(ggar.gt,
                                       probs = c(0, 0.025, 0.5, 0.975, 1),
                                       model = "RAW")

# save(divers.river, divers.region,dA.gt.river,dA.gt.pop, file = "Genetic_diversity.Rdata")
```

```
load("Genetic_diversity.Rdata")
strata <- as.character(ggar.gt@schemes$pop)
names(strata) <- as.character(ggar.gt@schemes$id)
strataG::setStrata(ggar.gt) <- strata

knitr::kable(
  divers.region,
  caption = "Nucleotide and haplotype diversity per region",
  format = "latex",
  digits = 6
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
knitr::kable(
  dA.gt.pop$within[order(dA.gt.pop$within$mean, decreasing = TRUE), -1],
  format = "latex",
  caption = "Nucleotide divergence within each region"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
knitr::kable(
  dA.gt.pop$between[order(dA.gt.pop$between$mean, decreasing = TRUE), -1],
  format = "latex",
  caption = "Nucleotide divergence between each region"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

Table 10: Nucleotide divergence between each region

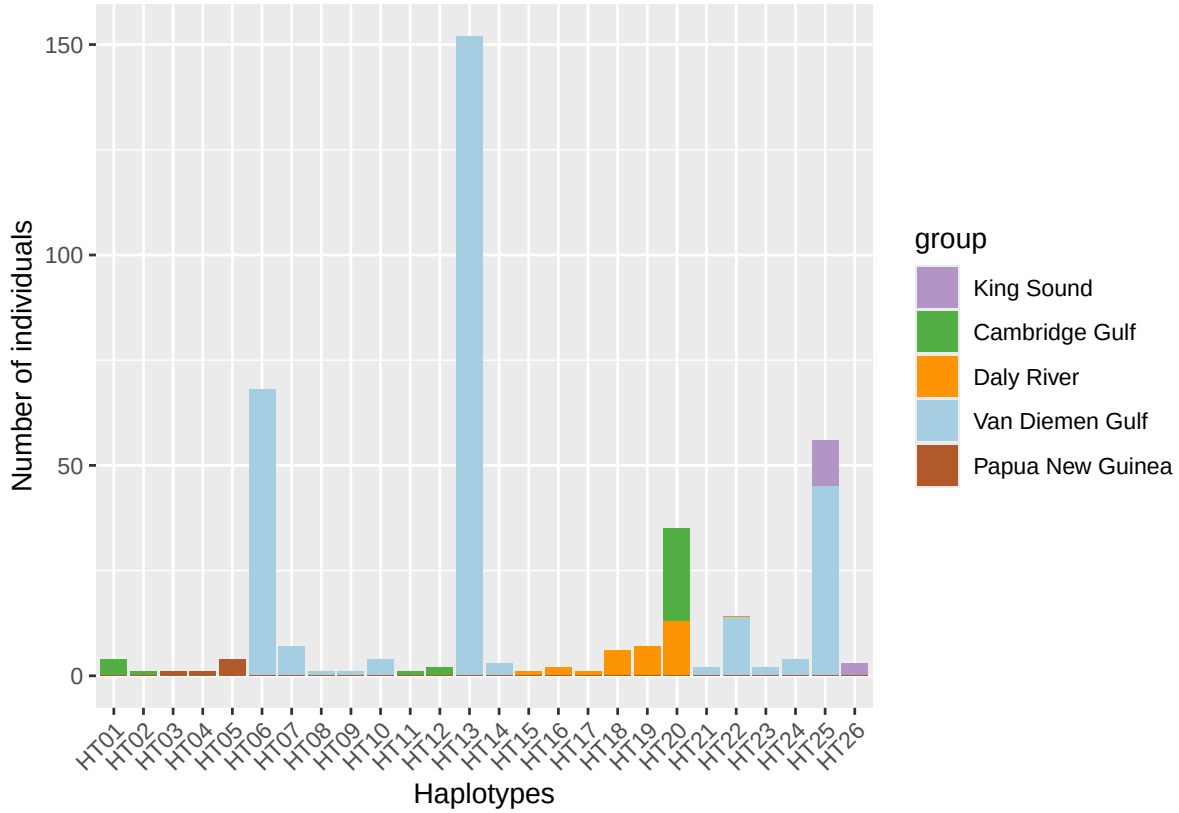
	strata.1	strata.2	dA	mean	q.0	q.0.025	q.0.5	q.0.975	q.1
6	Daly River	Papua New Guinea	0.0003016	0.0003393	0.0002994	0.0002994	0.0003592	0.0004191	0.0004191
3	Cambridge Gulf	Papua New Guinea	0.0002482	0.0003293	0.0002994	0.0002994	0.0002994	0.0004191	0.0004790
10	Papua New Guinea	Van Diemen Gulf	0.0002595	0.0003126	0.0002395	0.0002395	0.0002994	0.0003592	0.0004191
8	King Sound	Papua New Guinea	0.0002415	0.0002723	0.0002395	0.0002395	0.0002395	0.0003592	0.0003592
4	Cambridge Gulf	Van Diemen Gulf	0.0000686	0.0001629	0.0000599	0.0000599	0.0001197	0.0004191	0.0004790
7	Daly River	Van Diemen Gulf	0.0000821	0.0001330	0.0000599	0.0000599	0.0001197	0.0001796	0.0002395
2	Cambridge Gulf	King Sound	0.0000506	0.0001226	0.0000599	0.0000599	0.0000599	0.0003592	0.0004191
5	Daly River	King Sound	0.0000640	0.0000927	0.0000599	0.0000599	0.0000599	0.0001796	0.0001796
1	Cambridge Gulf	Daly River	0.0000109	0.0000898	0.0000000	0.0000000	0.0000599	0.0004191	0.0004790
9	King Sound	Van Diemen Gulf	0.0000220	0.0000660	0.0000000	0.0000000	0.0000599	0.0001197	0.0001796

```
hapcount <- pegas::haploFreq(ggar.dna@sequence, fac = ggar.gt@schemes$pop,
                             split = NULL, what = 1, haplo = NULL)
row.names(hapcount) <- paste0("HT", sprintf('%0.2d', 1:ggar.hap@nhap))
knitr::kable(hapcount,
             format = "latex",
             caption = "Haplotype counts per region according to pegas")  %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
group.levels <- levels(ggar.gi$other$ind.meta$pop)
group.names <- ggar.gi$other$ind.meta$pop

HTs <- paste0("HT", sprintf('%0.2d', 1:ggar.hap@nhap))
df <- data.frame(group = NULL,nind = NULL, hap = NULL, nhap = NULL,
                 hapfreq = NULL, stringsAsFactors = FALSE)
for (k in 1:length(group.levels)) {
  group.inds <- adegenet::indNames(ggar.gi)[group.names == group.levels[k]]
  for (i in 1:ggar.hap@nhap) {
    group <- group.levels[k]
    nind <- length(group.inds)
    hap <- HTs[i]
    nhap <- sum(group.inds %in% ggar.hap@haplist[[i]])
    hapfreq <- nhap/nind
    df <- plyr::rbind.fill(df, data.frame(group = group, nind = nind, hap = hap,
                                          nhap = nhap, hapfreq = hapfreq,
                                          stringsAsFactors = FALSE))
  }
}
df$group <- factor(df$group, levels = group.levels)
ggplot2::ggplot(data = df) +
  ggplot2::geom_bar(ggplot2::aes(x = hap, y = nhap, fill = group),
                    stat = "identity", position = "stack") +
  ggplot2::labs(x = "Haplotypes", y = "Number of individuals") +
  ggplot2::scale_fill_manual(values = colours.5) +
  ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 45, hjust = 1))
```

Table 11: Haplotype counts per region according to pegas

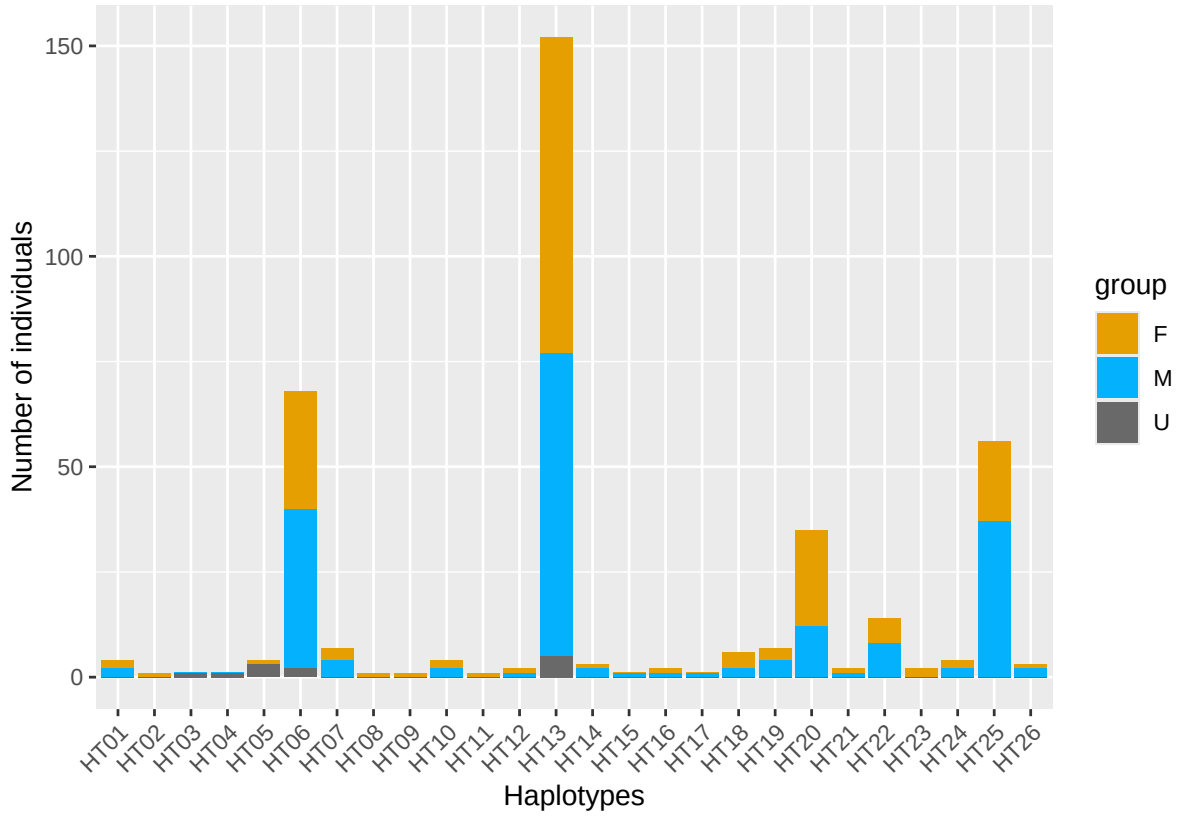
	King Sound	Cambridge Gulf	Daly River	Van Diemen Gulf	Papua New Guinea
HT01	0	4	0	0	0
HT02	0	1	0	0	0
HT03	0	0	0	0	1
HT04	0	0	0	0	1
HT05	0	0	0	0	4
HT06	0	0	0	68	0
HT07	0	0	0	7	0
HT08	0	0	0	1	0
HT09	0	0	0	1	0
HT10	0	0	0	4	0
HT11	0	1	0	0	0
HT12	0	2	0	0	0
HT13	0	0	0	152	0
HT14	0	0	0	3	0
HT15	0	0	1	0	0
HT16	0	0	2	0	0
HT17	0	0	1	0	0
HT18	0	0	6	0	0
HT19	0	0	7	0	0
HT20	0	22	13	0	0
HT21	0	0	0	2	0
HT22	0	0	0	14	0
HT23	0	0	0	2	0
HT24	0	0	0	4	0
HT25	11	0	0	45	0
HT26	3	0	0	0	0



4.4 Haplotypes by sex

```
sex <- ggar.gi$other$ind.meta$sex
sex[is.na(sex)] <- "U"
sex[sex == ""] <- "U"
sex <- factor(sex)
group.levels <- levels(sex)
group.names <- sex

HTs <- paste0("HT", sprintf('%0.2d', 1:ggar.hap@nhap))
df <- data.frame(group = NULL,nind = NULL, hap = NULL, nhap = NULL,
                 hapfreq = NULL, stringsAsFactors = FALSE)
for (k in 1:length(group.levels)) {
  group.inds <- adegenet::indNames(ggar.gi)[group.names == group.levels[k]]
  for (i in 1:ggar.hap@nhap) {
    group <- group.levels[k]
    nind <- length(group.inds)
    hap <- HTs[i]
    nhap <- sum(group.inds %in% ggar.hap@haplist[[i]])
    hapfreq <- nhap/nind
    df <- plyr::rbind.fill(df, data.frame(group = group, nind = nind, hap = hap,
                                          nhap = nhap, hapfreq = hapfreq,
                                          stringsAsFactors = FALSE))
  }
}

df$group <- factor(df$group, levels = group.levels)
ggplot2::ggplot(data = df) +
  ggplot2::geom_bar(ggplot2::aes(x = hap, y = nhap, fill = group),
                    stat = "identity",position = "stack") +
```

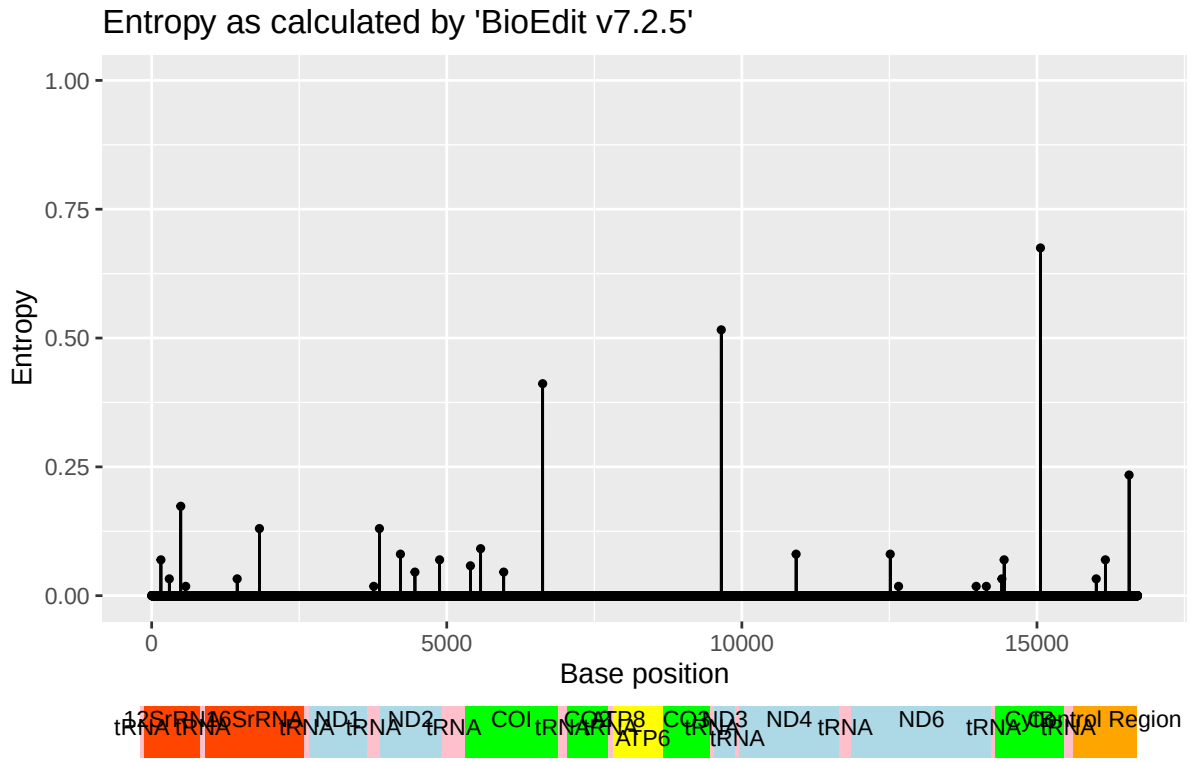
```
  ggplot2::labs(x = "Haplotypes", y = "Number of individuals") +
  ggplot2::scale_fill_manual(values = c("#E69F00","#03b1fc", "#696969")) +
  ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 45, hjust = 1))
```



4.5 Entropy plot

```
## [1] "Average entropy as calculated by 'BioEdit v7.2.5' =
0.000194762018799018"
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


Entropy as calculated by 'BioEdit v7.2.5'
Entropy
1.00
0.75
0.50
0.25
0.00
0
5000
10000
15000
Base position
tRNA
12SrRNA
tRNA
16SrRNA
tRNA
ND1
tRNA
ND2
tRNA
COI
tRNA
COII
tRNA
ATP8
ATP6
COIII
tRNA
ND3
tRNA
ND4
tRNA
ND6
tRNA
Cytb
tRNA
Control Region

---

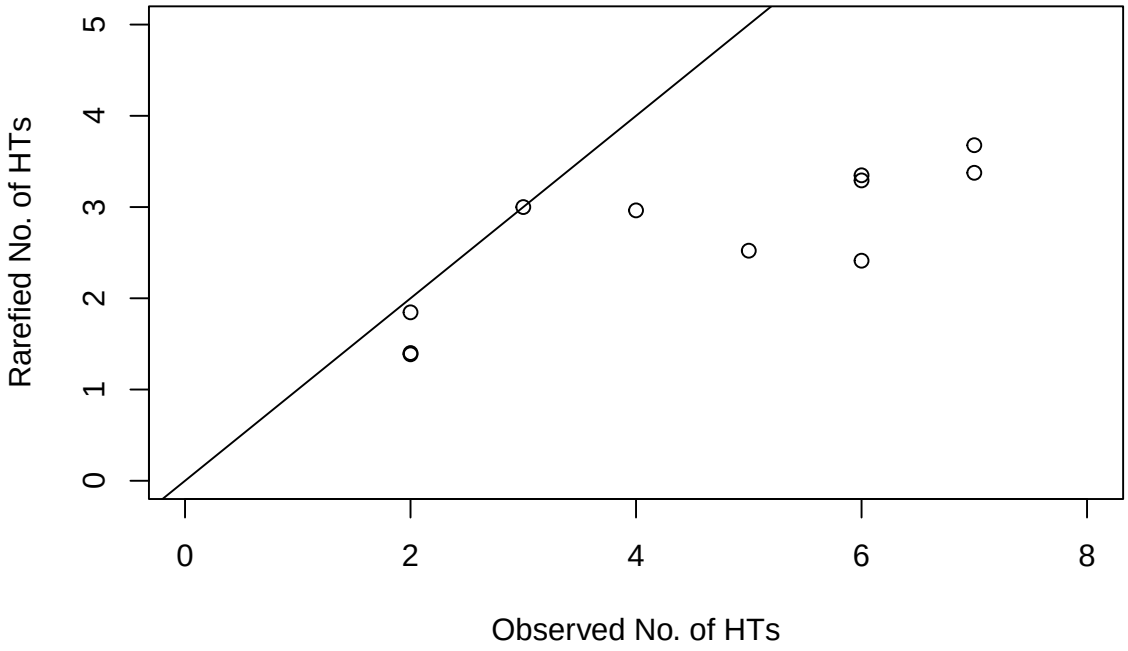
5 Haplotype network

5.1 Rarefaction

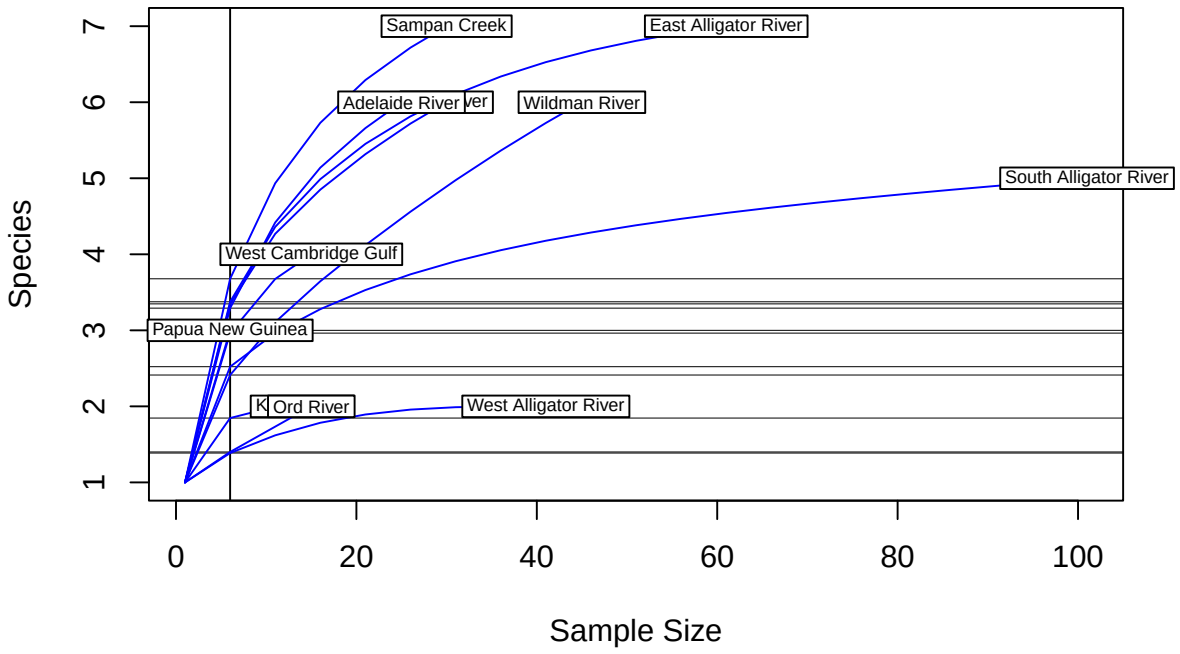
```
hapcount <- pegas::haploFreq(ggar.dna@sequence, fac = ggar.gt@schemes$river,
                             split = NULL, what = 1, haplo = NULL)
BCI <- t(hapcount)

S <- vegan::specnumber(BCI) ## observed number of HTs
(raremax <- min(rowSums(BCI)))
```

```
## [1] 6
```

```
Srare <- vegan::rarefy(BCI, raremax, se = FALSE)

plot(S, Srare, xlab = "Observed No. of HTs", ylab = "Rarefied No. of HTs",
     xlim = c(0,8),ylim = c(0,5))
abline(0, 1)
```



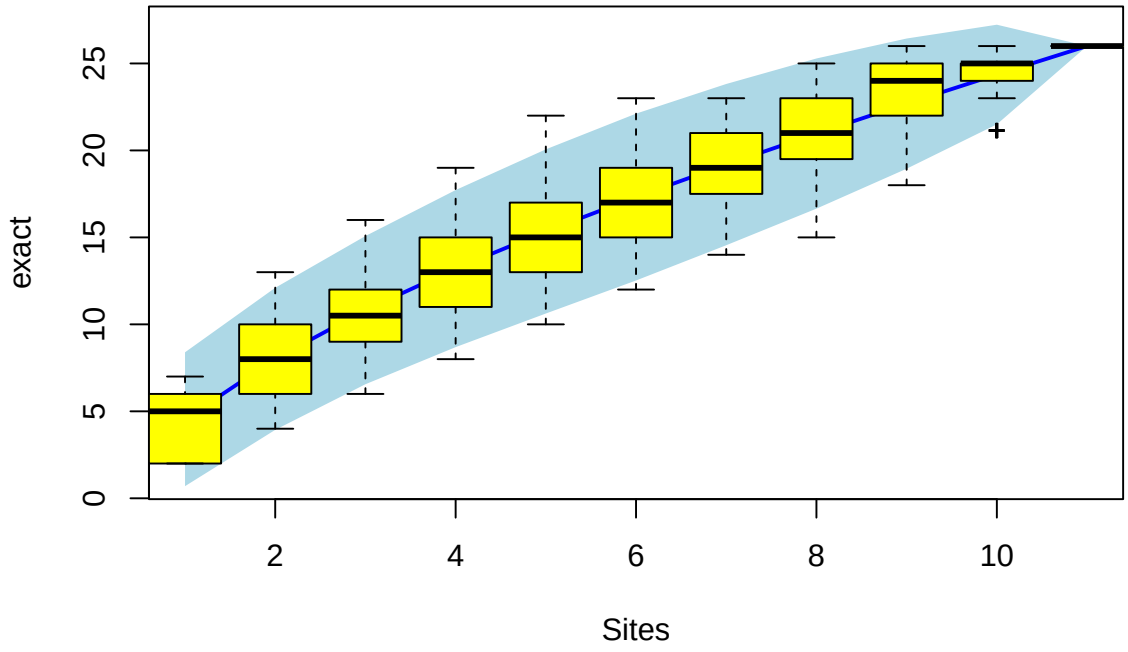

```
vegan::rarecurve(BCI, step = 5, sample = raremax, col = "blue", cex = 0.6)
```




```
## HT accumulation curve based on observed HTs
sp1 <- vegan::specaccum(BCI)
sp2 <- vegan::specaccum(BCI, "random")
plot(sp1, ci.type = "poly", col = "blue", lwd = 2, ci.lty = 0, ci.col = "lightblue")
boxplot(sp2, col = "yellow", add = TRUE, pch = "+")
```

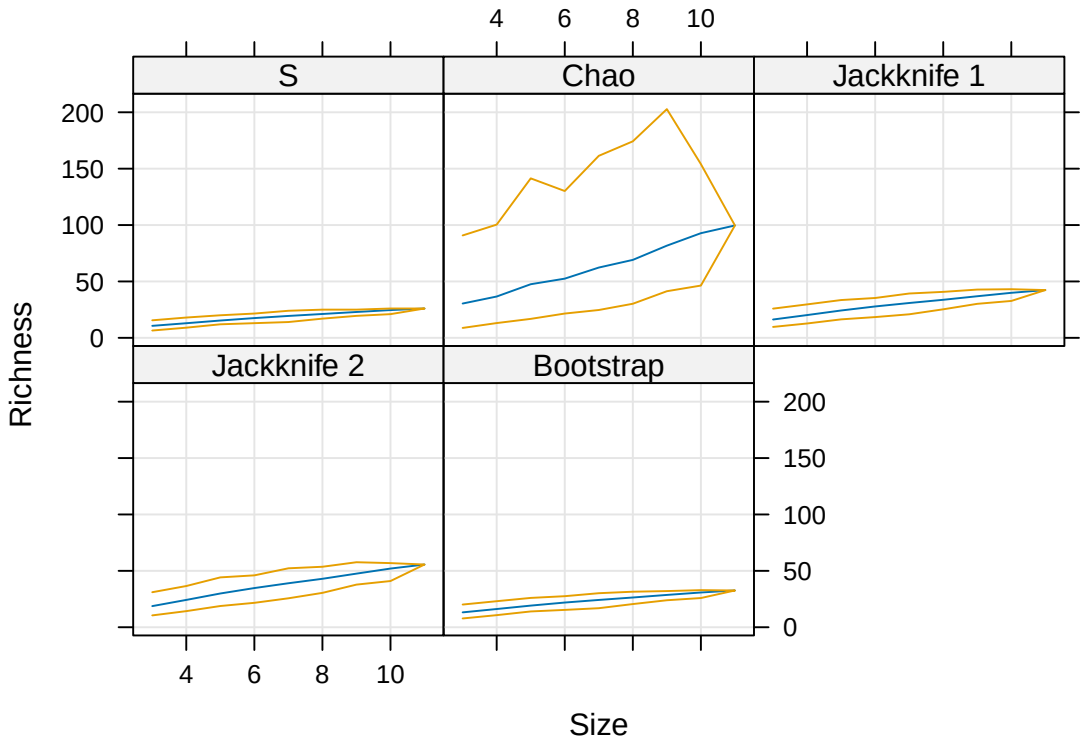
Table 12: Observed and extrapolated HT richness based on resampling

N	S	Chao	Jackknife 1	Jackknife 2	Bootstrap
3	10.63	30.39906	16.24333	18.72833	13.19630
4	12.95	36.63221	20.17250	24.22750	16.14305
5	15.35	47.52533	24.24600	29.92800	19.18039
6	17.52	52.47785	27.81167	34.73300	21.90044
7	19.40	62.30143	30.86857	38.99714	24.21349
8	21.13	69.10115	33.72125	42.96446	26.35998
9	22.91	81.71519	36.88333	47.53792	28.64473
10	24.60	92.78850	39.91800	52.04067	30.81977
11	26.00	99.63636	42.36364	55.61818	32.59653




```
## Extrapolated HT richness based on unobserved HTs
pool <- vegan::poolaccum(BCI)
knitr::kable(pool$means,
             format = "latex",
             caption = "Observed and extrapolated HT richness based on resampling") %>%
  kableExtra::kable_styling(full_width = FALSE)
```

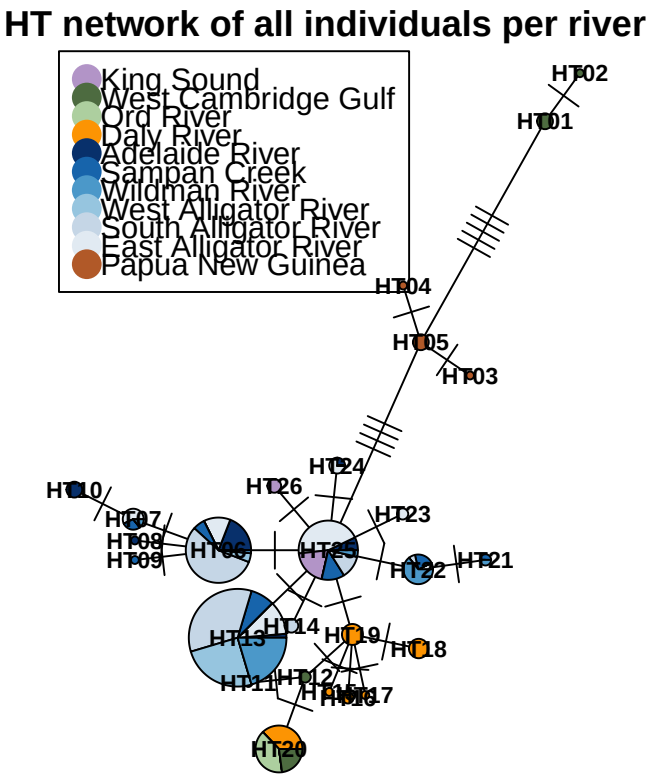
```
plot(pool)
```



5.2 Network

5.2.1 Overall

```
strata <- as.character(ggar.gt@schemes$river)
names(strata) <- as.character(ggar.gt@schemes$id)
strataG::setStrata(ggar.gt) <- strata

net <- pegas::haploNet(ggar.h)

ind.hap <- with(
    stack(setNames(attr(ggar.h, "index"), rownames(ggar.h))),
    table(hap = ind, pop = as.character(strata)[values]))

size.new <- attr(net, "freq")
size.new <- sqrt(size.new)
colours.11.new <- colours.11[order(names(colours.11))]

par(mar = c(1,1,1,1))
plot(net, size = size.new, scale.ratio = 6, pie = ind.hap,
     show.mutation = 1,threshold = 0,  srt = 1, cex = 0.75, bg = colours.11.new)
legend(-34, 63, names(colours.11), col = colours.11, pch = 19, ncol = 1,
       x.intersp = 0.5, y.intersp = 0.5, pt.cex = 2)
title(paste0("HT network of all individuals per river"))
```

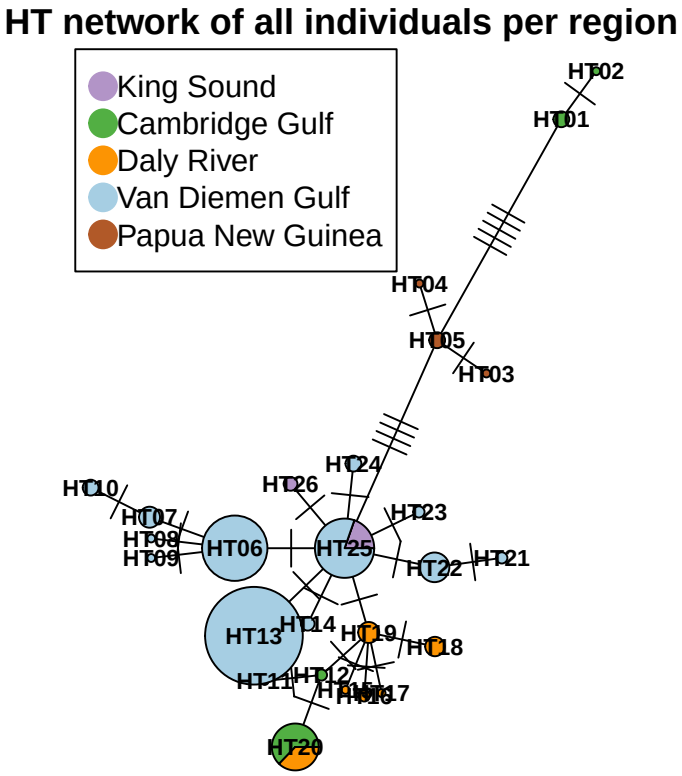


5.2.1.1 Per river

```
dev.print(
    device = png,
    file = "Haplotype_network_per_river.png",
    res = 300,
    width = 15,
    height = 15,
    units = "cm")
```

```
## pdf
##   2
```

```
strata <- as.character(ggar.gt@schemes$pop)
names(strata) <- as.character(ggar.gt@schemes$id)
# dimnames(ggar.bin)[[1]] == names(strata)

ind.hap <- with(
    stack(setNames(attr(ggar.h, "index"), rownames(ggar.h))),
    table(hap = ind, pop = as.character(strata)[values]))

size.new <- attr(net, "freq")
size.new <- sqrt(size.new)
colours.5.new <- colours.5[order(names(colours.5))]


par(mar = c(1,1,1,1))
plot(net, size = size.new, scale.ratio = 6, pie = ind.hap,
     show.mutation = 1,threshold = 0,  srt = 1, cex = 0.75, bg = colours.5.new)
legend(-34, 63, names(colours.5), col = colours.5, pch = 19, ncol = 1,
       x.intersp = 0.5, y.intersp = 1, pt.cex = 2)
title(paste0("HT network of all individuals per region"))
```



5.2.1.2 Per region

```
dev.print(
    device = png,
    file = "Haplotype_network_per_region.png",
    res = 300,
    width = 15,
    height = 15,
    units = "cm")
```

```
## pdf
##   2
```

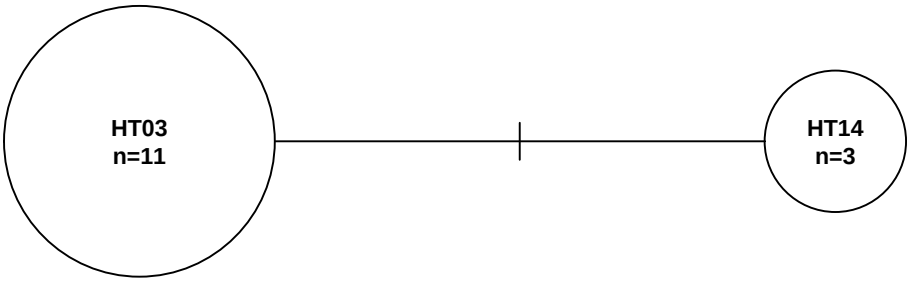
```
# xy <- replot()
# save(xy, file = "HT_network_replot.Rdata")
# load("HT_network_replot.Rdata")
# plot(net, xy = xy, size = size.new, scale.ratio = 0.2, pie = ind.hap, labels = FALSE,
    # show.mutation = 1,threshold = 0,  srt = 1, cex = 0.75, bg = colours.5.new)
# legend(-34, 63, names(colours.5), col = colours.5, pch = 19, ncol = 1,
#        x.intersp = 0.5, y.intersp = 1, pt.cex = 2)

# dev.print(
#     device = png,
#     file = "Haplotype_network_per_region3.png",
#     res = 600,
#     width = 15,
#     height = 30,
#     units = "cm")
```

5.2.2 HT network per river

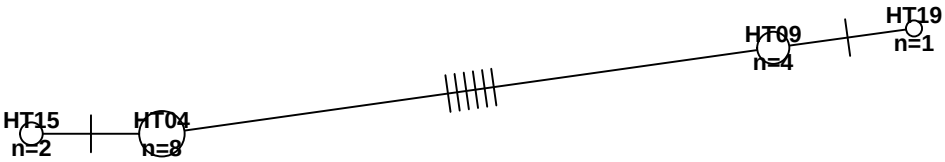
```
strata <- as.character(ggar.gt@schemes$river)
names(strata) <- as.character(ggar.gt@schemes$id)
strataG::setStrata(ggar.gt) <- strata
```

```r
ggar.haps <- strataG::labelHaplotypes(ggar.bin, prefix = "HT", use.indels = TRUE)
haps <- ggar.haps$haps

for (river in levels(ggar.gt@schemes$river)) {
  ggar.gt.sub <- ggar.gt[,,river, drop = TRUE]
  ind.names <- names(ggar.dna) %in% ggar.gt.sub@data$id
  ggar.dna.sub <- as.dna(ggar.dna[ind.names, , drop = TRUE])
  ggar.bin.sub <- ggar.bin[dimnames(ggar.bin)[[1]] %in%
                             ggar.gt.sub@data$id, , drop = TRUE]

  HTs <- unique(haps[names(haps) %in% dimnames(ggar.bin.sub)[[1]]])
  ggar.h.sub <- pegas::haplotype(ggar.bin.sub, HTs)
  ggar.h.sub <- sort(ggar.h.sub, what = "label")

  strata.sub <- as.character(ggar.gt.sub@data$stratum)
  names(strata.sub) <- as.character(ggar.gt.sub@data$id)

  net <- pegas::haploNet(ggar.h.sub)
  ind.hap <- with(
    stack(setNames(attr(ggar.h.sub, "index"), rownames(ggar.h.sub))),
    table(hap = ind, pop = strata.sub[values]))

  size.new <- attr(net, "freq")
  size.new <- sqrt(size.new)

  attr(net, "labels") <- paste0(attr(net, "labels"),"\nn=", attr(net, "freq"))

  plot(net, size = size.new, scale.ratio = 6, pie = NULL,
     show.mutation = 1,threshold = 0,  srt = 1, cex = 0.75)
  title(paste0(river))
}
```

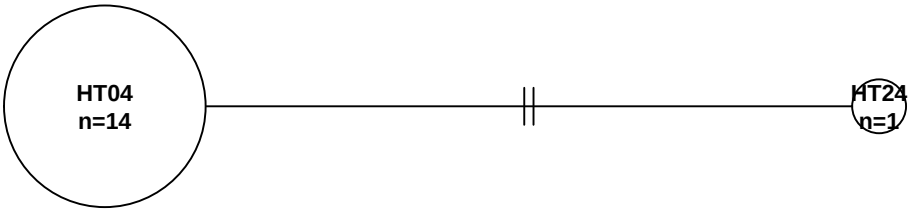
King Sound



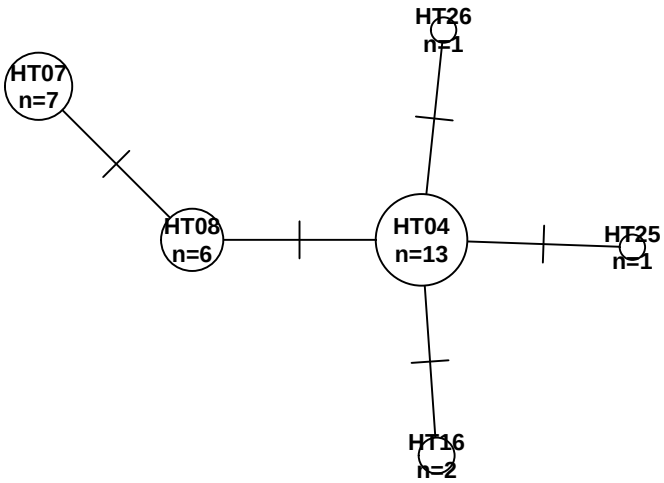
West Cambridge Gulf



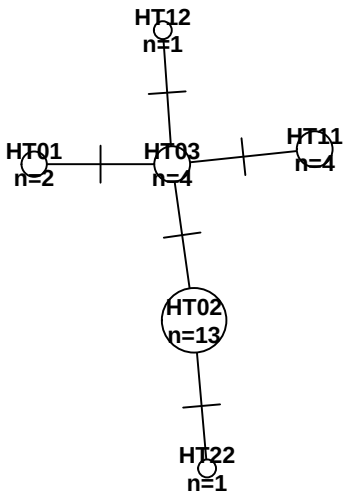
Ord River



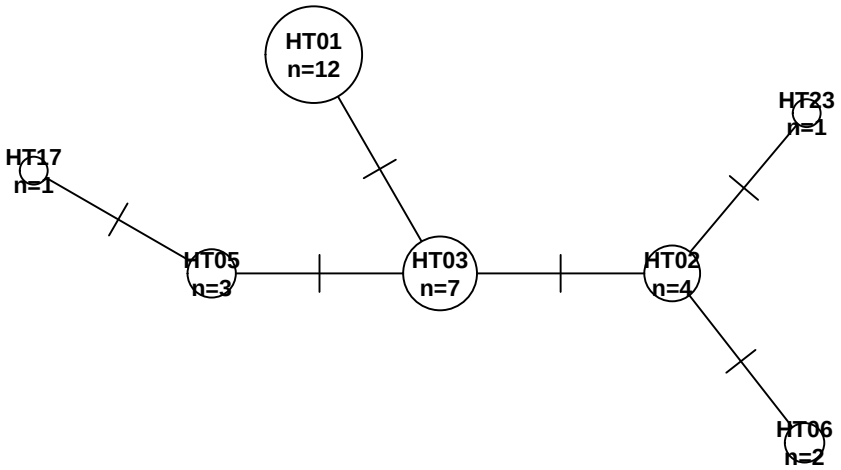
Daly River



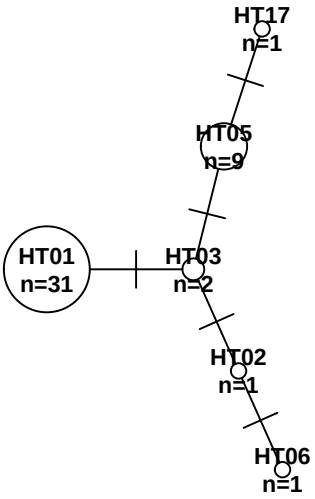
Adelaide River



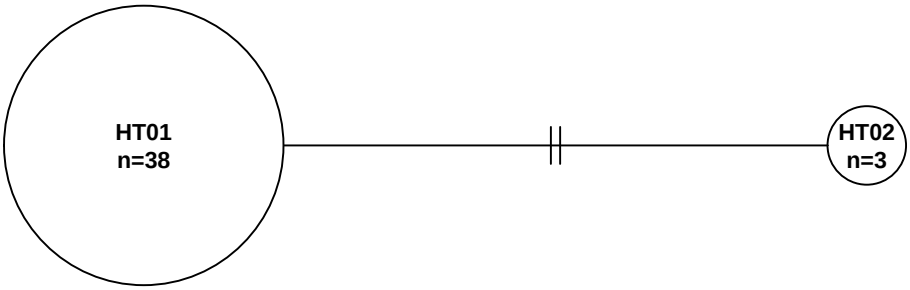
Sampan Creek



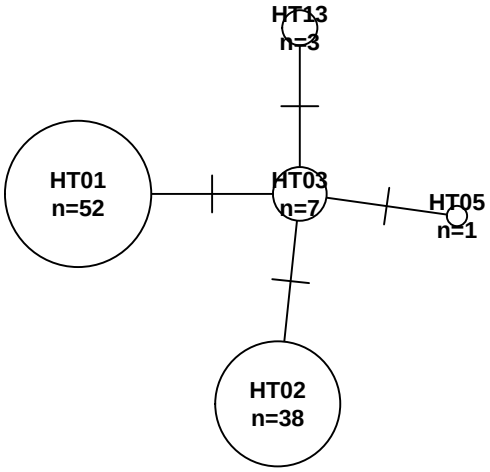
Wildman River



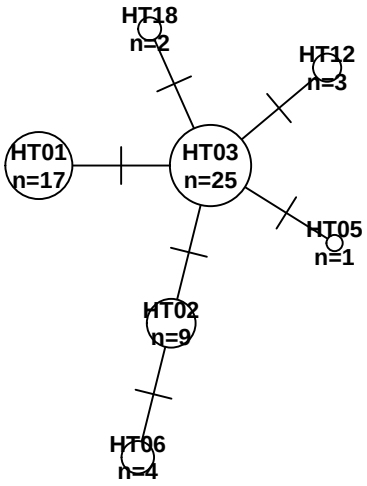
West Alligator River

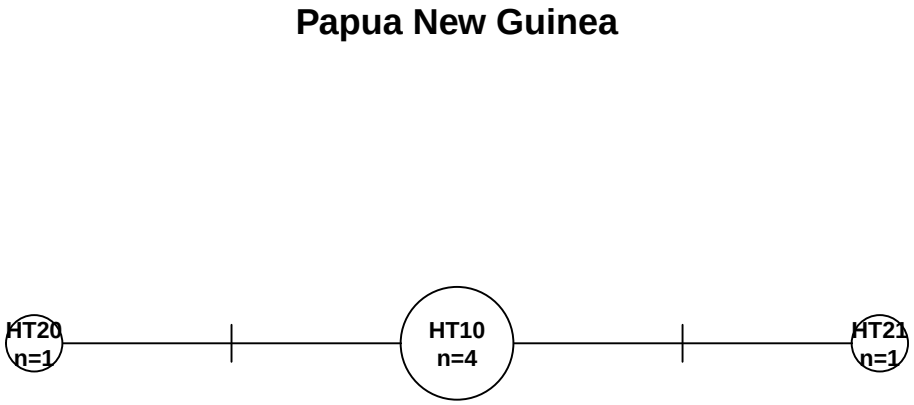


South Alligator River



East Alligator River

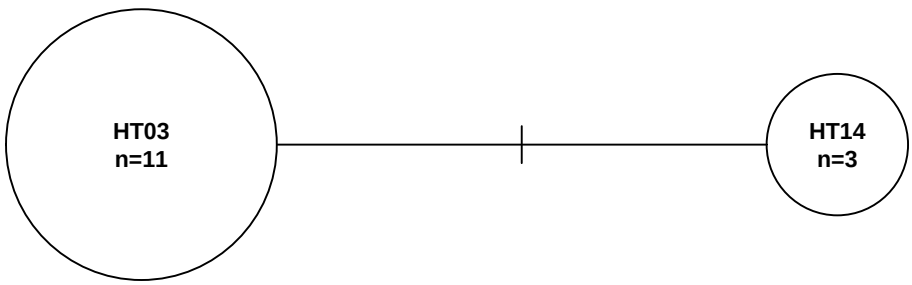




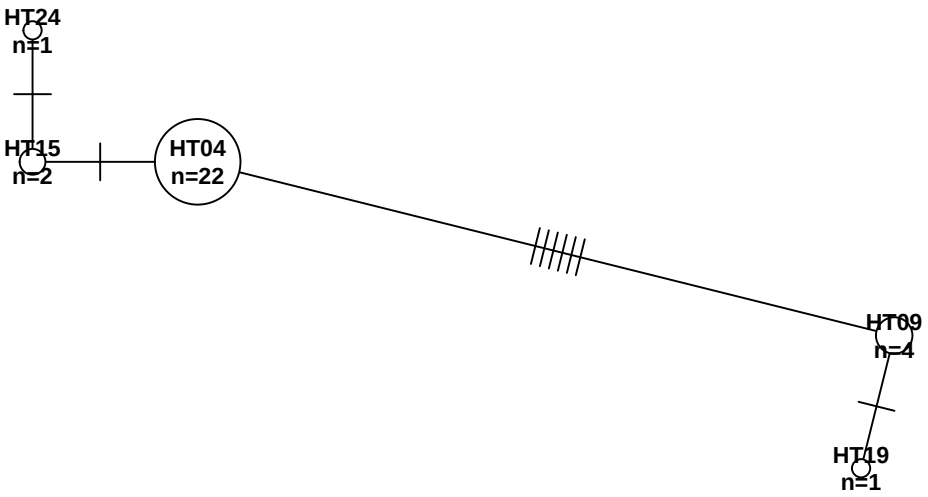
5.2.3 HT network per region

```
strata <- as.character(ggar.gt@schemes$pop)
names(strata) <- as.character(ggar.gt@schemes$id)
strataG::setStrata(ggar.gt) <- strata

ggar.haps <- strataG::labelHaplotypes(ggar.bin, prefix = "HT", use.indels = TRUE)
haps <- ggar.haps$haps

for (pop in levels(ggar.gt@schemes$pop)) {
  ggar.gt.sub <- ggar.gt[,,pop, drop = TRUE]
  ind.names <- names(ggar.dna) %in% ggar.gt.sub@data$id
  ggar.dna.sub <- as.dna(ggar.dna[ind.names, , drop = TRUE])
  ggar.bin.sub <- ggar.bin[dimnames(ggar.bin)[[1]] %in%
                             ggar.gt.sub@data$id, , drop = TRUE]

  HTs <- unique(haps[names(haps) %in% dimnames(ggar.bin.sub)[[1]]])
  ggar.h.sub <- pegas::haplotype(ggar.bin.sub, HTs)
  ggar.h.sub <- sort(ggar.h.sub, what = "label")

  strata.sub <- as.character(ggar.gt.sub@data$stratum)
  names(strata.sub) <- as.character(ggar.gt.sub@data$id)

  net <- pegas::haploNet(ggar.h.sub)
  ind.hap <- with(
    stack(setNames(attr(ggar.h.sub, "index"), rownames(ggar.h.sub))),
    table(hap = ind, pop = strata.sub[values]))

  size.new <- attr(net, "freq")
  size.new <- sqrt(size.new)
```

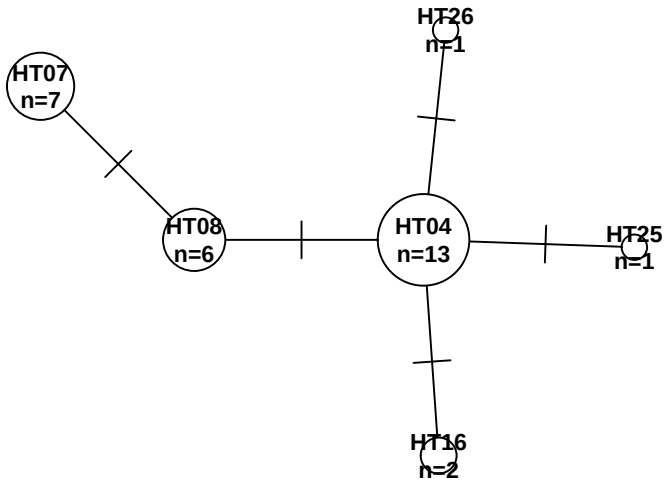
```
  attr(net, "labels") <- paste0(attr(net, "labels"),"\nn=", attr(net, "freq"))

  plot(net, size = size.new, scale.ratio = 6, pie = NULL,
     show.mutation = 1,threshold = 0,  srt = 1, cex = 0.75)
  title(paste0(pop))
}
```



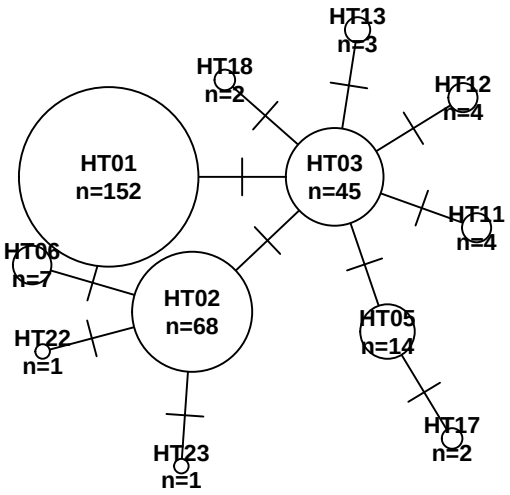
Cambridge Gulf



Daly River



Van Diemen Gulf



Papua New Guinea

Table 13: PHIst from pegas AMOVA using the pooled genetic distance

	regions	rivers
GLOBAL	0.5668958	0.7095871
regions	NA	0.3294620
rivers	NA	NA

6 Genetic differentiation

6.1 Overall - AMOVA

```
ggar.dist <- apex::dist.multidna(ggar.ml, pool = TRUE)

{amova.result <- pegas::amova(ggar.dist ~ regions/rivers,
                              data = adegenet::strata(ggar.gi), nperm = 10000)
print("AMOVA result from pegas using the pooled genetic distance:")
print(amova.result)
sig2 <- setNames(amova.result$varcomp$sigma2, rownames(amova.result$varcomp))
knitr::kable(
  pegas::getPhi(sig2),
  format = "latex",
  caption = "PHIst from pegas AMOVA using the pooled genetic distance"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
}
```

```
## [1] "AMOVA result from pegas using the pooled genetic
distance:"
##
## Analysis of Molecular Variance
##
## Call: pegas::amova(formula = ggar.dist ~ regions/rivers,
data = adegenet::strata(ggar.gi),
## nperm = 10000)
##
## SSD MSD df
## regions 1.148744e-06 2.871860e-07 4
## rivers 4.865332e-07 8.108887e-08 6
## Error 1.379628e-06 3.708676e-09 372
## Total 3.014905e-06 7.892422e-09 382
##
## Variance components:
## sigma2 P.value
## regions 7.2395e-09 0.0012
## rivers 1.8222e-09 0.0000
## Error 3.7087e-09
##
## Phi-statistics:
## regions.in.GLOBAL (Phi_CT) rivers.in.GLOBAL (Phi_ST)
## 0.5668958 0.7095871
## rivers.in.regions (Phi_SC)
## 0.3294620
##
## Variance coefficients:
## a b c
## 42.46480 18.51821 34.49608
```

Table 14: PHIst from pegas AMOVA using the euclidean distance

	regions	rivers
GLOBAL	0.2973134	0.4261045
regions	NA	0.1832838
rivers	NA	NA

```
D <- dist(tab(ggar.gi))

{amova.result <- pegas::amova(D ~ regions/rivers,
                              data = adegenet::strata(ggar.gi), nperm = 10000)
print("AMOVA result from pegas using the euclidean distance:")
print(amova.result)
sig2 <- setNames(amova.result$varcomp$sigma2, rownames(amova.result$varcomp))
knitr::kable(
  pegas::getPhi(sig2),
  format = "latex",
  caption = "PHIst from pegas AMOVA using the euclidean distance"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
}
```

```
## [1] "AMOVA result from pegas using the euclidean
distance:"
##
## Analysis of Molecular Variance
##
## Call: pegas::amova(formula = D ~ regions/rivers, data =
adegenet::strata(ggar.gi),
## nperm = 10000)
##
## SSD MSD df
## regions 52.06899 13.0172484 4
## rivers 35.71526 5.9525437 6
## Error 210.29407 0.5653066 372
## Total 298.07833 0.7803098 382
##
## Variance components:
## sigma2 P.value
## regions 0.29286 0.0013
## rivers 0.12686 0.0000
## Error 0.56531
##
## Phi-statistics:
## regions.in.GLOBAL (Phi_CT) rivers.in.GLOBAL (Phi_ST)
## 0.2973134 0.4261045
## rivers.in.regions (Phi_SC)
## 0.1832838
##
## Variance coefficients:
## a b c
## 42.46480 18.51821 34.49608
```

```
{amova.result <- vegan::adonis2(ggar.dist ~ regions/rivers,
                                data = adegenet::strata(ggar.gi),
                                permutations = 10000)
print("AMOVA result from vegan using the pooled genetic distance:")
```

```
print(amova.result)}
```

```
## [1] "AMOVA result from vegan using the pooled genetic
distance:"
## Df SumOfSqs R2 F Pr(>F)
## Model 10 1.635277e-06 0.5423976 44.09329 9.999e-05
## Residual 372 1.379628e-06 0.4576024 NA NA
## Total 382 3.014905e-06 1.0000000 NA NA
```

```
{amova.result <- vegan::adonis2(D ~ regions/rivers,
                                data = adegenet::strata(ggar.gi),
                                permutations = 10000)
print("AMOVA result from vegan using the euclidean distance:")
print(amova.result)}
```

```
## [1] "AMOVA result from vegan using the euclidean
distance:"
## Df SumOfSqs R2 F Pr(>F)
## Model 10 87.78426 0.2945006 15.52861 9.999e-05
## Residual 372 210.29407 0.7054994 NA NA
## Total 382 298.07833 1.0000000 NA NA
```

6.2 PHIst per river

```
strata <- as.character(ggar.gt@schemes$river)
names(strata) <- as.character(ggar.gt@schemes$id)
strataG::setStrata(ggar.gt) <- strata

fxdiff <- strataG::fixedDifferences(ggar.gt, count.indels = TRUE,
                                    consec.indels.as.one = TRUE,
  bases = c("a", "c", "g", "t", "-"))
knitr::kable(
  fxdiff$num.fixed[order(fxdiff$num.fixed$num.fixed, decreasing = TRUE),],
  format = "latex",
  caption = "Fixed differences between rivers"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
popstr.mt.river <-
  strataG::popStructTest(
    ggar.gt,
    nrep = 10000,
    stats = "all",
    type = "both",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores()-1,
    write.output = TRUE
  )
```

```
load("6.Ggar_Fst_objects.Rdata")
```

```
knitr::kable(
  popstr.mt.river$overall$result,
  digits = 4,
  format = "latex",
  caption = "Overall differentiation between rivers"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

Table 15: Fixed differences between rivers

	strata.1	strata.2	num.fixed
14	Daly River	Papua New Guinea	5
35	Ord River	Papua New Guinea	5
5	Adelaide River	Papua New Guinea	4
22	East Alligator River	Papua New Guinea	4
29	King Sound	Papua New Guinea	4
41	Papua New Guinea	Sampan Creek	4
42	Papua New Guinea	South Alligator River	4
43	Papua New Guinea	West Alligator River	4
45	Papua New Guinea	Wildman River	4
44	Papua New Guinea	West Cambridge Gulf	2
1	Adelaide River	Daly River	1
4	Adelaide River	Ord River	1
11	Daly River	East Alligator River	1
12	Daly River	King Sound	1
15	Daly River	Sampan Creek	1
16	Daly River	South Alligator River	1
17	Daly River	West Alligator River	1
19	Daly River	Wildman River	1
21	East Alligator River	Ord River	1
28	King Sound	Ord River	1
36	Ord River	Sampan Creek	1
37	Ord River	South Alligator River	1
38	Ord River	West Alligator River	1
40	Ord River	Wildman River	1
2	Adelaide River	East Alligator River	0
3	Adelaide River	King Sound	0
6	Adelaide River	Sampan Creek	0
7	Adelaide River	South Alligator River	0
8	Adelaide River	West Alligator River	0
9	Adelaide River	West Cambridge Gulf	0
10	Adelaide River	Wildman River	0
13	Daly River	Ord River	0
18	Daly River	West Cambridge Gulf	0
20	East Alligator River	King Sound	0
23	East Alligator River	Sampan Creek	0
24	East Alligator River	South Alligator River	0
25	East Alligator River	West Alligator River	0
26	East Alligator River	West Cambridge Gulf	0
27	East Alligator River	Wildman River	0
30	King Sound	Sampan Creek	0
31	King Sound	South Alligator River	0
32	King Sound	West Alligator River	0
33	King Sound	West Cambridge Gulf	0
34	King Sound	Wildman River	0
39	Ord River	West Cambridge Gulf	0
46	Sampan Creek	South Alligator River	0
47	Sampan Creek	West Alligator River	0
48	Sampan Creek	West Cambridge Gulf	0
49	Sampan Creek	Wildman River	0
50	South Alligator River	West Alligator River	0
51	South Alligator River	West Cambridge Gulf	0
52	South Alligator River	Wildman River	0
53	West Alligator River	West Cambridge Gulf	0
54	West Alligator River	Wildman River	0
55	West Cambridge Gulf	Wildman River	0

Table 16: Overall differentiation between rivers

	estimate	p.val
Chi2	3830.000	1e+00
Fst	0.000	1e+00
PHIst	0.446	1e-04

```
knitr::kable(
  popstr.mt.river$pairwise$result[order(popstr.mt.river$pairwise$result$PHIst,
                                        decreasing  = TRUE), -c(1:8)],
  format = "latex",
  digits = 4,
  caption = "Pairwise differentiation between rivers"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

6.3 PHIst per region

```
strata <- as.character(ggar.gt@schemes$pop)
names(strata) <- as.character(ggar.gt@schemes$id)
strataG::setStrata(ggar.gt) <- strata

fxdiff <- strataG::fixedDifferences(ggar.gt, count.indels = TRUE,
                                    consec.indels.as.one = TRUE,
                                    bases = c("a", "c", "g", "t", "-"))
knitr::kable(
  fxdiff$num.fixed[order(fxdiff$num.fixed$num.fixed, decreasing = TRUE), ],
  format = "latex",
  caption = "Fixed differences between regions"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
popstr.mt.pop <-
  strataG::popStructTest(
    ggar.gt,
    nrep = 10000,
    stats = "all",
    type = "both",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores()-1,
    write.output = TRUE
  )
```

```
knitr::kable(
  popstr.mt.pop$overall$result,
  digits = 4,
  format = "latex",
  caption = "Overall differentiation between regions"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
knitr::kable(
  popstr.mt.pop$pairwise$result[order(popstr.mt.pop$pairwise$result$PHIst,
  decreasing = TRUE), -c(1:8)],
  format = "latex",
  digits = 4,
```

Table 17: Pairwise differentiation between rivers

	PHIst	PHIst.p.val
Papua New Guinea (6) v. West Alligator River (41)	0.9384	0.0001
Ord River (15) v. Papua New Guinea (6)	0.9311	0.0001
King Sound (14) v. Papua New Guinea (6)	0.9006	0.0001
Daly River (30) v. Papua New Guinea (6)	0.8931	0.0001
Ord River (15) v. West Alligator River (41)	0.8712	0.0001
Adelaide River (25) v. Papua New Guinea (6)	0.8620	0.0001
Papua New Guinea (6) v. Wildman River (45)	0.8275	0.0001
Daly River (30) v. West Alligator River (41)	0.8223	0.0001
Papua New Guinea (6) v. South Alligator River (101)	0.8182	0.0001
East Alligator River (61) v. Papua New Guinea (6)	0.7980	0.0001
King Sound (14) v. Ord River (15)	0.7676	0.0001
Papua New Guinea (6) v. Sampan Creek (30)	0.7672	0.0001
King Sound (14) v. West Alligator River (41)	0.7499	0.0001
Adelaide River (25) v. Ord River (15)	0.7279	0.0001
Adelaide River (25) v. West Alligator River (41)	0.7147	0.0001
Adelaide River (25) v. Daly River (30)	0.7022	0.0001
Daly River (30) v. King Sound (14)	0.6712	0.0001
Daly River (30) v. Wildman River (45)	0.6545	0.0001
Ord River (15) v. Wildman River (45)	0.6527	0.0001
West Alligator River (41) v. West Cambridge Gulf (15)	0.6375	0.0001
Daly River (30) v. South Alligator River (101)	0.6183	0.0001
Ord River (15) v. South Alligator River (101)	0.6094	0.0001
Papua New Guinea (6) v. West Cambridge Gulf (15)	0.5736	0.0001
Daly River (30) v. Sampan Creek (30)	0.5610	0.0001
Daly River (30) v. East Alligator River (61)	0.5587	0.0001
East Alligator River (61) v. Ord River (15)	0.5448	0.0001
Ord River (15) v. Sampan Creek (30)	0.5426	0.0001
South Alligator River (101) v. West Cambridge Gulf (15)	0.5264	0.0001
West Cambridge Gulf (15) v. Wildman River (45)	0.4943	0.0001
Adelaide River (25) v. Wildman River (45)	0.4909	0.0001
Adelaide River (25) v. King Sound (14)	0.4726	0.0001
Adelaide River (25) v. West Cambridge Gulf (15)	0.4688	0.0001
East Alligator River (61) v. West Cambridge Gulf (15)	0.4250	0.0001
King Sound (14) v. Wildman River (45)	0.3916	0.0001
Sampan Creek (30) v. West Cambridge Gulf (15)	0.3703	0.0001
East Alligator River (61) v. West Alligator River (41)	0.3653	0.0001
King Sound (14) v. West Cambridge Gulf (15)	0.3365	0.0001
King Sound (14) v. South Alligator River (101)	0.3263	0.0002
Daly River (30) v. West Cambridge Gulf (15)	0.3131	0.0001
Sampan Creek (30) v. West Alligator River (41)	0.2939	0.0001
Adelaide River (25) v. South Alligator River (101)	0.2494	0.0003
Ord River (15) v. West Cambridge Gulf (15)	0.2432	0.0271
Adelaide River (25) v. Sampan Creek (30)	0.2325	0.0002
Adelaide River (25) v. East Alligator River (61)	0.2215	0.0001
South Alligator River (101) v. West Alligator River (41)	0.2213	0.0001
East Alligator River (61) v. Wildman River (45)	0.1801	0.0001
King Sound (14) v. Sampan Creek (30)	0.1784	0.0007
South Alligator River (101) v. Wildman River (45)	0.1404	0.0004
West Alligator River (41) v. Wildman River (45)	0.1269	0.0016
East Alligator River (61) v. King Sound (14)	0.1260	0.0042
East Alligator River (61) v. South Alligator River (101)	0.0823	0.0005
Sampan Creek (30) v. Wildman River (45)	0.0807	0.0130
Daly River (30) v. Ord River (15)	0.0603	0.0394
Sampan Creek (30) v. South Alligator River (101)	0.0357	0.0754
East Alligator River (61) v. Sampan Creek (30)	0.0047	0.2881

Table 18: Fixed differences between regions

	strata.1	strata.2	num.fixed
6	Daly River	Papua New Guinea	5
8	King Sound	Papua New Guinea	4
10	Papua New Guinea	Van Diemen Gulf	4
3	Cambridge Gulf	Papua New Guinea	2
5	Daly River	King Sound	1
7	Daly River	Van Diemen Gulf	1
1	Cambridge Gulf	Daly River	0
2	Cambridge Gulf	King Sound	0
4	Cambridge Gulf	Van Diemen Gulf	0
9	King Sound	Van Diemen Gulf	0

Table 19: Overall differentiation between regions

	estimate	p.val
Chi2	1532.0000	1e-04
Fst	0.0000	1e+00
PHIst	0.5457	1e-04

```
  caption = "Pairwise differentiation between regions"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

Table 20: Pairwise differentiation between regions

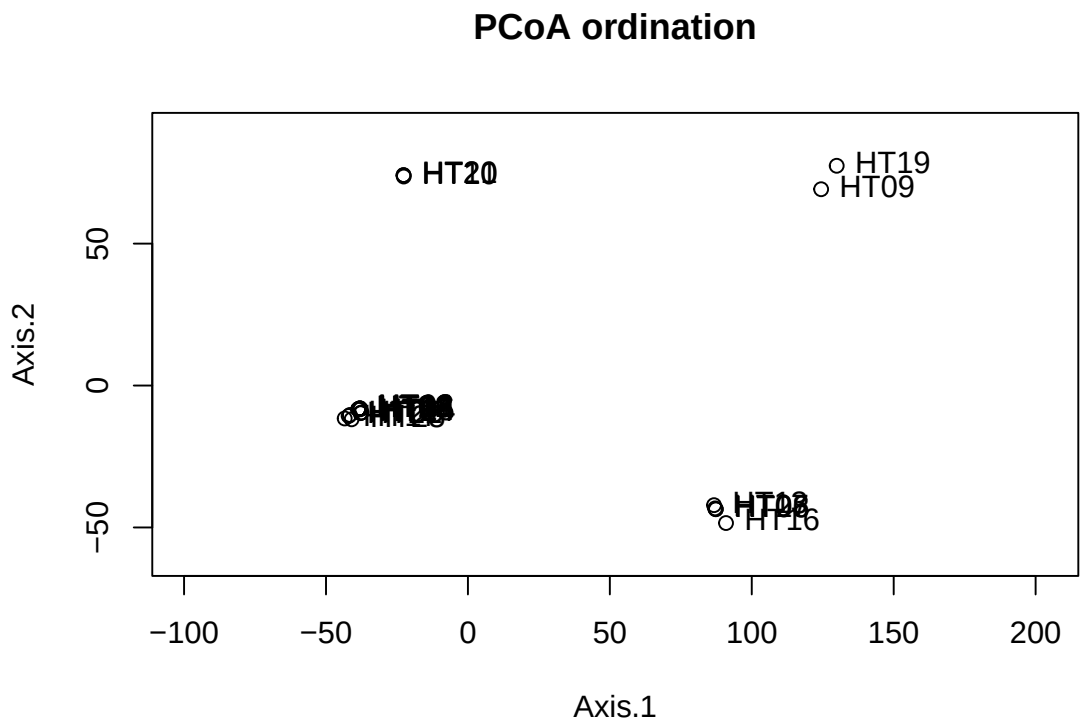
	PHIst	PHIst.p.val
King Sound (14) v. Papua New Guinea (6)	0.9006	1e-04
Daly River (30) v. Papua New Guinea (6)	0.8931	1e-04
Papua New Guinea (6) v. Van Diemen Gulf (303)	0.7965	1e-04
Cambridge Gulf (30) v. Papua New Guinea (6)	0.6875	1e-04
Daly River (30) v. King Sound (14)	0.6712	1e-04
Daly River (30) v. Van Diemen Gulf (303)	0.5625	1e-04
Cambridge Gulf (30) v. Van Diemen Gulf (303)	0.4940	1e-04
Cambridge Gulf (30) v. King Sound (14)	0.3475	1e-04
King Sound (14) v. Van Diemen Gulf (303)	0.2416	1e-04
Cambridge Gulf (30) v. Daly River (30)	0.1211	4e-04

7 PCA

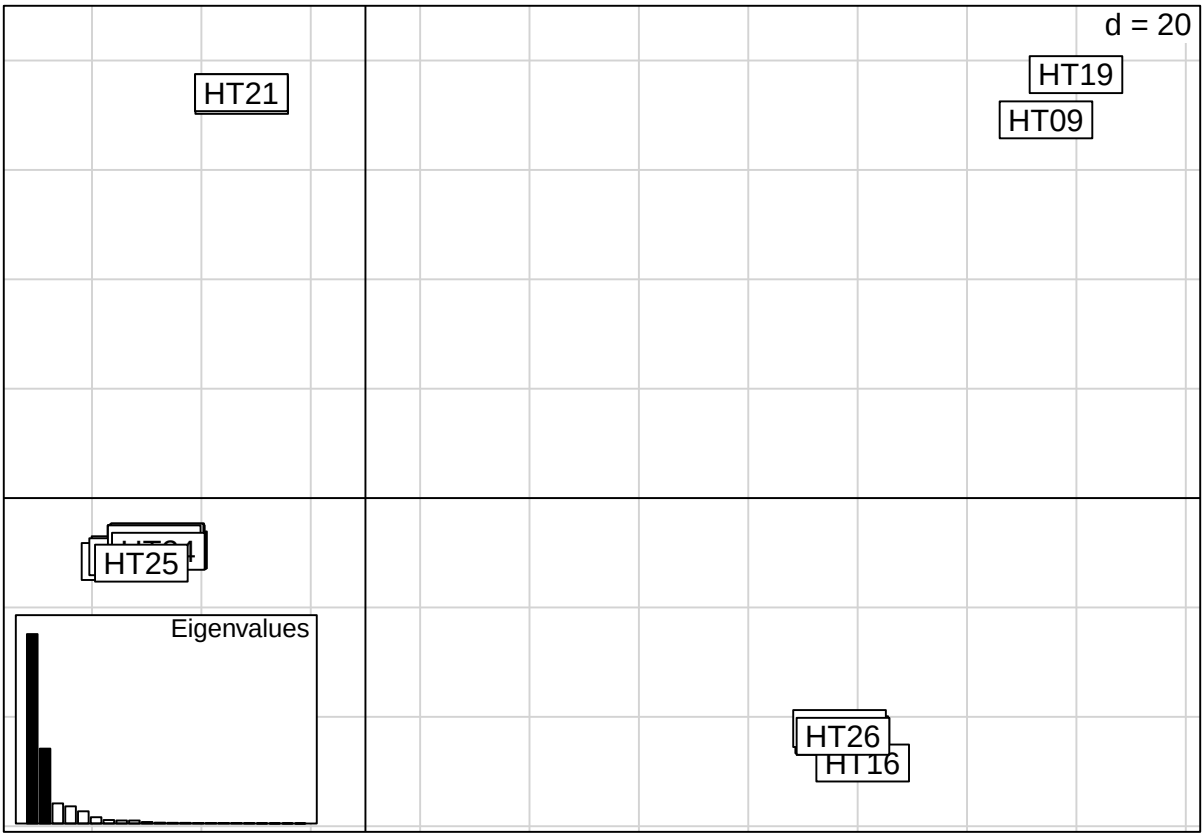
```
ggar.haps <- strataG::labelHaplotypes(x = ggar.bin, prefix = "HT",
                                      use.indels = TRUE)
ggar.hapdist <- dist(ggar.haps$hap.seqs)
table(ggar.haps$haps)
```

```
##
## HT01 HT02 HT03 HT04 HT05 HT06 HT07 HT08 HT09 HT10 HT11
HT12 HT13 HT14 HT15 HT16
## 152 68 56 35 14 7 7 6 4 4 4 4 3 3 2 2
## HT17 HT18 HT19 HT20 HT21 HT22 HT23 HT24 HT25 HT26
## 2 2 1 1 1 1 1 1 1 1
```

```
x <- ape::pcoa(ggar.hapdist, correction = "none", rn = NULL)
biplot(x, Y = NULL, plot.axes = c(1,2), dir.axis1 = 1,
       dir.axis2 = 1, rn = NULL, main = NULL)
```




```
pco1 <- dudi.pco(ggar.hapdist, scannf = FALSE, nf = 2)
scatter(pco1, posi = "bottomleft")
```


d = 20
HT21
HT19
HT09
HT25
Eigenvalues
HT26
HT16

8 Demographic analyses

The mismatch distribution and Watterson's estimator of theta (theta = 2Nefμ) were calculated with the 'MMD' and 'theta.s' functions as implemented in pegas. The mismatch distribution tests for population expansion were based on pairwise site differences between haplotypes, where mutation rate is assumed to be constant (Rogers and Harpending 1992). theta measures historical female effective population size (Nef) relative to mutation rate; a larger theta indicated a longer time to coalescence across population genealogies given the observed population growth.

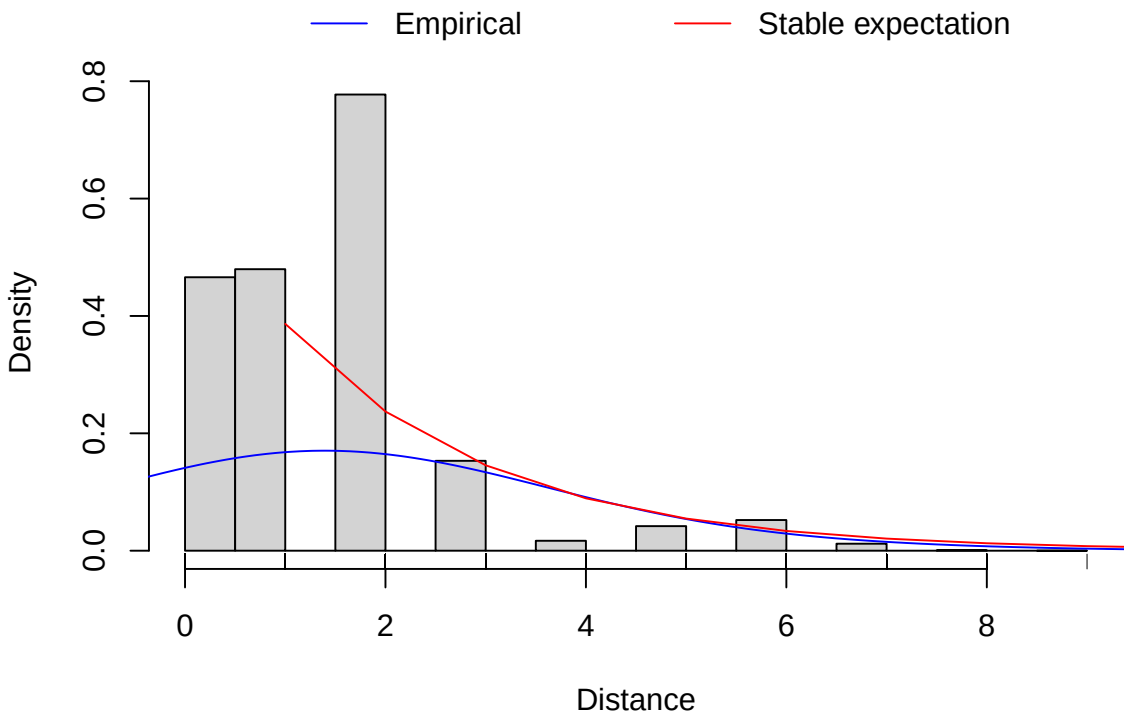
8.1 MMD

A histogram of the number of differences between pairs of haplotypes.

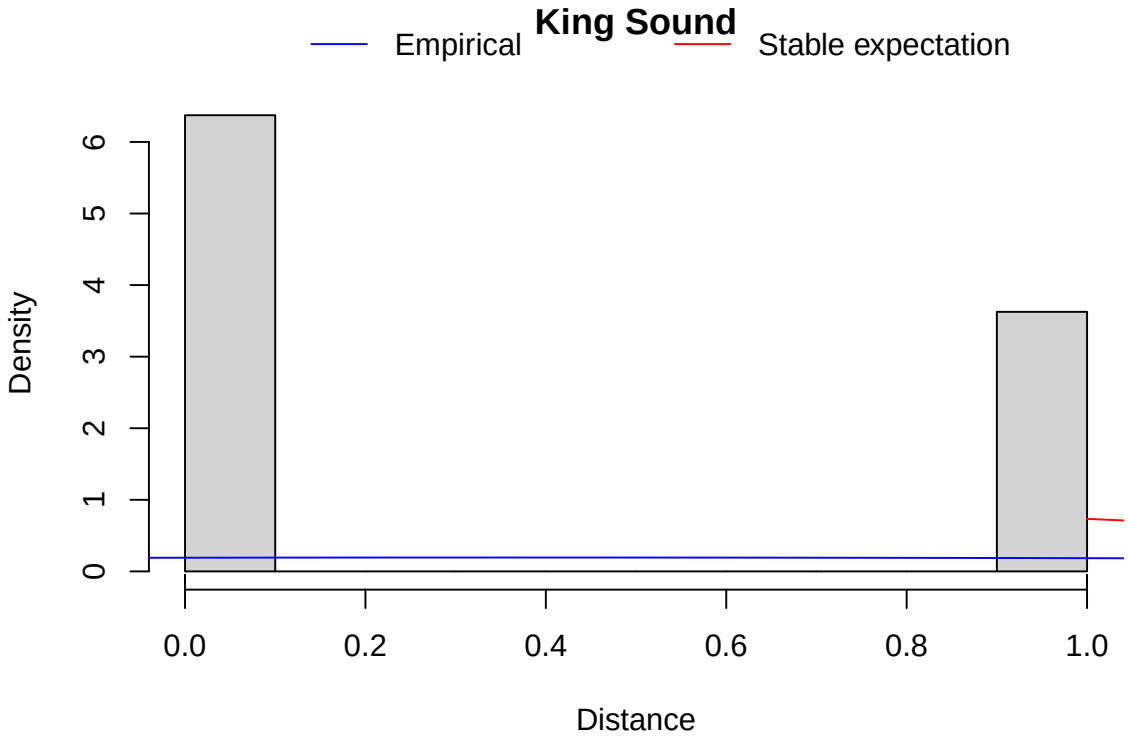
- Multimodal: demographic equilibrium or decline
- Unimodal: recent demographic expansion or range expansion with high migration rate.

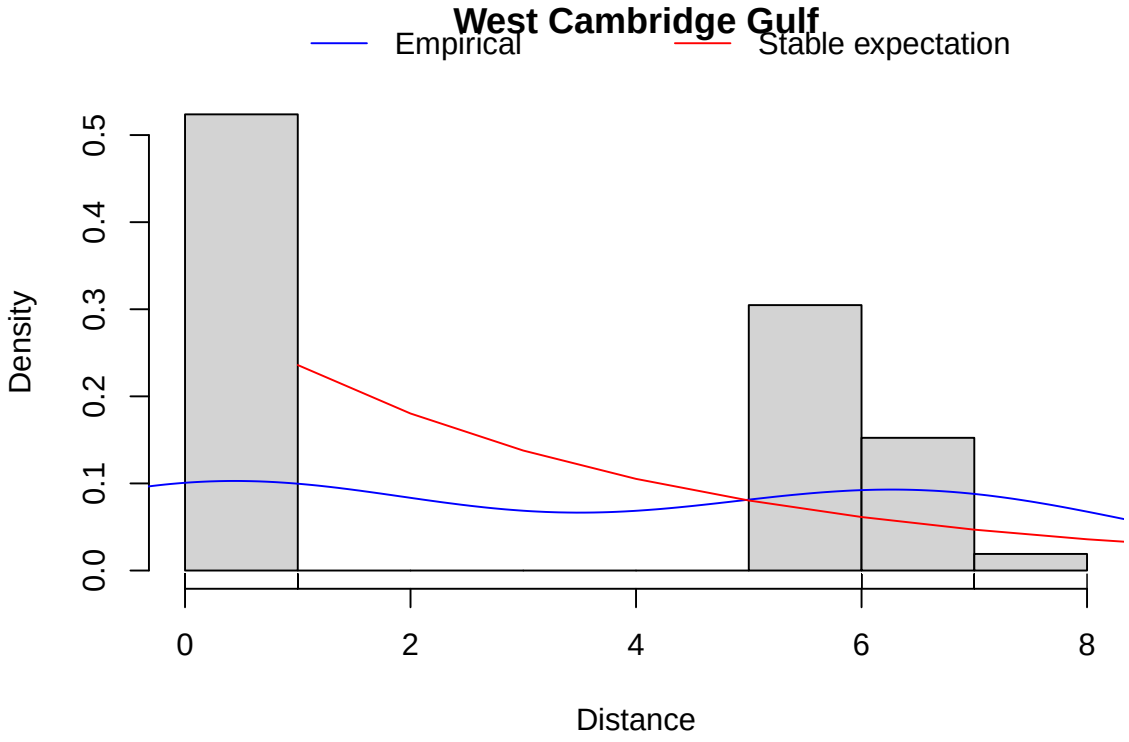
8.1.1 Per river

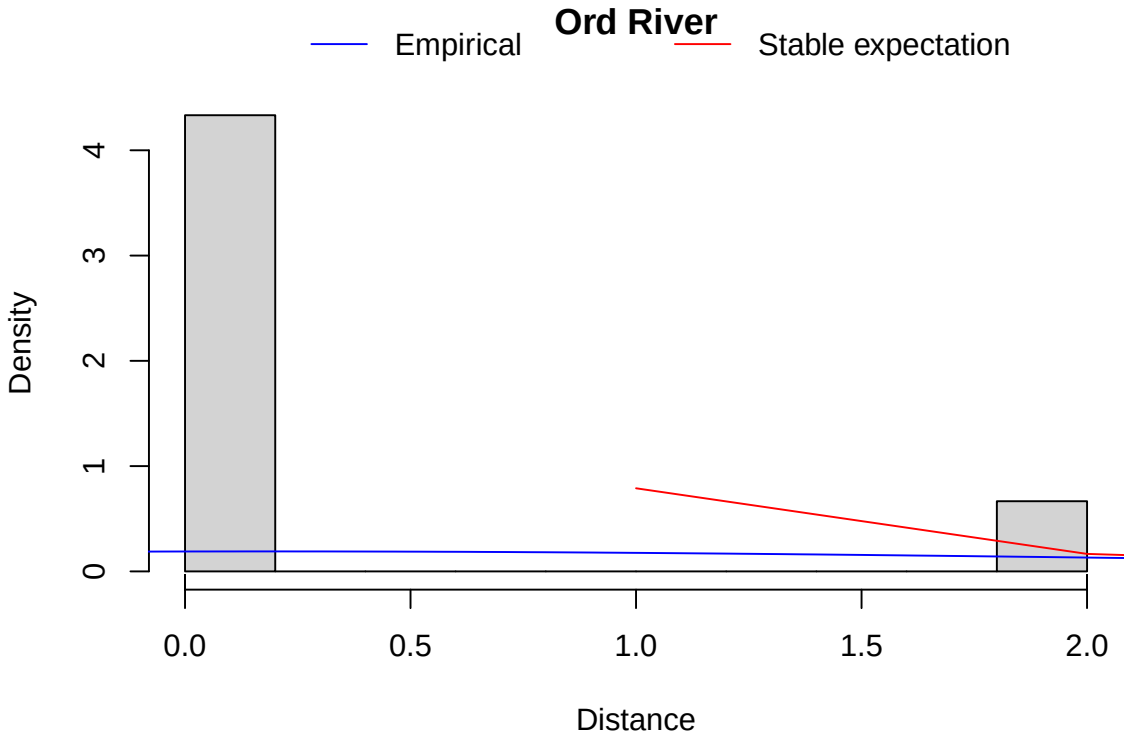
```
mmd <- pegas::MMD(ggar.bin, xlab = "Distance", main = "", rug = TRUE,
                  legend = TRUE, lcol = c("blue", "red"),
                  lty = c(1, 1), bw = 2)
```

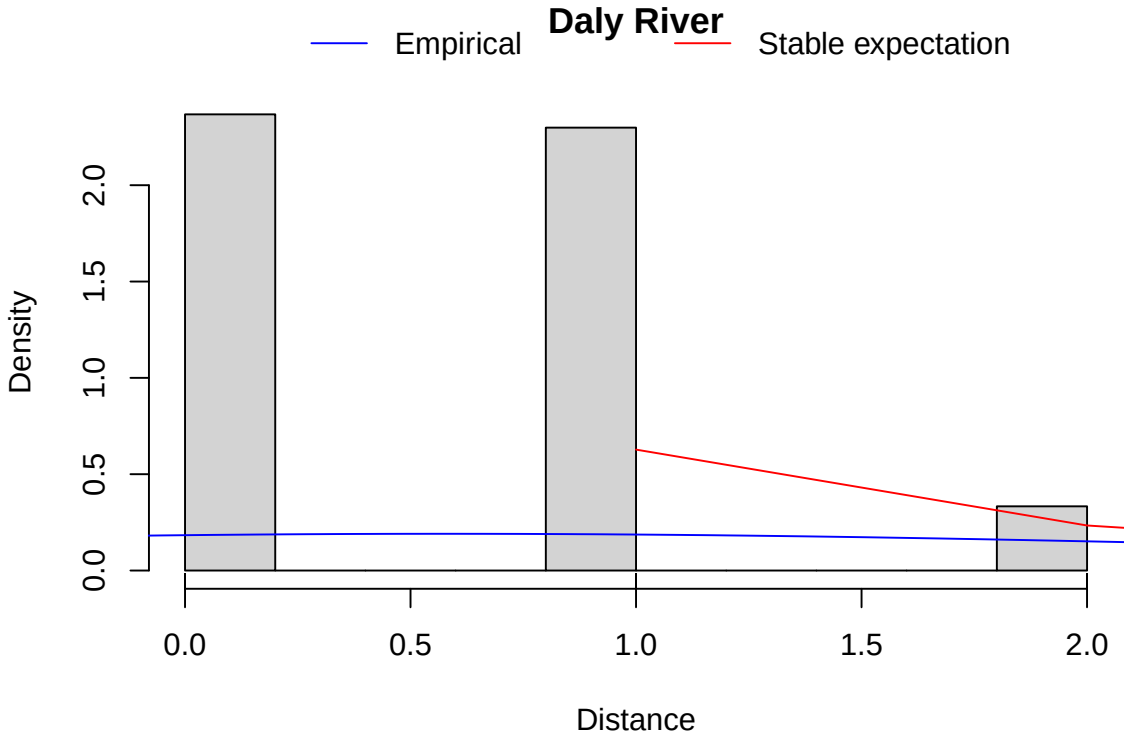


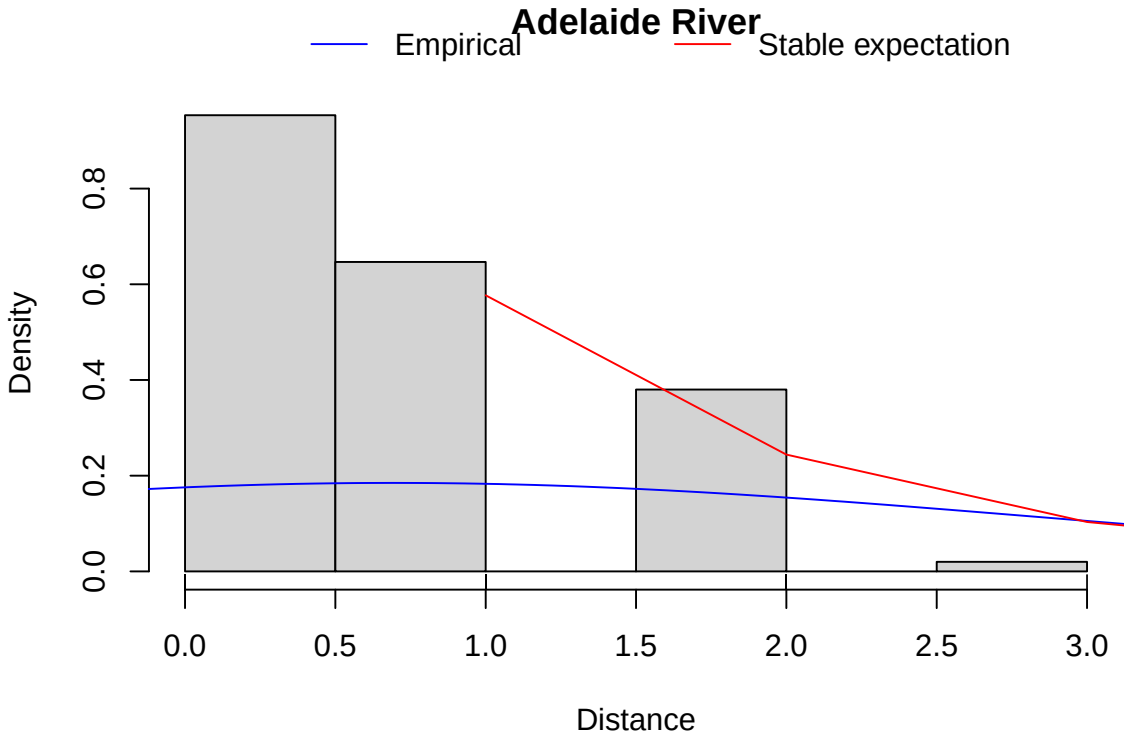

```
for (river in levels(ggar.gt@schemes$river)) {
  ind.name <- ggar.gt@schemes$id[ggar.gt@schemes$river == river]
  ggar.bin.mmd <- ggar.bin[dimnames(ggar.bin)[[1]] %in% ind.name,]
  pegas::MMD(ggar.bin.mmd, xlab = "Distance", main = river, rug = TRUE,
             legend = TRUE, lcol = c("blue", "red"), lty = c(1, 1), bw = 2)
}
```

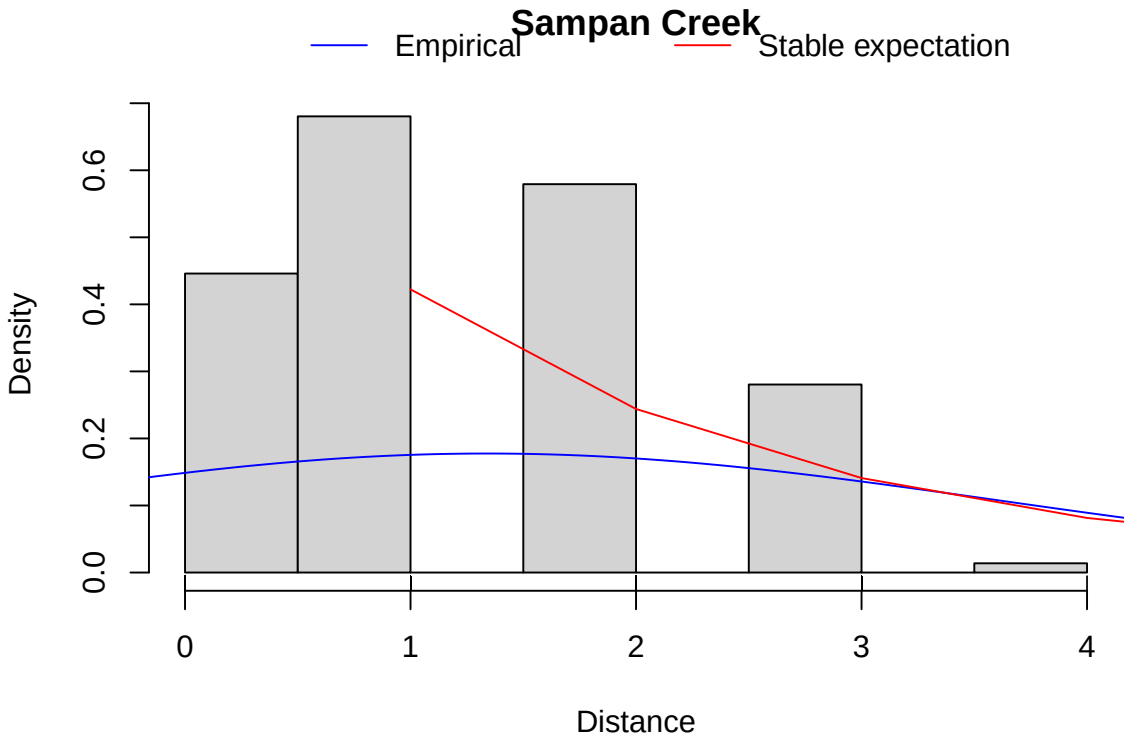

King Sound
Empirical
Stable expectation
Density
0
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.8
1.0
Distance

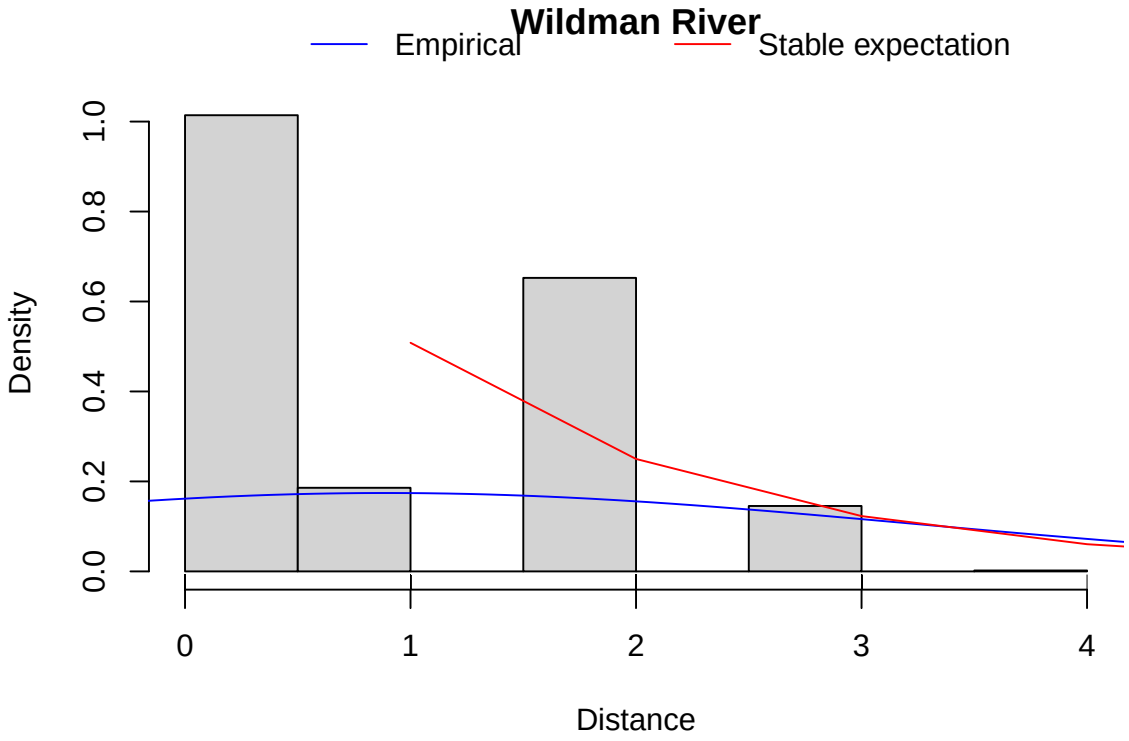

West Cambridge Gulf
Empirical
Stable expectation
Density
0.0
0.1
0.2
0.3
0.4
0.5
0
2
4
6
8
Distance

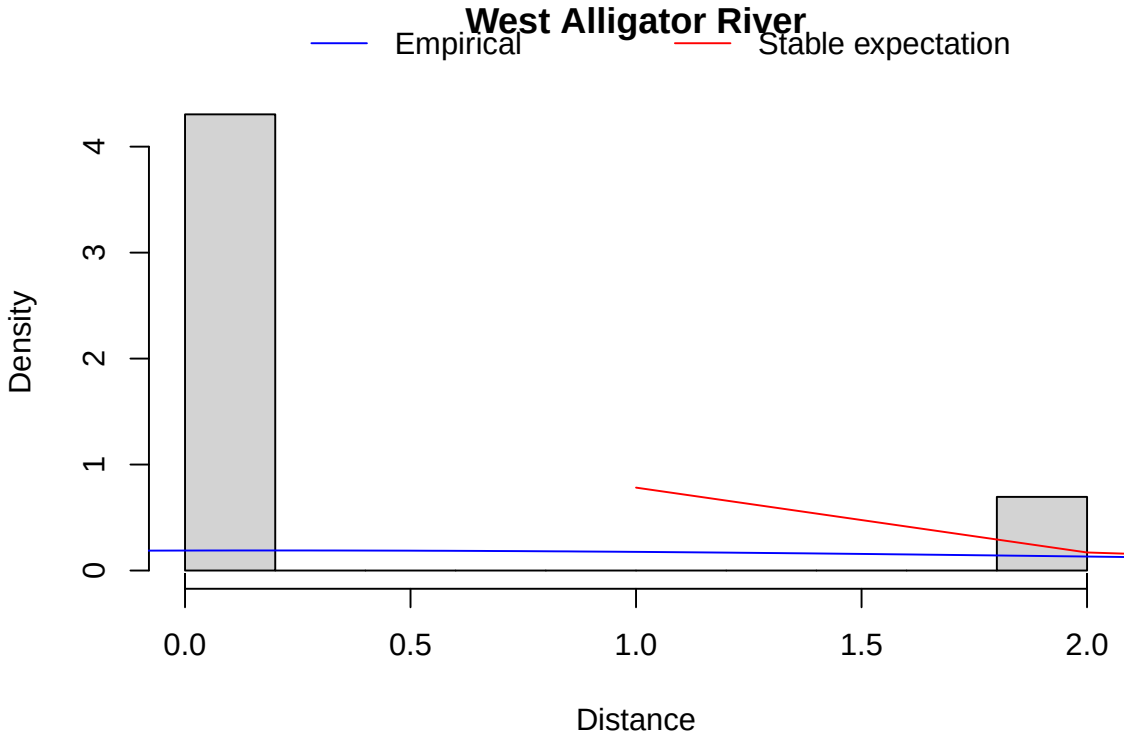

Ord River
Empirical
Stable expectation
Density
0
1
2
3
4
0.0
0.5
1.0
1.5
2.0
Distance

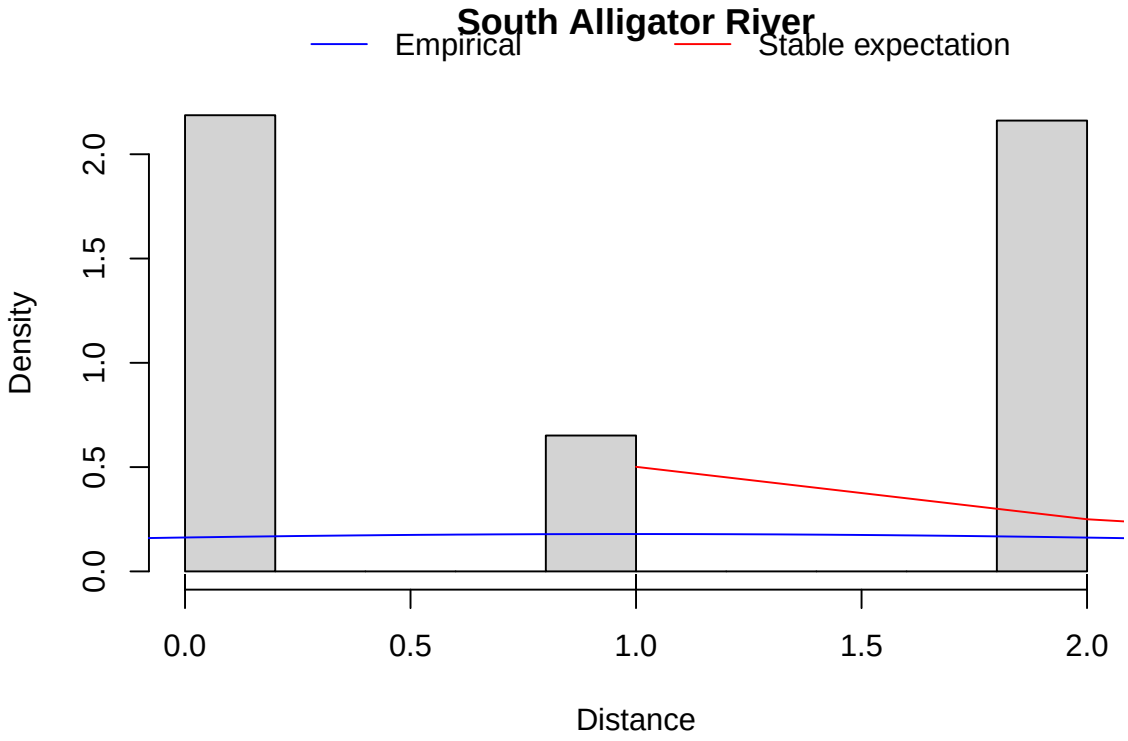

Daly River
Empirical
Stable expectation
Density
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
2.0
Distance

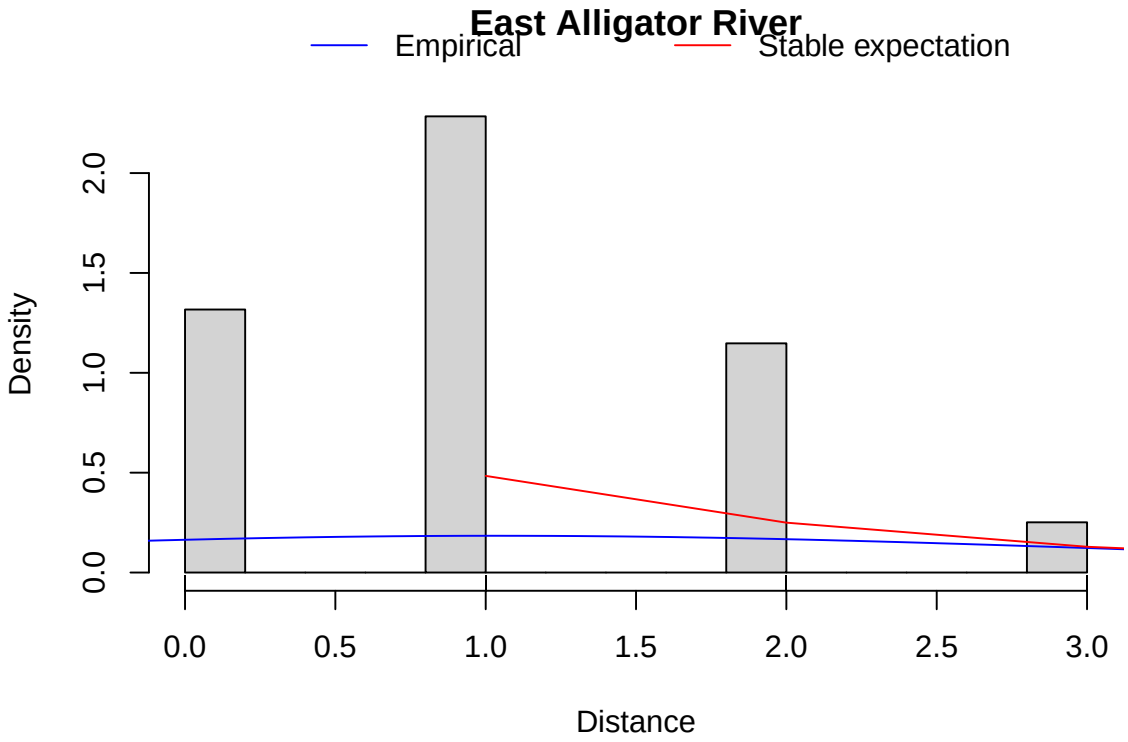

Adelaide River
Empirical
Stable expectation
Density
0.0
0.2
0.4
0.6
0.8
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Distance

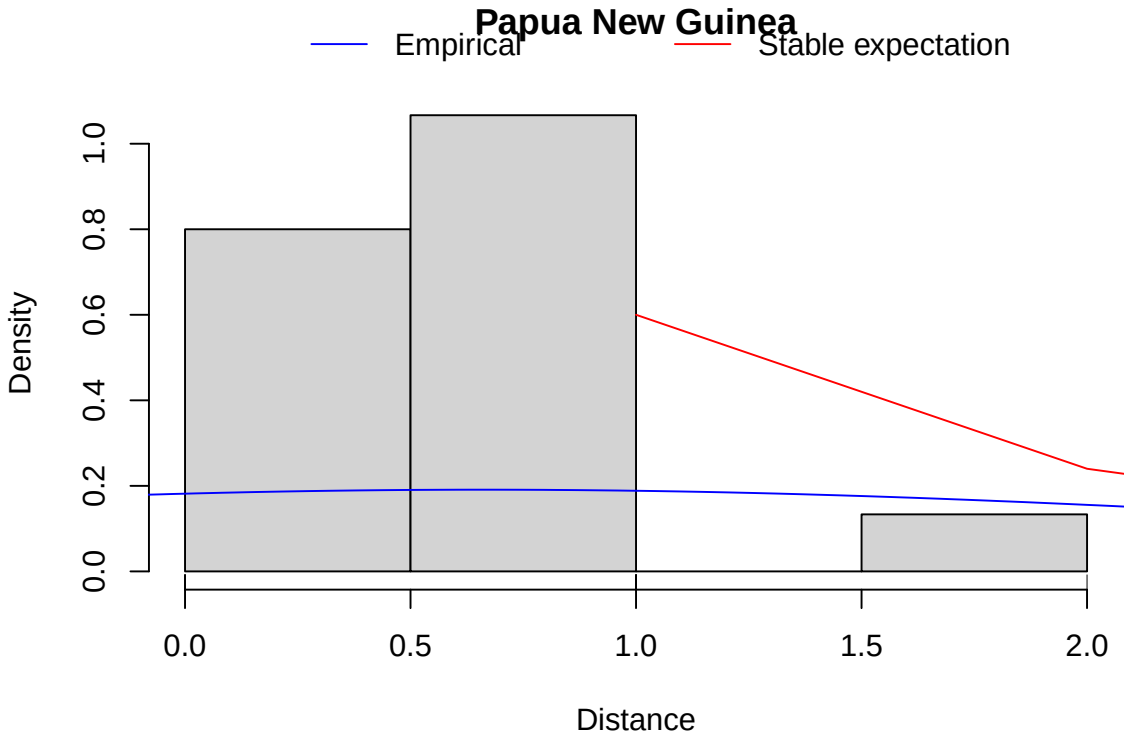

Sampan Creek
Empirical
Stable expectation
Density
0.0
0.2
0.4
0.6
0
1
2
3
4
Distance


Wildman River
Empirical
Stable expectation
Density
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
Distance


West Alligator River
Empirical
Stable expectation
Density
0
1
2
3
4
0.0
0.5
1.0
1.5
2.0
Distance

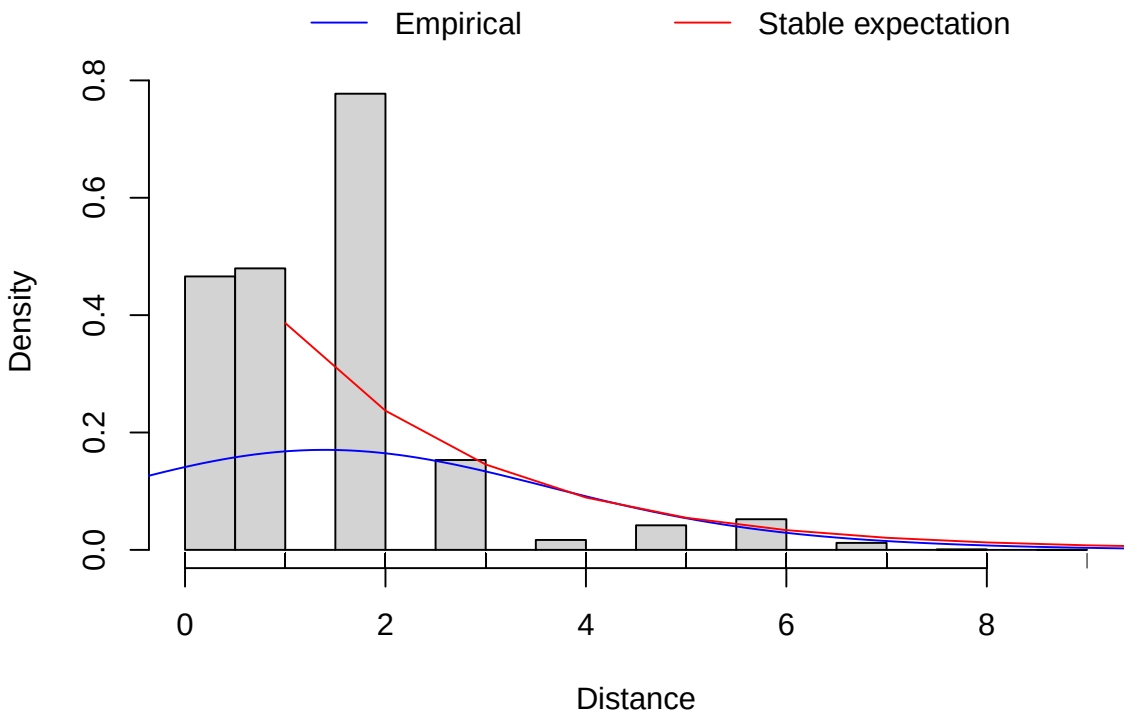

South Alligator River
Empirical
Stable expectation
Density
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
2.0
Distance


East Alligator River
Empirical
Stable expectation
Density
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Distance

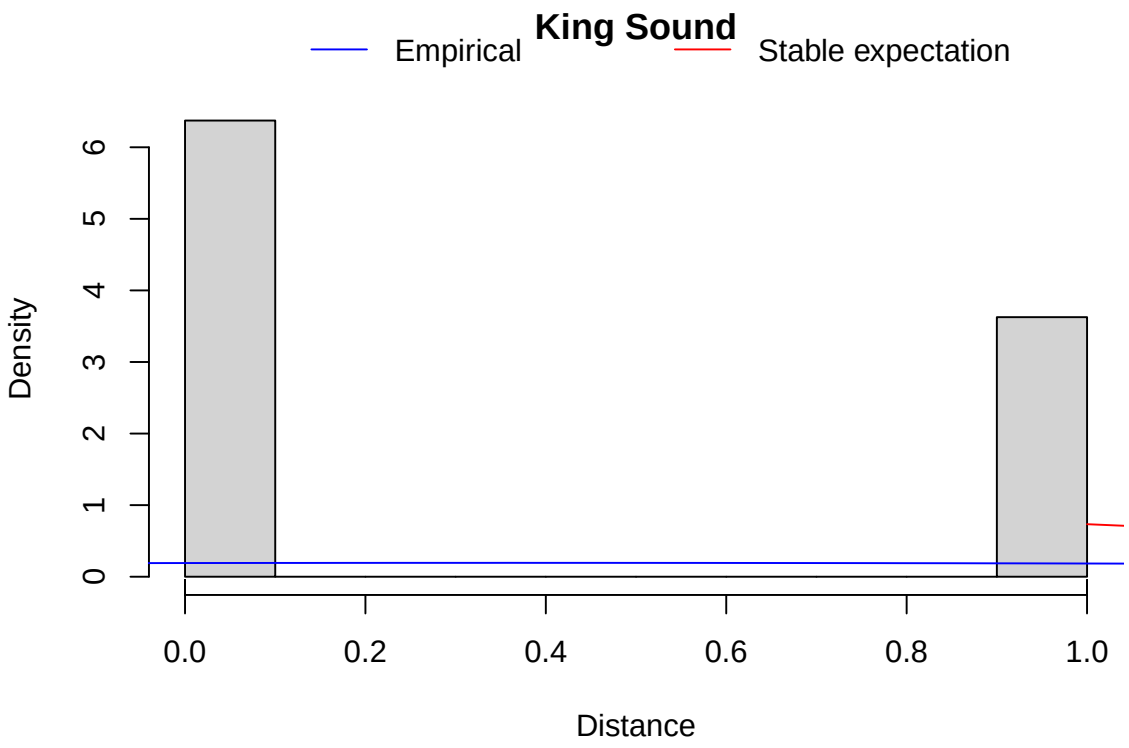


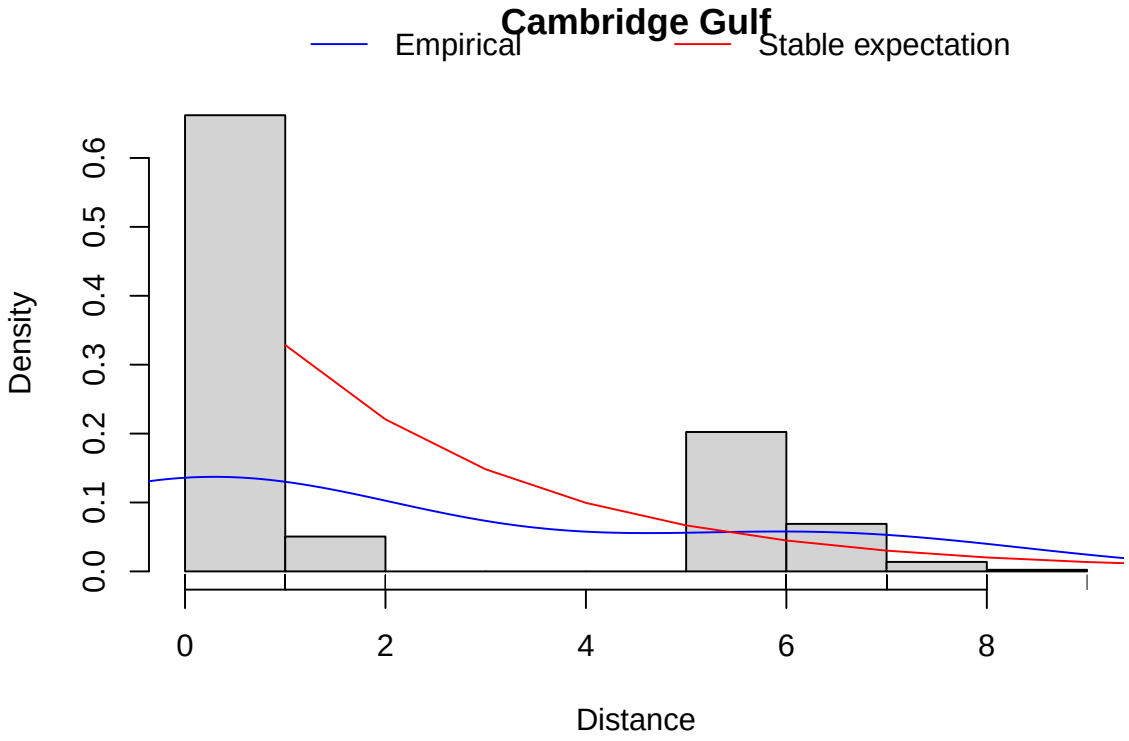
8.1.2 Per region

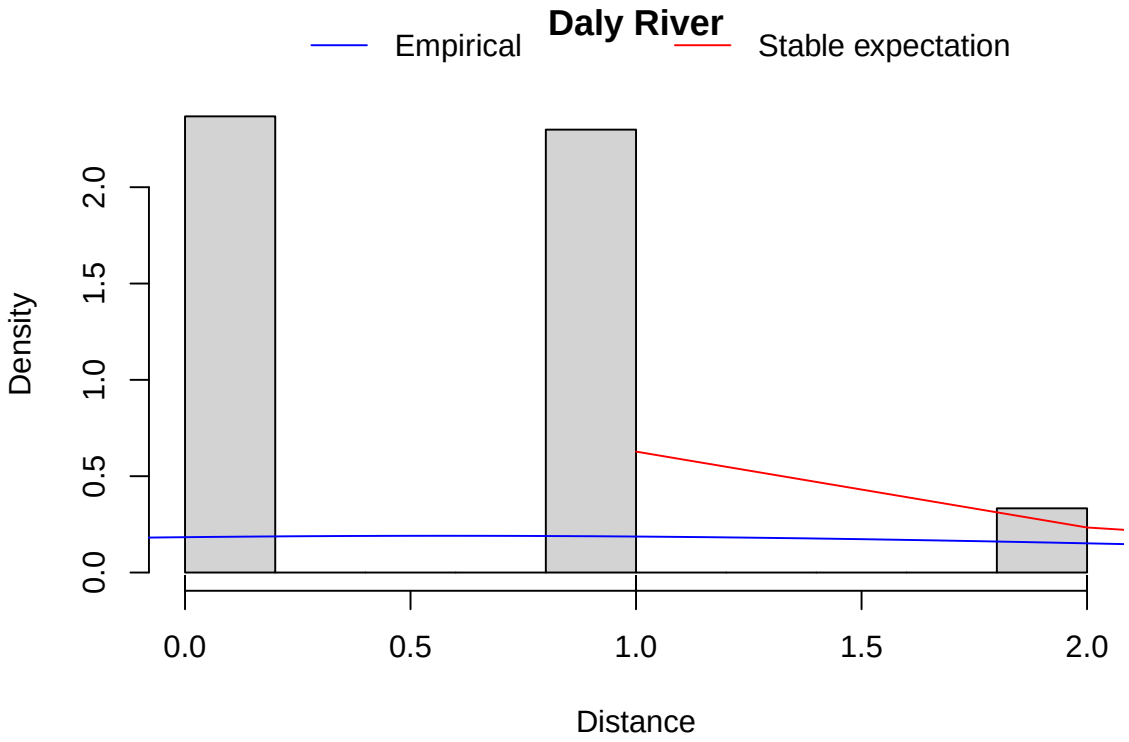
```
mmd <- pegas::MMD(ggar.bin, xlab = "Distance", main = "", rug = TRUE,
                  legend = TRUE, lcol = c("blue", "red"), lty = c(1, 1), bw = 2)
```

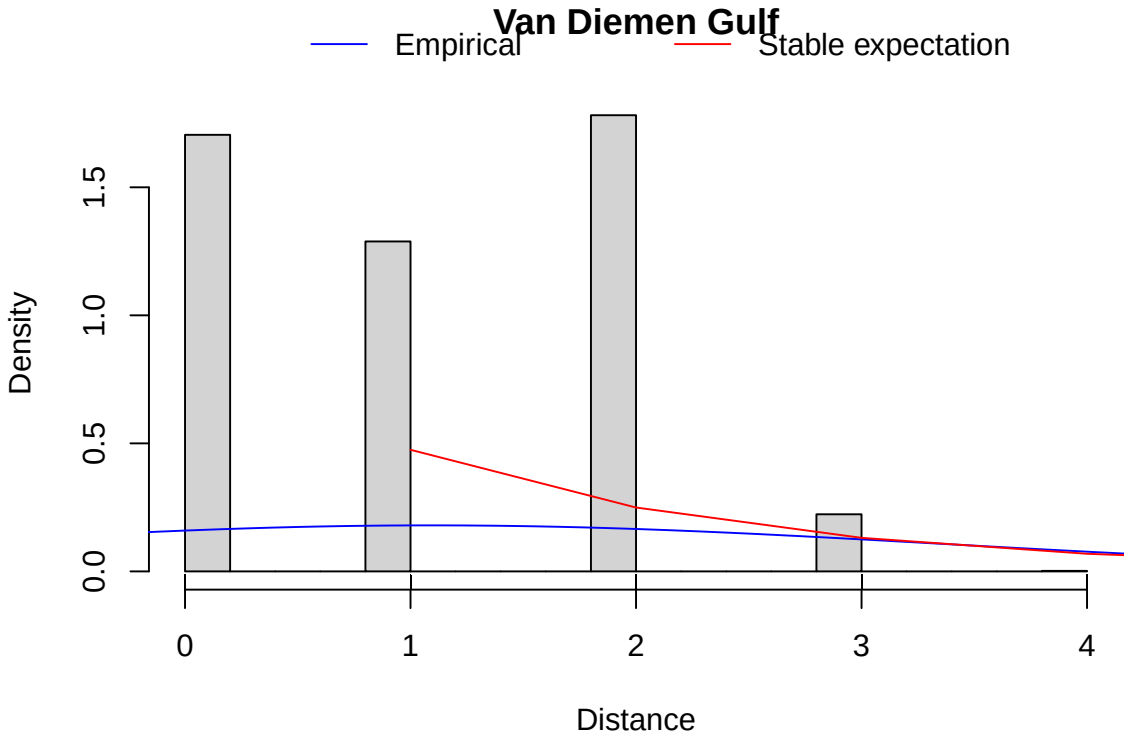


```
for (pop in levels(ggar.gt@schemes$pop)) {
  ind.name <- ggar.gt@schemes$id[ggar.gt@schemes$pop == pop]
  ggar.bin.mmd <- ggar.bin[dimnames(ggar.bin)[[1]] %in% ind.name,]
  pegas::MMD(ggar.bin.mmd, xlab = "Distance", main = pop, rug = TRUE,
             legend = TRUE, lcol = c("blue", "red"), lty = c(1, 1), bw = 2)
}
```




Cambridge Gulf
Empirical
Stable expectation
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
2
4
6
8
Distance


Daly River
Empirical
Stable expectation
Density
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
2.0
Distance


Van Diemen Gulf
Empirical
Stable expectation
Density
0.0
0.5
1.0
1.5
0
1
2
3
4
Distance

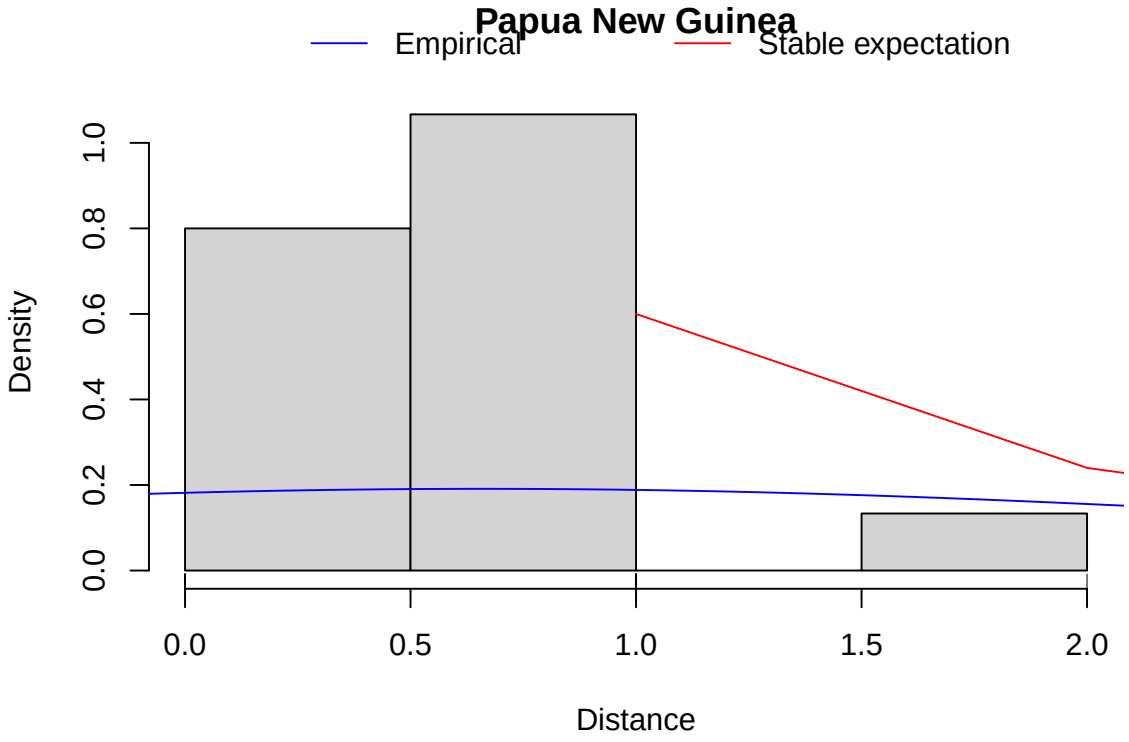

Papua New Guinea
Empirical
Stable expectation
Density
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
Distance

Table 21: Theta of segragating site per region

	River	Theta.segregating.sites	Variance.of.theta
1	King Sound	0.3144522	0.1142393
11	West Cambridge Gulf	2.7679002	1.9932672
12	Ord River	0.6150889	0.2455630
13	Daly River	1.2620992	0.4820872
14	Adelaide River	1.3241672	0.5479574
15	Sampan Creek	1.5145190	0.6177467
16	Wildman River	1.1434515	0.3724408
17	West Alligator River	0.4674488	0.1285942
18	South Alligator River	0.7711025	0.1847776
19	East Alligator River	1.2820868	0.3961742
110	Papua New Guinea	0.8759124	0.5989928

8.2 Theta

8.2.1 Per river

```
theta.river <- data.frame(River = character(),
                          `Theta segregating sites` = numeric(),
                          `Variance of theta` = numeric())
for (river in levels(ggar.gt@schemes$river)) {
  ind.name <- ggar.gt@schemes$id[ggar.gt@schemes$river == river]
  ggar.bin.seg <- ggar.bin[dimnames(ggar.bin)[[1]] %in% ind.name,]
  seg <- ape::seg.sites(ggar.bin.seg)
  s <- length(seg)
  n <- length(ind.name)
  res <- pegas::theta.s(s, n , variance = TRUE)
  res <- data.frame(river, res[1], res[2])
  colnames(res) <- c("River", "Theta segregating sites", "Variance of theta")
  theta.river <- rbind(theta.river, res)
}
knitr::kable(theta.river,
             format = "latex",
             caption = "Theta of segragating site per region") %>%
  kableExtra::kable_styling(full_width = FALSE)
```

8.2.2 Per region

```
theta.pop <- data.frame(Region = character(),
                          `Theta segregating sites` = numeric(),
                          `Variance of theta` = numeric())
for (pop in levels(ggar.gt@schemes$pop)) {
  ind.name <- ggar.gt@schemes$id[ggar.gt@schemes$pop == pop]
  ggar.bin.seg <- ggar.bin[dimnames(ggar.bin)[[1]] %in% ind.name,]
  seg <- ape::seg.sites(ggar.bin.seg)
  s <- length(seg)
  n <- length(ind.name)
  res <- pegas::theta.s(s, n , variance = TRUE)
  res <- data.frame(pop, res[1], res[2])
  colnames(res) <- c("Region", "Theta segregating sites", "Variance of theta")
  theta.pop <- rbind(theta.pop, res)
}
knitr::kable(theta.pop,
             format = "latex",
             caption = "Theta of segragating site per region") %>%
```

Table 22: Theta of segragating site per region

	Region	Theta.segregating.sites	Variance.of.theta
1	King Sound	0.3144522	0.1142393
11	Cambridge Gulf	2.5241984	1.2911912
12	Daly River	1.2620992	0.4820872
13	Van Diemen Gulf	1.7490030	0.4050475
14	Papua New Guinea	0.8759124	0.5989928

```
kableExtra::kable_styling(full_width = FALSE)
```

8.3 Neutrality tests

Neutrality was tested with 'TajimasD' and 'Fusfs' functions from strataG and the 'R2.test' function from pegas. Tajima's D and R2 compare theta values based on nucleotide site differences and Fu's FS uses pairwise differences (Ramos-Onsins and Rozas 2002). Significantly negative FS or D values and R2 values close to zero indicated an excess of low-frequency mutations, as expected under population expansion. Fu's FS, Tajima's D, and R2 are arguably more robust than other tests, such as mismatch distributions (Ramos-Onsins and Rozas 2002).

- Fu's Fs: Based on Ewens' sampling distribution, taking into account the number of different haplotypes in the sample.
  - StrataG issue: Currently, this function is limited to calculating Fs for fewer than approximately 172 sequences due to numerical overflow issues. NaN will be returned for larger data sets. THAT IS WHY I SAMPLE 170 INDIVIDUALS
  - Negative = recent population expansion and an excess of rare alleles
  - Positive = recent population bottleneck potential based on deficiency of rare alleles
- R2: Comparison of the difference between the number of singleton mutations and the average number of nucleotide differences.
  - Closer to 0 = population expansion based on the distribution of singleton mutations
- Tajimas' D: Comparison of estimates of the number of segregating sites and the mean pairwise difference between sequences.
  - Negative = recent population expansion or purifying selection (historical bottleneck/selective sweep)
  - Positive = recent population contraction or selection maintaining diversity

8.3.1 full mitogenome

```
strataG::TiTvRatio(ggar.gt) # No transversions
```

8.3.1.1 Per river

```
##          Ti          Tv Ti.Tv.ratio
##          25           0         Inf
```

```
neutral.river <- data.frame(River = character(), `Fu's Fs` = numeric(),
                            `Theta for R2 test` = numeric(), R2 = numeric(),
                            `R2 p-value` = numeric(), `Tajima's D` = numeric(),
                            `Tajima's p-value` = numeric(),
                            `Taj D lower 95% CI` = numeric(),
                            `Taj D upper 95% CI` = numeric())
for (river in levels(ggar.gt@schemes$river)) {
  ggar.gt.neu <- ggar.gt[ggar.gt@schemes$river %in% river,,,drop = TRUE]
  ind.name <- ggar.gt.neu@data$id
  ggar.bin.neu <- ggar.bin[dimnames(ggar.bin)[[1]] %in% ind.name,]
  if (length(ggar.gt.neu@data$id) > 172) {
    fu <- strataG::fusFs(ggar.gt.neu[sample(x = 1:length(ggar.gt.neu@data$id),
                                            size = 170, replace = FALSE),,,
                                     drop = TRUE])
  } else {
    fu <- strataG::fusFs(ggar.gt.neu)
  }
  seg <- ape::seg.sites(ggar.bin.seg)
  s <- length(seg)
  n <- length(ind.name)
  res <- pegas::theta.s(s, n , variance = TRUE)
  res <- data.frame(river, res[1], res[2])
```

```
  colnames(res) <- c("River", "Theta segregating sites", "Variance of theta")
  theta <- theta.river[match(river,levels(ggar.gt@schemes$river)),2]
  RR <- pegas::R2.test(ggar.bin.neu, B = 1000, theta = theta, plot = FALSE,
                       quiet = TRUE)
  taj <- strataG::tajimasD(ggar.gt.neu, CI = 0.95)
  res <- data.frame(river, fu[[1]], theta, RR[[1]], RR[[2]], taj[,-5])
  colnames(res) <- c("River", "Fu's Fs", "Theta for R2 test", "R2", "R2 p-value",
                     "Tajima's D", "Tajima's p-value", "Taj D lower 95% CI",
                     "Taj D upper 95% CI")
  neutral.river <- rbind(neutral.river, res)
}
knitr::kable(neutral.river, digits = 4,
             caption = "Neutrality tests per river") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

Table 23: Neutrality tests per river

	River	Fu.s.Fs	Theta.for.R2.test	R2	R2.p.value	Tajima.s.D	Tajima.s.p.value	Taj.D.lower.95..CI	Taj.D.upper.95..CI
1	King Sound	-0.9039	0.3145	0.1630	0.4097	0.2170	0.4094	-1.7829	1.9849
11	West Cambridge Gulf	-3.6837	2.7679	0.1132	0.2980	-0.3771	0.3766	-1.7910	1.9836
12	Ord River	-0.7735	0.6151	0.1413	0.3917	-0.3340	0.3920	-1.7910	1.9836
13	Daly River	-1.2446	1.2621	0.0564	0.0070	-1.7307	0.0321	-1.8066	2.0199
14	Adelaide River	-1.9961	1.3242	0.1116	0.4270	-0.4311	0.3544	-1.8071	2.0102
15	Sampan Creek	-1.1515	1.5145	0.1161	0.5536	-0.1740	0.4494	-1.8066	2.0199
16	Wildman River	-1.2452	1.1435	0.0971	0.4314	-0.3766	0.3740	-1.8018	2.0386
17	West Alligator River	-1.6309	0.4674	0.0862	0.2381	-1.0202	0.1624	-1.8034	2.0343
18	South Alligator River	-6.2149	0.7711	0.0540	0.1341	-1.1246	0.1325	-1.7812	2.0735
19	East Alligator River	-2.8953	1.2821	0.0543	0.0790	-1.4322	0.0703	-1.7950	2.0523
110	Papua New Guinea	1.4196	0.8759	0.2658	0.9001	-0.3508	0.4230	-1.4779	1.9992

```
strataG::TiTvRatio(ggar.gt) # No transversions
```

8.3.1.2 Per region

```
##          Ti          Tv Ti.Tv.ratio
##          25           0         Inf
```

```
neutral.pop <- data.frame(Region = character(), `Fu's Fs` = numeric(),
                          `Theta for R2 test` = numeric(), R2 = numeric(),
                          `R2 p-value` = numeric(), `Tajima's D` = numeric(),
                          `Tajima's p-value` = numeric(),
                          `Taj D lower 95% CI` = numeric(),
                          `Taj D upper 95% CI` = numeric())
for (pop in levels(ggar.gt@schemes$pop)) {
  ggar.gt.neu <- ggar.gt[ggar.gt@schemes$pop %in% pop,,,drop = TRUE]
  ind.name <- ggar.gt.neu@data$id
  ggar.bin.neu <- ggar.bin[dimnames(ggar.bin)[[1]] %in% ind.name,]
  if (length(ggar.gt.neu@data$id) > 172) {
    fu <- strataG::fusFs(ggar.gt.neu[sample(x = 1:length(ggar.gt.neu@data$id),
                                            size = 170, replace = FALSE),,,
                                     drop = TRUE])
  } else {
    fu <- strataG::fusFs(ggar.gt.neu)
  }
  seg <- ape::seg.sites(ggar.bin.seg)
  s <- length(seg)
  n <- length(ind.name)
  res <- pegas::theta.s(s, n , variance = TRUE)
  res <- data.frame(pop, res[1], res[2])
  colnames(res) <- c("Region", "Theta segregating sites", "Variance of theta")
  theta <- theta.pop[match(pop,levels(ggar.gt@schemes$pop)),2]
  RR <- pegas::R2.test(ggar.bin.neu, B = 1000, theta = theta, plot = FALSE,
                       quiet = TRUE)
  taj <- strataG::tajimasD(ggar.gt.neu, CI = 0.95)
  res <- data.frame(pop, fu[[1]], theta, RR[[1]], RR[[2]], taj[,-5])
  colnames(res) <- c("Region", "Fu's Fs", "Theta for R2 test",
                     "R2", "R2 p-value",
                     "Tajima's D", "Tajima's p-value",
                     "Taj D lower 95% CI",
                     "Taj D upper 95% CI")
  neutral.pop <- rbind(neutral.pop, res)
}
knitr::kable(neutral.pop, digits = 4,
             caption = "Neutrality tests per region") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

Table 24: Neutrality tests per region

	Region	Fu.s.Fs	Theta.for.R2.test	R2	R2.p.value	Tajima.s.D	Tajima.s.p.value	Taj.D.lower.95..CI	Taj.D.upper.95..CI
1	King Sound	-0.9039	0.3145	0.1630	0.4131	0.2170	0.4094	-1.7829	1.9849
11	Cambridge Gulf	-4.6777	2.5242	0.0718	0.0730	-1.1728	0.1253	-1.8066	2.0199
12	Daly River	-1.2446	1.2621	0.0564	0.0060	-1.7307	0.0321	-1.8066	2.0199
13	Van Diemen Gulf	-9.0540	1.7490	0.0309	0.0340	-1.5790	0.0433	-1.7482	2.1139
14	Papua New Guinea	1.4196	0.8759	0.2658	0.9091	-0.3508	0.4230	-1.4779	1.9992

8.3.2 Subset per gene

```
ggar.gi <- ggar.gi[order(match(indNames(ggar.gi), ggar.ml@labels)),]
river.all <- as.character(ggar.gi$other$ind.meta$River[order(
  match(indNames(ggar.gi), ggar.ml@labels))])
pop.all <- as.character(ggar.gi$other$ind.meta$pop[order(
  match(indNames(ggar.gi), ggar.ml@labels))])
gene.names <- apex::getLocusNames(ggar.ml)

neutral.pop <- data.frame(
  Region = character(),
  gene = character(),
  gene.length = numeric(),
  `Fu's Fs` = numeric(),
  `Theta for R2 test` = numeric(),
  R2 = numeric(),
  `R2 p-value` = numeric(),
  `Tajima's D` = numeric(),
  `Tajima's p-value` = numeric(),
  `Taj D lower 95% CI` = numeric(),
  `Taj D upper 95% CI` = numeric()
)

for (pop in levels(ggar.gt@schemes$pop)) {
    ggar.gt.pop <- ggar.gt[ggar.gt@schemes$pop %in% pop,,,drop = TRUE]
    ggar.gt.pop@schemes <- ggar.gt.pop@schemes[ggar.gt.pop@schemes$pop %in% pop,]
  for (i in 1:length(gene.names)) {
    print(ggar.ml@gene.info[i,])
    ggar.bin.neu <- ggar.ml@dna[[i]]
    ggar.bin.neu <- ggar.bin.neu[dimnames(ggar.bin.neu)[[1]] %in%
                                   ggar.gt.pop@data$id, , drop = TRUE]

    river.neu <- ggar.gt.pop@schemes$river[order(match(ggar.gt.pop@schemes$id,
                                                       dimnames(ggar.bin.neu)[[1]]))]
    pop.neu <- ggar.gt.pop@schemes$pop[order(match(ggar.gt.pop@schemes$id,
                                                   dimnames(ggar.bin.neu)[[1]]))]

    strata.schemes <- data.frame(pop.neu, river.neu)
    row.names(strata.schemes) <- dimnames(ggar.bin.neu)[[1]]

    ggar.gt.neu <- strataG::sequence2gtypes(ggar.bin.neu, strata = river.neu,
                                            seq.names = gene.names[i],
                                        schemes = strata.schemes,
                                        description = NULL, other = NULL)

    if (length(ggar.gt.neu@data$id) > 172) {
      fu <- strataG::fusFs(ggar.gt.neu[sample(x = 1:length(ggar.gt.neu@data$id),
                                              size = 170, replace = FALSE),,,
                                       drop = TRUE])
    } else {
      fu <- strataG::fusFs(ggar.gt.neu)
    }
    seg <- ape::seg.sites(ggar.bin.neu)
    s <- length(seg)
    n <- length(dimnames(ggar.bin.neu)[[1]])
    theta <- pegas::theta.s(s, n , variance = TRUE)
    theta <- data.frame(pop, theta[1], theta[2])
    colnames(theta) <- c("Region", "Theta segregating sites", "Variance of theta")
```

```
    if (theta[,2] > 0){
      RR <- pegas::R2.test(ggar.bin.neu, B = 1000, theta = theta[,2], plot = FALSE,
                           quiet = TRUE)
      taj <- strataG::tajimasD(ggar.gt.neu, CI = 0.95)
    } else {
      RR <- list(R2 = NaN, P.val = NaN)
      taj <- data.frame(D = NaN, p.value = NaN, LCI = NaN,UCI = NaN)
      }

    res <- data.frame(pop, gene.names[i], ggar.ml@gene.info[i,2],
                      fu[[1]], theta[2], RR[[1]], RR[[2]], taj[,-5])
    colnames(res) <- c("Region", "gene", "gene.length", "Fu's Fs",
                       "Theta for R2 test", "R2", "R2 p-value",
                       "Tajima's D", "Tajima's p-value", "Taj D lower 95% CI",
                       "Taj D upper 95% CI")
    neutral.pop <- rbind(neutral.pop, res)
  }
    print(neutral.pop[neutral.pop$Region == pop,])
}

save(neutral.pop, file = "Neutrality_per_gene.Rdata")
```

```
print(load("Neutrality_per_gene.Rdata"))
```

```
## [1] "neutral.pop"
```

```
kableExtra::kbl(neutral.pop, digits = 4, longtable = TRUE,
             caption = "Neutrality tests per region per gene") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            font_size = 7) %>%
  kableExtra::landscape()
```

Table 25: Neutrality tests per region per gene

	Region	gene	gene.length	Fu.s.Fs	Theta.for.R2.test	R2	R2.p.value	Tajima.s.D	Tajima.s.p.value	Taj.D.lower.95..CI	Taj.D.upper.95..CI
1	King Sound	tRNA1	69	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
11	King Sound	12SrRNA	953	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
12	King Sound	tRNA2	72	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
13	King Sound	16SrRNA	1668	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
14	King Sound	tRNA3	75	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
15	King Sound	ND1	975	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
16	King Sound	tRNA4	212	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
17	King Sound	ND2	1045	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
18	King Sound	tRNA5	388	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
19	King Sound	COI	1558	-0.2071	0.6289	0.1693	0.6246	-0.2006	0.4411	-1.7829	1.9849
110	King Sound	tRNA6	144	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
111	King Sound	CO2	698	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
112	King Sound	tRNA7	74	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
113	King Sound	ATP8	169	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
114	King Sound	ATP6	674	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
115	King Sound	CO3	785	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
116	King Sound	tRNA8	72	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
117	King Sound	ND3	349	1.1394	0.3145	0.2473	0.8402	1.2122	0.1276	-1.7829	1.9849
118	King Sound	tRNA9	70	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
119	King Sound	ND4	1671	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
120	King Sound	tRNA10	208	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
121	King Sound	ND5	1830	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
122	King Sound	ND6	517	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
123	King Sound	tRNA11	70	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
124	King Sound	CytB	1147	0.1857	0.3145	0.1319	0.1964	-0.3414	0.3903	-1.7829	1.9849
125	King Sound	tRNA12	143	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
126	King Sound	CR	1069	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
127	Cambridge Gulf	tRNA1	69	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
128	Cambridge Gulf	12SrRNA	953	-1.2114	0.2524	0.1795	0.7346	-1.1470	0.1315	-1.8066	2.0199
129	Cambridge Gulf	tRNA2	72	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
130	Cambridge Gulf	16SrRNA	1668	0.6649	0.2524	0.1437	0.5711	0.2157	0.4064	-1.8066	2.0199
131	Cambridge Gulf	tRNA3	75	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
132	Cambridge Gulf	ND1	975	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
133	Cambridge Gulf	tRNA4	212	0.6649	0.2524	0.1437	0.5503	0.2157	0.4064	-1.8066	2.0199
134	Cambridge Gulf	ND2	1045	-0.5944	0.5048	0.1150	0.4148	-0.6119	0.2900	-1.8066	2.0199
135	Cambridge Gulf	tRNA5	388	NaN	0.2524	0.1826	0.8323	-1.5587	0.0530	-1.8066	2.0199
136	Cambridge Gulf	COI	1558	-1.5064	0.7573	0.0862	0.1613	-0.9888	0.1724	-1.8066	2.0199
137	Cambridge Gulf	tRNA6	144	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
138	Cambridge Gulf	CO2	698	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
139	Cambridge Gulf	tRNA7	74	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
140	Cambridge Gulf	ATP8	169	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
141	Cambridge Gulf	ATP6	674	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
142	Cambridge Gulf	CO3	785	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
143	Cambridge Gulf	tRNA8	72	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN

144	Cambridge Gulf	ND3	349	1.0799	0.2524	0.1851	0.8262	0.7267	0.2399	-1.8066	2.0199
145	Cambridge Gulf	tRNA9	70	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
146	Cambridge Gulf	ND4	1671	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
147	Cambridge Gulf	tRNA10	208	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
148	Cambridge Gulf	ND5	1830	0.6649	0.2524	0.1437	0.5316	0.2157	0.4064	-1.8066	2.0199
149	Cambridge Gulf	ND6	517	-1.2114	0.2524	0.1795	0.7308	-1.1470	0.1315	-1.8066	2.0199
150	Cambridge Gulf	tRNA11	70	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
151	Cambridge Gulf	CytB	1147	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
152	Cambridge Gulf	tRNA12	143	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
153	Cambridge Gulf	CR	1069	-2.3549	0.5048	0.1247	0.5057	-1.5074	0.0605	-1.8066	2.0199
154	Daly River	tRNA1	69	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
155	Daly River	12SrRNA	953	-0.4393	0.2524	0.0644	0.0711	-0.7637	0.2396	-1.8066	2.0199
156	Daly River	tRNA2	72	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
157	Daly River	16SrRNA	1668	-0.4393	0.2524	0.0644	0.0599	-0.7637	0.2396	-1.8066	2.0199
158	Daly River	tRNA3	75	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
159	Daly River	ND1	975	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
160	Daly River	tRNA4	212	-0.4393	0.2524	0.0644	0.0746	-0.7637	0.2396	-1.8066	2.0199
161	Daly River	ND2	1045	-0.4393	0.2524	0.0644	0.0625	-0.7637	0.2396	-1.8066	2.0199
162	Daly River	tRNA5	388	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
163	Daly River	COI	1558	-0.4393	0.2524	0.0644	0.0424	-0.7637	0.2396	-1.8066	2.0199
164	Daly River	tRNA6	144	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
165	Daly River	CO2	698	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
166	Daly River	tRNA7	74	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
167	Daly River	ATP8	169	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
168	Daly River	ATP6	674	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
169	Daly River	CO3	785	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
170	Daly River	tRNA8	72	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
171	Daly River	ND3	349	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
172	Daly River	tRNA9	70	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
173	Daly River	ND4	1671	0.3880	0.2524	0.1195	0.3684	-0.0824	0.4837	-1.8066	2.0199
174	Daly River	tRNA10	208	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
175	Daly River	ND5	1830	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
176	Daly River	ND6	517	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
177	Daly River	tRNA11	70	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
178	Daly River	CytB	1147	-1.6686	0.5048	0.0953	0.2552	-1.2555	0.1068	-1.8066	2.0199
179	Daly River	tRNA12	143	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
180	Daly River	CR	1069	-3.6260	0.5048	0.0898	0.1594	-1.7779	0.0276	-1.8066	2.0199
181	Van Diemen Gulf	tRNA1	69	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
182	Van Diemen Gulf	12SrRNA	953	-1.7301	0.4770	0.0222	0.0421	-1.0898	0.1388	-1.7482	2.1139
183	Van Diemen Gulf	tRNA2	72	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
184	Van Diemen Gulf	16SrRNA	1668	-4.0776	0.3180	0.0082	0.0103	-1.1523	0.1229	-1.7482	2.1139
185	Van Diemen Gulf	tRNA3	75	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
186	Van Diemen Gulf	ND1	975	NaN	0.1590	0.0574	0.4151	-0.9096	0.1901	-1.7482	2.1139
187	Van Diemen Gulf	tRNA4	212	-1.9492	0.1590	0.0098	0.0625	-0.8316	0.2147	-1.7482	2.1139
188	Van Diemen Gulf	ND2	1045	-4.8443	0.3180	0.0082	0.0133	-1.1523	0.1229	-1.7482	2.1139
189	Van Diemen Gulf	tRNA5	388	NaN	0.1590	0.0000	0.0000	-0.9490	0.1782	-1.7482	2.1139
190	Van Diemen Gulf	COI	1558	-0.8850	0.3180	0.0391	0.2227	-0.6535	0.2755	-1.7482	2.1139

191	Van Diemen Gulf	tRNA6	144	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
192	Van Diemen Gulf	CO2	698	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
193	Van Diemen Gulf	tRNA7	74	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
194	Van Diemen Gulf	ATP8	169	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
195	Van Diemen Gulf	ATP6	674	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
196	Van Diemen Gulf	CO3	785	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
197	Van Diemen Gulf	tRNA8	72	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
198	Van Diemen Gulf	ND3	349	1.7472	0.1590	0.1709	0.8952	1.0915	0.1440	-1.7482	2.1139
199	Van Diemen Gulf	tRNA9	70	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1100	Van Diemen Gulf	ND4	1671	-1.9492	0.1590	0.0066	0.0280	-0.8705	0.2023	-1.7482	2.1139
1101	Van Diemen Gulf	tRNA10	208	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1102	Van Diemen Gulf	ND5	1830	-2.6684	0.1590	0.0574	0.3949	-0.9096	0.1901	-1.7482	2.1139
1103	Van Diemen Gulf	ND6	517	-2.6684	0.1590	0.0574	0.3796	-0.9096	0.1901	-1.7482	2.1139
1104	Van Diemen Gulf	tRNA11	70	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1105	Van Diemen Gulf	CytB	1147	1.1618	0.4770	0.0890	0.6476	0.1614	0.4140	-1.7482	2.1139
1106	Van Diemen Gulf	tRNA12	143	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1107	Van Diemen Gulf	CR	1069	-6.1902	0.3180	0.0065	0.0000	-1.1793	0.1164	-1.7482	2.1139
1108	Papua New Guinea	tRNA1	69	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1109	Papua New Guinea	12SrRNA	953	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1110	Papua New Guinea	tRNA2	72	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1111	Papua New Guinea	16SrRNA	1668	-0.0027	0.4380	0.3727	0.8224	-0.9330	0.2201	-1.4779	1.9992
1112	Papua New Guinea	tRNA3	75	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1113	Papua New Guinea	ND1	975	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1114	Papua New Guinea	tRNA4	212	-0.0027	0.4380	0.3727	0.8103	-0.9330	0.2201	-1.4779	1.9992
1115	Papua New Guinea	ND2	1045	-0.0027	0.4380	0.3727	0.8142	-0.9330	0.2201	-1.4779	1.9992
1116	Papua New Guinea	tRNA5	388	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1117	Papua New Guinea	COI	1558	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1118	Papua New Guinea	tRNA6	144	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1119	Papua New Guinea	CO2	698	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1120	Papua New Guinea	tRNA7	74	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1121	Papua New Guinea	ATP8	169	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1122	Papua New Guinea	ATP6	674	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1123	Papua New Guinea	CO3	785	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1124	Papua New Guinea	tRNA8	72	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1125	Papua New Guinea	ND3	349	0.7952	0.4380	0.3000	0.8172	1.4451	0.1031	-1.4779	1.9992
1126	Papua New Guinea	tRNA9	70	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1127	Papua New Guinea	ND4	1671	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1128	Papua New Guinea	tRNA10	208	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1129	Papua New Guinea	ND5	1830	-0.0027	0.4380	0.3727	0.8135	-0.9330	0.2201	-1.4779	1.9992
1130	Papua New Guinea	ND6	517	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1131	Papua New Guinea	tRNA11	70	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1132	Papua New Guinea	CytB	1147	0.6254	0.4380	0.2667	0.7914	0.8506	0.2278	-1.4779	1.9992
1133	Papua New Guinea	tRNA12	143	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN
1134	Papua New Guinea	CR	1069	NaN	0.0000	NaN	NaN	NaN	NaN	NaN	NaN

8.4 Phylogenetic analysis

8.4.1 Metadata with outgroup

```
f.in <- "../3.Data/6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis.fasta"
ggar.ml <- apex::read.multiFASTA(f.in)

metafile <- "../3.Data/Glyphis garricki_Metadata2_ALL_10-05-2019.csv"
meta <- readr::read_csv(metafile)
```

```
## New names:
## Rows: 571 Columns: 34
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (24): SubsetID, id, pop, River, sample location, sex, tl_mm, date, Sampl... dbl
## (10): txtlat, txtlon, lat, lon, SampleLat, SampleLon, Conc. (ng/ul), 260...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `Mitogenome selected` -> `Mitogenome selected...28`
## * `Mitogenome selected` -> `Mitogenome selected...29`
```

```
meta <- meta[meta$id %in% ggar.ml@labels,]

pop.levels1 <- c("King Sound", "West Cambridge Gulf", "Ord River",
                 "Daly River", "Adelaide River", "Sampan Creek", "Wildman River",
                 "West Alligator River", "South Alligator River",
                 "East Alligator River", "Papua New Guinea"
                 ,"G. glyphis"
                 )

meta$River <- factor(x = meta$River, levels = pop.levels1)

table(meta$River)
```

```
##
## King Sound West Cambridge Gulf Ord River
## 14 15 15
## Daly River Adelaide River Sampan Creek
## 30 25 30
## Wildman River West Alligator River South Alligator River
## 45 41 101
## East Alligator River Papua New Guinea G. glyphis
## 61 6 1
```

```
pop.levels2 <- c("King Sound", "Cambridge Gulf", "Daly River",
                 "Van Diemen Gulf", "Papua New Guinea"
                 ,"G. glyphis"
                 )
meta$pop <- factor(x = meta$pop, levels = pop.levels2)
table(meta$pop)
```

```
##
## King Sound Cambridge Gulf Daly River Van Diemen Gulf
## 14 30 30 303
## Papua New Guinea G. glyphis
## 6 1
```

8.4.2 Input files by gene

```r
# f.in <- "../3.Data/6.ggar_Nucleotide_alignment_edited_HTPsites.fasta"
f.in <- "../3.Data/6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis.fasta"

ggar.ml <- apex::read.multiFASTA(f.in)
ggar.bin.phy <- ape::read.dna(f.in, format = "fasta", as.matrix = TRUE)

genes <- list(`tRNA1` = ggar.bin.phy[,1:69],
              `12SrRNA` = ggar.bin.phy[,70:1022],
              `tRNA2` = ggar.bin.phy[,1023:1094],
              `16SrRNA` = ggar.bin.phy[,1094:2761],
              `tRNA3` = ggar.bin.phy[,2762:2836],
              `ND1` = ggar.bin.phy[,2837:3811],
              `tRNA4` = ggar.bin.phy[,3812:4023],
              `ND2` = ggar.bin.phy[,4024:5068],
              `tRNA5` = ggar.bin.phy[,5069:5456],
              `COI` = ggar.bin.phy[,5457:7014],
              `tRNA6` = ggar.bin.phy[,7015:7158],
              `CO2` = ggar.bin.phy[,7159:7856],
              `tRNA7` = ggar.bin.phy[,7857:7930],
              `ATP8` = ggar.bin.phy[,7931:8099],
              `ATP6` = ggar.bin.phy[,8100:8773],
              `CO3` = ggar.bin.phy[,8774:9558],
              `tRNA8` = ggar.bin.phy[,9559:9630],
              `ND3` = ggar.bin.phy[,9631:9979],
              `tRNA9` = ggar.bin.phy[,9980:10049],
              `ND4` = ggar.bin.phy[,10050:11720],
              `tRNA10` = ggar.bin.phy[,11721:11928],
              `ND5` = ggar.bin.phy[,11929:13758],
              `ND6` = ggar.bin.phy[,13759:14275],
              `tRNA11` = ggar.bin.phy[,14276:14345],
              `CytB` = ggar.bin.phy[,14346:15492],
              `tRNA12` = ggar.bin.phy[,15493:15635],
              `CR` = ggar.bin.phy[,15636:16707]
              )
ggar.ml <- new("multidna", genes)
ggar.aln <- multidna2alignment(ggar.ml, genes = TRUE)
ggar.phy <- multidna2multiphyDat(ggar.ml)
ggar.aln2 <- multiphyDat2alignment(ggar.phy, genes = TRUE)



genes <- getLocusNames(ggar.ml)
for (i in 1:length(genes)) {
  ape::write.dna(ggar.ml@dna[[i]], file = paste0("Ggar_with_outgroup_",genes[i],".fasta"),
                 format = "fasta", append = FALSE, colsep = "", colw = 100, indent = NULL,blocksep =
  fsta <- phylotools::read.fasta(file =  paste0("Ggar_with_outgroup_",genes[i],".fasta"), clean_name
  phylotools::dat2phylip(fsta, outfile = paste0("Ggar_with_outgroup_",genes[i],".phy"))
}

f.in <- "../3.Data/6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis.fasta"
fsta <- phylotools::read.fasta(file =  f.in, clean_name = FALSE)
phylotools::dat2phylip(fsta, outfile = "../3.Data/6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyp

f.in <- "../3.Data/6.ggar_Nucleotide_alignment_edited_HTPsites.fasta"
fsta <- phylotools::read.fasta(file = f.in, clean_name = FALSE)
phylotools::dat2phylip(fsta, outfile = "../3.Data/6.ggar_Nucleotide_alignment_edited_HTPsites.phy")
```

8.4.3 Metadata without outgroup

```
f.in <- "../3.Data/6.ggar_Nucleotide_alignment_edited_HTPsites.fas"
ggar.ml <- apex::read.multiFASTA(f.in)

metafile <- "../3.Data/Glyphis garricki_Metadata2_ALL_10-05-2019.csv"
meta <- readr::read_csv(metafile)
```

```
## New names:
## Rows: 571 Columns: 34
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (24): SubsetID, id, pop, River, sample location, sex, tl_mm, date, Sampl... dbl
## (10): txtlat, txtlon, lat, lon, SampleLat, SampleLon, Conc. (ng/ul), 260...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `Mitogenome selected` -> `Mitogenome selected...28`
## * `Mitogenome selected` -> `Mitogenome selected...29`
```

```
meta <- meta[meta$id %in% ggar.ml@labels,]

pop.levels1 <- c("King Sound", "West Cambridge Gulf", "Ord River",
                 "Daly River", "Adelaide River", "Sampan Creek", "Wildman River",
                 "West Alligator River", "South Alligator River",
                 "East Alligator River", "Papua New Guinea"
                 )

meta$River <- factor(x = meta$River, levels = pop.levels1)

table(meta$River)
```

```
##
## King Sound West Cambridge Gulf Ord River
## 14 15 15
## Daly River Adelaide River Sampan Creek
## 30 25 30
## Wildman River West Alligator River South Alligator River
## 45 41 101
## East Alligator River Papua New Guinea
## 61 6
```

```
pop.levels2 <- c("King Sound", "Cambridge Gulf", "Daly River",
                 "Van Diemen Gulf", "Papua New Guinea"
                 )
meta$pop <- factor(x = meta$pop, levels = pop.levels2)
table(meta$pop)
```

```
##
## King Sound Cambridge Gulf Daly River Van Diemen Gulf
## 14 30 30 303
## Papua New Guinea
## 6
```

8.4.4 IQtree - dated phylogeny

We used the estimated time of divergence between *Glyphis glyphis* and *Glyphis garricki* of 15 Mya as the root date (Brée et al. 2022).

8.4.4.1 Running IQtree

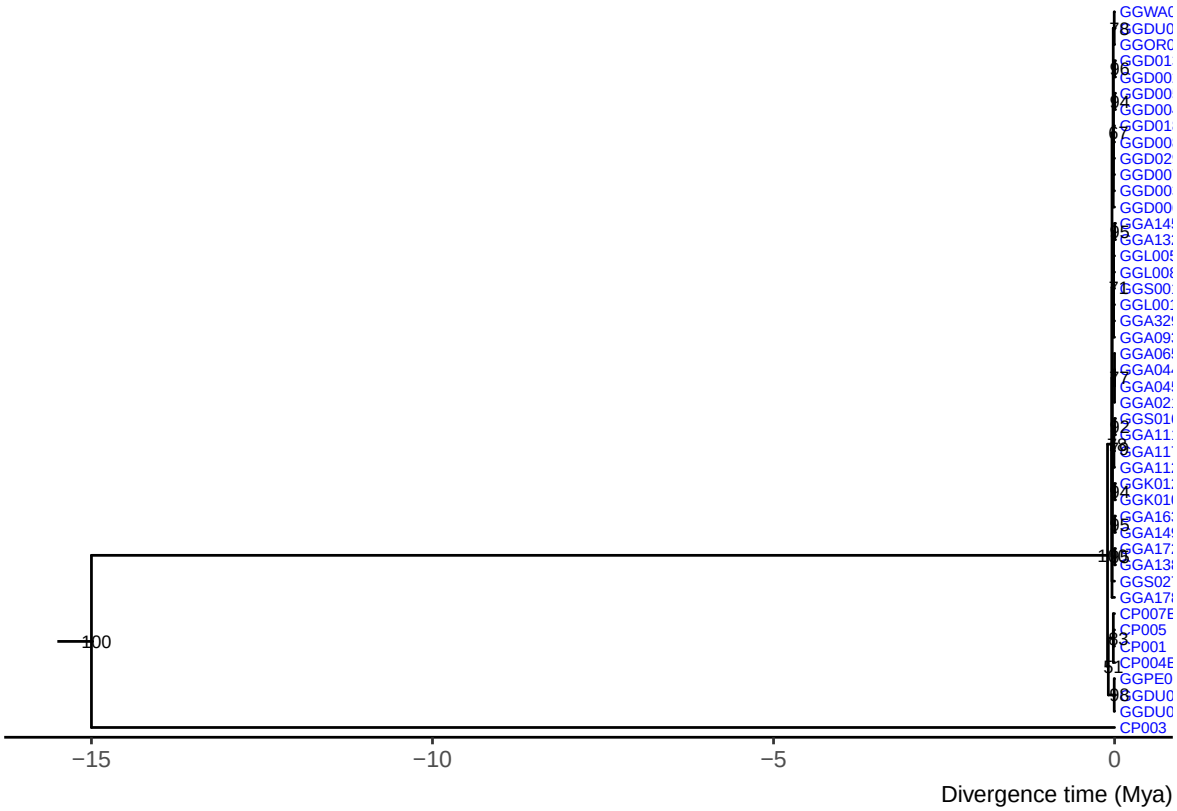
Identical sequences are ignored.

```
#!/bin/bash
#SBATCH --time=1:0:0
#SBATCH --nodes=1
#SBATCH --mem-per-cpu=1GB
#SBATCH --ntasks-per-node=1
#SBATCH --account=000000

start=$(date)
echo $start
SECONDS=0


module load iqtree/2.2.0.5
cd /datasets/work/ncmi-toa-rrbs/work/BEAST/Mitogenomes_Ggar/



PREFIX=iqtree_results_no_outgroup
ALIGNMENT=6.ggar_Nucleotide_alignment_edited_HTPsites.phy
FOLDER=/datasets/work/ncmi-toa-rrbs/work/BEAST/Mitogenomes_Ggar/${PREFIX}
mkdir $FOLDER
rsync -r ${ALIGNMENT} ${FOLDER}/${ALIGNMENT}
cd ${FOLDER}

iqtree -s ${ALIGNMENT} --seqtype DNA -T 1 -m MF
iqtree -s ${ALIGNMENT} --seqtype DNA -T 1 -m HKY+F+I --ufboot 10000 -bnni -redo #without outgroup
### HKY+F+I was best model according to BIC



PREFIX=iqtree_results_with_outgroup
ALIGNMENT=6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis.phy
FOLDER=/datasets/work/ncmi-toa-rrbs/work/BEAST/Mitogenomes_Ggar/${PREFIX}
mkdir $FOLDER
rsync -r ${ALIGNMENT} ${FOLDER}/${ALIGNMENT}
cd ${FOLDER}

iqtree -s ${ALIGNMENT} --seqtype DNA -T 1 -m MF
iqtree -s ${ALIGNMENT} --seqtype DNA -T 1 -m TN+F+I -o CP003 --ufboot 10000 -bnni -redo #with outgro
### TN+F+I was best model according to BIC



# PREFIX=iqtree_results_with_outgroup_dated
# FOLDER=/datasets/work/ncmi-toa-rrbs/work/BEAST/Mitogenomes_Ggar/${PREFIX}
# ALIGNMENT=6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis.phy.uniqueseq.phy
# mkdir $FOLDER
# rsync -r ${ALIGNMENT} ${FOLDER}/${ALIGNMENT}
# cd ${FOLDER}
# iqtree -s ${ALIGNMENT} --date-root -15 --date-tip 0 --date-ci 100 --seqtype DNA -T 1 -o CP003 -m T
```

```
duration=$SECONDS
echo "$(($duration / 3600)) hours, $((($duration / 60) % 60)) minutes and $(($duration % 60)) second
```

```
ggar.nex <- "iqtree_results2/6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis_uniqueseq.phy.tim
ggar.tree <- ape::read.nexus(ggar.nex)

boot_trees <- ape::read.tree("iqtree_results2/6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis_
boot_support <- ape::prop.clades(ggar.tree, boot_trees)
boot_support <- boot_support/100
boot_support <- round(boot_support, 0)
nodes <- paste0("t",1:18)

ggar.tree$root.edge <- 0.5

p1 <- ggtree::ggtree(tr = ggar.tree, color = "black")
p1$data$label[p1$data$isTip == FALSE] <- nodes
bs <- data.frame(label = p1$data$label, bootstrap = c(rep(NA, 45),boot_support))
bs$bootstrap[is.na(bs$bootstrap)] <- 100
p1 <- p1 %<+% bs

p1 <- p1 +
  ggtree::geom_tiplab(size = 2, color = 'blue') +
  ggtree::geom_rootedge() +
  ggtree::geom_nodelab(aes(label = bootstrap
                           ), size = 2.5,
                       ) +
  ggtree::theme_tree2() +
  labs(caption = "Divergence time (Mya)")
p2 <- ggtree::revts(p1)
print(p2)
```

8.4.4.2 IQtree results with outgroup

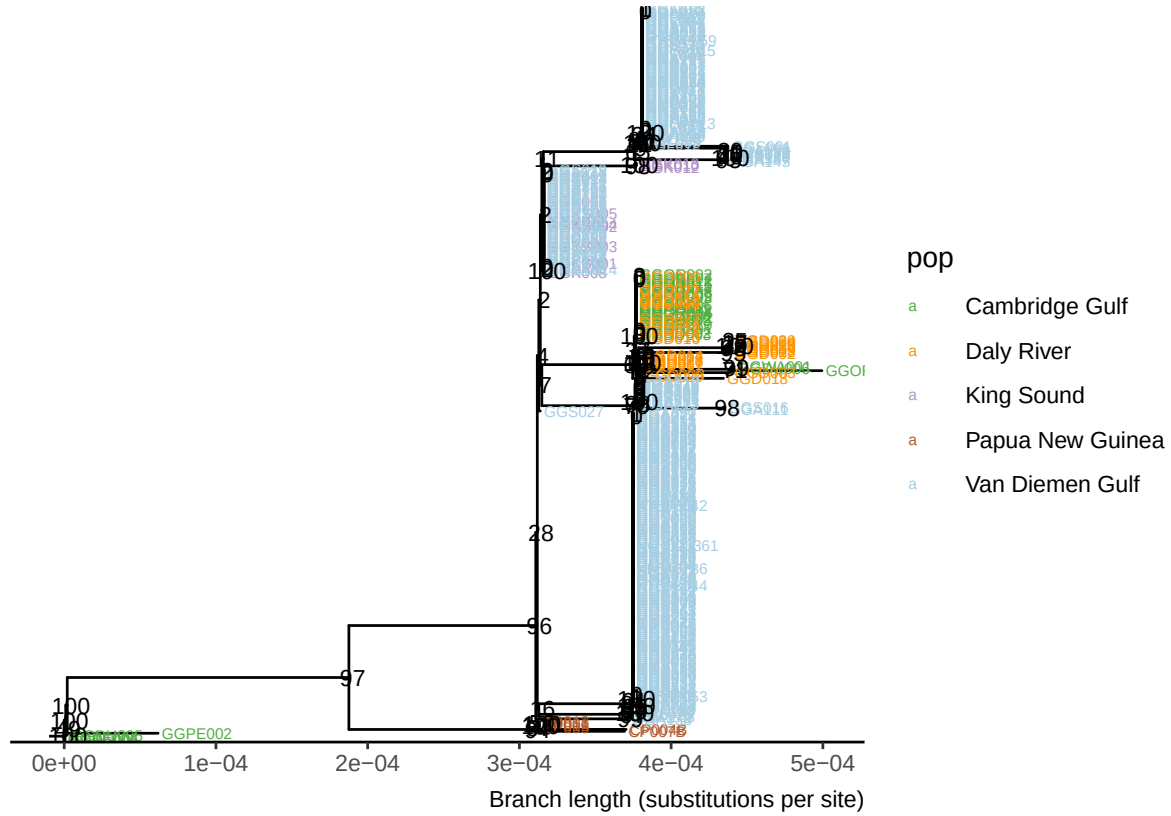



```
ggplot2::ggsave(plot = p2, filename = "IQTREE_outgroup_timetree_regular_ggtree.png",
                device = "png", width = 30, height = 15, units = "cm")
```

```
ggar.nex <- "iqtree_results_no_outgroup/6.ggar_Nucleotide_alignment_edited_HTPsites.phy.treefile"
ggar.tree <- ape::read.tree(ggar.nex)

boot_trees <- ape::read.tree("iqtree_results_no_outgroup/6.ggar_Nucleotide_alignment_edited_HTPsites
boot_support <- ape::prop.clades(ggar.tree, boot_trees)
boot_support <- boot_support/100
boot_support <- round(boot_support, 0)
nodes <- paste0("t",1:381)

ggar.tree$root.edge <- 1e-5

p1 <- ggtree::ggtree(tr = ggar.tree, color = "black")
p1$data$label[p1$data$isTip == FALSE] <- nodes
pop <- as.character(meta$pop[meta$id %in% p1$data$label[p1$data$isTip == TRUE]])
id <- as.character(meta$id[meta$id %in% p1$data$label[p1$data$isTip == TRUE]])
pop <- pop[order(match(id,p1$data$label[p1$data$isTip == TRUE]))]
p1$data$pop <- NA
p1$data$pop[p1$data$isTip == TRUE] <- pop
bs <- data.frame(label = p1$data$label, bootstrap = c(rep(NA, 383),boot_support))
bs$bootstrap[is.na(bs$bootstrap)] <- NA
p1 <- p1 %<+% bs

p1 <- p1 +
  ggtree::geom_tiplab(ggplot2::aes(color = pop),size = 2) +
  ggplot2::scale_color_manual(
    values = colours.5
    # values = colours.11
```

```
    ) +
  ggtree::geom_rootedge() +
  ggtree::geom_nodelab(ggplot2::aes(label = bootstrap
                               ), size = 3,
                       ) +
  ggtree::theme_tree2() +
  ggplot2::labs(caption = "Branch length (substitutions per site)")
print(p1)
```

8.4.4.3 IQtree results without outgroup




```
ggplot2::ggsave(plot = p1, filename = "IQTREE_no_outgroup_regular_ggtree.png",
                device = "png", width = 30, height = 70, units = "cm")
```

8.4.5 BEAST

```
beast2_folder = beastier::get_default_beast2_folder()
mauricer::get_beast2_pkg_names(beast2_folder = beastier::get_default_beast2_folder())

if (mauricer::is_beast2_pkg_installed(name = "NS", beast2_folder = beast2_folder)) {
  stop(
    "Cannot install installed package '",
    name,
    "'at '",
    beast2_folder,
    "'.",
    "Tip: this means the package is already installed :-)"
  )
}
show_warnings = TRUE
verbose = TRUE
result <- beastier::is_beast2_input_file(
  filename = "C:/Users/dev093/AppData/Local/R/win-library/4.2/mauricer/extdata/nested_sampling.xml",
  # mauricer::get_mrc_path("nested_sampling.xml"),
  show_warnings = show_warnings,
  verbose = verbose,
  beast2_path = "C:/Users/dev093/AppData/Local/BEAST/BEAST2.exe"
  # beastier::get_default_beast2_bin_path(beast2_folder = beast2_folder)
)

df <- mauricer::get_beast2_pkg_names(beast2_folder = beast2_folder)
df[df$name == name, ]$installed_version != "NA"

babette::create_test_ns_output()
```

8.4.6 BEAST - divergence time analysis - with outgroup

We used the estimated time of divergence between *Glyphis glyphis* and *Glyphis garricki* of 15 Mya as the root date (Brée et al. 2022). The mutation rate was based on the Silky shark (0.62% substitution per site per Ma) from Galván-Tirado et al. (2013) and the references therein (scalloped hammerhead shark: 0.43%, blacktip shark: 0.67%, and nurse shark: 0.57%). The generation time of 27 years was based on Feutry et al. (2020).

```
phy_filename <- "6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis_uniqueseq.phy"
fasta_filename <- "6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis_uniqueseq.fasta"
pop_1 <- "ALL_outgroup_CPP"

phy.dat <- phylotools::read.phylip(infile = phy_filename, clean_name = TRUE)
phylotools::dat2fasta(dat = phy.dat, outfile = fasta_filename)
```

8.4.6.1 Create BEAUTI file

```
##
6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis_uniqueseq.fasta
has been saved to C:/Users/dev093/OneDrive - University of
Tasmania/PhD/7.SBD_garricki/4.Results
```

```
inference_model <-
  beautier::create_inference_model(
    site_model = beautier::create_site_model_hky( #best scheme according to IQTREE = TN+F+I
      gamma_site_model = beautier::create_gamma_site_model(
        gamma_cat_count = "0",
```

```
        gamma_shape = "1.0",
        prop_invariant = "0.759", # from IQtree
        gamma_shape_prior_distr = NA,
        freq_equilibrium = "estimated",
        freq_prior_uniform_distr_id = 1000
      ),
      freq_equilibrium = "estimated"
    ),
    # tree_prior = beautier::create_yule_tree_prior(
      # birth_rate_distr = beautier::create_uniform_distr(value = 0.2, lower = 0, upper = 15)
    #log(n/2)/t = 0.20: t = expected root height (15), n = number of taxa(45).or t= 0.1/u per site p
    #log(0.2) = -1.61
    # ),
    ## TRIED YULE, BUT BAD MODEL ESS https://beast2-dev.github.io/hmc/hmc/Priors/YuleBirthRatePrior/
     tree_prior = beautier::create_tree_prior_ccp(#Coalescent constant population size tree prior fo
      pop_size_distr = beautier::create_distr_log_normal(m = 6.91, s = 2.35,
                                                         value = 6.91, lower = 2.30)
     # pop_size_distr = beautier::create_distr_uniform(value = 1000, lower = 10, upper = 50000)
    ),
    clock_model = beautier::create_strict_clock_model(
      clock_rate_param = beautier::create_clock_rate_param(
        value = 6.2e-3, #0.62% per site per million years (Galván-Tirado et al 2013)
        estimate = TRUE,
      ),
      # clock_rate_distr = beautier::create_distr_log_normal(m = -18.88, s = 1.175, value = -18.88,
      #                                                      lower = 1e-20, upper = 1),
      clock_rate_distr = beautier::create_distr_log_normal(m = -5.083, s = 1.175, value = -5.083,
                                                           lower = 0, upper = 1),
      rate_scaler_factor = 1
    ),
    mrca_prior = beautier::create_mrca_prior( # which tips share a common ancerstor
      name = "Gglyphis_root",
      is_monophyletic = TRUE,
      mrca_distr = beautier::create_normal_distr( # to set the crown age
        mean = 15, # 15 Ma: based on Bree et al. 2022
        value = 15.8,
        sigma = 3)
      ), #hist(rnorm(1000, mean = 15, sd = 3), breaks = 40)
      mcmc = beautier::create_mcmc(
        chain_length = 1e+07,
        store_every = 5e+03,
        pre_burnin = 1e+04,
        n_init_attempts = 3,
        tracelog = beautier::create_tracelog(filename =
                                               paste0("tracelog_", pop_1, ".log")),
        screenlog = beautier::create_screenlog(filename =
                                                 paste0("screenlog_", pop_1, ".log")),
        treelog = beautier::create_treelog(filename =
                                             paste0("treelog_", pop_1, ".log"))
      ),
      beauti_options = beautier::create_beauti_options_v2_6()
    )


beast2_input_file <- paste0("beast2_",pop_1,".xml")
beautier::create_beast2_input_file_from_model(
  input_filename = fasta_filename,
  inference_model = inference_model,
```

```
  output_filename = beast2_input_file
)

print(head(readLines(beast2_input_file)))
```

```
## [1] "<?xml version=\"1.0\" encoding=\"UTF-8\"
standalone=\"no\"?><beast beautitemplate='Standard'
beautistatus=''
namespace=\"beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:bea
required=\"\" version=\"2.6\">"
## [2] " "
## [3] " <data"
## [4]
"id=\"6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis_uniqueseq\""
## [5] "spec=\"Alignment\""
## [6] "name=\"alignment\">"
```

```
if (beastier::is_beast2_installed()) {
  beastier::is_beast2_input_file(beast2_input_file,
                                 show_warnings = FALSE,
                                 verbose = FALSE)
}
```

```
## [1] FALSE
```

```
if (beastier::is_beast2_installed()) {
  beast2_options <- beastier::create_beast2_options(
    input_filename = beast2_input_file,
    output_state_filename = paste0("beast2_",pop_1,".xml.state"),
    n_threads = 4,
    verbose = TRUE
  )
  beastier::check_can_create_file(beast2_options$output_state_filename)
  beastier::check_can_create_treelog_file(beast2_options)
  run_beast2_from_options(
    beast2_options = beast2_options
  )
  testthat::expect_true(file.exists(beast2_options$output_state_filename))
}

if (beastier::is_beast2_installed()) {
  out <- babette::bbt_run_from_model(
    fasta_filename = fasta_filename,
    inference_model = inference_model,
    beast2_options = beast2_options)
}
```

8.4.6.2 Run BEAST

```
tracelog <- "beast_divergence_outgroup_CPP/tracelog_ALL_outgroup_CPP.log"
treelog <- "beast_divergence_outgroup_CPP/treelog_ALL_outgroup_CPP.log"
output_state <- "beast_divergence_outgroup_CPP/beast2_ALL_outgroup_CPP.xml.state"
pop_1 <- "ALL_outgroup_CPP"


estimates <- tracerer::parse_beast_tracelog_file(
  tracelog_filename = tracelog
```

```
)
estimates_burn <- tracerer::remove_burn_ins(estimates, burn_in_fraction = 0.1)
esses <- tracerer::calc_esses(estimates_burn, sample_interval = 5000)
table <- t(esses)
colnames(table) <- c("ESS")
knitr::kable(table)%>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")
```

8.4.6.3 Summary stats

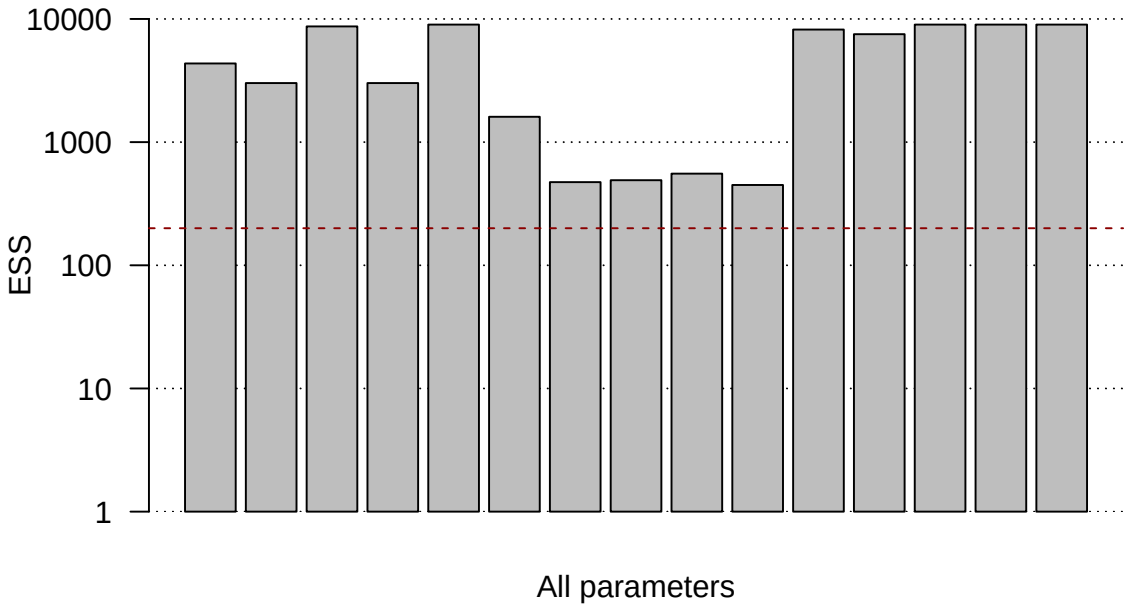
```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

	ESS
posterior	4253
likelihood	2998
prior	8097
treeLikelihood.6.ggar_Nucleotide_alignment_edited_HTPsites2_Gglyphis_uniqueseq	2998
TreeHeight	9001
kappa	1439
freqParameter.1	508
freqParameter.2	466
freqParameter.3	532
freqParameter.4	474
popSize	7983
CoalescentConstant	7080
logP.mrca.Gglyphis_root..	9001
mrca.age.Gglyphis_root.	9001
clockRate	9001

```
bdsky_trace   <- beastio::readLog(tracelog, burnin = 0.1)
beastio::checkESS(bdsky_trace)
```

```
## named numeric(0)
```

```
beastio::checkESS(bdsky_trace, cutoff = 200, plot = TRUE, log = 'y',
                  ylim = c(1,10000), title = "All parameters", plot.grid = TRUE)
```




```
## named numeric(0)
```

```
Posterior <- tracerer::calc_summary_stats(
  estimates_burn$posterior,
  sample_interval = 5000
)
table <- t(Posterior)
colnames(table) <- c("Posterior")
knitr::kable(table)
```

	Posterior
mean	-25505.59
stderr_mean	0.1165292
stdev	7.59985
variance	57.75772
median	-25505.32
mode	n/a
geom_mean	n/a
hpd_interval_low	-25520.49
hpd_interval_high	-25491.27
act	10582.02
ess	4252.967

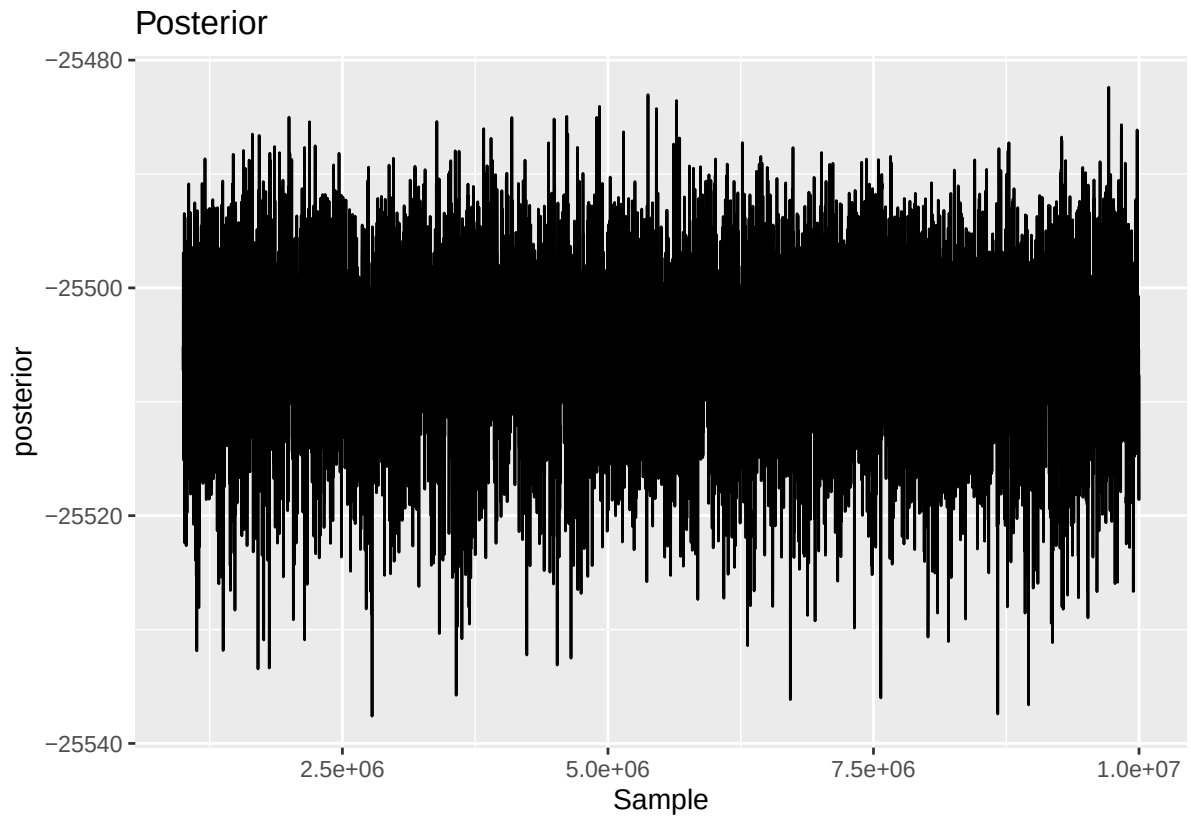
```
sum_stats <- tracerer::calc_summary_stats(
  estimates_burn,
  sample_interval = 5000
)
knitr::kable(sum_stats) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
```

```
  kableExtra::column_spec(1, width = "5cm")  %>%
  kableExtra::landscape()
```

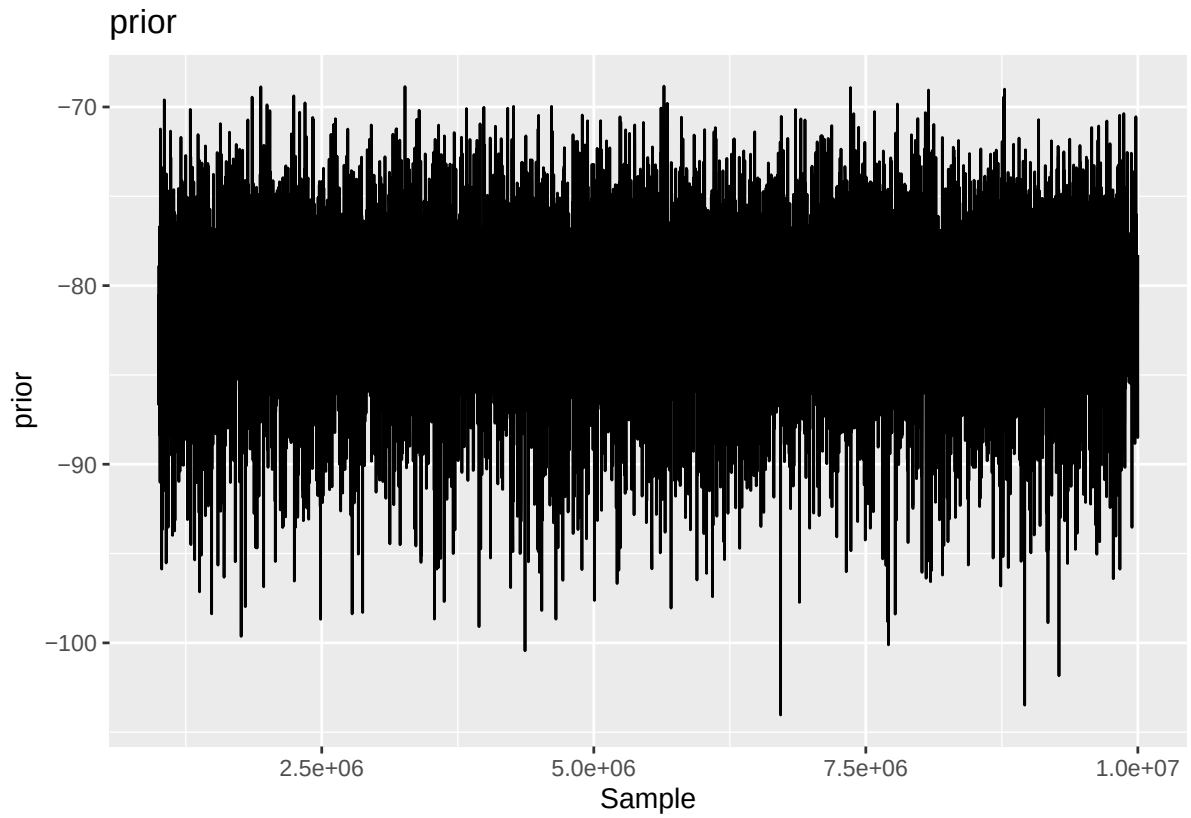
```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

	mean	stderr_mean	stdev	variance	median	mode	geom_mean	hpd_interval_lo
posterior	-2.550559e+04	0.1165292	7.5998499	57.7577186	-2.550532e+04	n/a	n/a	-2.552049e+0
likelihood	-2.542430e+04	0.0940909	5.1524997	26.5482534	-2.542399e+04	n/a	n/a	-2.543497e+0
prior	-8.129531e+01	0.0533938	4.8047388	23.0855153	-8.093884e+01	n/a	n/a	-9.091074e+0
treeLikelihood.6.ggar_Nucleotide_alignment_edited_HTD040909Gglyphis_unique6cp	-2.542430e+04	0.0940909	5.1524997	26.5482534	-2.542399e+04	n/a	n/a	-2.543497e+0
TreeHeight	2.232668e+01	0.0243981	2.3148644	5.3585974	2.226949e+01	n/a	22.2061508892689	1.785056e+0
kappa	3.469371e+01	0.1471201	5.5817647	31.1560970	3.392084e+01	36.1494715626347	34.2591441548738	2.482309e+0
freqParameter.1	3.124879e-01	0.0001526	0.0034379	0.0000118	3.125741e-01	0.313393387045323	0.31246897492753	3.064862e-0
freqParameter.2	2.639570e-01	0.0001547	0.0033399	0.0000112	2.638701e-01	0.262414595162655	0.263935877086023	2.571665e-0
freqParameter.3	1.263295e-01	0.0001071	0.0024714	0.0000061	1.261868e-01	0.124782469481515	0.126305384135099	1.212732e-0
freqParameter.4	2.972256e-01	0.0001629	0.0035444	0.0000126	2.970470e-01	0.29823445312944	0.297204421344917	2.906817e-0
popSize	2.426912e+00	0.0014243	0.1272634	0.0161960	2.388583e+00	n/a	2.42376847580225	2.300027e+0
CoalescentConstant	-6.331396e+01	0.0386814	3.2550086	10.5950809	-6.317711e+01	n/a	n/a	-6.988494e+0
logP.mrca.Gglyphis_root..	-5.297450e+00	0.0207524	1.9689690	3.8768388	-4.953414e+00	n/a	n/a	-9.028493e+0
mrca.age.Gglyphis_root.	2.232668e+01	0.0243981	2.3148644	5.3585974	2.226949e+01	n/a	22.2061508892689	1.785056e+0
clockRate	9.456000e-04	0.0000011	0.0001082	0.0000000	9.375000e-04	n/a	0.000939517568010223	7.345000e-0

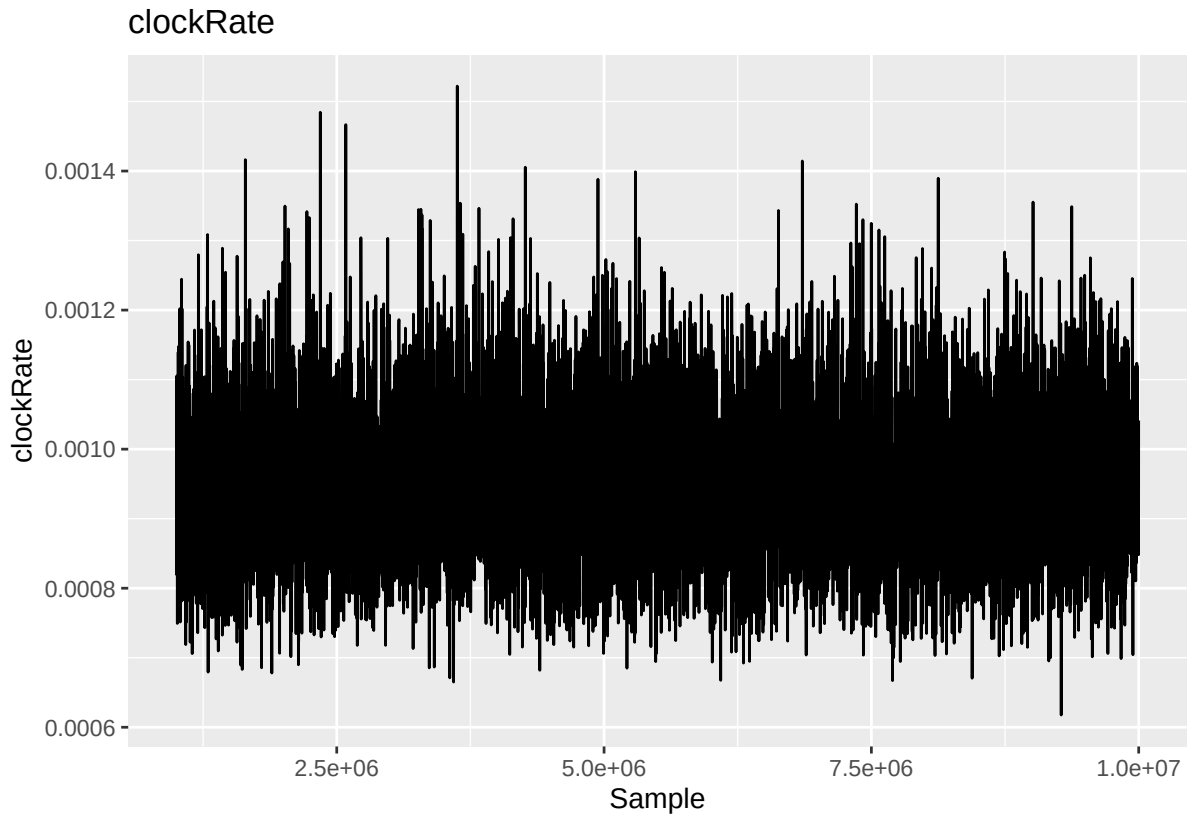
```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = posterior)) +
  ggplot2::ggtitle("Posterior")
```



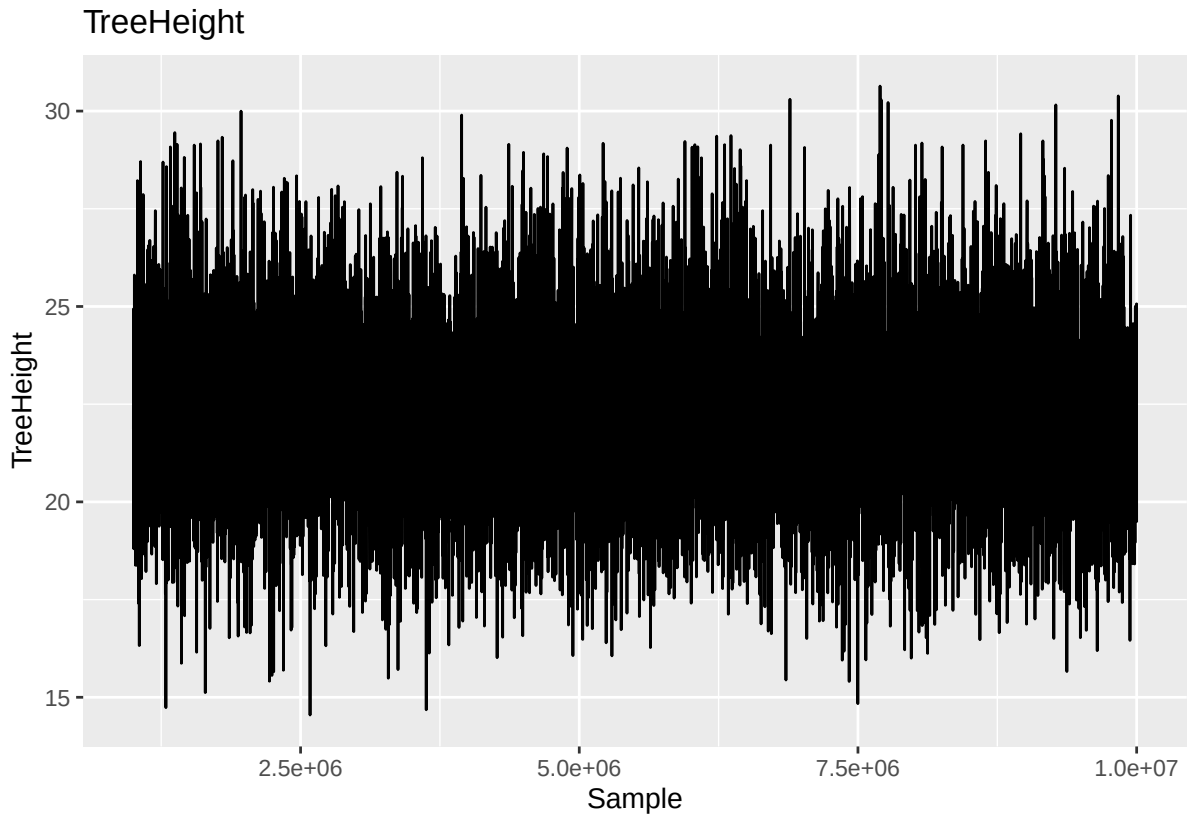

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = prior)) +
  ggplot2::ggtitle("prior")
```



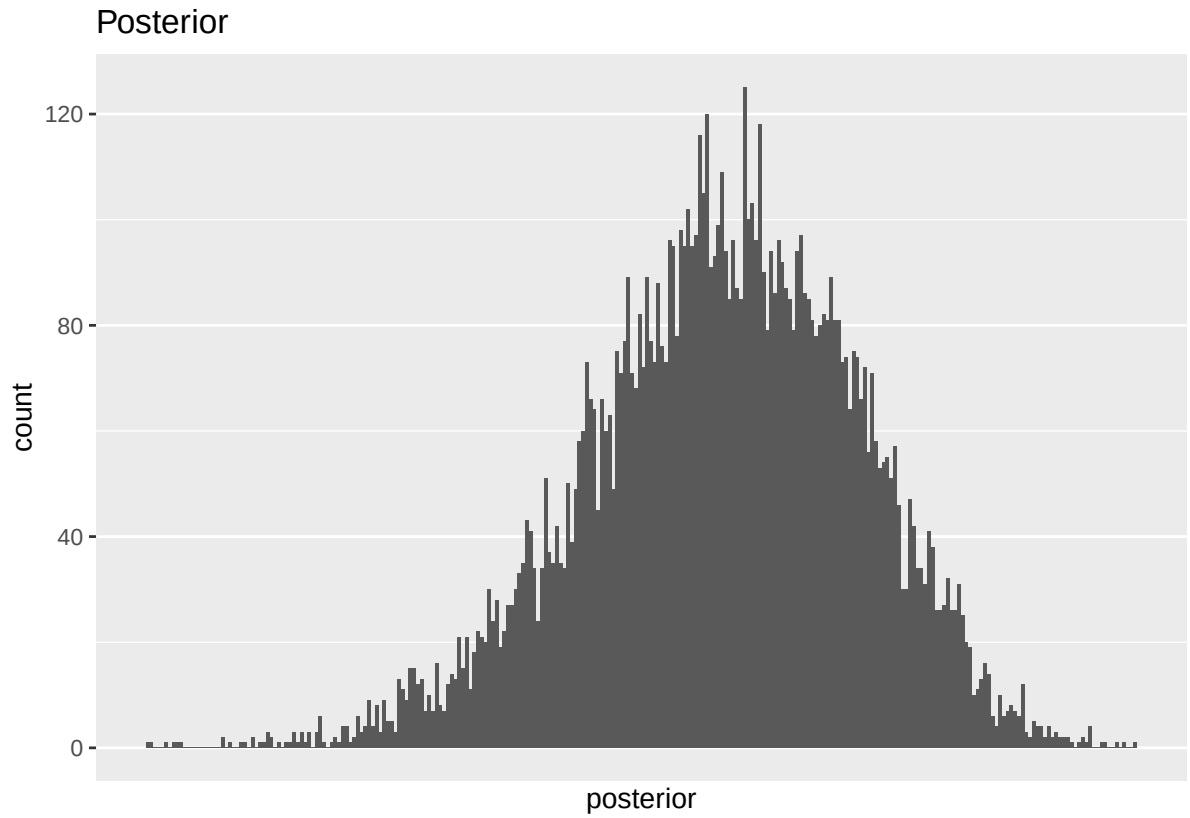

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = clockRate)) +
  ggplot2::ggtitle("clockRate")
```




```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = TreeHeight)) +
  ggplot2::ggtitle("TreeHeight")
```

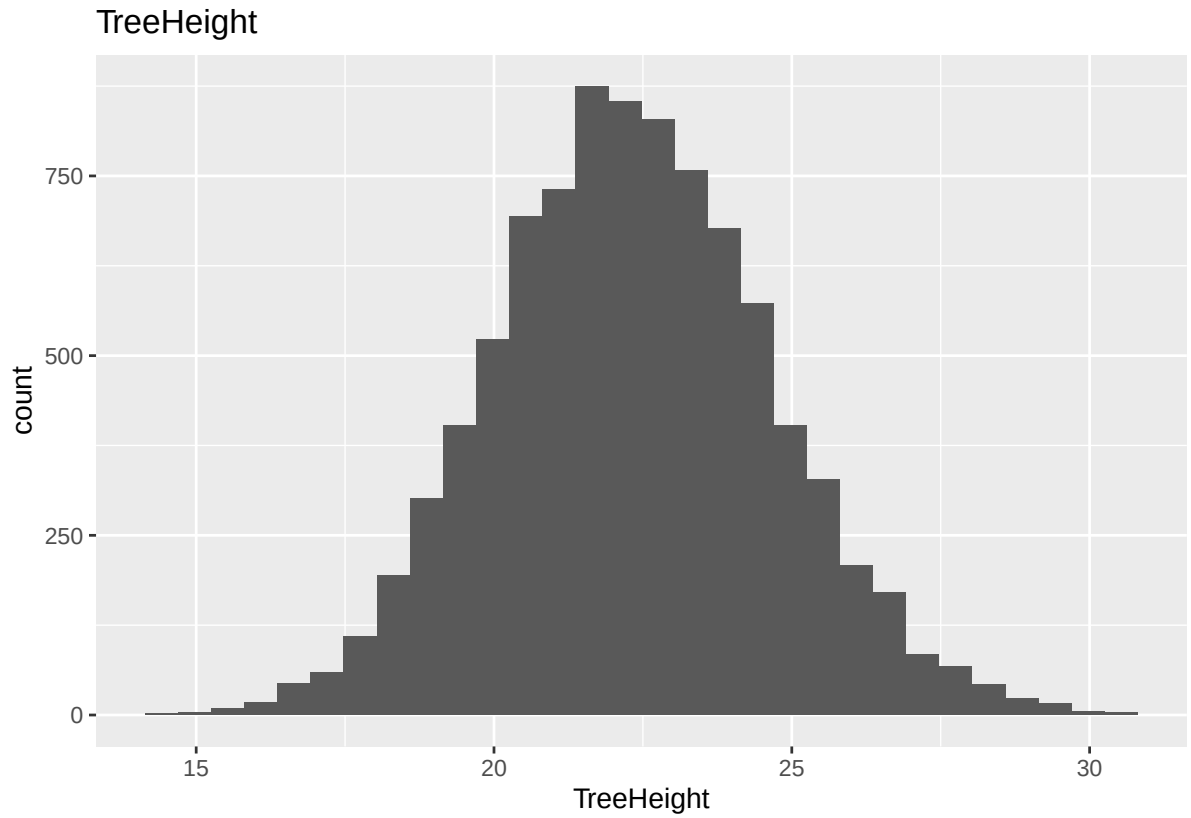



```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(posterior)) +
  ggplot2::geom_histogram(binwidth = 0.21) +
  ggplot2::scale_x_continuous(breaks = seq(-75, -68)) +
  ggplot2::ggtitle("Posterior")
```



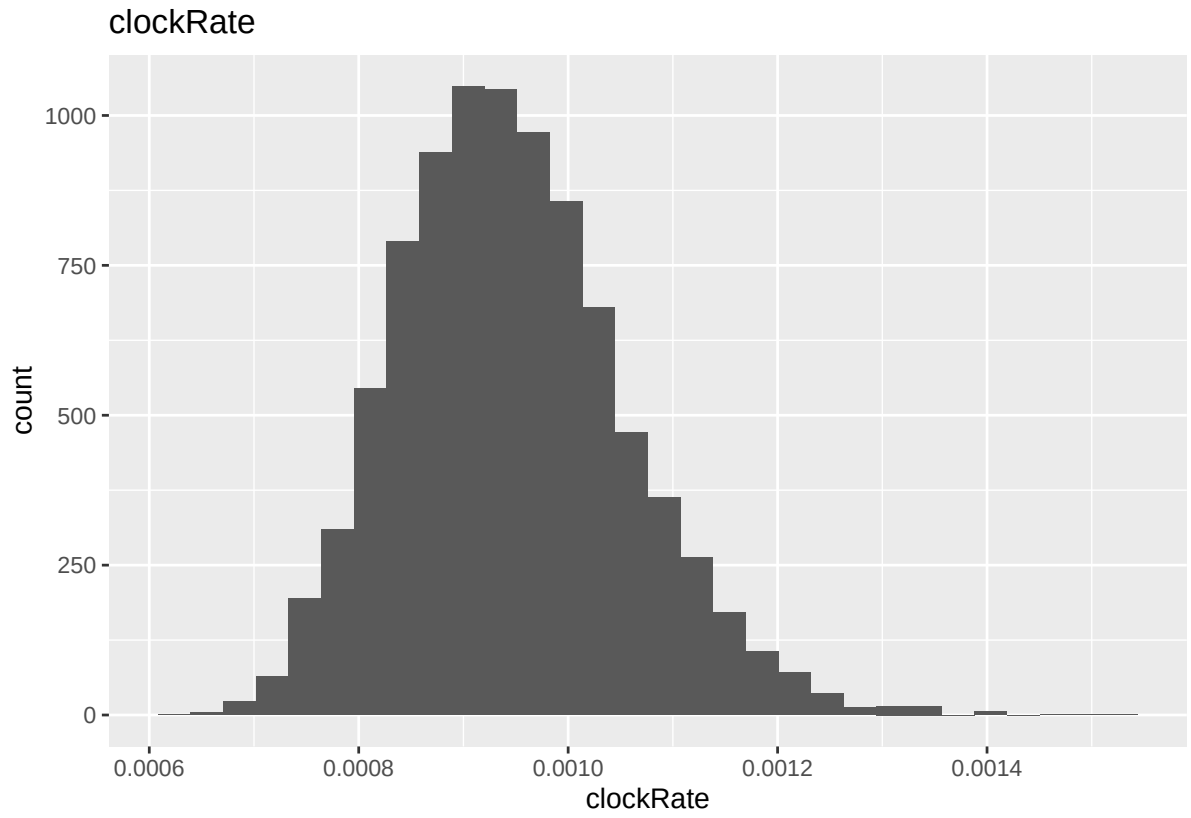

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(TreeHeight)) +
  ggplot2::geom_histogram() +
  ggplot2::ggtitle("TreeHeight")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

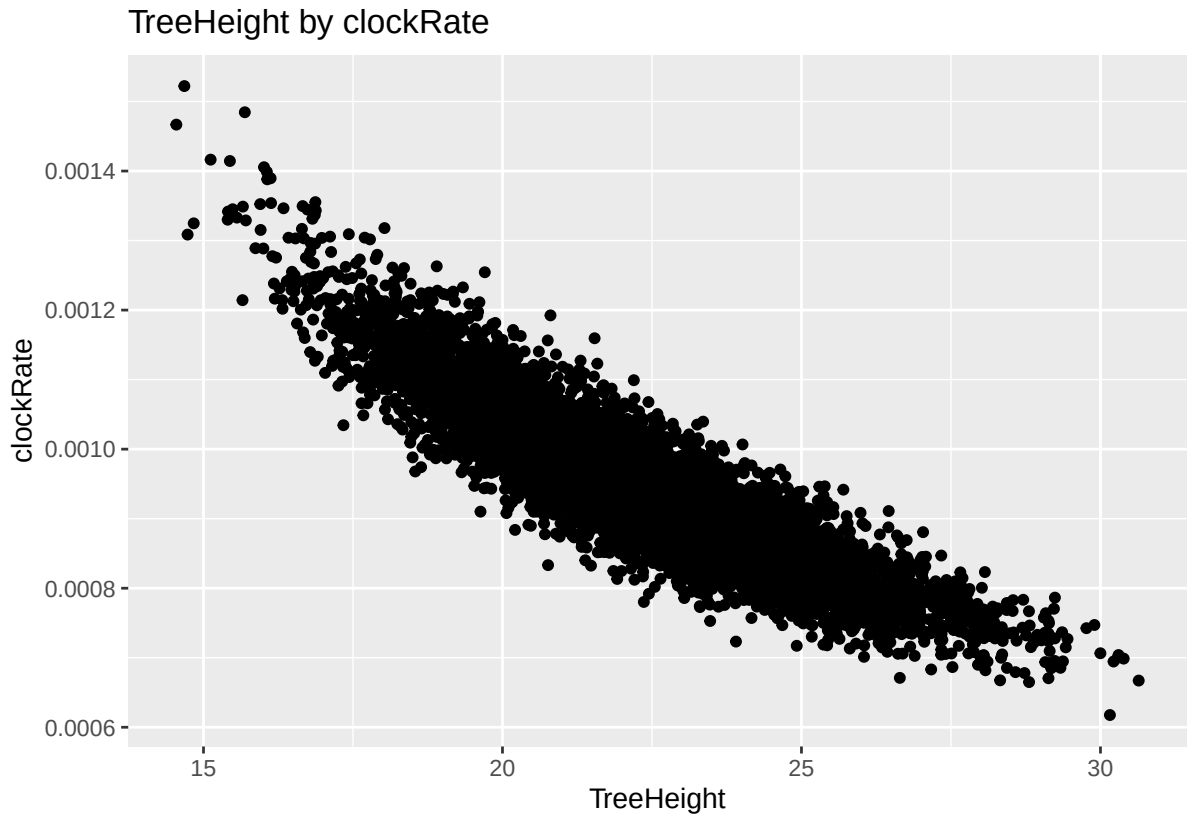



```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(clockRate)) +
  ggplot2::geom_histogram() +
  ggplot2::ggtitle("clockRate")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

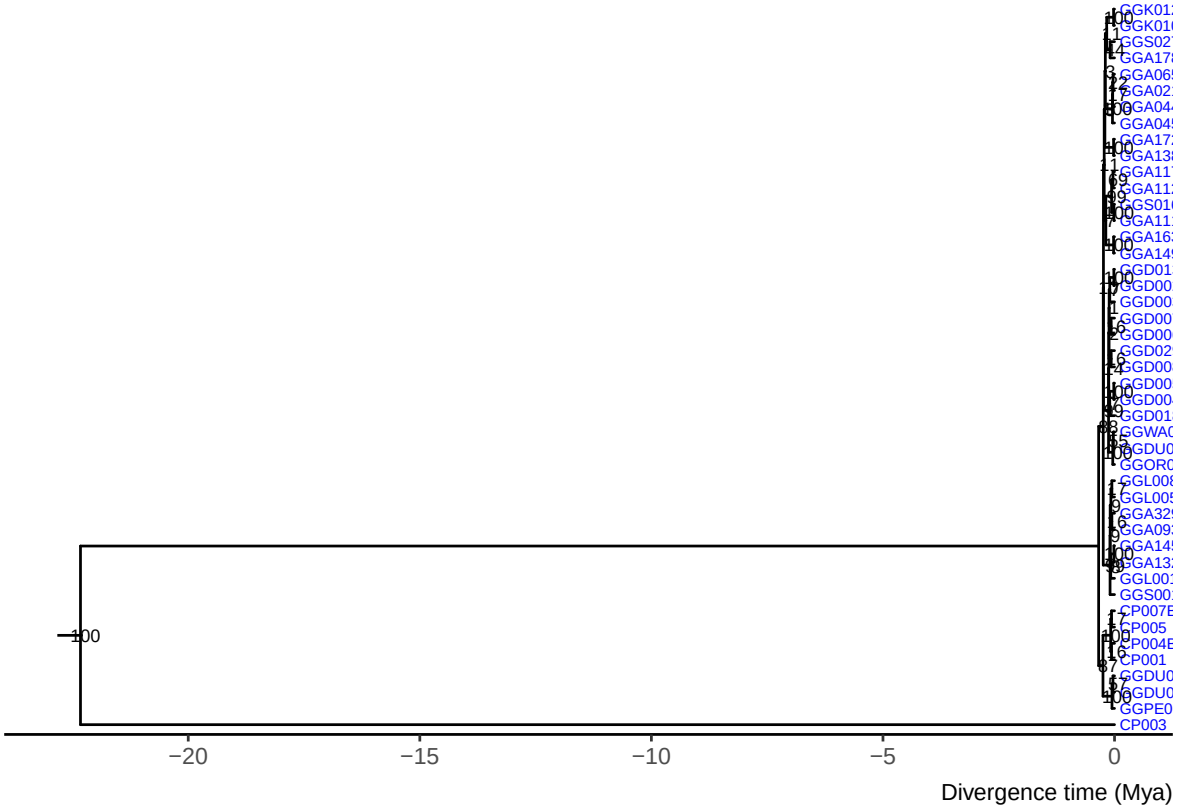



```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = TreeHeight, y = clockRate)) +
  ggplot2::geom_point() +
  ggplot2::ggtitle("TreeHeight by clockRate")
```




```
trees <- tracerer::parse_beast_trees(
  filename = treelog
)
# tracerer::save_beast_trees(trees, filename = treefile)
treefile <- "beast_divergence_outgroup_CPP/Result_trees.tre" #with treeAnnotator
# phyloch::treeannotator(input = treelog, )

ggar.tree <- ape::read.nexus(treefile)
boot_support <- ape::prop.clades(ggar.tree, trees)
boot_support <- boot_support/100
boot_support <- round(boot_support, 0)
nodes <- paste0("t",1:44)

ggar.tree$root.edge <- 0.5

p1 <- ggtree::ggtree(tr = ggar.tree, color = "black")
p1$data$label[p1$data$isTip == FALSE] <- nodes
bs <- data.frame(label = p1$data$label, bootstrap = c(rep(NA, 45),boot_support))
# bs$bootstrap[is.na(bs$bootstrap)] <- 100
p1 <- p1 %<+% bs

p1 <- p1 +
  ggtree::geom_tiplab(size = 2, color = 'blue') +
  ggtree::geom_rootedge() +
  ggtree::geom_nodelab(aes(label = bootstrap), size = 2.5) +
  ggtree::theme_tree2() +
  labs(caption = "Divergence time (Mya)")
p2 <- ggtree::revts(p1)
print(p2)
```

8.4.6.4 Plot Tree




```
ggplot2::ggsave(plot = p2, filename = "BEAST_timetree_outgroup_ggtree.png",
                device = "png", width = 30, height = 15, units = "cm")
```

8.4.7 BEAST - divergence time analysis - without outgroup

```
phy_filename <- "6.ggar_Nucleotide_alignment_edited_HTPsites.phy.uniqueseq.phy"
fasta_filename <- "6.ggar_Nucleotide_alignment_edited_HTPsites.phy.uniqueseq.fasta"
pop_1 <- "ALL_NoOutgroup_CPP"

phy.dat <- phylotools::read.phylip(infile = phy_filename, clean_name = TRUE)
phylotools::dat2fasta(dat = phy.dat, outfile = fasta_filename)
```

8.4.7.1 Create BEAUTI file

```
##
6.ggar_Nucleotide_alignment_edited_HTPsites.phy.uniqueseq.fasta
has been saved to C:/Users/dev093/OneDrive - University of
Tasmania/PhD/7.SBD_garricki/4.Results
```

```
inference_model <-
  beautier::create_inference_model(
    site_model = beautier::create_site_model_hky( #best scheme according to IQTREE = TN+F+I
      gamma_site_model = beautier::create_gamma_site_model(
        gamma_cat_count = "0",
        gamma_shape = "1.0",
        prop_invariant = "0.9925",
        gamma_shape_prior_distr = NA,
        freq_equilibrium = "empirical",
        freq_prior_uniform_distr_id = 1000
      ),
      freq_equilibrium = "empirical",
    ),
    tree_prior = beautier::create_tree_prior_ccp(#Coalescent constant population size tree prior for
      # pop_size_distr = beautier::create_distr_uniform(value = 1000, lower = 10, upper = 50000)
      pop_size_distr = beautier::create_distr_log_normal(m = 6.91, s = 2.35,
                                                         value = 6.91, lower = 2.30,
                                                         upper = 10.81)
    ),
    # clock_model = beautier::create_strict_clock_model(
    #   clock_rate_param = beautier::create_clock_rate_param(value = 1, estimate = FALSE),
    #   rate_scaler_factor = 1), # branch lengths in the tree are  in substitutions per site
    # clock_model = beautier::create_clock_model_rln(
    #   mean_rate_prior_distr = beautier::create_distr_log_normal(m = 0, s = 0.1, lower = 0, upper =
    #   rate_scaler_factor = 1
    # ),
    clock_model = beautier::create_strict_clock_model(
      clock_rate_param = beautier::create_clock_rate_param(
        value = 6.3e-3, #0.62% per site per million years (Galván-Tirado et al 2013)
        estimate = TRUE,
        id = NA
      ),
    # clock_rate_distr = beautier::create_distr_uniform(value = 6.3e-9, lower = 0, upper = 1),
    # clock_rate_distr = beautier::create_distr_log_normal(m = -18.88, s = 1.175, value = -18.88,
    #                                                      lower = 1e-20, upper = 1),
    clock_rate_distr = beautier::create_distr_log_normal(m = -5.083, s = 1.175, value = -5.083,
                                                         lower = 0, upper = 1),
      rate_scaler_factor = 1
    ),
      mcmc = beautier::create_mcmc(
        chain_length = 1e+07,
        store_every = 5e+03,
        pre_burnin = 1e+04,
```

```
        n_init_attempts = 3,
        tracelog = beautier::create_tracelog(filename =
                                               paste0("tracelog_", pop_1, ".log")),
        screenlog = beautier::create_screenlog(filename =
                                                 paste0("screenlog_", pop_1, ".log")),
        treelog = beautier::create_treelog(filename =
                                             paste0("treelog_", pop_1, ".log"))
      ),
      beauti_options = beautier::create_beauti_options_v2_6()
    )


beast2_input_file <- paste0("beast2_",pop_1,".xml")
beautier::create_beast2_input_file_from_model(
  input_filename = fasta_filename,
  inference_model = inference_model,
  output_filename = beast2_input_file
)

print(head(readLines(beast2_input_file)))
```

```
## [1] "<?xml version=\"1.0\" encoding=\"UTF-8\"
standalone=\"no\"?><beast beautitemplate='Standard'
beautistatus=''
namespace=\"beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:bea
required=\"\" version=\"2.6\">"
## [2] " "
## [3] " <data"
## [4]
"id=\"6.ggar_Nucleotide_alignment_edited_HTPsites.phy.uniqueseq\""
## [5] "spec=\"Alignment\""
## [6] "name=\"alignment\">"
```

```
if (beastier::is_beast2_installed()) {
  beastier::is_beast2_input_file(beast2_input_file,
                                 show_warnings = FALSE,
                                 verbose = FALSE)
}
```

```
## [1] TRUE
```

```
if (beastier::is_beast2_installed()) {
  beast2_options <- beastier::create_beast2_options(
    input_filename = beast2_input_file,
    output_state_filename = paste0("beast2_",pop_1,".xml.state"),
    n_threads = 4,
    verbose = TRUE
  )
  beastier::check_can_create_file(beast2_options$output_state_filename)
  beastier::check_can_create_treelog_file(beast2_options)
  run_beast2_from_options(
    beast2_options = beast2_options
  )
  testthat::expect_true(file.exists(beast2_options$output_state_filename))
}

if (beastier::is_beast2_installed()) {
  out <- babette::bbt_run_from_model(
```

```
    fasta_filename = fasta_filename,
    inference_model = inference_model,
    beast2_options = beast2_options)
}
```

8.4.7.2 Run BEAST

```
tracelog <- "beast_divergence_NoOutgroup_CPP/tracelog_ALL_NoOutgroup_CPP.log"
treelog <- "beast_divergence_NoOutgroup_CPP/treelog_ALL_NoOutgroup_CPP.log"
output_state <- "beast_divergence_NoOutgroup_CPP/beast2_ALL_NoOutgroup_CPP.xml.state"


estimates <- tracerer::parse_beast_tracelog_file(
  tracelog_filename = tracelog
)
estimates_burn <- tracerer::remove_burn_ins(estimates, burn_in_fraction = 0.1)
esses <- tracerer::calc_esses(estimates_burn, sample_interval = 5000)
table <- t(esses)
colnames(table) <- c("ESS")
knitr::kable(table)
```

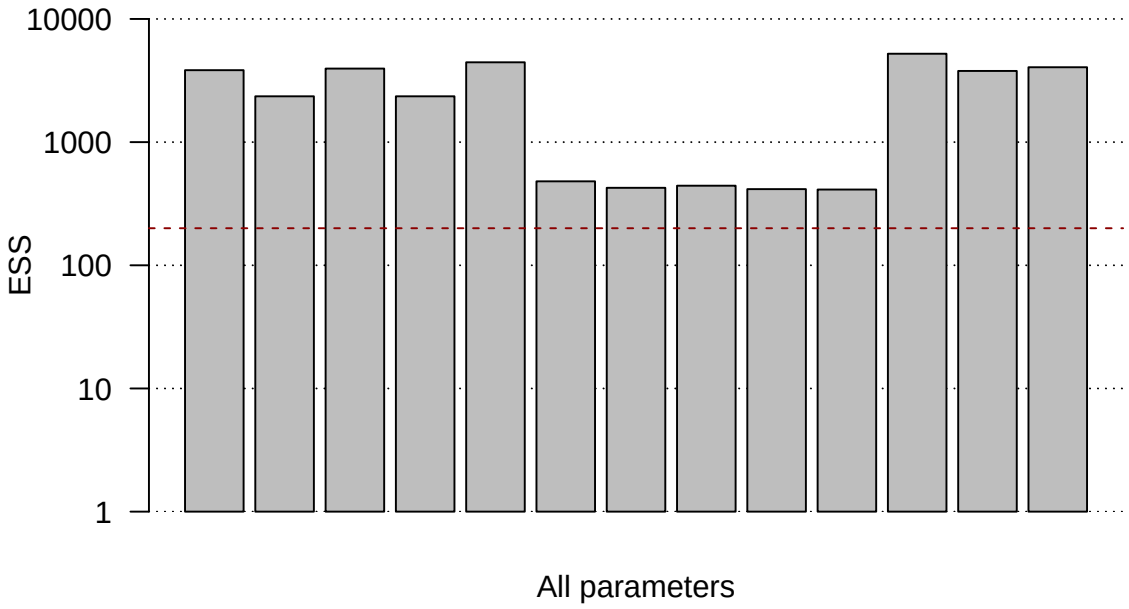
8.4.7.3 Summary stats

	ESS
posterior	3794
likelihood	2030
prior	3852
treeLikelihood	2030
TreeHeight	4498
kappa	554
freqParameter.1	265
freqParameter.2	442
freqParameter.3	438
freqParameter.4	294
clockRate	5017
popSize	3842
CoalescentConstant	4123

```
bdsky_trace   <- beastio::readLog(tracelog, burnin = 0.1)
beastio::checkESS(bdsky_trace)
```

```
## named numeric(0)
```

```
beastio::checkESS(bdsky_trace, cutoff = 200, plot = TRUE, log = 'y',
                  ylim = c(1,10000), title = "All parameters", plot.grid = TRUE)
```




```
## named numeric(0)
```

```
Posterior <- tracerer::calc_summary_stats(
  estimates_burn$posterior,
  sample_interval = 5000
)
table <- t(Posterior)
colnames(table) <- c("Posterior")
knitr::kable(table)
```

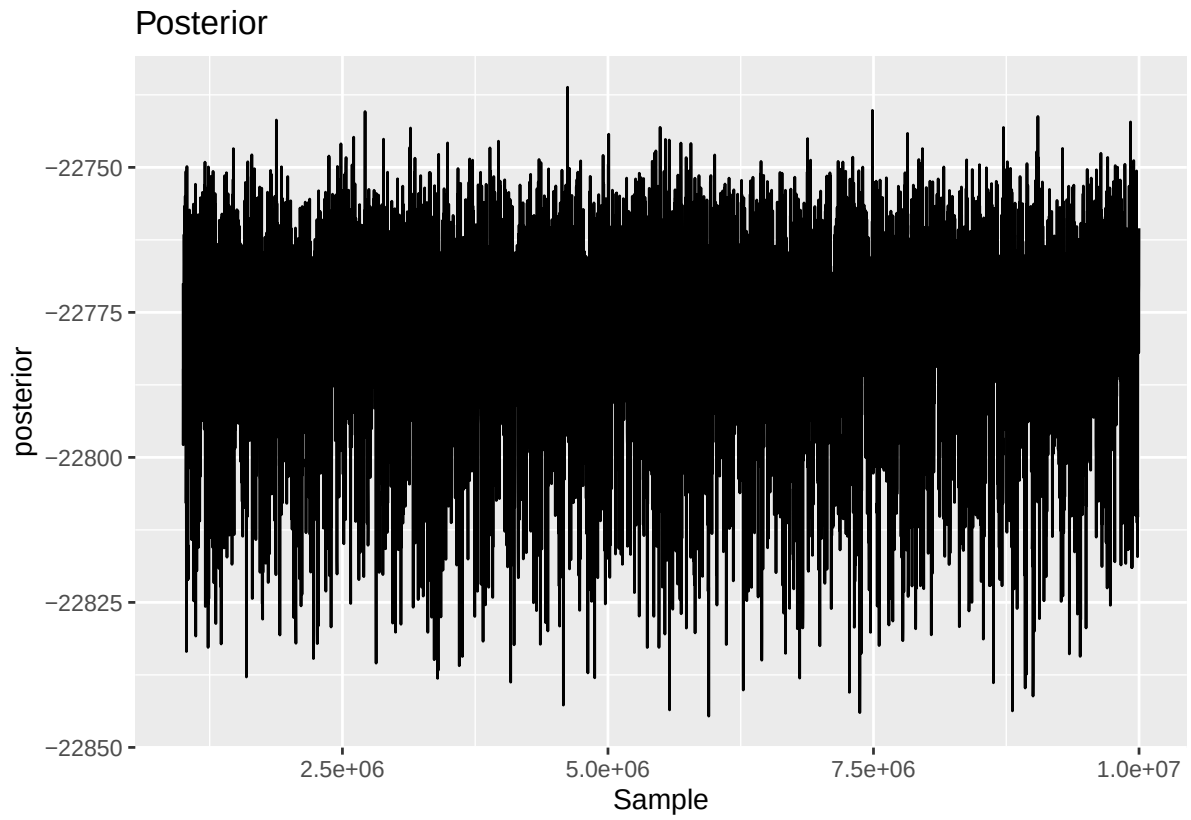
	Posterior
mean	-22779.93
stderr_mean	0.27963
stdev	17.22471
variance	296.6906
median	-22777.37
mode	n/a
geom_mean	n/a
hpd_interval_low	-22815.84
hpd_interval_high	-22750.79
act	11862.4
ess	3793.919

```
sum_stats <- tracerer::calc_summary_stats(
  estimates_burn,
  sample_interval = 5000
)
knitr::kable(sum_stats) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

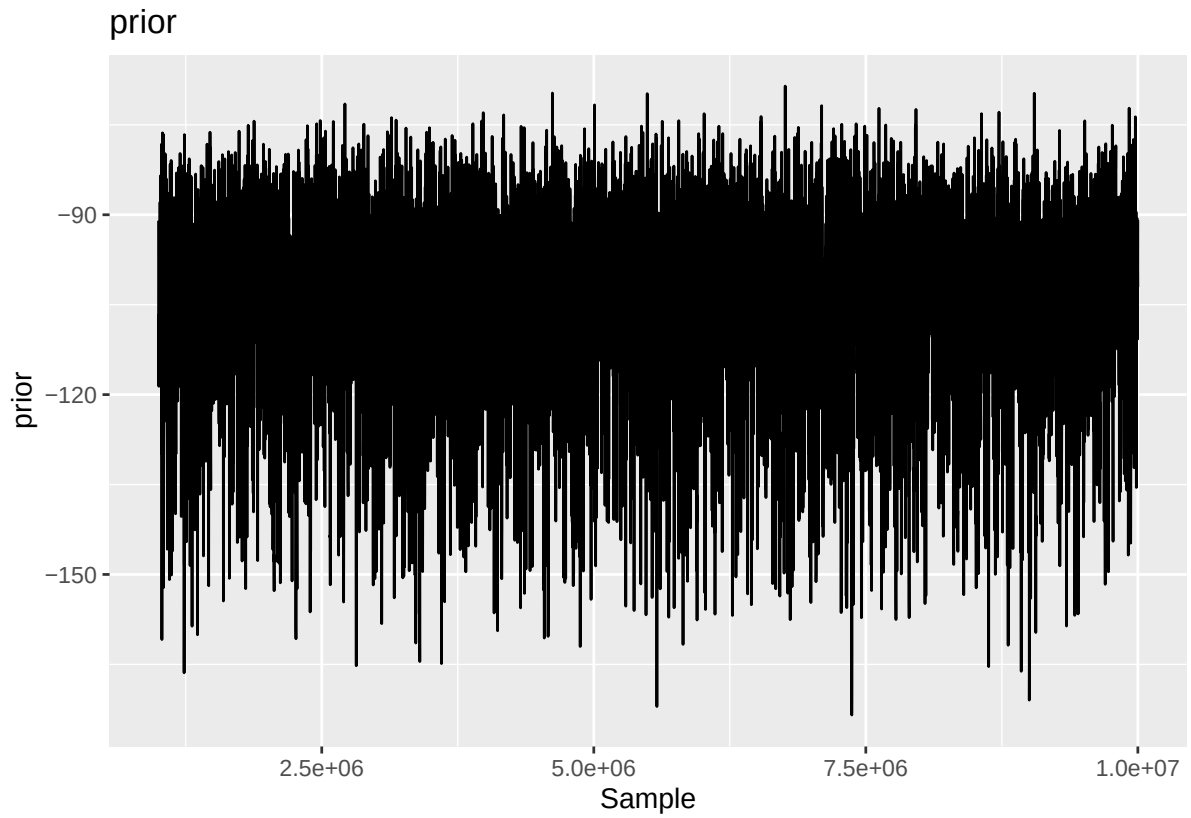
```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

	mean	stderr_mean	stdev	variance	median	mode	geom_mean	hpd_interval_low	hpd_i
posterior	-2.277993e+04	0.2796300	17.2247092	296.6906071	-2.277737e+04	n/a	n/a	-2.281584e+04	-2
likelihood	-2.267510e+04	0.1153701	5.1987267	27.0267598	-2.267445e+04	n/a	n/a	-2.268584e+04	-2
prior	-1.048380e+02	0.2666296	16.5500380	273.9037589	-1.018504e+02	n/a	n/a	-1.403306e+02	-7
treeLikelihood	-2.267510e+04	0.1153701	5.1987267	27.0267598	-2.267445e+04	n/a	n/a	-2.268584e+04	-2
TreeHeight	2.807542e+00	0.0223744	1.5006690	2.2520076	2.411138e+00	n/a	2.50837098173729	8.629460e-01	5
kappa	4.989072e+01	1.7292585	40.7001245	1656.5001320	3.839770e+01	59.8467608763125	39.2494563984526	7.300690e+00	1
freqParameter.1	2.492876e-01	0.0118115	0.1924373	0.0370321	2.079152e-01	0.0839703944500121	0.158202929812485	3.023000e-04	0
freqParameter.2	2.372412e-01	0.0089271	0.1876262	0.0352036	1.847490e-01	0.173705637052645	0.149177752008778	1.550000e-05	0
freqParameter.3	2.583018e-01	0.0092471	0.1936269	0.0374914	2.215576e-01	0.0923881290017638	0.167907632694165	2.080000e-04	0
freqParameter.4	2.551694e-01	0.0109370	0.1874960	0.0351548	2.247822e-01	0.174830838004216	0.172192662146243	1.174000e-04	0
clockRate	1.098000e-04	0.0000006	0.0000440	0.0000000	1.056000e-04	n/a	0.0001009112355395	3.220000e-05	
popSize	3.716663e+00	0.0246872	1.5302644	2.3417093	3.193135e+00	n/a	3.48296781604499	2.300210e+00	6
CoalescentConstant	-9.357834e+01	0.2445039	15.6999717	246.4891123	-9.070315e+01	n/a	n/a	-1.270006e+02	-6

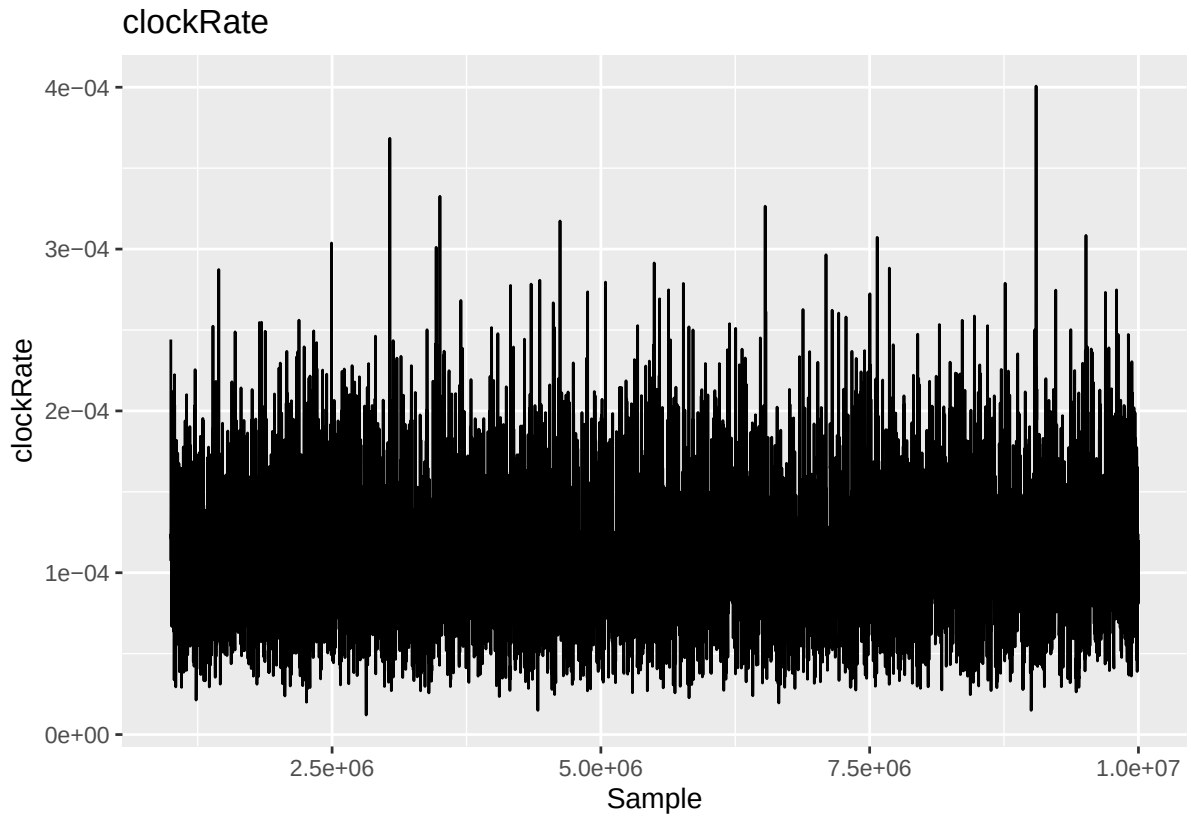
```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = posterior)) +
  ggplot2::ggtitle("Posterior")
```



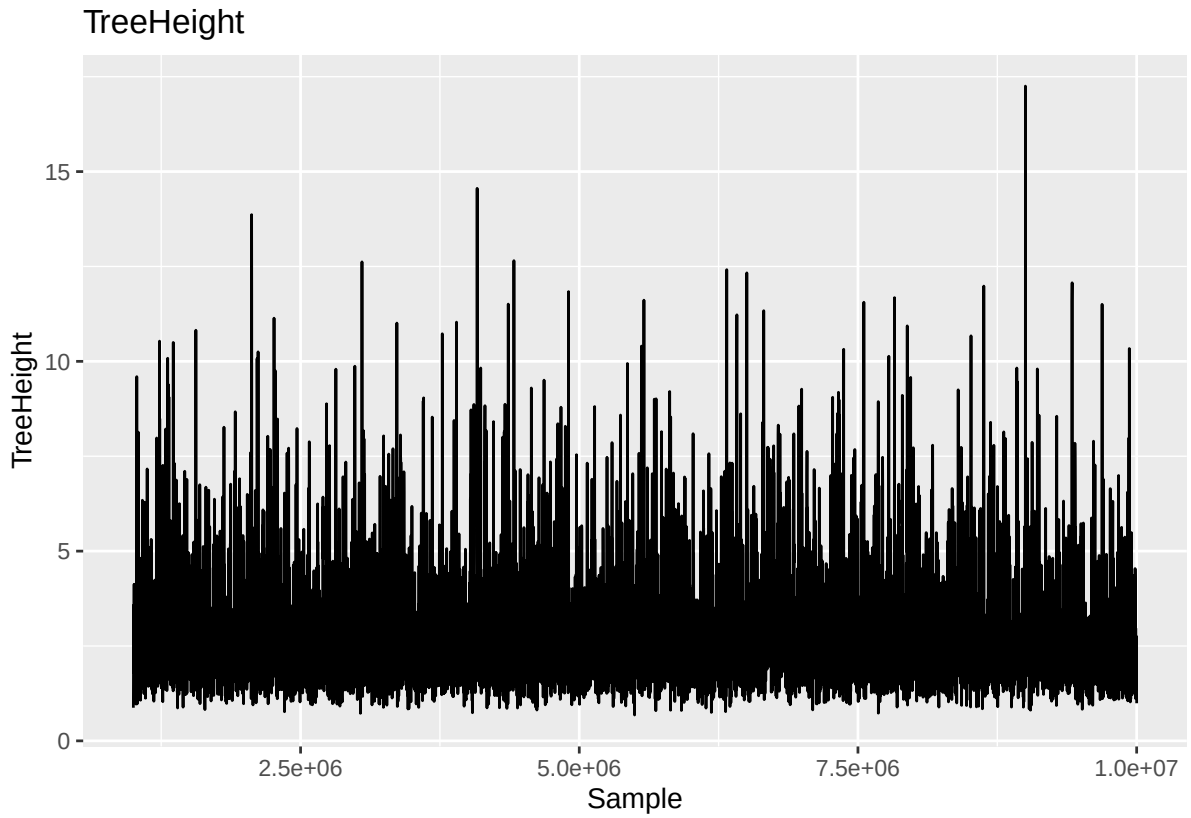

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = prior)) +
  ggplot2::ggtitle("prior")
```



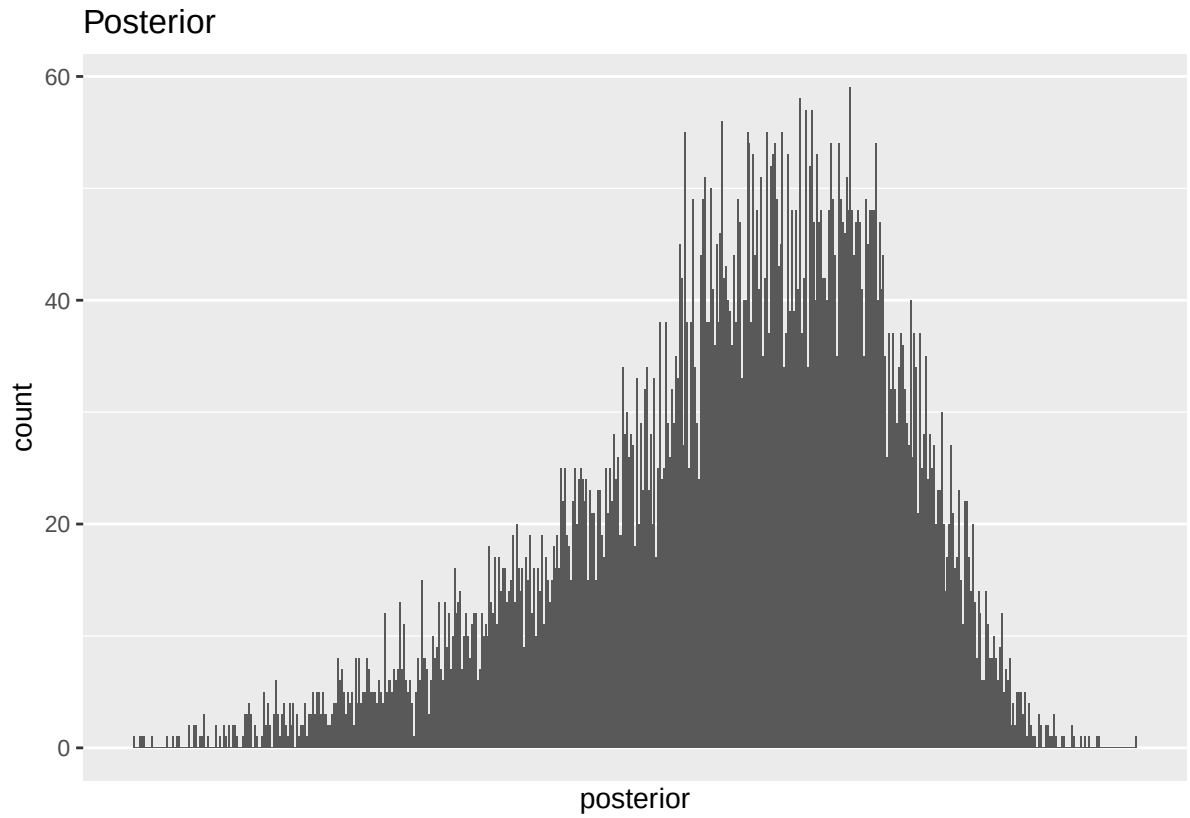

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = clockRate)) +
  ggplot2::ggtitle("clockRate")
```




```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = TreeHeight)) +
  ggplot2::ggtitle("TreeHeight")
```

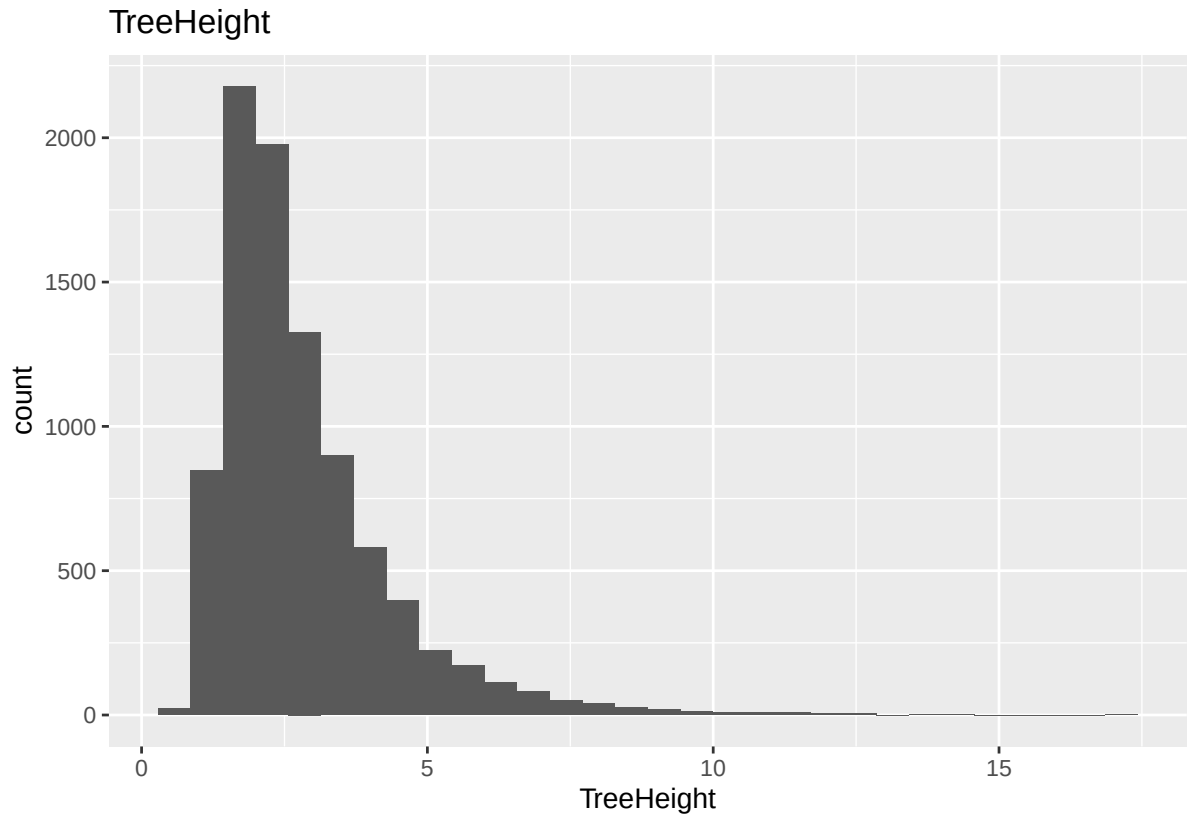



```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(posterior)) +
  ggplot2::geom_histogram(binwidth = 0.21) +
  ggplot2::scale_x_continuous(breaks = seq(-75, -68)) +
  ggplot2::ggtitle("Posterior")
```



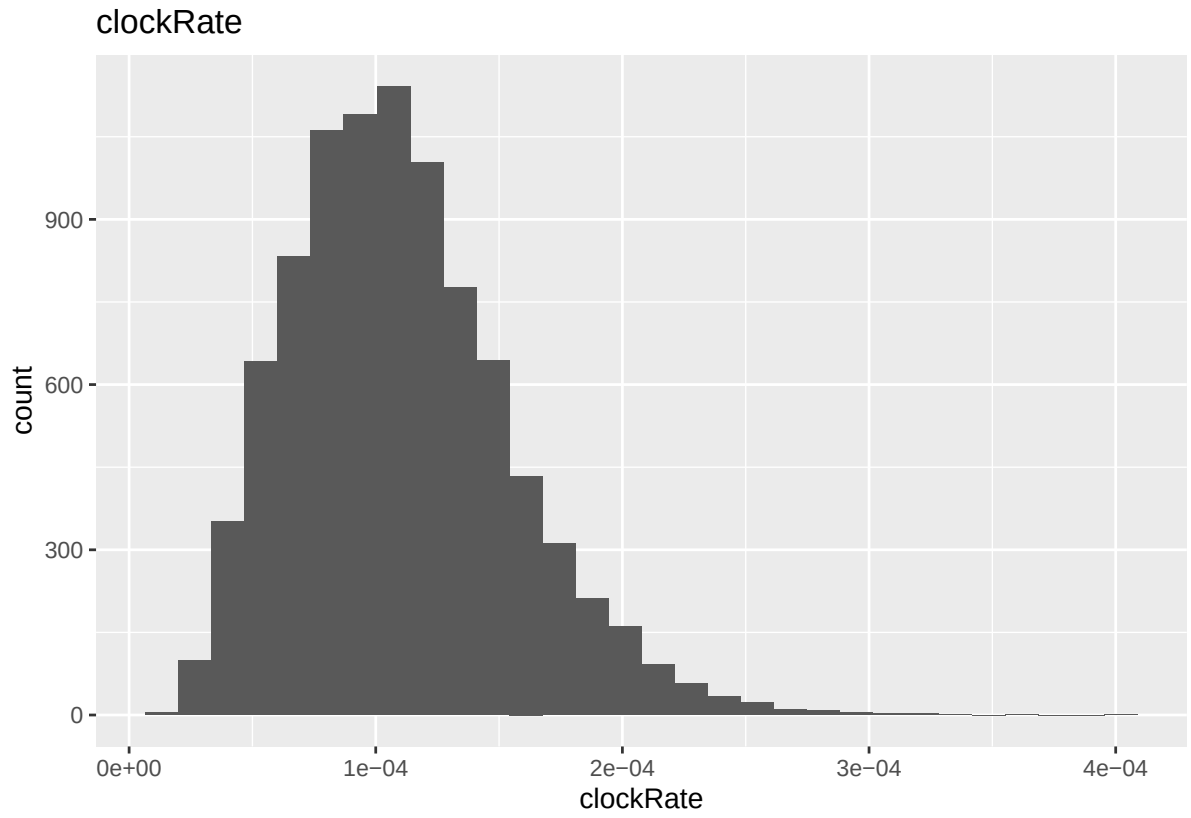

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(TreeHeight)) +
  ggplot2::geom_histogram() +
  ggplot2::ggtitle("TreeHeight")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

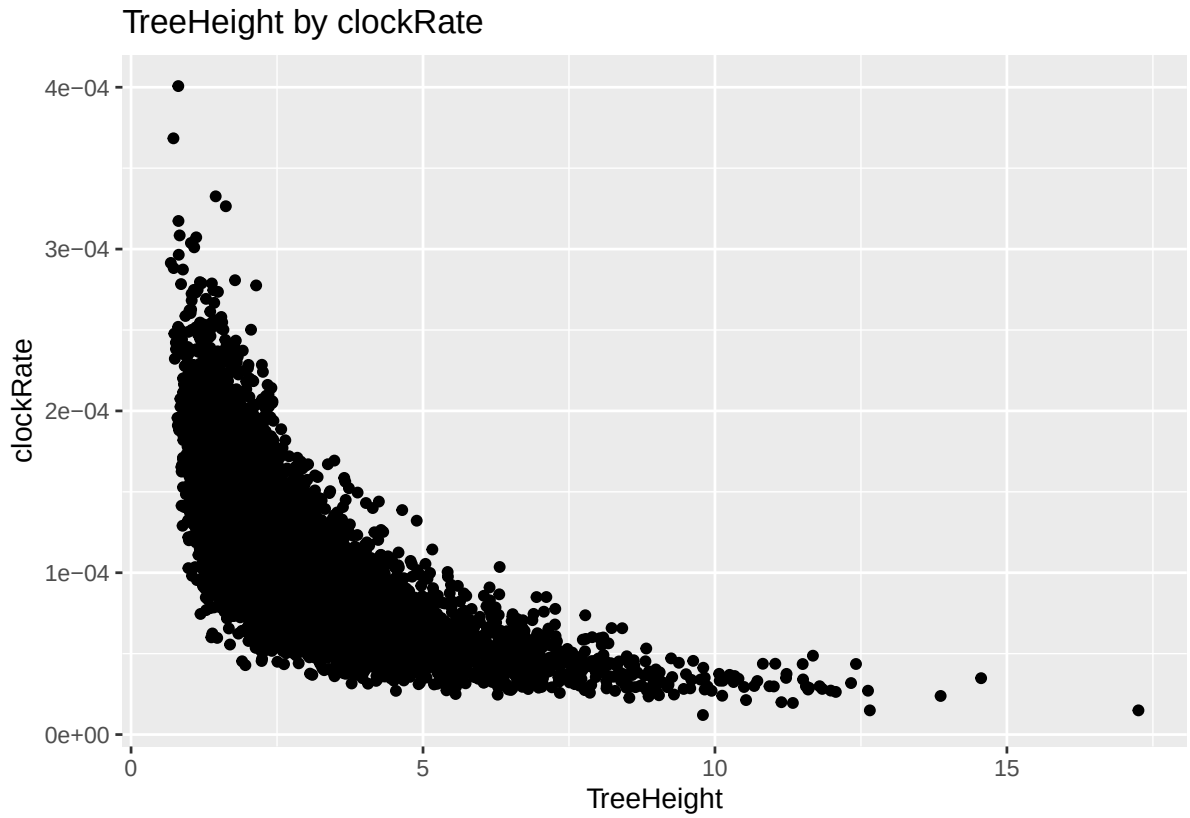



```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(clockRate)) +
  ggplot2::geom_histogram() +
  ggplot2::ggtitle("clockRate")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

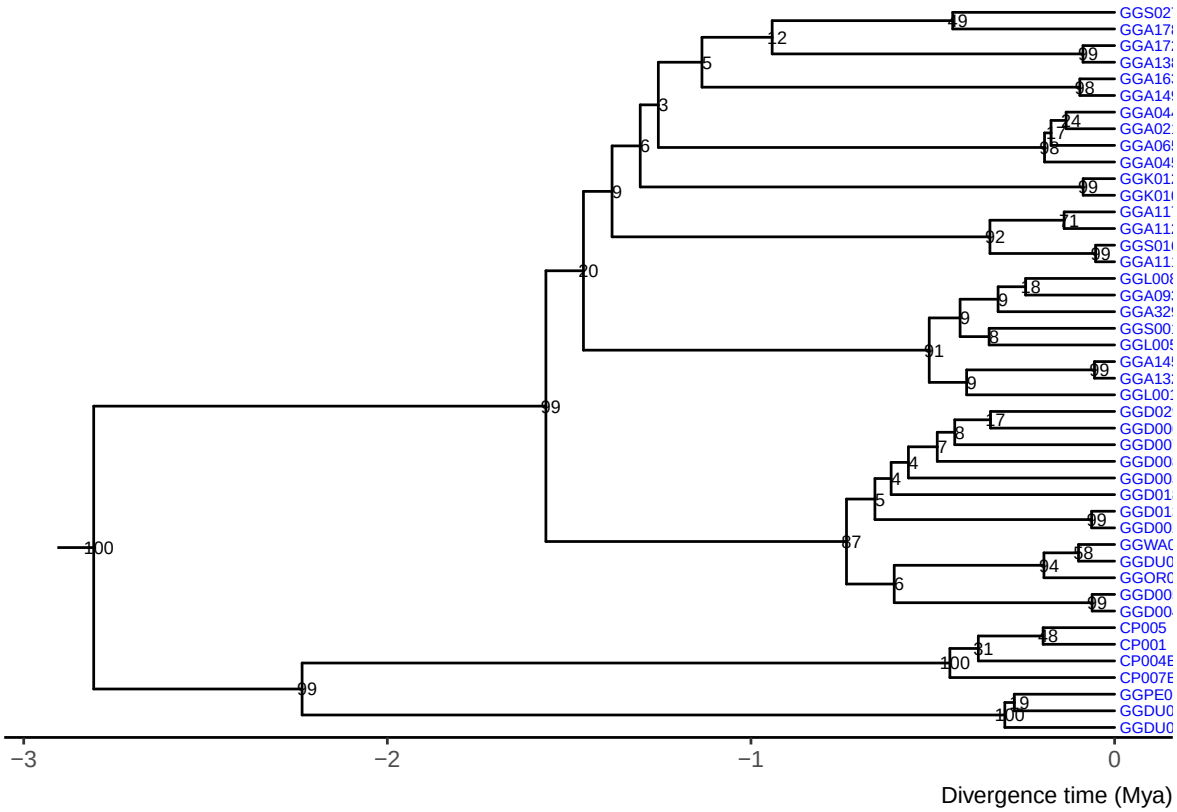



```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = TreeHeight, y = clockRate)) +
  ggplot2::geom_point() +
  ggplot2::ggtitle("TreeHeight by clockRate")
```




```
trees <- tracerer::parse_beast_trees(
  filename = treelog
)
treefile <- "beast_divergence_NoOutgroup_CPP/Result_trees.tre" #with treeAnnotator
# phyloch::treeannotator(input = treelog, )

ggar.tree <- ape::read.nexus(treefile)
boot_support <- ape::prop.clades(ggar.tree, trees)
boot_support <- boot_support/100
boot_support <- round(boot_support, 0)
nodes <- paste0("t",1:43)

ggar.tree$root.edge <- 0.1


p1 <- ggtree::ggtree(tr = ggar.tree, color = "black")
p1$data$label[p1$data$isTip == FALSE] <- nodes
bs <- data.frame(label = p1$data$label, bootstrap = c(rep(NA, 44),boot_support))
# bs$bootstrap[is.na(bs$bootstrap)] <- 100
p1 <- p1 %<+% bs

p1 <- p1 +
  ggtree::geom_tiplab(size = 2, color = 'blue') +
  ggtree::geom_rootedge() +
  ggtree::geom_nodelab(aes(label = bootstrap), size = 2.5) +
  ggtree::theme_tree2() +
  labs(caption = "Divergence time (Mya)")
p2 <- ggtree::revts(p1)
print(p2)
```

8.4.7.4 Plot Tree




```
ggplot2::ggsave(plot = p2, filename = "BEAST_timetree_NoOutgroup_ggtree_FINAL.png",
                device = "png", width = 30, height = 15, units = "cm")
```

8.4.8 BEAST - Bayesian skyline for all regions

This analysis is likely not appropriate, since we have evidence of structured populations (not accounted for in Bayesian Skyline analysis).

```
pop_1 <- "bsp_ALL_unique"
print(pop_1)
```

8.4.8.1 Create BEAUTI file

```
## [1] "bsp_ALL_unique"
```

```
phy_filename <- "6.ggar_Nucleotide_alignment_edited_HTPsites.phy.uniqueseq.phy"
fasta_filename <- "6.ggar_Nucleotide_alignment_edited_HTPsites.phy.uniqueseq.fasta"

phy.dat <- phylotools::read.phylip(infile = phy_filename, clean_name = TRUE)
phylotools::dat2fasta(dat = phy.dat, outfile = fasta_filename)
```

```
##
6.ggar_Nucleotide_alignment_edited_HTPsites.phy.uniqueseq.fasta
has been saved to C:/Users/dev093/OneDrive - University of
Tasmania/PhD/7.SBD_garricki/4.Results
```

```
inference_model <-
  beautier::create_inference_model(
    site_model = beautier::create_site_model_hky(
      # based on #IQtree: HKY+F+I
      gamma_site_model = beautier::create_gamma_site_model(
        gamma_cat_count = "4",
        gamma_shape = "1.0",
        prop_invariant = "0.9925",
        gamma_shape_prior_distr = NA,
        freq_equilibrium = "empirical",
        freq_prior_uniform_distr_id = 1000
      ),
      freq_equilibrium = "empirical",
    ),
    tree_prior = beautier::create_tree_prior_cbs(
      group_sizes_dimension = 5,
      b_pop_sizes_param = beautier::create_b_pop_sizes_param(
        id = NA,
        upper = "50000.0",
        value = beautier::create_distr_log_normal(m = 6.91, s = 2.35,
                                                  value = 6.91, lower = 2.30,
                                                  upper = 10.81
                                                  ) #curve(dlnorm(x, meanlog=10, sdlog=1), from=0, t
        # value = beautier::create_distr_uniform( value = 10000,
        #                                         lower = 50,
        #                                         upper = 25000)
      ),
      pop_sizes_scaler_scale_factor = 0.5 #for mtDNA
    ),
    clock_model = beautier::create_strict_clock_model(
      clock_rate_param = beautier::create_clock_rate_param(
        value = 6.2e-9, #based on divergence tree
        estimate = TRUE,
        id = NA

      ),
    clock_rate_distr = beautier::create_distr_log_normal(m = -18.88, s = 1.175,
```

```
                                                    value = -18.88,
                                                    lower = 1e-20,
                                                    upper = 1),
    # clock_rate_distr = beautier::create_distr_log_normal(m = -9.11, s = 1.25, value = -9.11,
    #                                                      lower = 0, upper = 1),
      rate_scaler_factor = 1
    ),
    # mrca_prior = beautier::create_mrca_prior(
    #   mrca_distr = beautier::create_normal_distr(mean = 3e6,  sigma = 5e5)# based on divergence tr
    #   ),
    mcmc = beautier::create_mcmc(
      chain_length = 1e+07,
      store_every = 5e+03,
      pre_burnin = 1e4,
      n_init_attempts = 3,
      tracelog = beautier::create_tracelog(filename =
                                             paste0("tracelog_", pop_1, ".log")),
      screenlog = beautier::create_screenlog(filename =
                                               paste0("screenlog_", pop_1, ".log")),
      treelog = beautier::create_treelog(filename =
                                           paste0("treelog_", pop_1, ".log"))
    ),
    beauti_options = beautier::create_beauti_options()
  )



beast2_input_file <- paste0("beast2_", pop_1, ".xml")
beautier::create_beast2_input_file_from_model(
  input_filename = fasta_filename,
  inference_model = inference_model,
  output_filename = beast2_input_file
)

if (beastier::is_beast2_installed()) {
  beastier::is_beast2_input_file(beast2_input_file,
                                 show_warnings = FALSE,
                                 verbose = FALSE)
}
```

```
## [1] TRUE
```

```
if (beastier::is_beast2_installed()) {
  beast2_options <- beastier::create_beast2_options(
    input_filename = beast2_input_file,
    output_state_filename = paste0("beast2_",pop_1,".xml.state"),
    n_threads = 4,
    verbose = TRUE
  )
  beastier::check_can_create_file(beast2_options$output_state_filename)
  beastier::check_can_create_treelog_file(beast2_options)
  run_beast2_from_options(
    beast2_options = beast2_options
  )
  testthat::expect_true(file.exists(beast2_options$output_state_filename))
}
```

```
if (beastier::is_beast2_installed()) {
  out <- babette::bbt_run_from_model(
    fasta_filename = fasta_filename,
    inference_model = inference_model,
    beast2_options = beast2_options)
}
```

8.4.8.2 Run BEAST

```
tracelog <- "beast_bspe_ALL_unique/tracelog_bsp_ALL_unique.log" #clock rate in MYA
treelog <- "beast_bspe_ALL_unique/treelog_bsp_ALL_unique.log"
output_state <- "beast_bspe_ALL_unique/beast2_bsp_ALL_unique.xml.state"
# tracelog <- "beast_bspe_ALL_unique_2/tracelog_bsp_ALL_unique.log" ##clock rate in Years
# treelog <- "beast_bspe_ALL_unique_2/treelog_bsp_ALL_unique.log"
# output_state <- "beast_bspe_ALL_unique_2/beast2_bsp_ALL_unique.xml.state"


estimates <- tracerer::parse_beast_tracelog_file(
  tracelog_filename = tracelog
)
estimates_burn <- tracerer::remove_burn_ins(estimates, burn_in_fraction = 0.1)
esses <- tracerer::calc_esses(estimates_burn, sample_interval = 5000)
esses <- t(esses)
colnames(esses) <- c("ESS")
knitr::kable(esses)
```

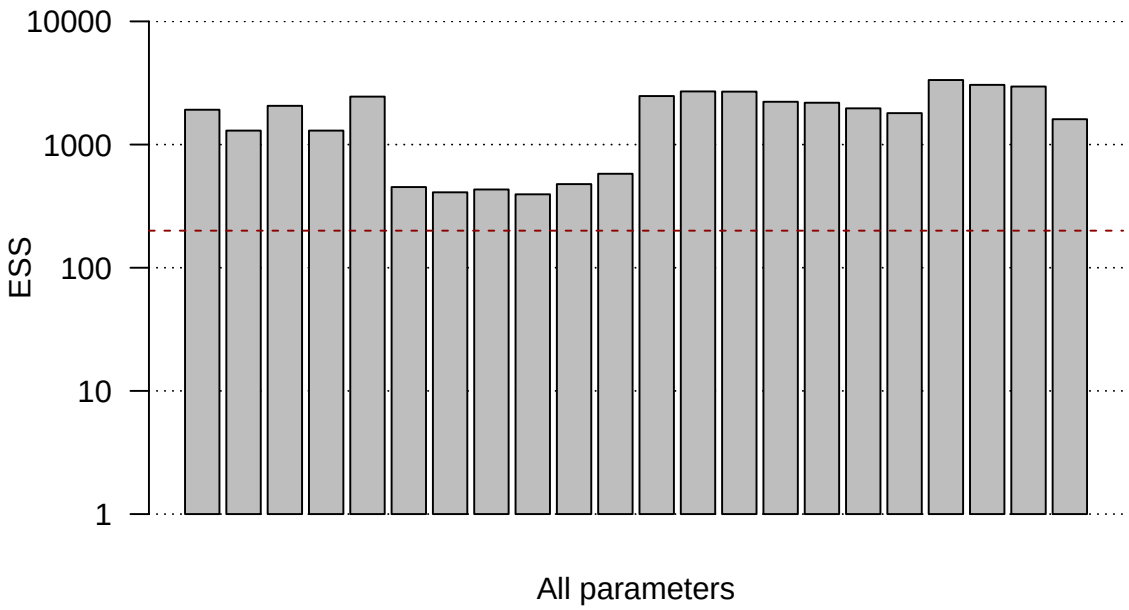
8.4.8.3 Summary stats

	ESS
posterior	1798
likelihood	1086
prior	2087
treeLikelihood	1086
TreeHeight	2127
kappa	340
freqParameter.1	338
freqParameter.2	472
freqParameter.3	405
freqParameter.4	445
gammaShape	587
BayesianSkyline	2521
bPopSizes.1	2713
bPopSizes.2	2586
bPopSizes.3	2008
bPopSizes.4	1854
bPopSizes.5	1941
bGroupSizes.1	1423
bGroupSizes.2	2955
bGroupSizes.3	3124
bGroupSizes.4	2665
bGroupSizes.5	1420

```
bdsky_trace   <- beastio::readLog(tracelog, burnin = 0.1)
beastio::checkESS(bdsky_trace)
```

```
## named numeric(0)
```

```
beastio::checkESS(bdsky_trace, cutoff = 200, plot = TRUE, log = 'y',
                  ylim = c(1,10000), title = "All parameters", plot.grid = TRUE)
```




```
## named numeric(0)
```

```
Posterior <- tracerer::calc_summary_stats(
  estimates_burn$posterior,
  sample_interval = 5000
)
Posterior <- t(Posterior)
colnames(Posterior) <- c("Posterior")
knitr::kable(Posterior)
```

	Posterior
mean	-22582.6
stderr_mean	0.3350352
stdev	14.20845
variance	201.8801
median	-22582.06
mode	n/a
geom_mean	n/a
hpd_interval_low	-22609.67
hpd_interval_high	-22554
act	25026.29
ess	1798.309

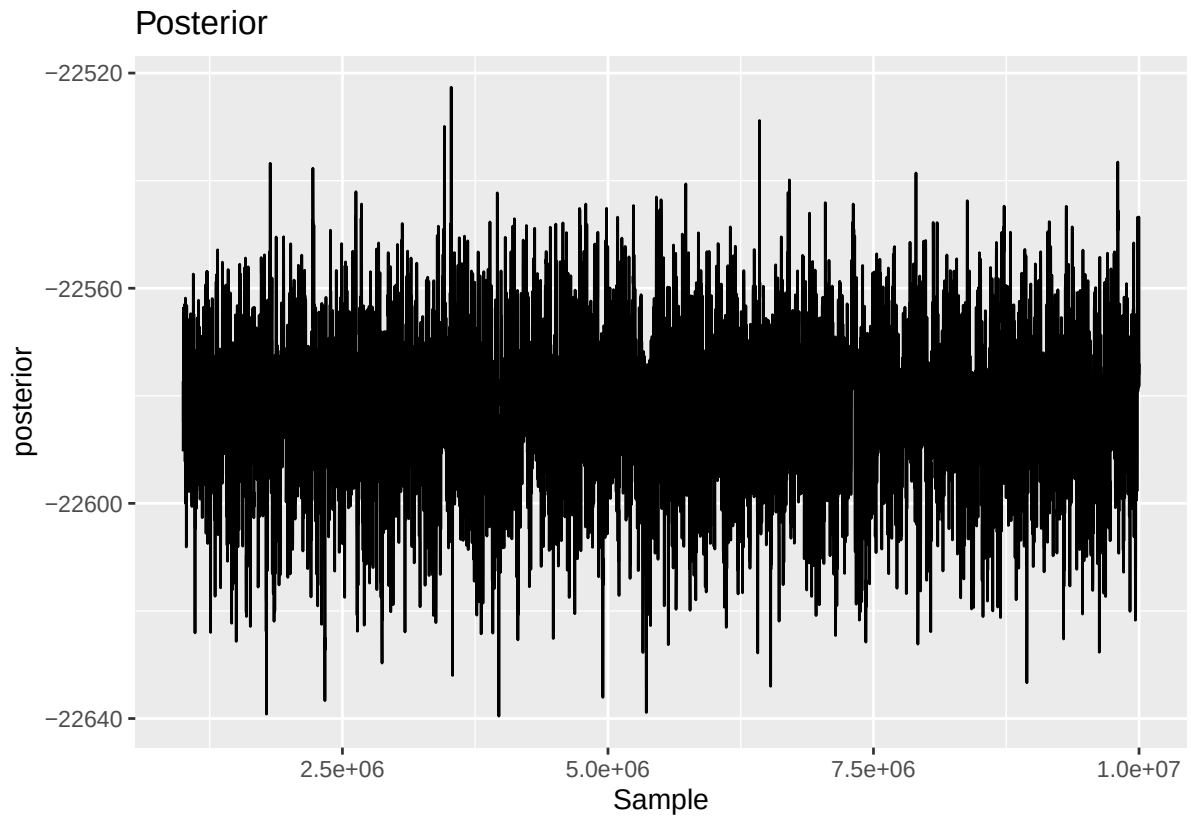
```
sum_stats <- tracerer::calc_summary_stats(
  estimates_burn,
  sample_interval = 5000
)
```

```
knitr::kable(sum_stats) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

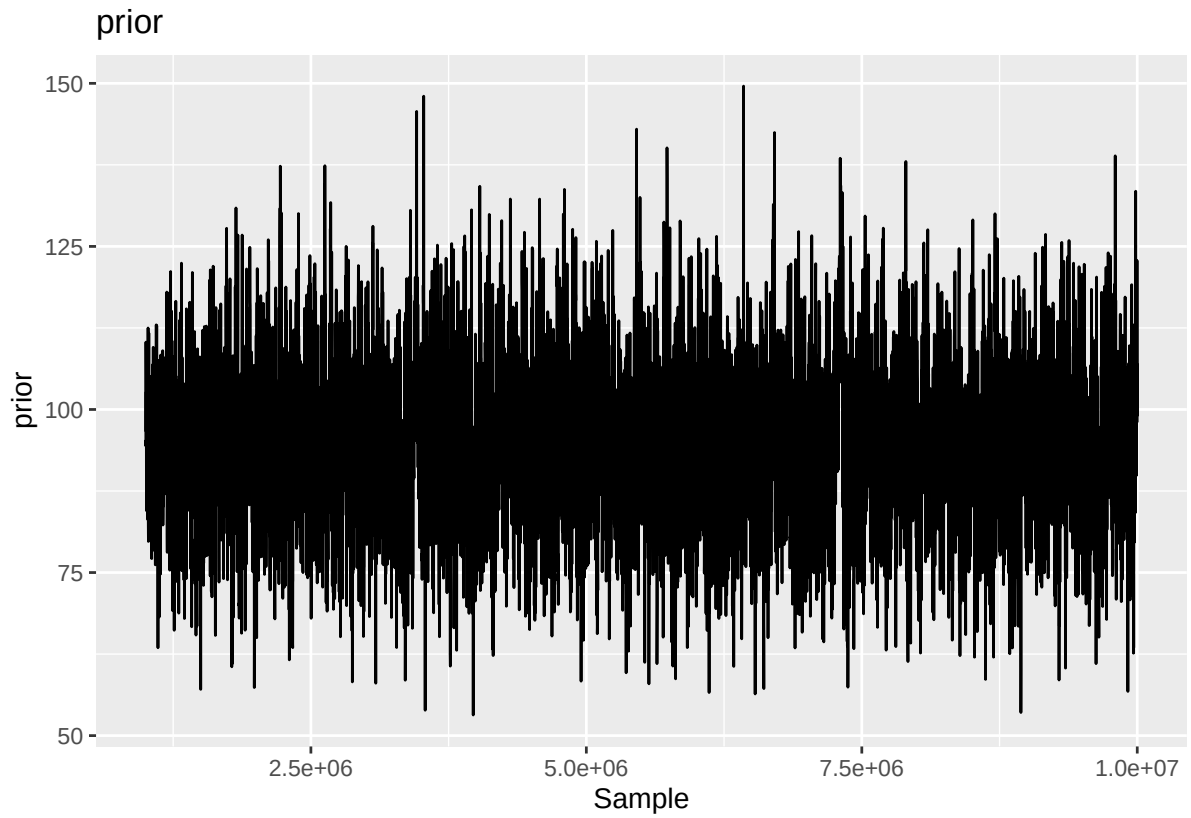
```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

	mean	stderr_mean	stdev	variance	median	mode	geom_mean	hpd_interval_low	hpd_inter
posterior	-2.258260e+04	0.3350352	14.2084513	201.8800895	-2.258206e+04	n/a	n/a	-2.260967e+04	-2.255
likelihood	-2.267696e+04	0.1753328	5.7784674	33.3906852	-2.267635e+04	n/a	n/a	-2.268833e+04	-2.266
prior	9.435274e+01	0.2756018	12.5912670	158.5400035	9.404087e+01	n/a	93.5042975330348	7.068561e+01	1.198
treeLikelihood	-2.267696e+04	0.1753328	5.7784674	33.3906852	-2.267635e+04	n/a	n/a	-2.268833e+04	-2.266
TreeHeight	3.050130e-02	0.0002315	0.0106797	0.0001141	2.895230e-02	n/a	0.0287487649285555	1.241550e-02	5.10
kappa	5.044288e+01	2.2322252	41.1785650	1695.6742159	3.838575e+01	n/a	39.4765331572733	4.552848e+00	1.293
freqParameter.1	2.366842e-01	0.0106445	0.1957396	0.0383140	1.870478e-01	n/a	0.142663138257315	8.900000e-05	6.38
freqParameter.2	2.639978e-01	0.0092409	0.2008262	0.0403312	2.157116e-01	0.037464440054446	0.174379406942813	2.117000e-04	6.58
freqParameter.3	2.482809e-01	0.0091916	0.1850597	0.0342471	2.034475e-01	0.0546114464602024	0.168289364852218	2.999800e-03	5.98
freqParameter.4	2.510370e-01	0.0091964	0.1940893	0.0376706	1.999372e-01	0.0432908932418333	0.168804892216275	1.499900e-03	6.49
gammaShape	1.641273e+00	0.0438253	1.0619604	1.1277599	1.388952e+00	1.89203285089849	1.33127795790062	2.064165e-01	3.776
BayesianSkyline	9.264780e+01	0.2210022	11.0972995	123.1500563	9.239701e+01	n/a	91.9792548634406	7.142304e+01	1.149
bPopSizes.1	1.275969e-01	0.0030434	0.1585235	0.0251297	9.138320e-02	n/a	0.0920444560319592	6.362300e-03	3.30
bPopSizes.2	8.778530e-02	0.0013975	0.0710728	0.0050513	7.126040e-02	n/a	0.0677406015384853	3.259700e-03	2.12
bPopSizes.3	6.056750e-02	0.0012352	0.0553595	0.0030647	4.692950e-02	n/a	0.0433091017051147	2.510000e-05	1.52
bPopSizes.4	3.513540e-02	0.0008146	0.0350767	0.0012304	2.581230e-02	n/a	0.0241299374506324	1.400000e-06	9.33
bPopSizes.5	1.754680e-02	0.0003085	0.0135928	0.0001848	1.510340e-02	n/a	0.012024482657042	7.000000e-07	4.15
bGroupSizes.1	9.079658e+00	0.1822936	6.8763999	47.2848761	7.000000e+00	1	6.43043295601244	1.000000e+00	2.300
bGroupSizes.2	9.095656e+00	0.1243246	6.7590486	45.6847378	7.000000e+00	1	6.51330131082587	1.000000e+00	2.200
bGroupSizes.3	8.557494e+00	0.1183112	6.6128032	43.7291663	7.000000e+00	1	6.01797297064216	1.000000e+00	2.200
bGroupSizes.4	8.350628e+00	0.1227101	6.3356787	40.1408243	7.000000e+00	1	5.9308934975556	1.000000e+00	2.100
bGroupSizes.5	7.916565e+00	0.1642001	6.1877867	38.2887045	6.000000e+00	1	5.54341875595614	1.000000e+00	2.000

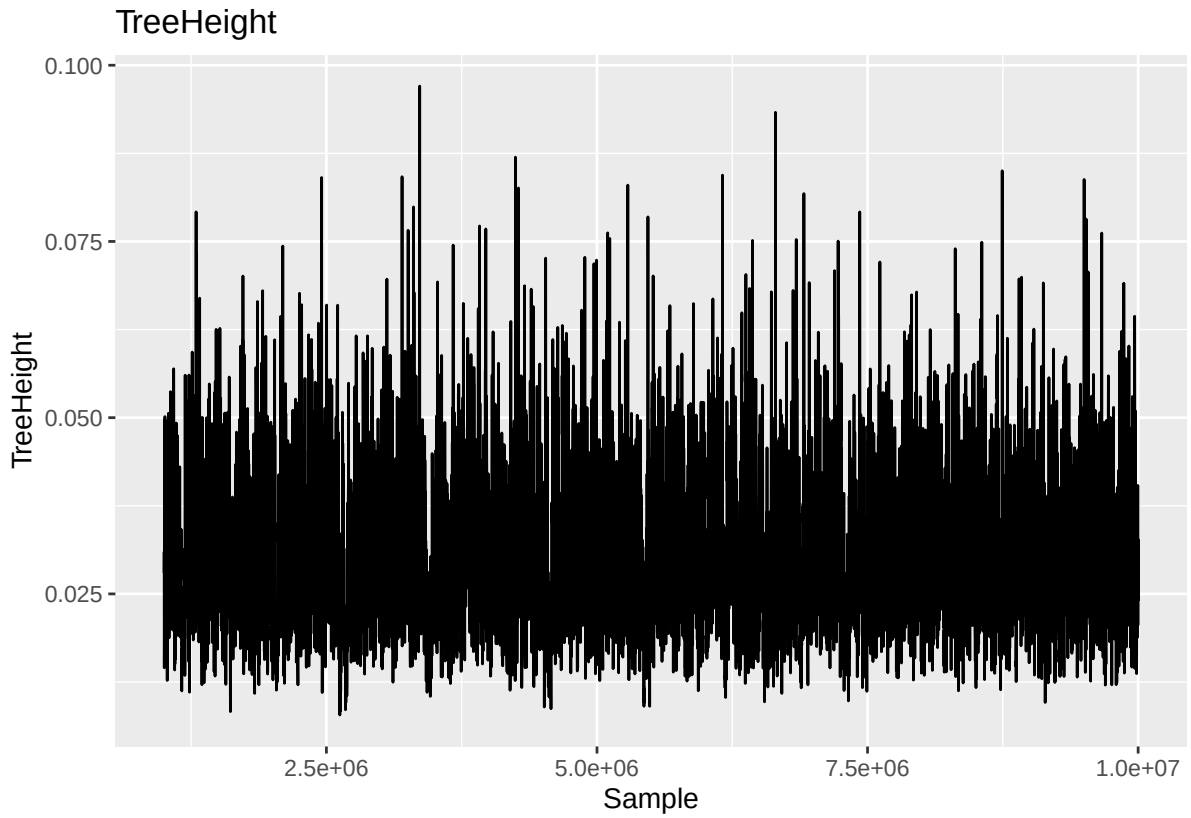
```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = posterior)) +
  ggplot2::ggtitle("Posterior")
```



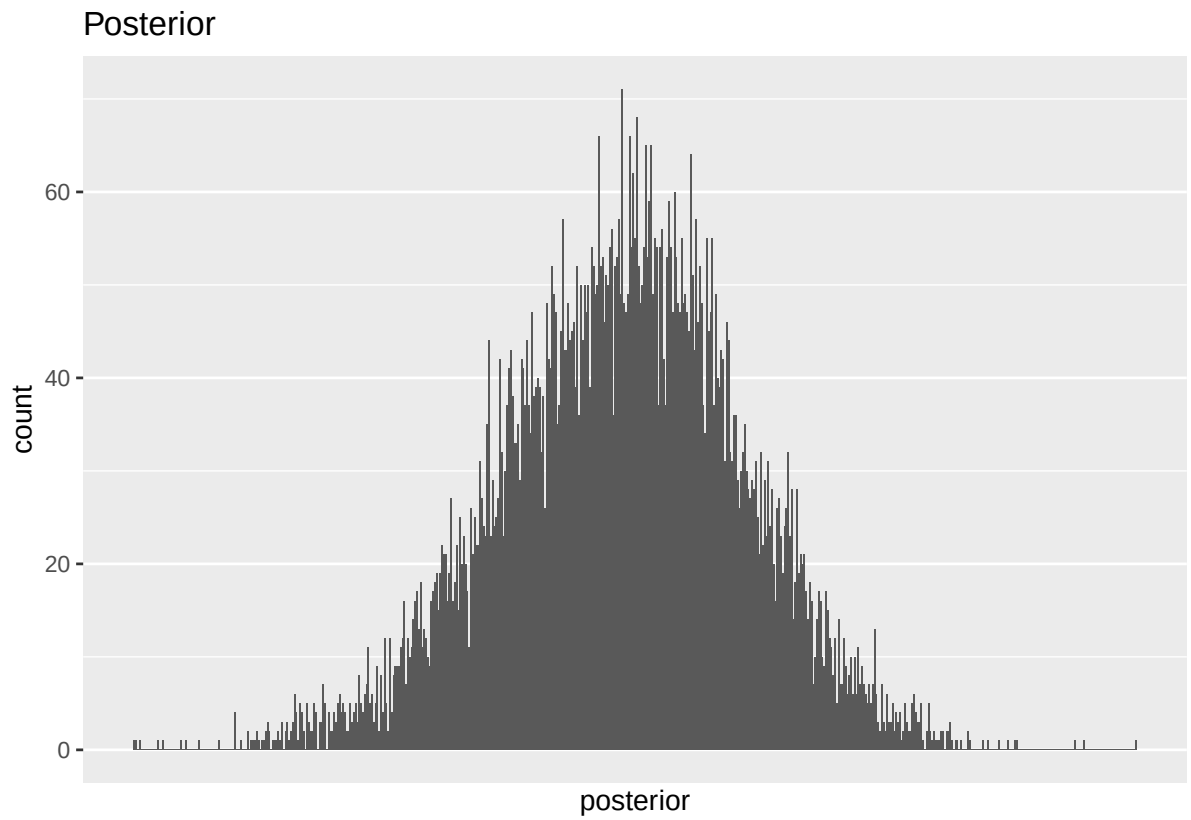

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = prior)) +
  ggplot2::ggtitle("prior")
```




```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(x = Sample)) +
  ggplot2::geom_line(ggplot2::aes(y = TreeHeight)) +
  ggplot2::ggtitle("TreeHeight")
```

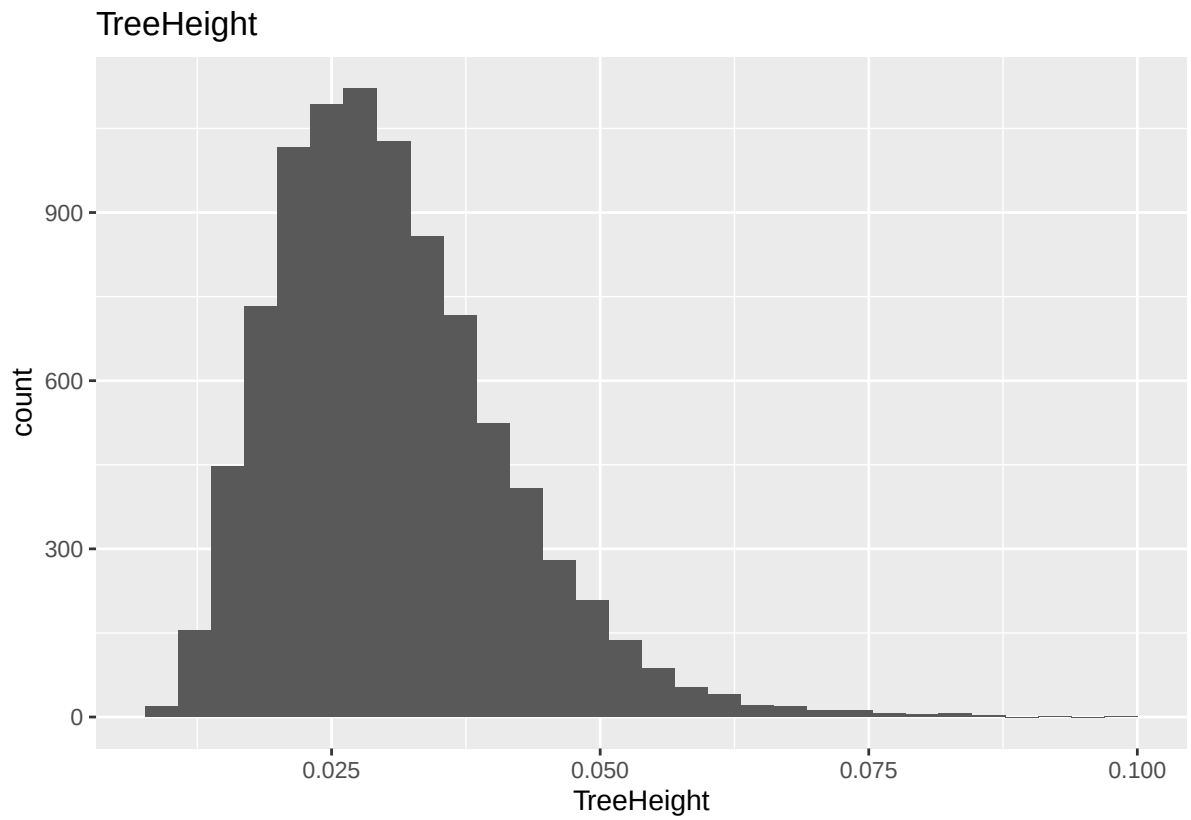



```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(posterior)) +
  ggplot2::geom_histogram(binwidth = 0.21) +
  ggplot2::scale_x_continuous(breaks = seq(-75, -68)) +
  ggplot2::ggtitle("Posterior")
```



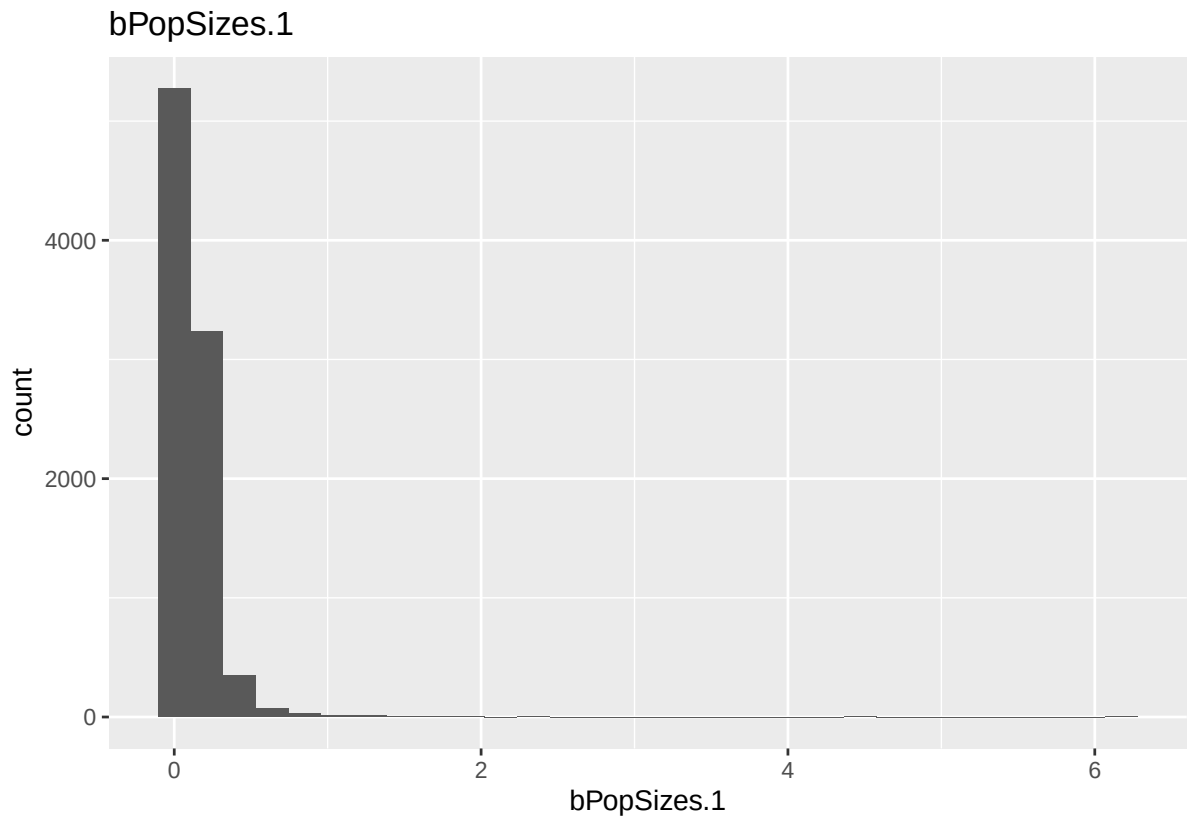

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(TreeHeight)) +
  ggplot2::geom_histogram() +
  ggplot2::ggtitle("TreeHeight")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



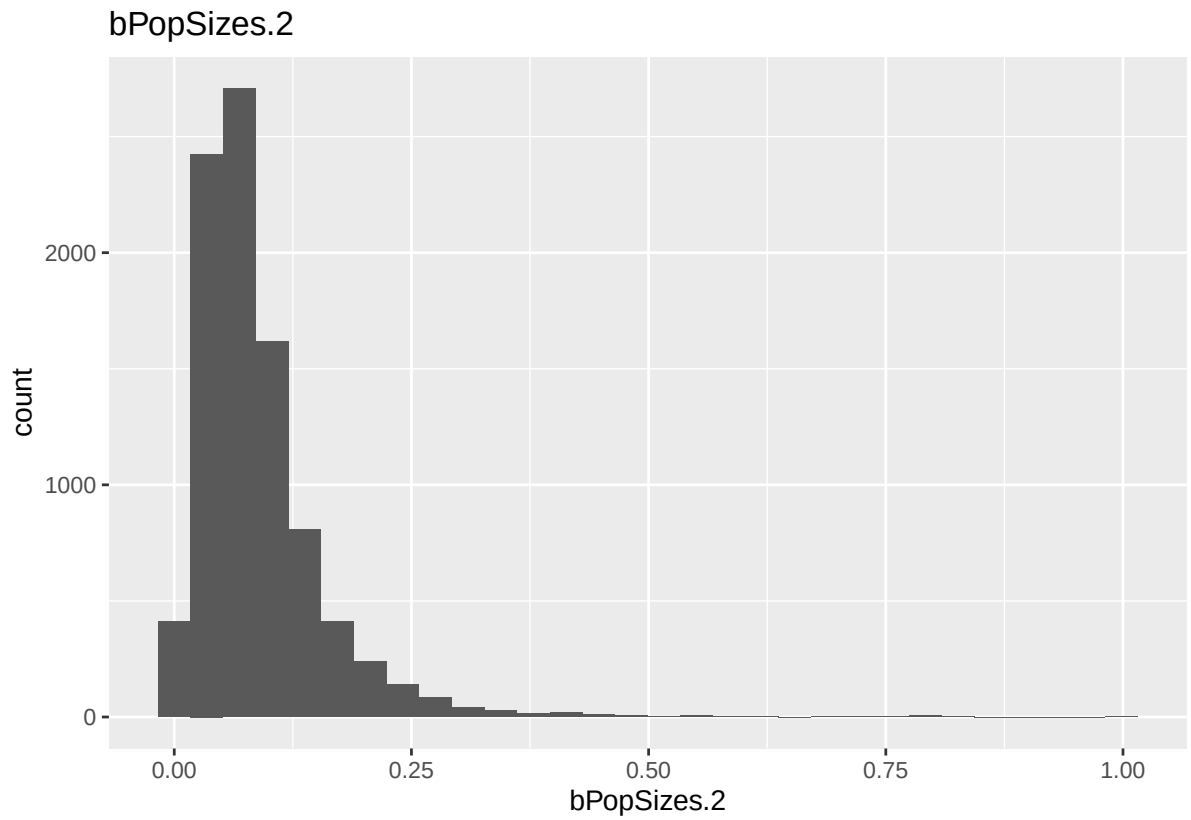

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(bPopSizes.1)) +
  ggplot2::geom_histogram() +
  ggplot2::ggtitle("bPopSizes.1")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



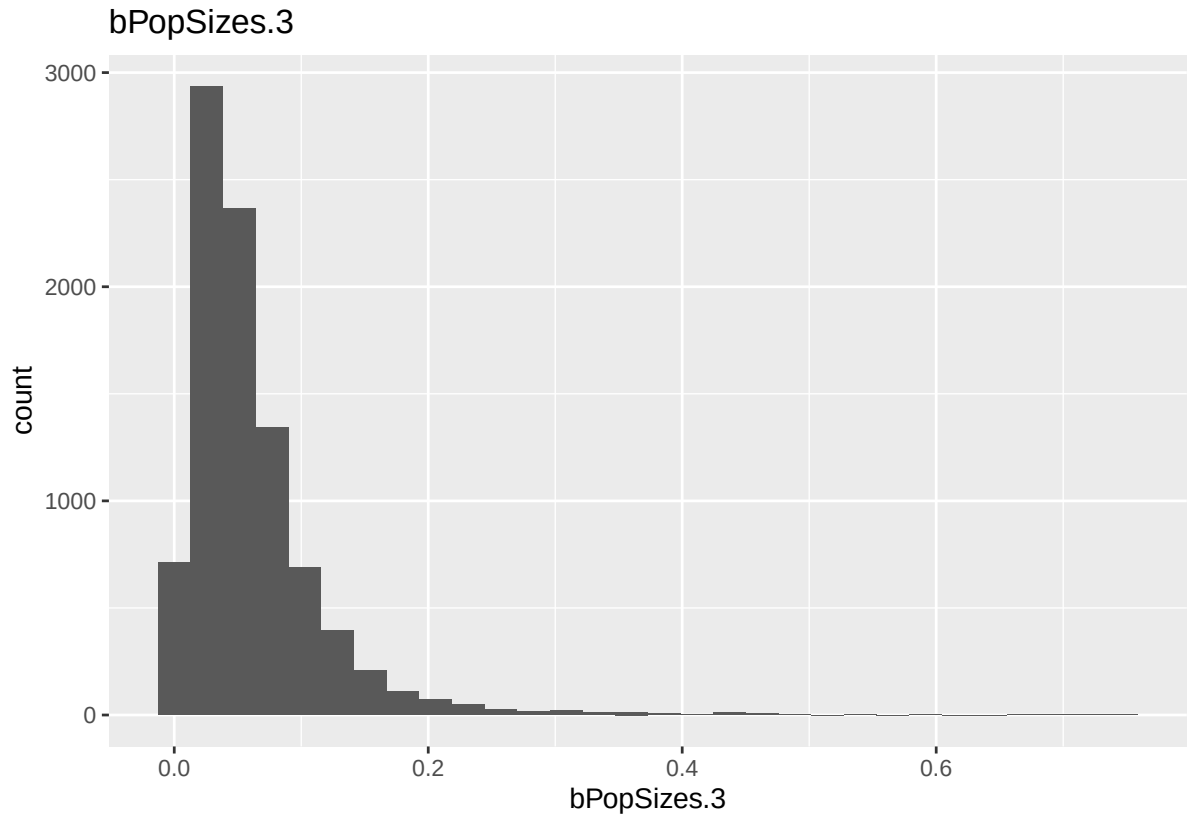

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(bPopSizes.2)) +
  ggplot2::geom_histogram() +
  ggplot2::ggtitle("bPopSizes.2")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



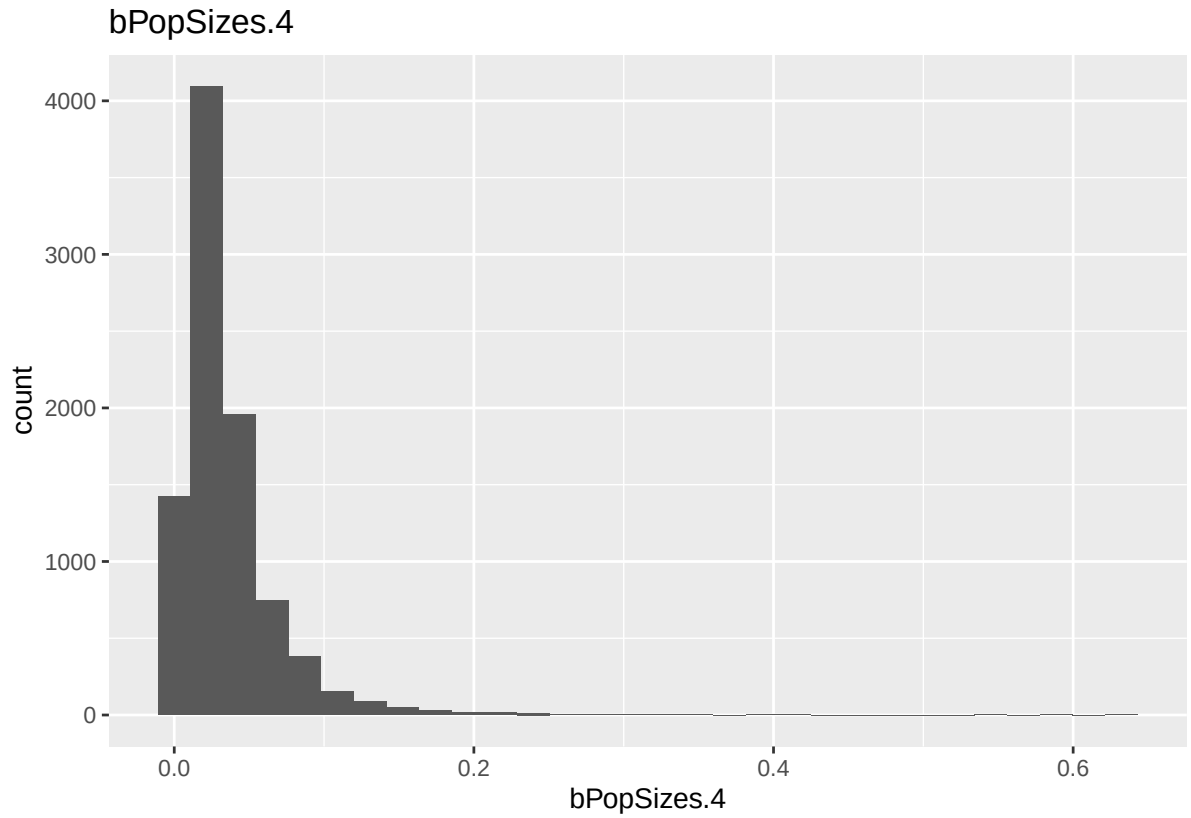

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(bPopSizes.3)) +
  ggplot2::geom_histogram() +
  ggplot2::ggtitle("bPopSizes.3")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



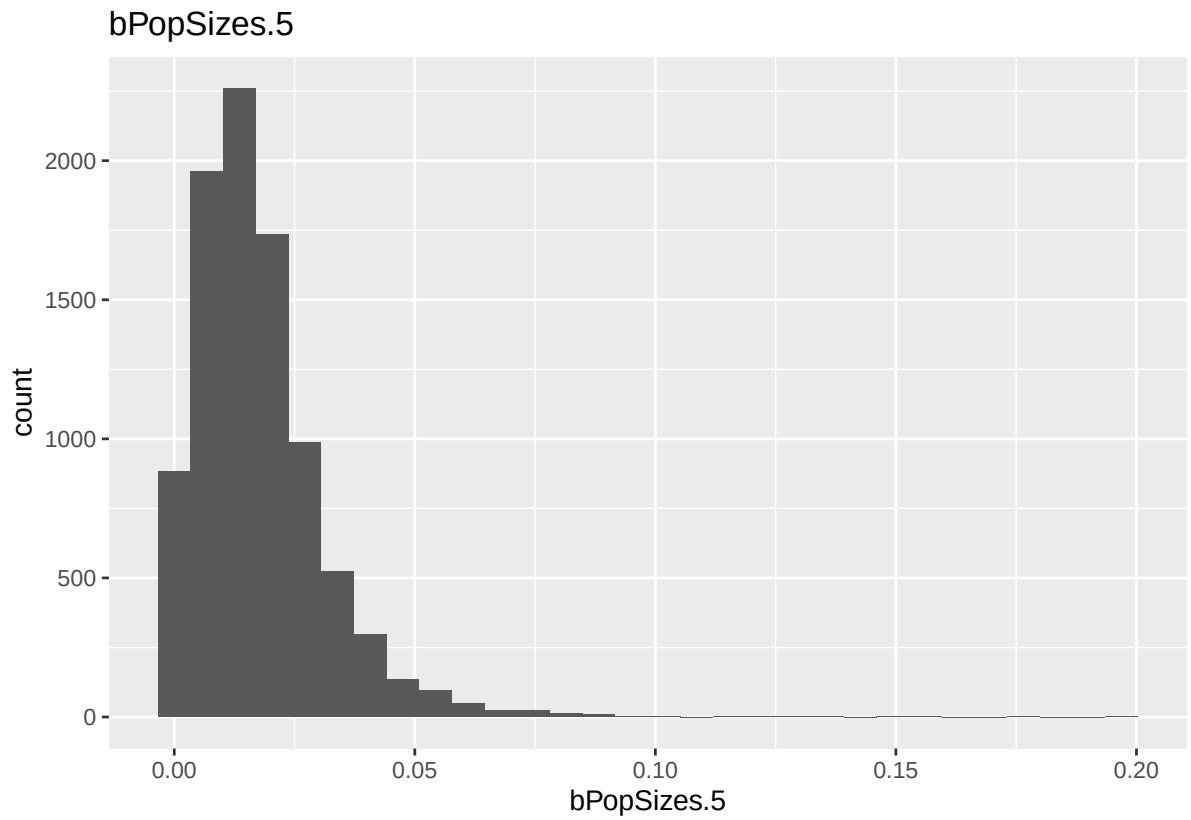

```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(bPopSizes.4)) +
  ggplot2::geom_histogram() +
  ggplot2::ggtitle("bPopSizes.4")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot2::ggplot(
  data = estimates_burn,
  ggplot2::aes(bPopSizes.5)) +
  ggplot2::geom_histogram() +
  ggplot2::ggtitle("bPopSizes.5")
```

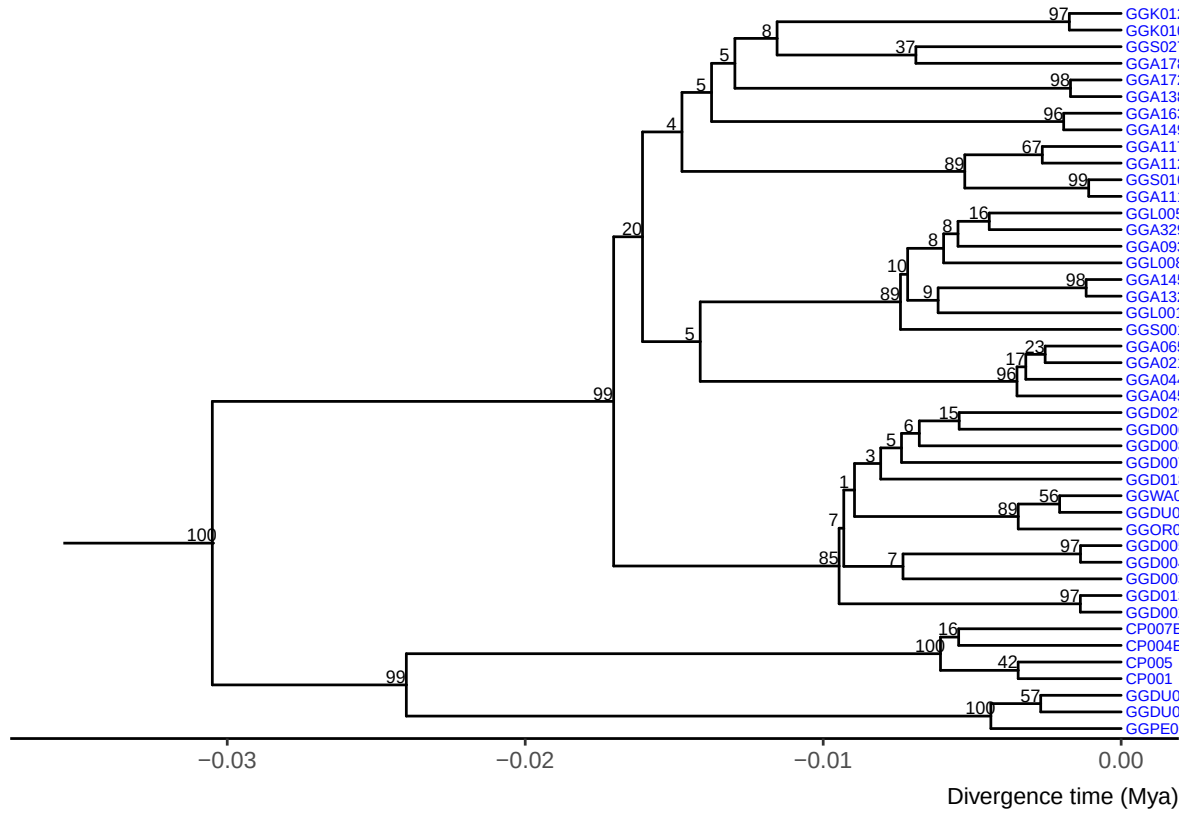
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
trees <- tracerer::parse_beast_trees(
  filename = treelog
)
treefile <- "beast_bspe_ALL_unique/Result_trees.tre" #with treeAnnotator
# treefile <- "beast_bspe_ALL_unique_2/Result_trees.tre" #with treeAnnotator
# phyloch::treeannotator(input = treelog, )

ggar.tree <- ape::read.nexus(treefile)
boot_support <- ape::prop.clades(ggar.tree, trees)
boot_support <- boot_support/100
boot_support <- round(boot_support, 0)
nodes <- paste0("t",1:43)

ggar.tree$root.edge <- 0.005
# ggar.tree$root.edge <- 500


p1 <- ggtree::ggtree(tr = ggar.tree, color = "black")
p1$data$label[p1$data$isTip == FALSE] <- nodes
bs <- data.frame(label = p1$data$label, bootstrap = c(rep(NA, 44),boot_support))
p1 <- p1 %<+% bs

p1 <- p1 +
  ggtree::geom_tiplab(size = 2, color = 'blue') +
  ggtree::geom_rootedge() +
  ggtree::geom_nodelab(aes(label = bootstrap), size = 2.5,
                       nudge_x = -0.0005, nudge_y = 0.5) +
  ggtree::theme_tree2() +
  ggplot2::labs(caption = "Divergence time (Mya)")
# ggplot2::labs(caption = "Divergence time (years)")
```

```
p2 <- ggtree::revts(p1)
print(p2)
```

8.4.8.4 Plot Tree




```
ggplot2::ggsave(plot = p2, filename = "BEAST_bsp_ALL_Unique_NoOutgroup_ggtree.png",
                device = "png", width = 30, height = 15, units = "cm")
```

```
summary(bdsky_trace)
```

8.4.8.5 Plot BSP

```
##
## Iterations = 999000:1e+07
## Thinning interval = 1000
## Number of chains = 1
## Sample size per chain = 9002
##
## 1. Empirical mean and standard deviation for each
variable,
## plus standard error of the mean:
##
## Mean SD Naive SE Time-series SE
## posterior -2.258e+04 14.20810 0.1497499 0.3244215
## likelihood -2.268e+04 5.77815 0.0609003 0.1603519
## prior 9.435e+01 12.59112 0.1327073 0.2772096
## treeLikelihood -2.268e+04 5.77815 0.0609003 0.1603519
## TreeHeight 3.050e-02 0.01068 0.0001126 0.0002157
## kappa 5.044e+01 41.17806 0.4340067 1.9373334
## freqParameter.1 2.367e-01 0.19573 0.0020630 0.0096712
## freqParameter.2 2.640e-01 0.20082 0.0021166 0.0096629
```

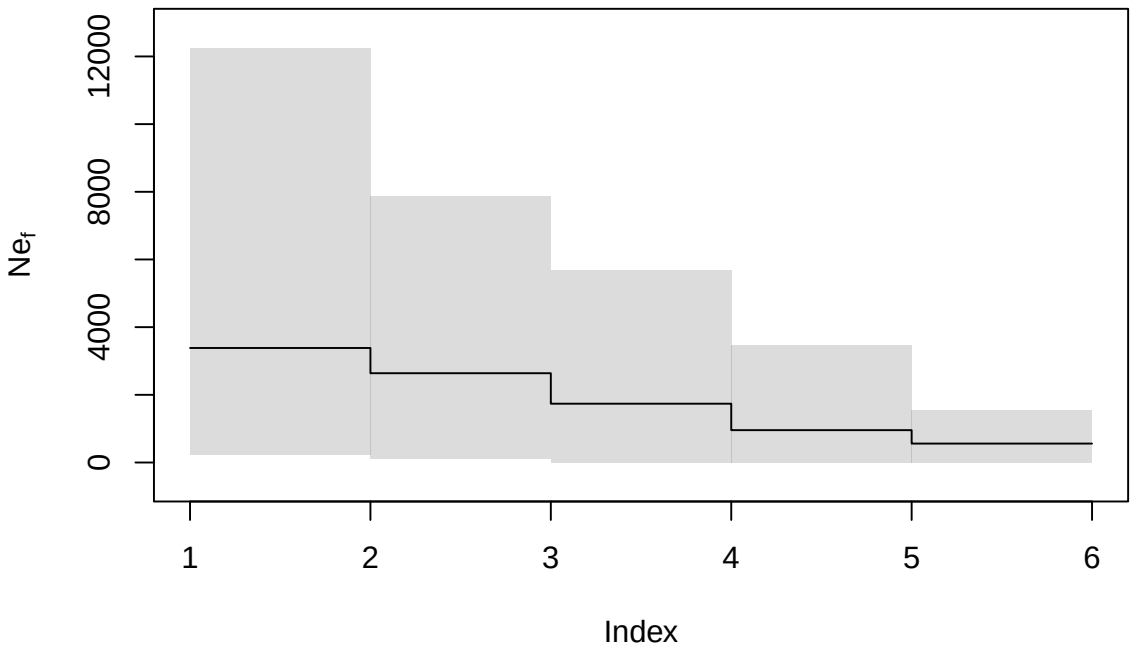
```
## freqParameter.3 2.483e-01 0.18505 0.0019504 0.0093175
## freqParameter.4 2.511e-01 0.19411 0.0020458 0.0088836
## gammaShape 1.641e+00 1.06191 0.0111923 0.0441228
## BayesianSkyline 9.265e+01 11.09674 0.1169569 0.2229926
## bPopSizes.1 1.276e-01 0.15852 0.0016707 0.0030500
## bPopSizes.2 8.778e-02 0.07107 0.0007491 0.0013701
## bPopSizes.3 6.056e-02 0.05536 0.0005835 0.0011734
## bPopSizes.4 3.513e-02 0.03508 0.0003697 0.0007503
## bPopSizes.5 1.755e-02 0.01359 0.0001433 0.0003064
## bGroupSizes.1 9.081e+00 6.87666 0.0724783 0.1621117
## bGroupSizes.2 9.096e+00 6.75887 0.0712368 0.1169094
## bGroupSizes.3 8.557e+00 6.61254 0.0696946 0.1196733
## bGroupSizes.4 8.350e+00 6.33568 0.0667765 0.1164574
## bGroupSizes.5 7.916e+00 6.18758 0.0652156 0.1543718
##
## 2. Quantiles for each variable:
##
## 2.5% 25% 50% 75% 97.5%
## posterior -2.261e+04 -2.259e+04 -2.258e+04 -2.257e+04
-2.256e+04
## likelihood -2.269e+04 -2.268e+04 -2.268e+04 -2.267e+04
-2.267e+04
## prior 7.069e+01 8.577e+01 9.404e+01 1.026e+02 1.198e+02
## treeLikelihood -2.269e+04 -2.268e+04 -2.268e+04
-2.267e+04 -2.267e+04
## TreeHeight 1.442e-02 2.273e-02 2.895e-02 3.643e-02
5.514e-02
## kappa 1.087e+01 2.512e+01 3.839e+01 6.147e+01 1.648e+02
## freqParameter.1 6.831e-03 7.215e-02 1.869e-01 3.526e-01
7.140e-01
## freqParameter.2 1.126e-02 1.018e-01 2.157e-01 3.918e-01
7.463e-01
## freqParameter.3 1.288e-02 9.470e-02 2.034e-01 3.747e-01
6.706e-01
## freqParameter.4 1.680e-02 9.709e-02 1.999e-01 3.603e-01
7.127e-01
## gammaShape 3.207e-01 8.987e-01 1.389e+00 2.144e+00
4.200e+00
## BayesianSkyline 7.167e+01 8.510e+01 9.240e+01 9.999e+01
1.153e+02
## bPopSizes.1 1.974e-02 5.682e-02 9.138e-02 1.473e-01
4.463e-01
## bPopSizes.2 1.319e-02 4.449e-02 7.125e-02 1.089e-01
2.621e-01
## bPopSizes.3 6.707e-03 2.631e-02 4.691e-02 7.656e-02
1.978e-01
## bPopSizes.4 3.183e-03 1.473e-02 2.581e-02 4.362e-02
1.233e-01
## bPopSizes.5 5.960e-04 8.327e-03 1.510e-02 2.337e-02
5.087e-02
## bGroupSizes.1 1.000e+00 4.000e+00 7.000e+00 1.300e+01
2.600e+01
## bGroupSizes.2 1.000e+00 4.000e+00 7.000e+00 1.300e+01
2.500e+01
## bGroupSizes.3 1.000e+00 3.000e+00 7.000e+00 1.200e+01
2.500e+01
## bGroupSizes.4 1.000e+00 3.000e+00 7.000e+00 1.200e+01
2.400e+01
```

```
## bGroupSizes.5 1.000e+00 3.000e+00 6.000e+00 1.200e+01
2.300e+01
```

```
Ne_sky <- beastio::getLogFileSubset(bdsky_trace, "bPopSizes")
Ne_hpd <- t(beastio::getHPDMedian(Ne_sky))

generation_time <- 27
gen_per_Ma <- 1e+06 / generation_time #37037.04 generations in 1 million year
# gen_per_Ma <- 1 / generation_time #0.03703704 generations in 1  year

Ne_hpd_gen <- Ne_hpd * gen_per_Ma

tryCatch({ #Error in clip(xlims[1], xlims[2], ylims[1], ylims[2])
  bdskytools::plotSkyline(1:5, Ne_hpd_gen, type = 'step',
                          ylab =  expression("Ne"[f]),new = TRUE)
}, error=function(e){})
```

```
## NULL
```

```
bdskytools::plotSkyline(1:5, Ne_hpd_gen, type = 'step',
                        ylab =  expression("Ne"[f]),new = FALSE)
```

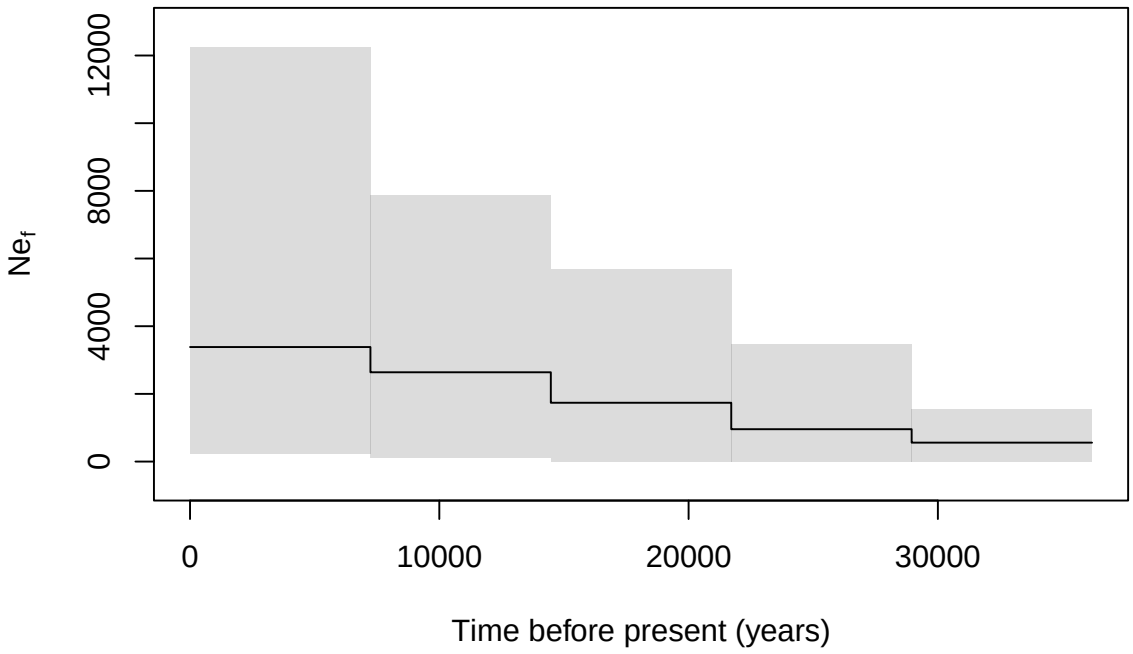



```
# delta_hpd <- beastio::getHPDMedian(bdsky_trace[, "mrca.age.auto_name_1."])
delta_hpd <- beastio::getHPDMedian(bdsky_trace[, "TreeHeight"]) #0.02894753 Ma or  28947.53 yr

med_age <- delta_hpd[2] * 1e6 #in years
# med_age <- delta_hpd[2]

time.seq <- seq(from = 0,
                to = med_age,
                length.out = 5)
# times.mrca <- 15e6 - time.seq
tryCatch({ #Error in clip(xlims[1], xlims[2], ylims[1], ylims[2])
```

```
  bdskytools::plotSkyline(time.seq,Ne_hpd_gen,type = 'step',
                          ylab =  expression("Ne"[f]),
                          xlab = "Time before present (years)",new = TRUE)
}, error=function(e){})
```

```
## NULL
```

```
bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'step',
                        ylab =  expression("Ne"[f]),
                        xlab = "Time before present (years)", new = FALSE)
```




```
dev.print(
  device = png,
  file = paste0("Bayesian_Skyline_plot_stepwise_bsp_ALL_unique.png"),
  res = 300,
  width = 30,
  height = 15,
  units = "cm"
)
```

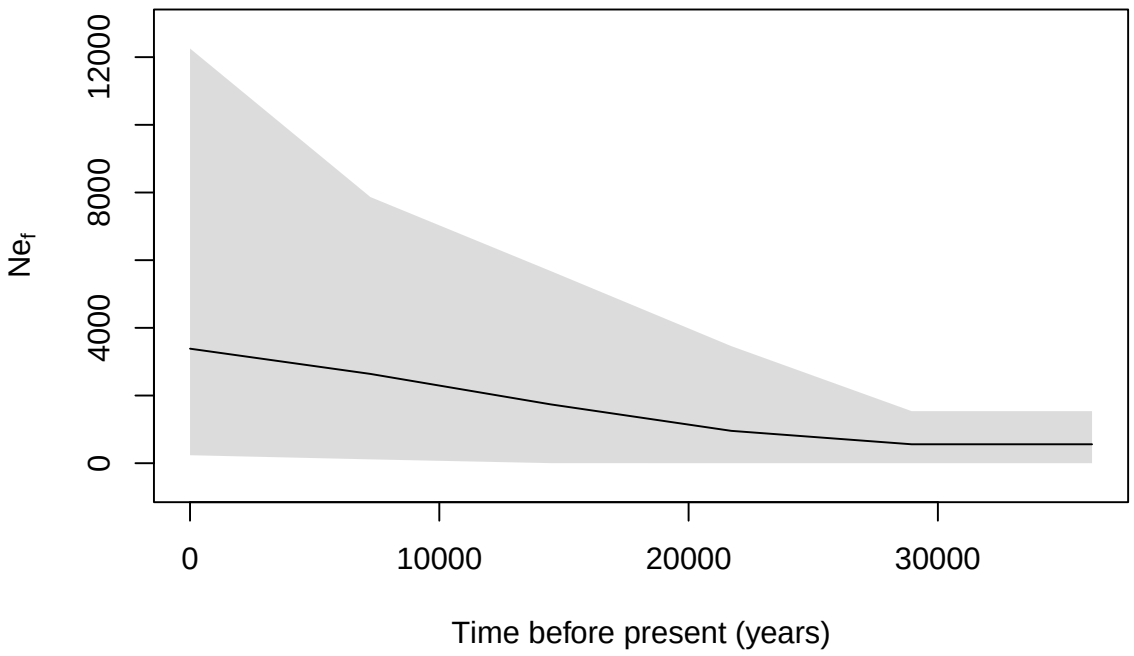
```
## pdf
##   2
```

```
tryCatch({ #Error in clip(xlims[1], xlims[2], ylims[1], ylims[2])
  bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'smooth',
                          ylab = expression("Ne"[f]),
                          xlab = "Time before present (years)", new = TRUE)
}, error=function(e){})
```

```
## NULL
```

```
bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'smooth',
                        ylab = expression("Ne"[f]),
```

```
                        xlab = "Time before present (years)", new = FALSE)
```




```
dev.print(
  device = png,
  file = paste0("Bayesian_Skyline_plot_smooth_bsp_ALL_unique.png"),
  res = 300,
  width = 30,
  height = 15,
  units = "cm"
)
```

```
## pdf
##   2
```

```
bsp.df <- data.frame(time = time.seq, lower = Ne_hpd_gen[1,],
                     med = Ne_hpd_gen[2,], upper = Ne_hpd_gen[3,],
                     pop = pop_1)
rownames(bsp.df) <- paste0(rownames(bsp.df), "_",bsp.df$pop)

# save(bsp.df, file = "Baysian_skyline_plot_combined.Rdata")
```

8.4.9 BEAST - Bayesian skyline per Region

```
for (pop in unique(meta$pop)) {
    print(pop)
    id <- meta$id[meta$pop == pop ]
    ggar.dna.new <- ggar.ml[attributes(ggar.ml)$labels %in% id, ,drop = TRUE]
    pop_1 <- gsub(pattern = " ", "_", x = pop)
    ape::write.FASTA(ggar.dna.new@dna$`6.ggar_Nucleotide_alignment_edited_HTPsites`,
                     file = paste0("Ggar_alignment_",pop_1, ".fasta"),
                     header = NULL, append = FALSE)
    fsta <- phylotools::read.fasta(file =  paste0("Ggar_alignment_",pop_1,
                                                  ".fasta"), clean_name = FALSE)
    phylotools::dat2phylip(fsta, outfile = paste0("Ggar_alignment_",pop_1,
                                                  ".phy"))
}
```

8.4.9.1 Subset by fasta files

```
## [1] "Papua New Guinea"
## [1] "Van Diemen Gulf"
## [1] "Daly River"
## [1] "Cambridge Gulf"
## [1] "King Sound"
```

```
for (pop in unique(meta$pop)) {
    pop_1 <- gsub(pattern = " ", "_", x = pop)
    print(pop_1)
    fasta_filename <- paste0("Ggar_alignment_", pop_1, ".fasta")

    inference_model <-
      beautier::create_inference_model(
        site_model = beautier::create_site_model_hky(# based on IQtree & Li et al. 2015
          gamma_site_model = beautier::create_gamma_site_model(
            gamma_cat_count = "4",
            gamma_shape = "1.0",
            prop_invariant = "0.0",
            gamma_shape_prior_distr = NA,
            freq_equilibrium = "estimated",
            freq_prior_uniform_distr_id = 1000
          ),
          freq_equilibrium = "estimated"
        ),
        tree_prior = beautier::create_tree_prior_cbs(
          group_sizes_dimension = 5,
          b_pop_sizes_param = beautier::create_b_pop_sizes_param(
            id = NA,
            upper = "50000.0",
            value = beautier::create_distr_log_normal(m = 6.91, s = 2.35,
                                                      value = 6.91, lower = 2.30,
                                                      upper = 10.81
                                                      )
            #curve(dlnorm(x, meanlog=10, sdlog=1), from=0, to=50000)
          ),
          pop_sizes_scaler_scale_factor = 0.5 #for mtDNA
        ),
        clock_model = beautier::create_strict_clock_model(
          clock_rate_param = beautier::create_clock_rate_param(
```

```
            value = 6.2e-9,
            #0.5-1% mutations per site per Ma (martin et al. 1992)
            #6.3e-3 mutations per site per Ma (2.0 × 10-3 Keeney and Heist 2006; 6.05 × 10-3 Nance e
            #0.62% per site per million years (Galván-Tirado et al 2013)
            estimate = FALSE,
            id = NA
          ),
          clock_rate_distr = beautier::create_distr_log_normal(m = -18.88,
                                                               s = 1.175,
                                                               value = -18.88,
                                                               lower = 1e-20,
                                                               upper = 1),
          # clock_rate_distr = beautier::create_distr_log_normal(m = -9.11, s = 1.25, value = -9.11,
          #                                                      lower = 0, upper = 1),
          rate_scaler_factor = 1
        ),
        mcmc = beautier::create_mcmc(
          chain_length = 1e+07,
          store_every = 5e+03,
          pre_burnin = 1e4,
          n_init_attempts = 3,
          tracelog = beautier::create_tracelog(filename =
                                                 paste0("tracelog_", pop_1, ".log")),
          screenlog = beautier::create_screenlog(filename =
                                                   paste0("screenlog_", pop_1, ".log")),
          treelog = beautier::create_treelog(filename =
                                               paste0("treelog_", pop_1, ".log"))
        ),
        beauti_options = beautier::create_beauti_options()
      )

    beast2_input_file <- paste0("beast2_", pop_1, ".xml")
    beautier::create_beast2_input_file_from_model(
      input_filename = fasta_filename,
      inference_model = inference_model,
      output_filename = beast2_input_file
    )


    if (beastier::is_beast2_installed()) {
      beastier::is_beast2_input_file(beast2_input_file,
                                     show_warnings = FALSE,
                                     verbose = FALSE)
    }
}
```

8.4.9.2 Create BEAUTI file

```
## [1] "Papua_New_Guinea"
## [1] "Van_Diemen_Gulf"
## [1] "Daly_River"
## [1] "Cambridge_Gulf"
## [1] "King_Sound"
```

```
for (pop in unique(meta$pop)) {
    pop_1 <- gsub(pattern = " ", "_", x = pop)
    print(pop_1)
```

```
    if (beastier::is_beast2_installed()) {
      beast2_options <- beastier::create_beast2_options(
        input_filename = paste0("beast2_", pop_1, ".xml"),
        output_state_filename = paste0("beast2_", pop_1, ".xml.state"),
        n_threads = 4,
        verbose = TRUE
      )
      beastier::check_can_create_file(beast2_options$output_state_filename)
      beastier::check_can_create_treelog_file(beast2_options)
      run_beast2_from_options(beast2_options = beast2_options)
      testthat::expect_true(file.exists(beast2_options$output_state_filename))
    }

    if (beastier::is_beast2_installed()) {
      out <- babette::bbt_run_from_model(
        fasta_filename = fasta_filename,
        inference_model = inference_model,
        beast2_options = beast2_options
      )
    }
}
```

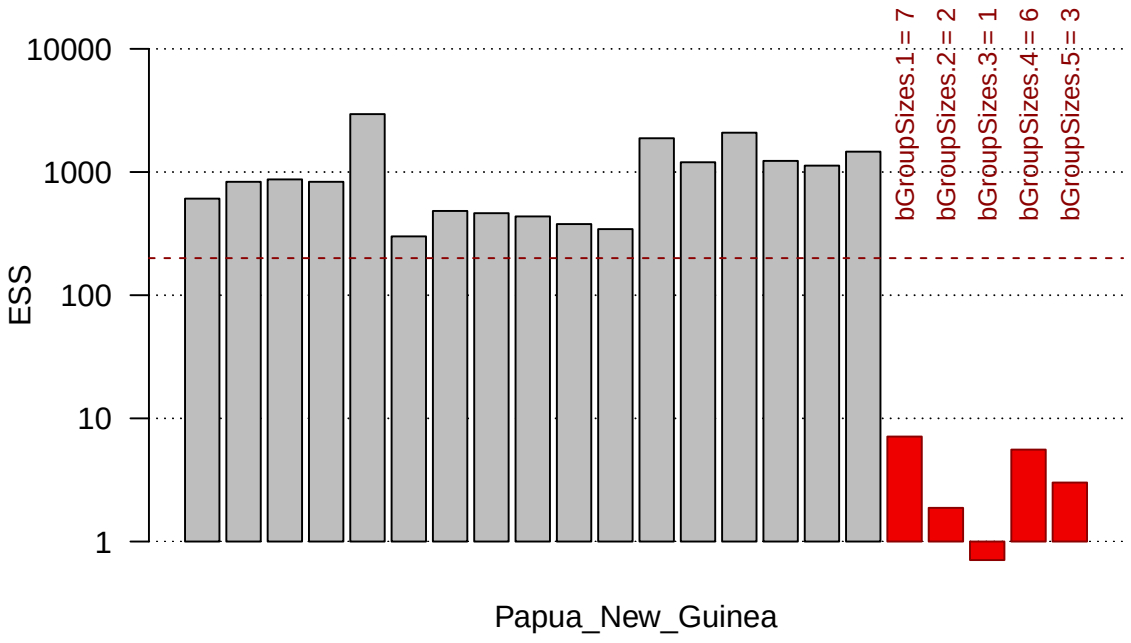
8.4.9.3 Run BEAST

```
for (pop in unique(meta$pop)) {
  pop_1 <- gsub(pattern = " ", "_", x = pop)
  print(pop_1)
  print(paste0("sample size = ", length(meta$pop[meta$pop == pop])))

  folder_1 <- paste0("beast_bspe_", pop_1, "/")
  tracelog <- paste0(folder_1, "tracelog_", pop_1, ".log")
  estimates <- tracerer::parse_beast_tracelog_file(tracelog_filename = tracelog)
  estimates_burn <- tracerer::remove_burn_ins(estimates, burn_in_fraction = 0.1)
  esses <- tracerer::calc_esses(estimates_burn, sample_interval = 1000)
  esses <- t(esses)
  colnames(esses) <- c("ESS")
  knitr::kable(esses)

  bdsky_trace   <- beastio::readLog(tracelog, burnin = 0.1)
  beastio::checkESS(bdsky_trace)
  beastio::checkESS(
    bdsky_trace,
    cutoff = 200,
    plot = TRUE,
    log = 'y',
    ylim = c(1, 10000),
    title = pop_1,
    plot.grid = TRUE
  )

  sum_stats <- tracerer::calc_summary_stats(estimates_burn, sample_interval = 5000)
  knitr::kable(sum_stats) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")


  ggplot2::ggplot(data = estimates_burn, ggplot2::aes(x = Sample)) +
```

```
    ggplot2::geom_line(ggplot2::aes(y = posterior)) +
    ggplot2::ggtitle("Posterior")


  ggplot2::ggplot(data = estimates_burn, ggplot2::aes(x = posterior)) +
    ggplot2::geom_histogram(binwidth = 0.21) +
    ggplot2::ggtitle(paste0(pop_1, ": Posterior distribution")) +
    ggplot2::scale_x_continuous() +
    ggplot2::theme_bw()


  ggplot2::ggplot(data = estimates_burn, ggplot2::aes(x = Sample,
                                                      y = posterior)) +
    ggplot2::geom_line() +
    ggplot2::ggtitle(paste0(pop_1, ": Trace plot")) +
    ggplot2::theme_bw()


  ggplot2::ggplot(data = estimates_burn, ggplot2::aes(x = TreeHeight)) +
    ggplot2::geom_histogram() +
    ggplot2::ggtitle(paste0(pop_1, ": TreeHeight")) +
    ggplot2::theme_bw()

}
```

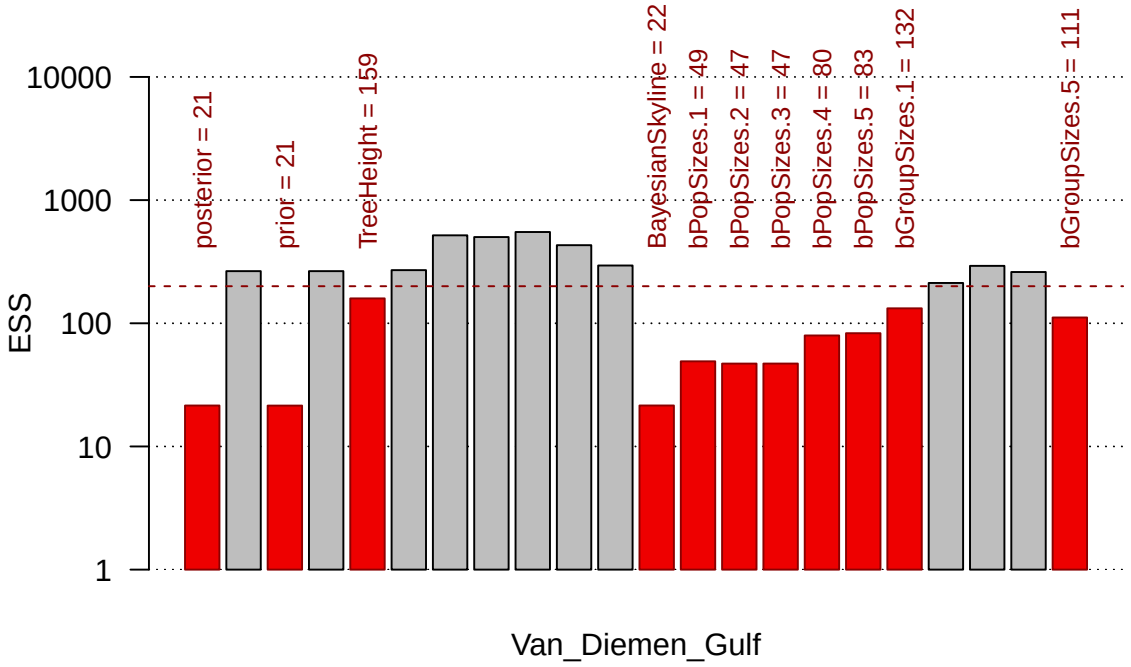
8.4.9.4 Summary stats

```
## [1] "Papua_New_Guinea"
## [1] "sample size = 6"
```



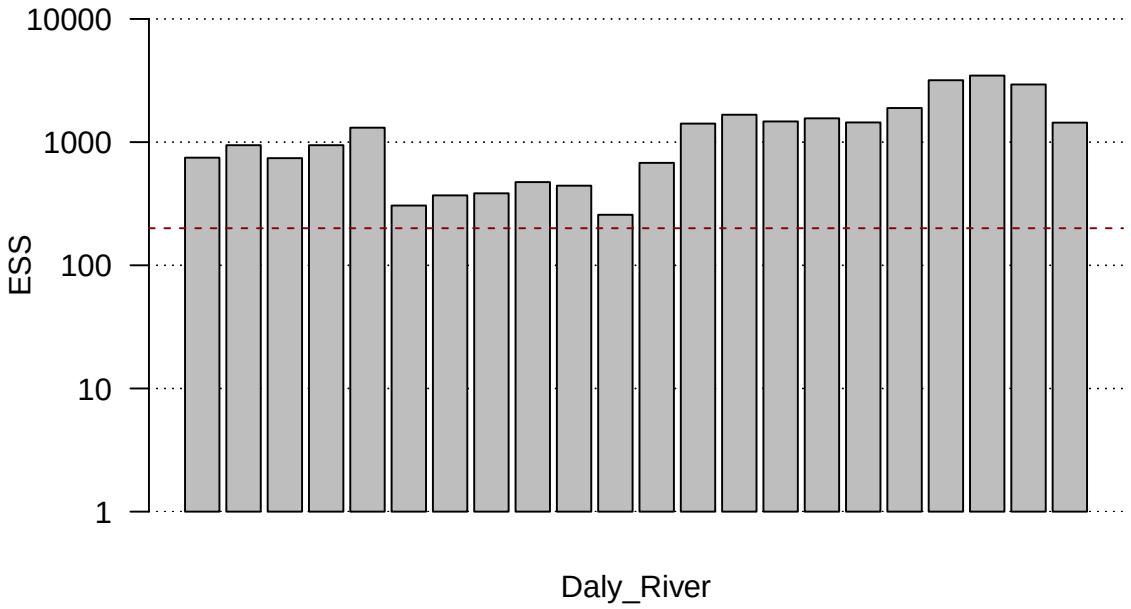

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "Van_Diemen_Gulf"
## [1] "sample size = 303"
```



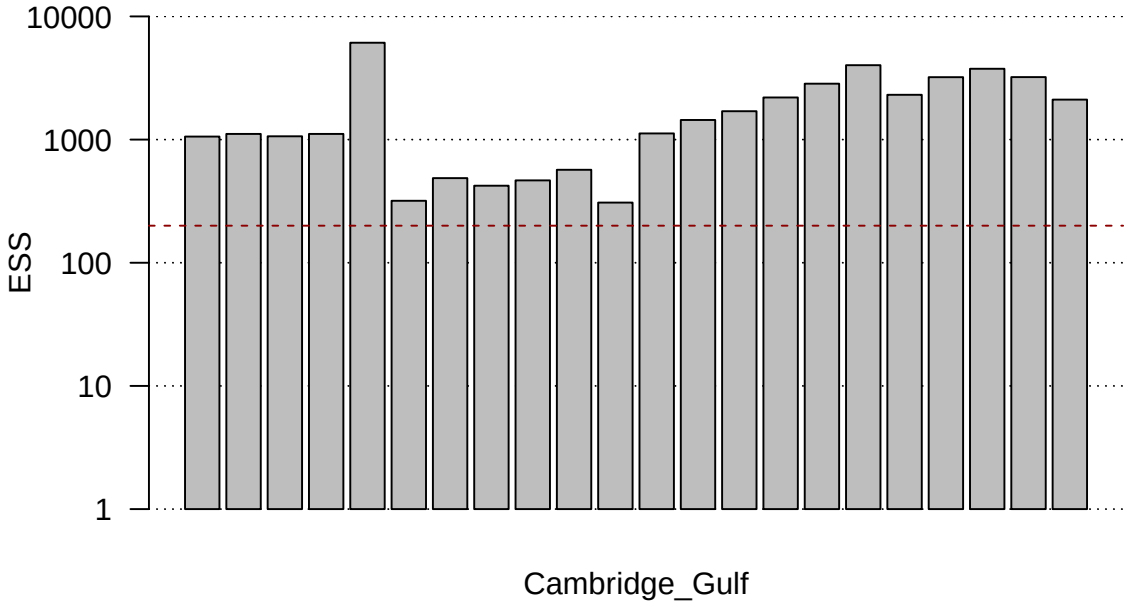

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "Daly_River"
## [1] "sample size = 30"
```



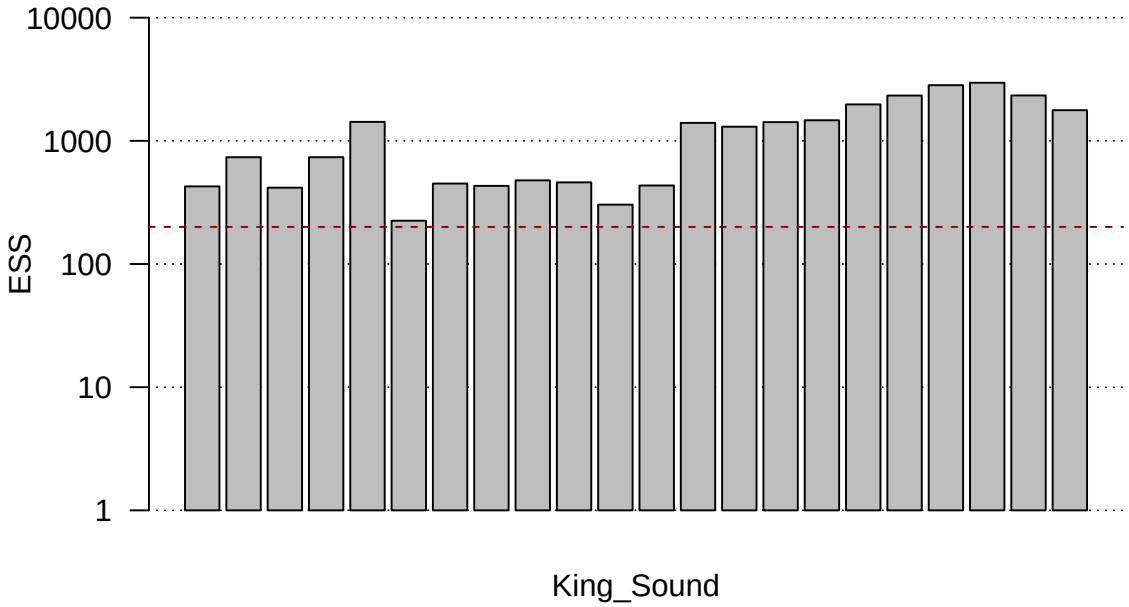

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "Cambridge_Gulf"
## [1] "sample size = 30"
```



```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "King_Sound"
## [1] "sample size = 14"
```



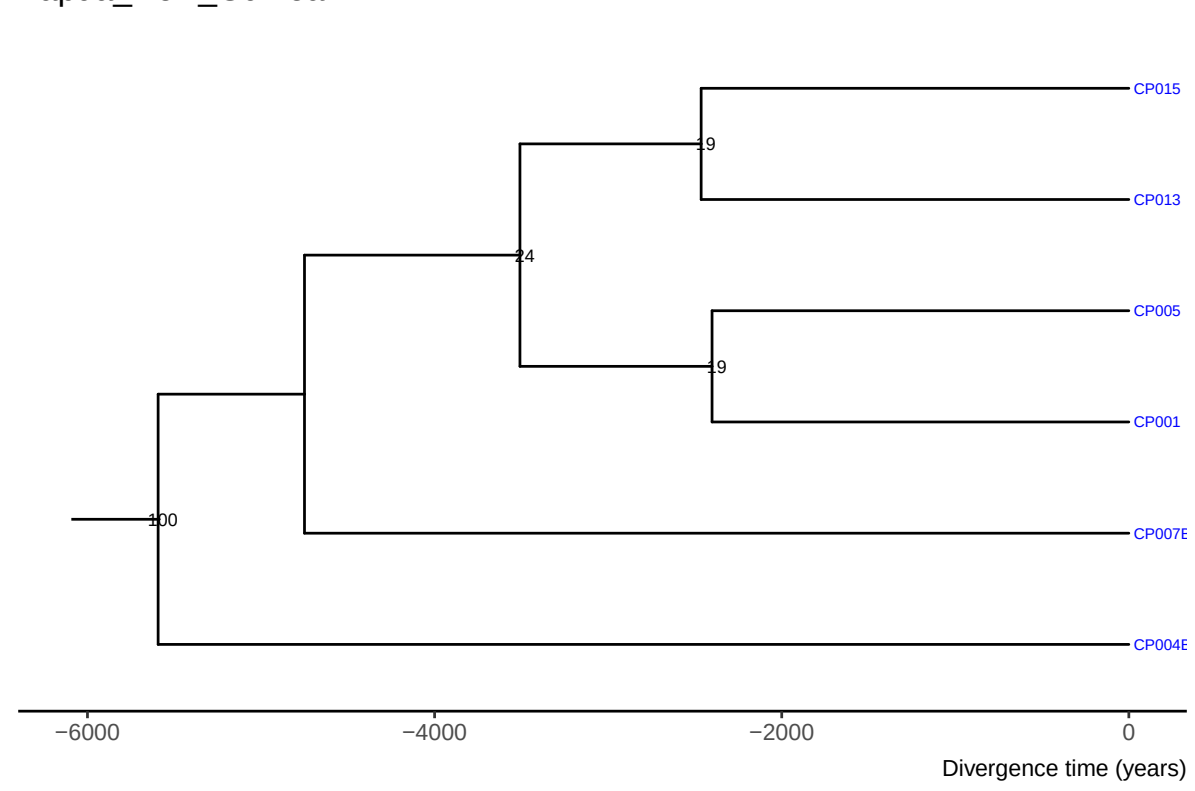

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
for (pop in unique(meta$pop)) {
  pop_1 <- gsub(pattern = " ", "_", x = pop)
  print(pop_1)

  folder_1 <- paste0("beast_bspe_", pop_1, "/")
  treelog <- paste0(folder_1, "treelog_", pop_1, ".log")

  trees <- tracerer::parse_beast_trees(filename = treelog)
  treefile <- paste0(folder_1, "Result_trees.tre") #with treeAnnotator
  # phyloch::treeannotator(input = treelog)

  ggar.tree <- ape::read.nexus(treefile)
  boot_support <- ape::prop.clades(ggar.tree, trees)
  boot_support <- boot_support / 100
  boot_support <- round(boot_support, 0)

  ggar.tree$root.edge <- 500


  p1 <- ggtree::ggtree(tr = ggar.tree, color = "black")
  num_tips <- length(p1$data$label[p1$data$isTip == TRUE])
  num_nodes <- length(p1$data$label[p1$data$isTip == FALSE])
```

```
  nodes <- paste0("t", 1:num_nodes)
  p1$data$label[p1$data$isTip == FALSE] <- nodes
  bs <- data.frame(label = p1$data$label,
                   bootstrap = c(rep(NA, num_tips), boot_support))
  p1 <- p1 %<+% bs

  p1 <- p1 +
    ggtree::geom_tiplab(size = 2, color = 'blue') +
    ggtree::geom_rootedge() +
    ggtree::geom_nodelab(aes(label = bootstrap), size = 2.5, ) +
    ggtree::theme_tree2() +
    ggplot2::ggtitle(pop_1) +
    ggplot2::labs(caption = "Divergence time (years)")
  p2 <- ggtree::revts(p1)
  print(p2)

  ggplot2::ggsave(
    plot = p2,
    filename = paste0("BEAST_bsp_", pop_1, "_ggtree.png"),
    device = "png",
    width = 30,
    height = 15,
    units = "cm"
  )
}
```

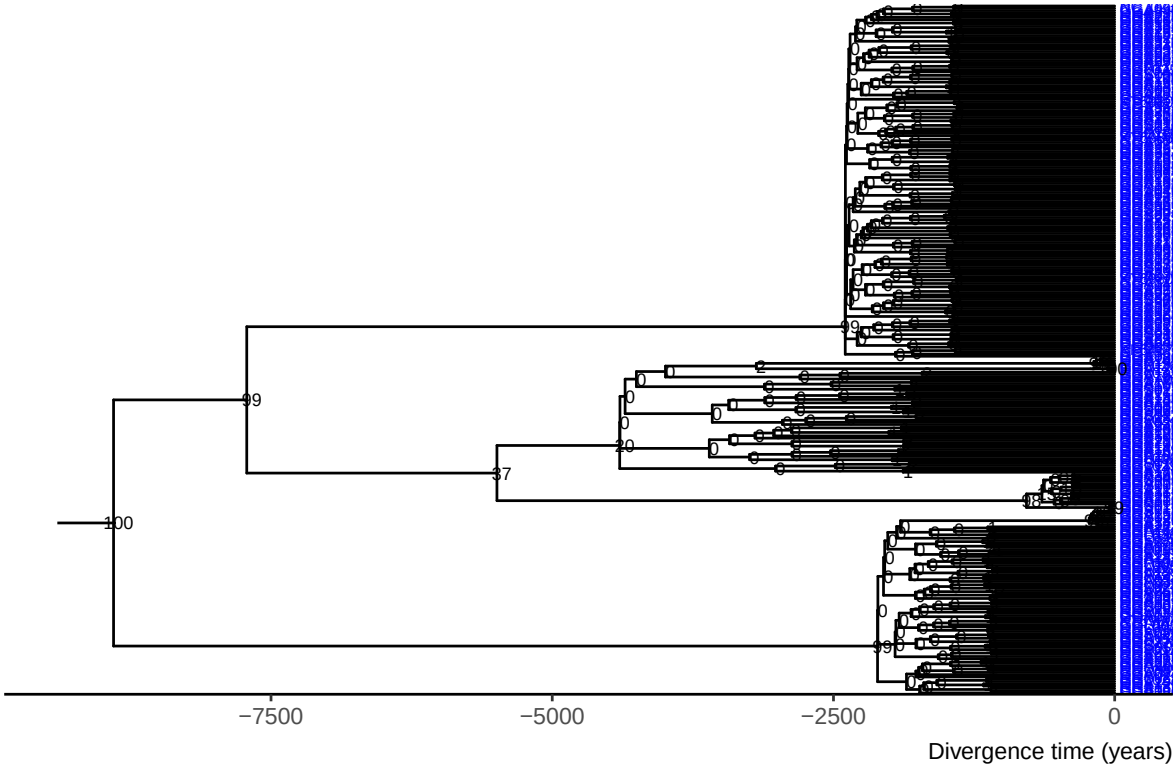
8.4.9.5 Plot Tree

```
## [1] "Papua_New_Guinea"
```



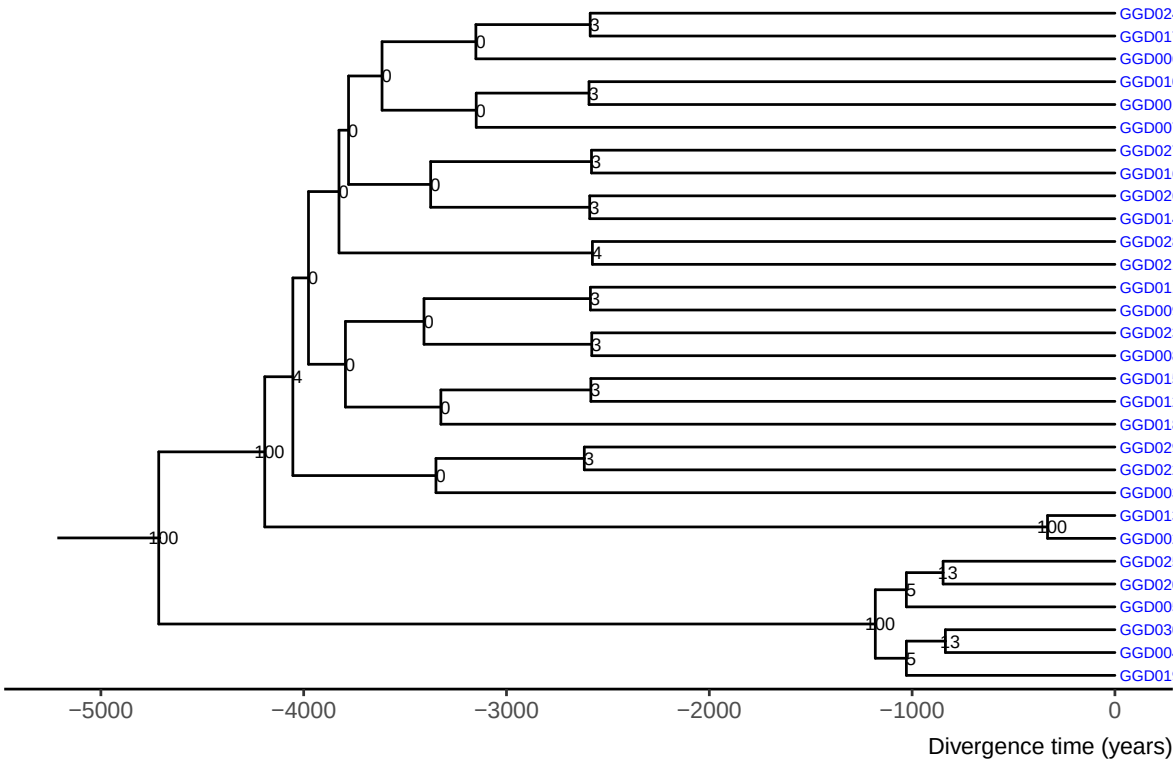

```
## [1] "Van_Diemen_Gulf"
```

Van_Diemen_Gulf



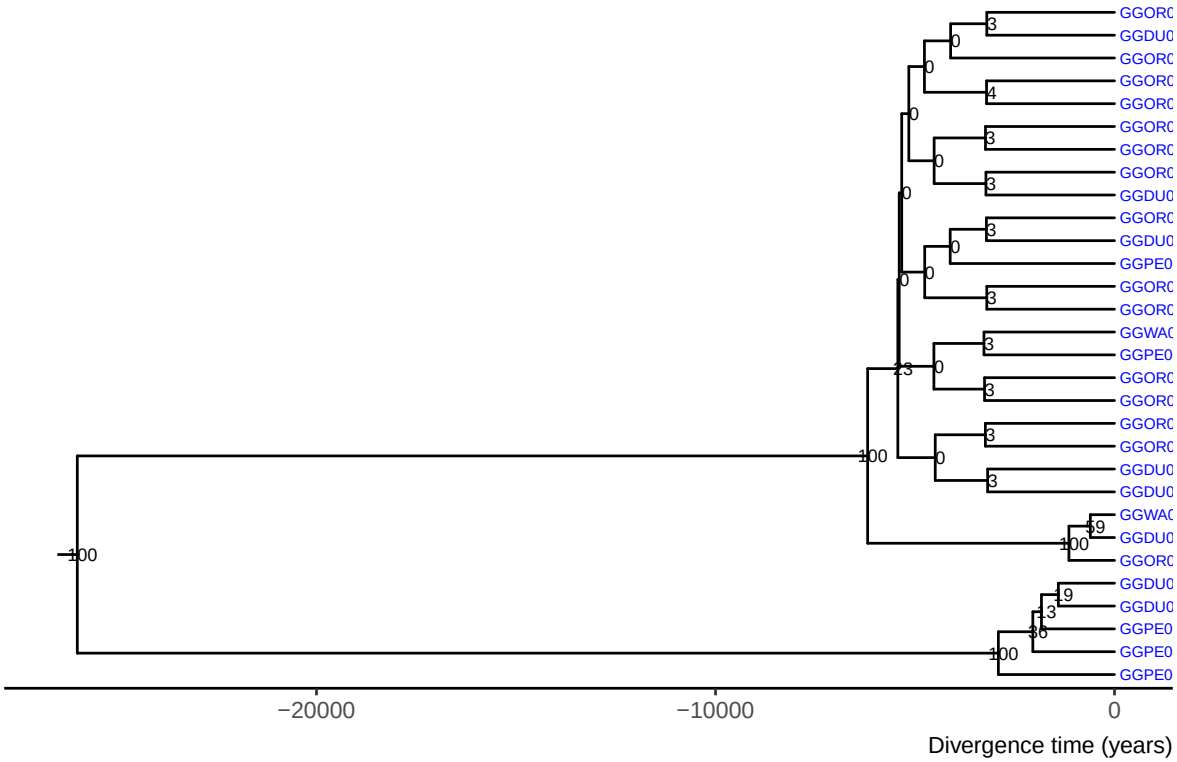

```
## [1] "Daly_River"
```

Daly_River



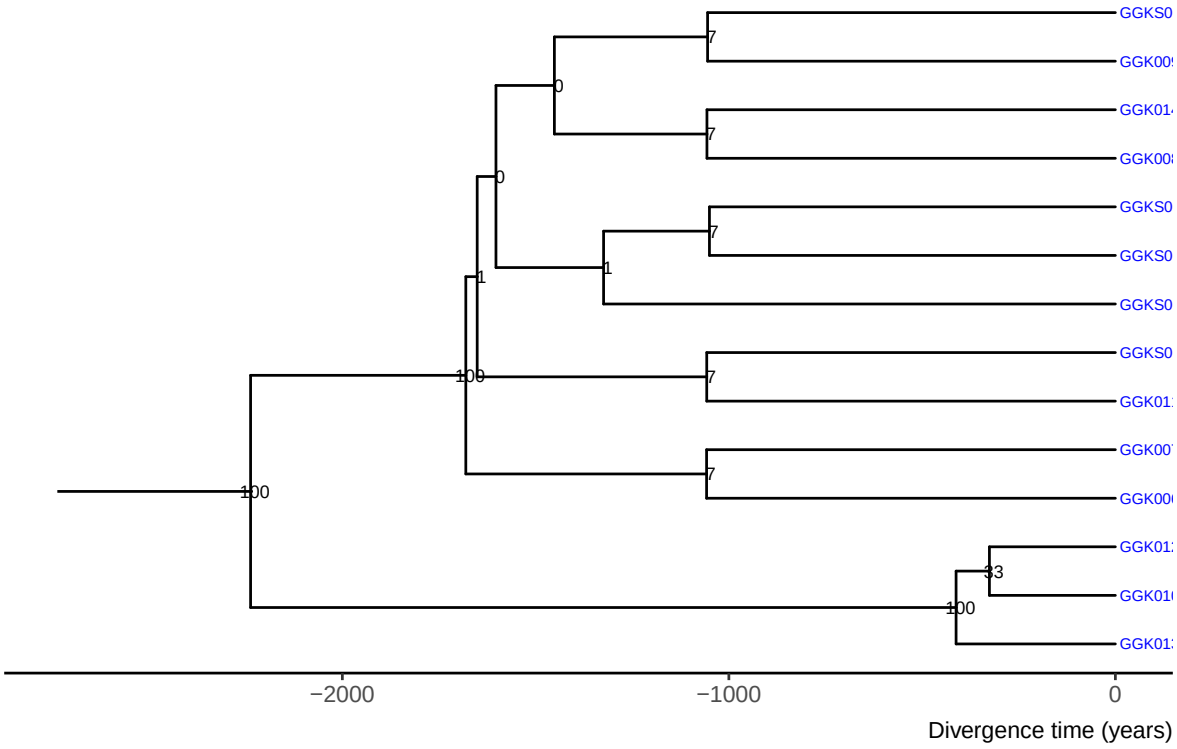

```
## [1] "Cambridge_Gulf"
```

Cambridge_Gulf




```
## [1] "King_Sound"
```

King_Sound




```
print(load("Baysian_skyline_plot_combined.Rdata"))
```

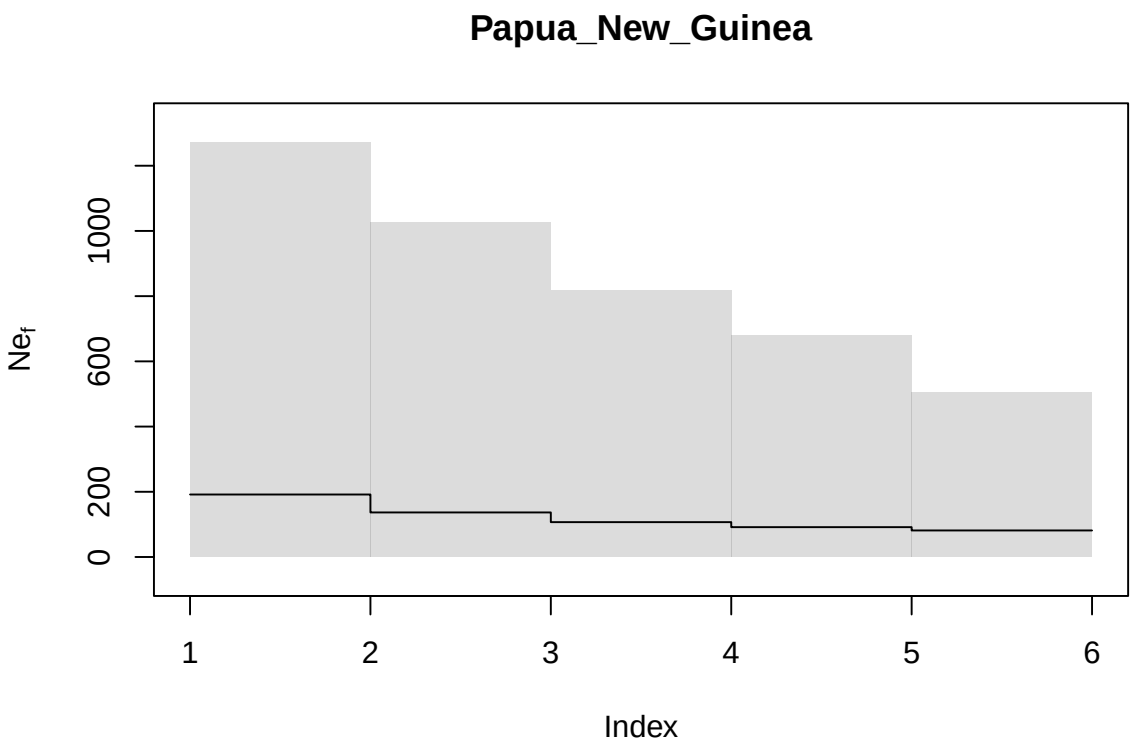
8.4.9.6 Plot BSP

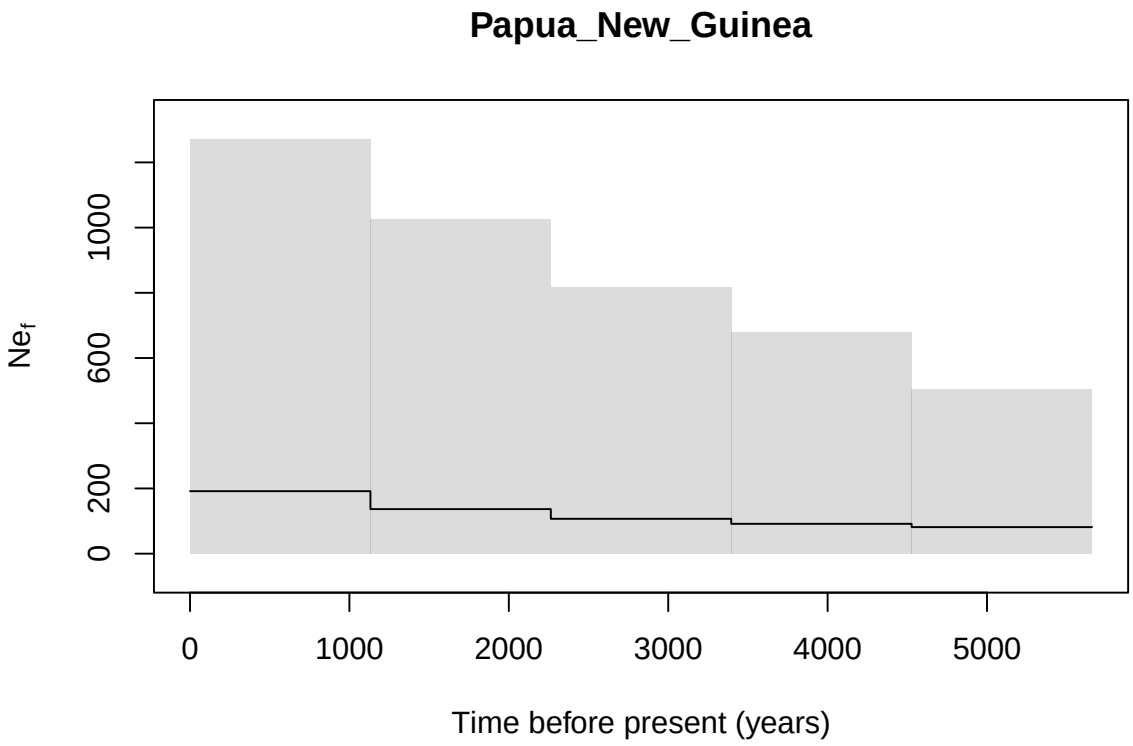
```
## [1] "bsp.df"
```

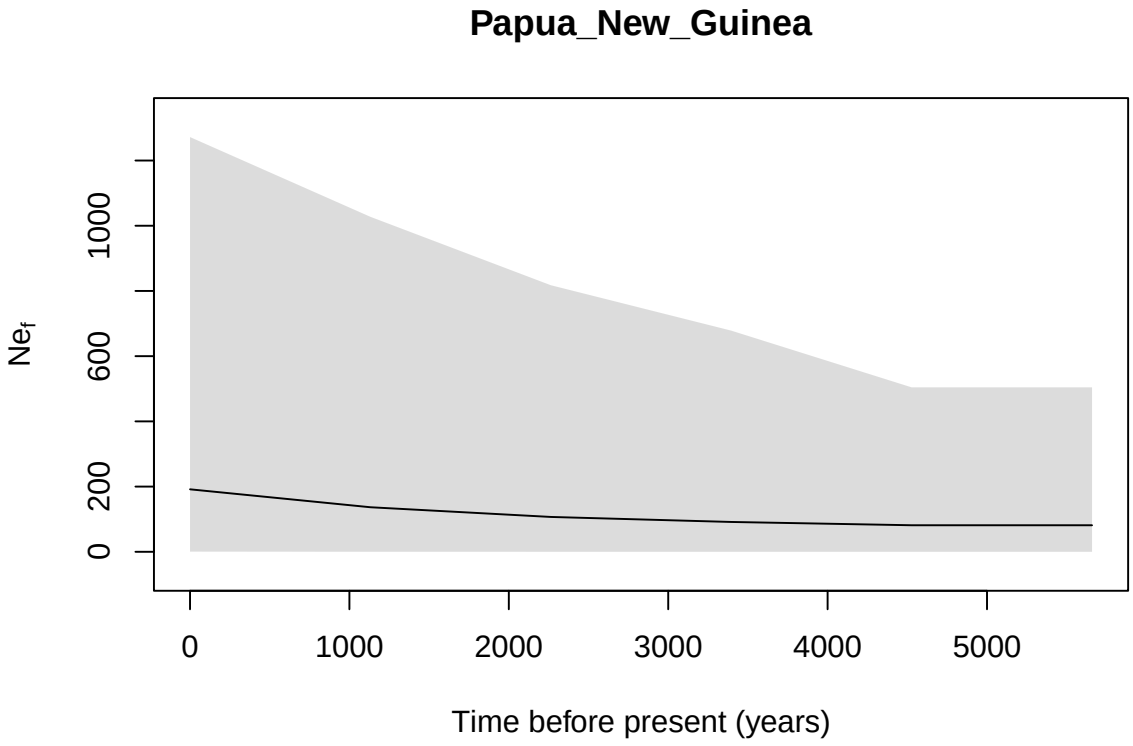
```
for (pop in unique(meta$pop)) {
  pop_1 <- gsub(pattern = " ", "_", x = pop)
  print(pop_1)
  print(paste0("sample size = ", length(meta$pop[meta$pop == pop])))
  folder_1 <- paste0("beast_bspe_", pop_1, "/")
  tracelog <- paste0(folder_1, "tracelog_", pop_1, ".log")
  bdsky_trace   <- beastio::readLog(tracelog, burnin = 0.1)

  summary(bdsky_trace)

  Ne_sky <- beastio::getLogFileSubset(bdsky_trace, "bPopSizes")
  Ne_hpd <- t(beastio::getHPDMedian(Ne_sky))

  generation_time <- 27
  # gen_per_Ma <- 1e+06 / generation_time #37037.04 generations in 1 million year
  gen_per_Ma <- 1 / generation_time #0.03703704 generations in 1  year
  Ne_hpd_gen <- Ne_hpd*gen_per_Ma
  tryCatch({ #Error in clip(xlims[1], xlims[2], ylims[1], ylims[2])
    bdskytools::plotSkyline(1:5, Ne_hpd_gen, type = 'step',  main = pop_1,
                            ylab =  expression("Ne"[f]), new = TRUE)
  }, error=function(e){})
  bdskytools::plotSkyline(1:5, Ne_hpd_gen, type = 'step',
                          ylab =  expression("Ne"[f]), new = FALSE)

  delta_hpd <- beastio::getHPDMedian(bdsky_trace[, "TreeHeight"]) #0.027243441 Ma or  27243.44 yr
  # med_age <- delta_hpd[2] * 1e6
  med_age <- delta_hpd[2]
  time.seq <- seq(from = 0, to = med_age, length.out = 5)
  # times.mrca <- 15e6 - time.seq
  tryCatch({
    bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'step',  main = pop_1,
                            ylab =  expression("Ne"[f]),
                            xlab = "Time before present (years)", new = TRUE)
  }, error=function(e){})
  bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'step',
                          ylab =  expression("Ne"[f]),
                          xlab = "Time before present (years)", new = FALSE)
  dev.print(
    device = png,
    file = paste0("Bayesian_Skyline_plot_stepwise_", pop_1, ".png"),
    res = 300,
    width = 30,
    height = 15,
    units = "cm"
  )
  tryCatch({
    bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'smooth', main = pop_1,
                            ylab = expression("Ne"[f]),
                            xlab = "Time before present (years)", new = TRUE)
  }, error=function(e){})
  bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'smooth',
                          ylab = expression("Ne"[f]),
                          xlab = "Time before present (years)", new = FALSE)
  dev.print(
    device = png,
```

```
    file = paste0("Bayesian_Skyline_plot_smooth_", pop_1, ".png"),
    res = 300,
    width = 30,
    height = 15,
    units = "cm"
  )
  bsp.df.pop <- data.frame(time = time.seq, lower = Ne_hpd_gen[1,],
                           med = Ne_hpd_gen[2,], upper = Ne_hpd_gen[3,],
                           pop = pop_1)
  rownames(bsp.df.pop) <- paste0(rownames(bsp.df.pop), "_",bsp.df.pop$pop)
  bsp.df <- rbind(bsp.df, bsp.df.pop)
}
```

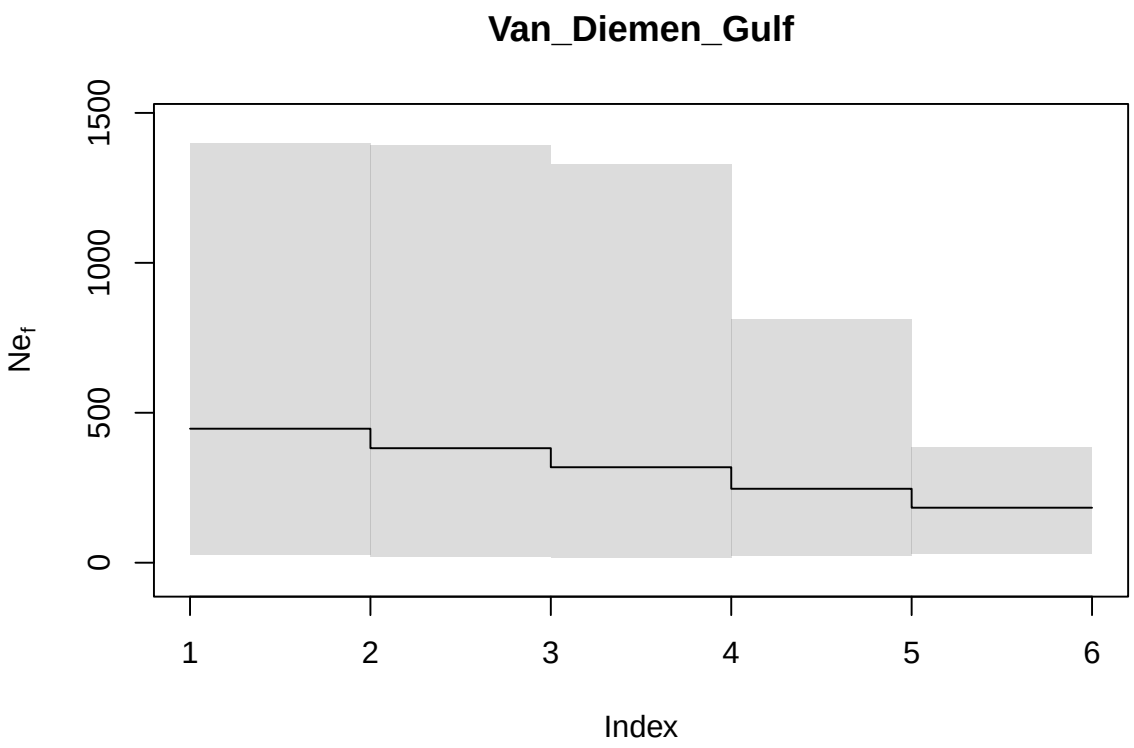
```
## [1] "Papua_New_Guinea"
## [1] "sample size = 6"
```

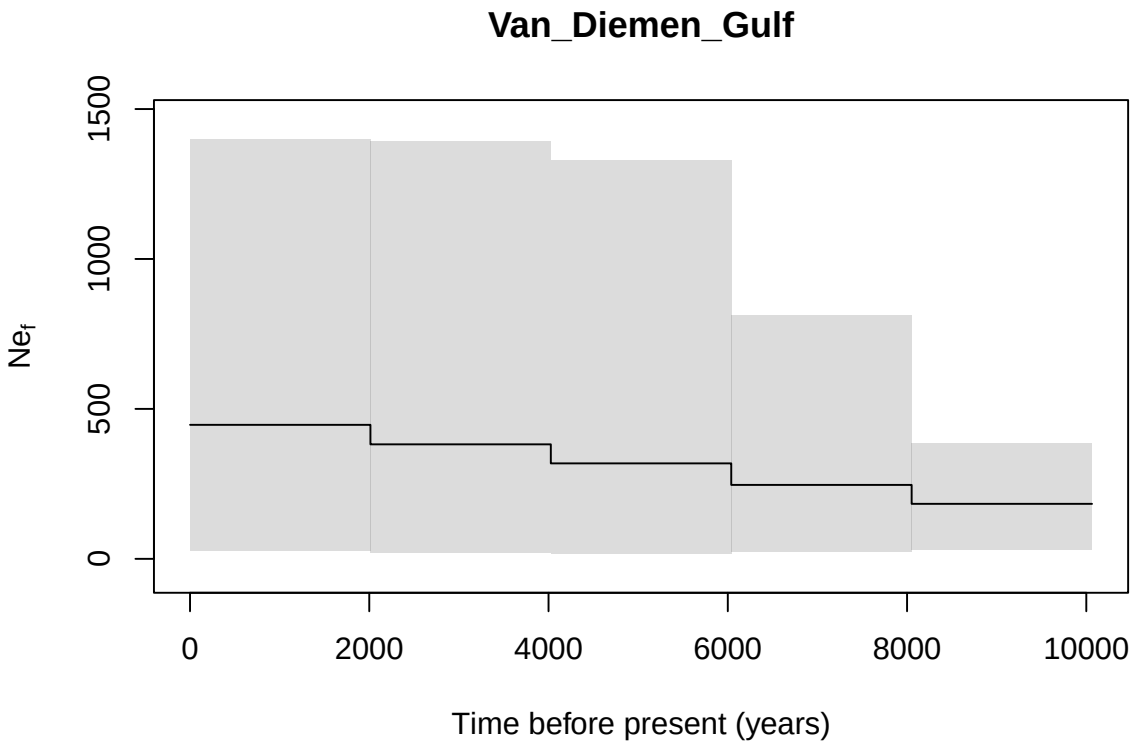


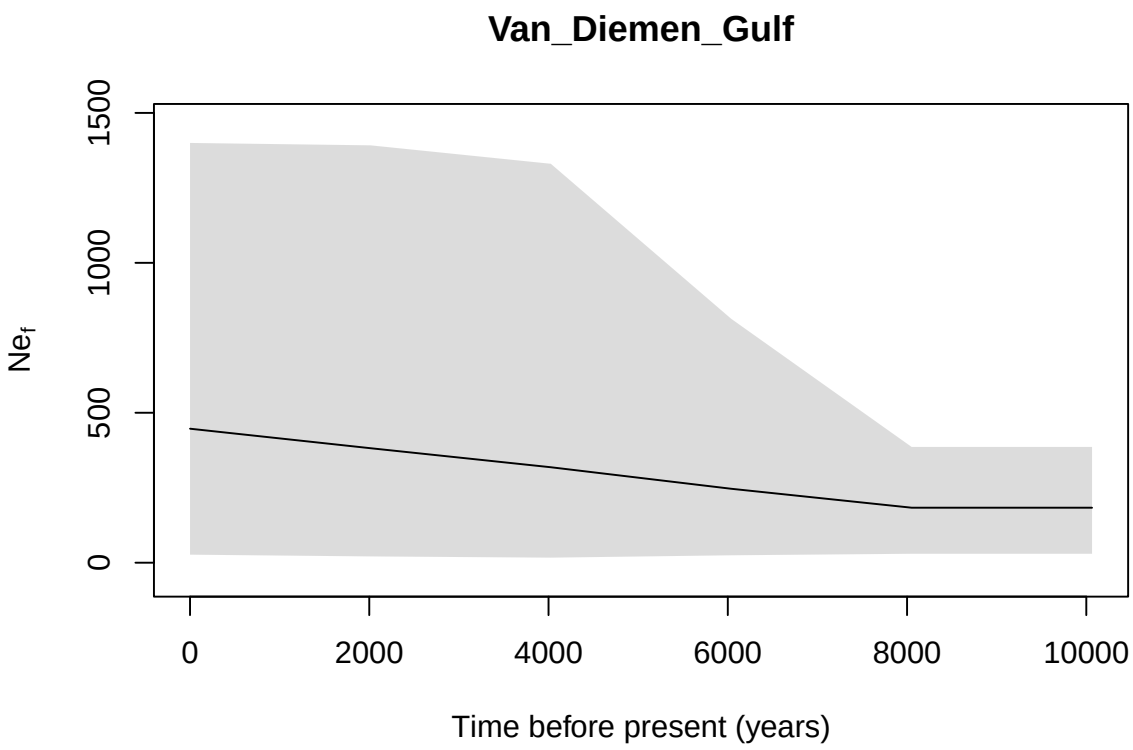




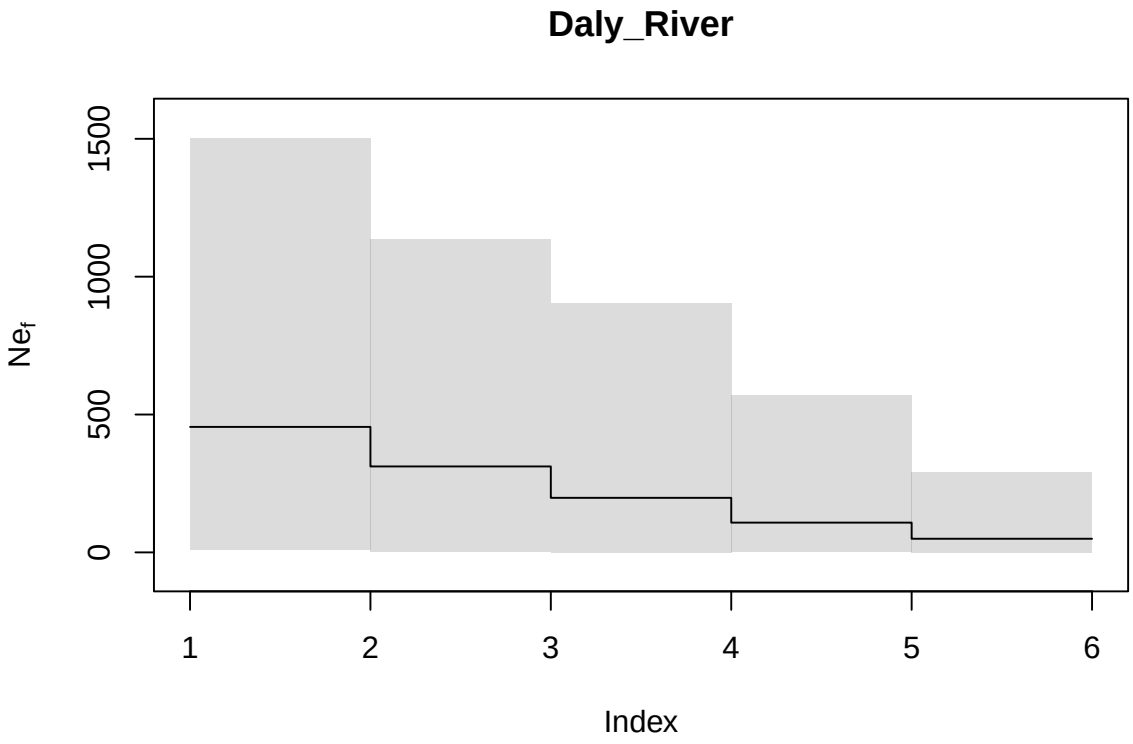

```
## [1] "Van_Diemen_Gulf"
## [1] "sample size = 303"
```

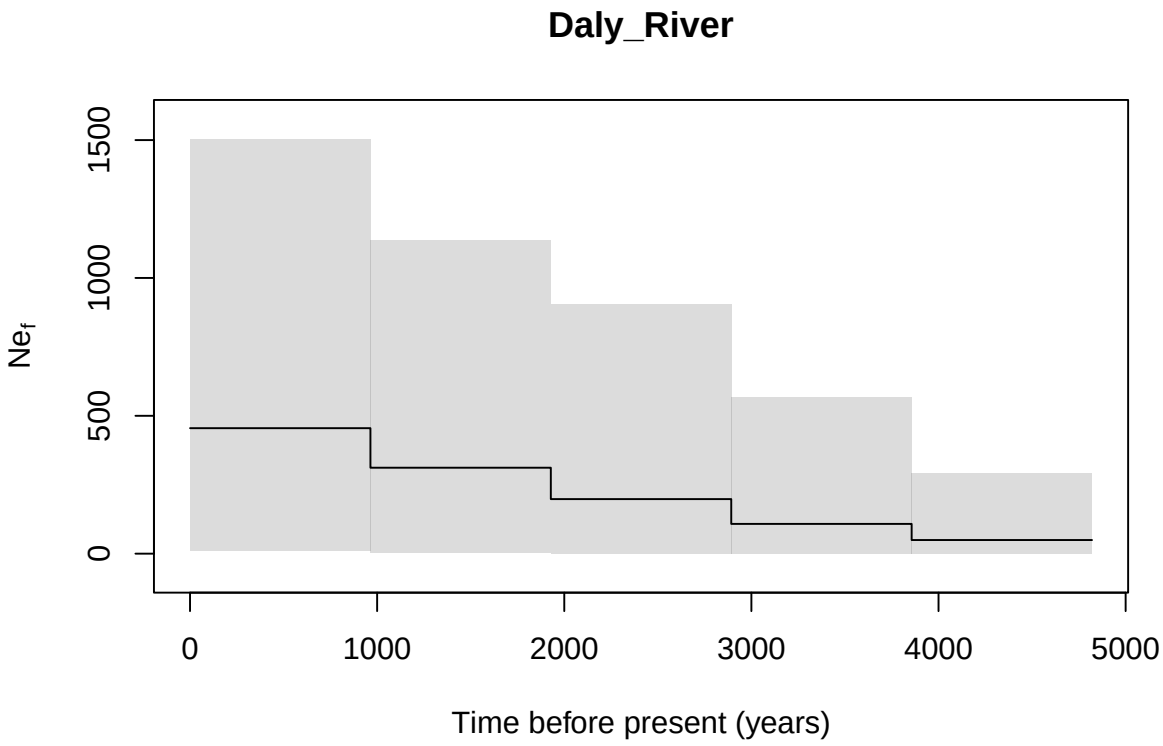

Van_Diemen_Gulf
Ne_f
0
500
1000
1500
1
2
3
4
5
6
Index

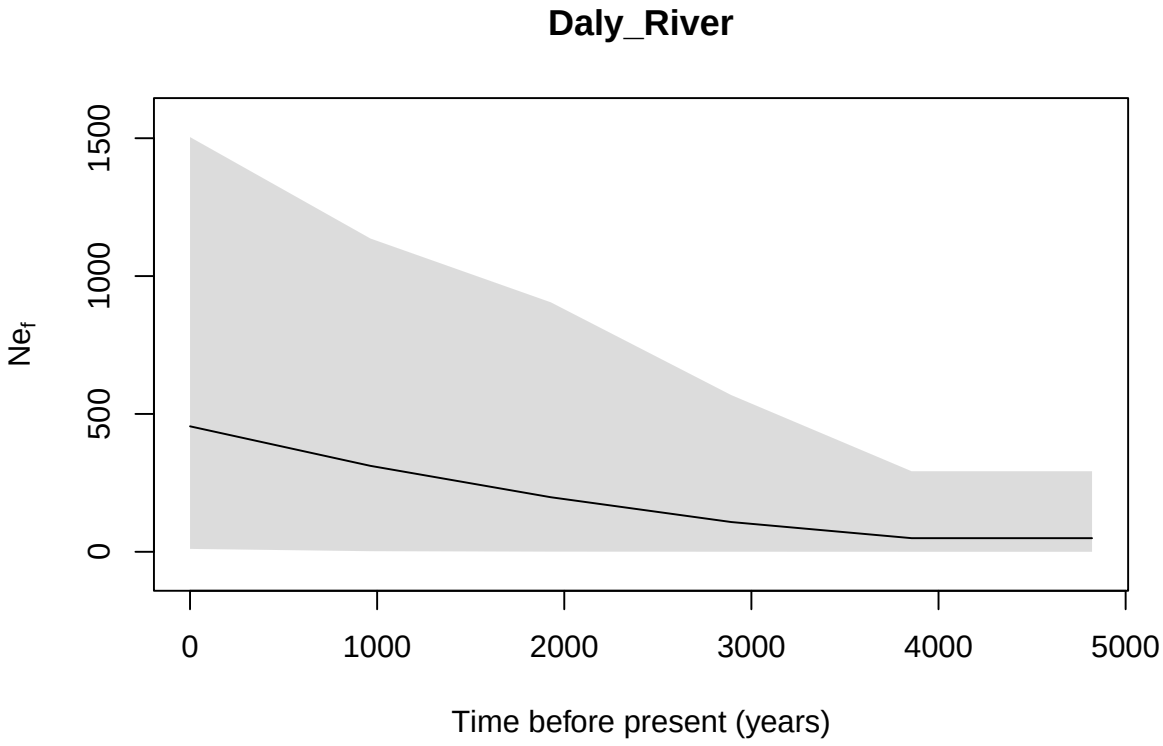

Van_Diemen_Gulf
Ne_f
0
500
1000
1500
0
2000
4000
6000
8000
10000
Time before present (years)



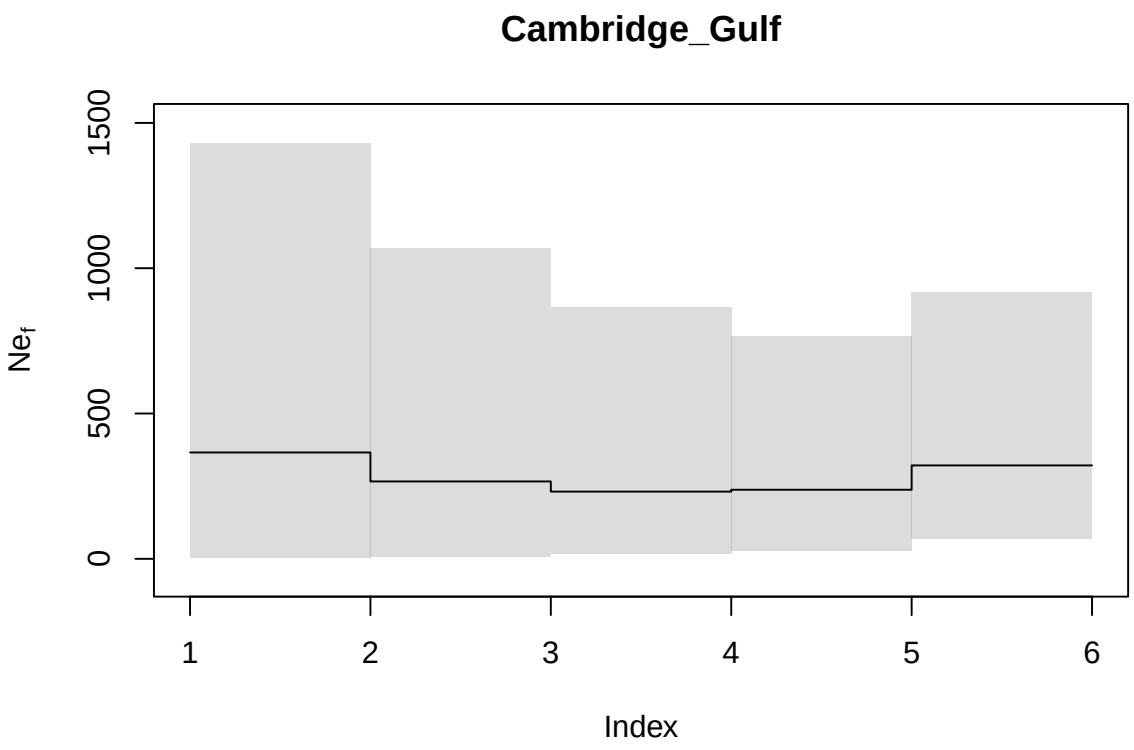

```
## [1] "Daly_River"
## [1] "sample size = 30"
```

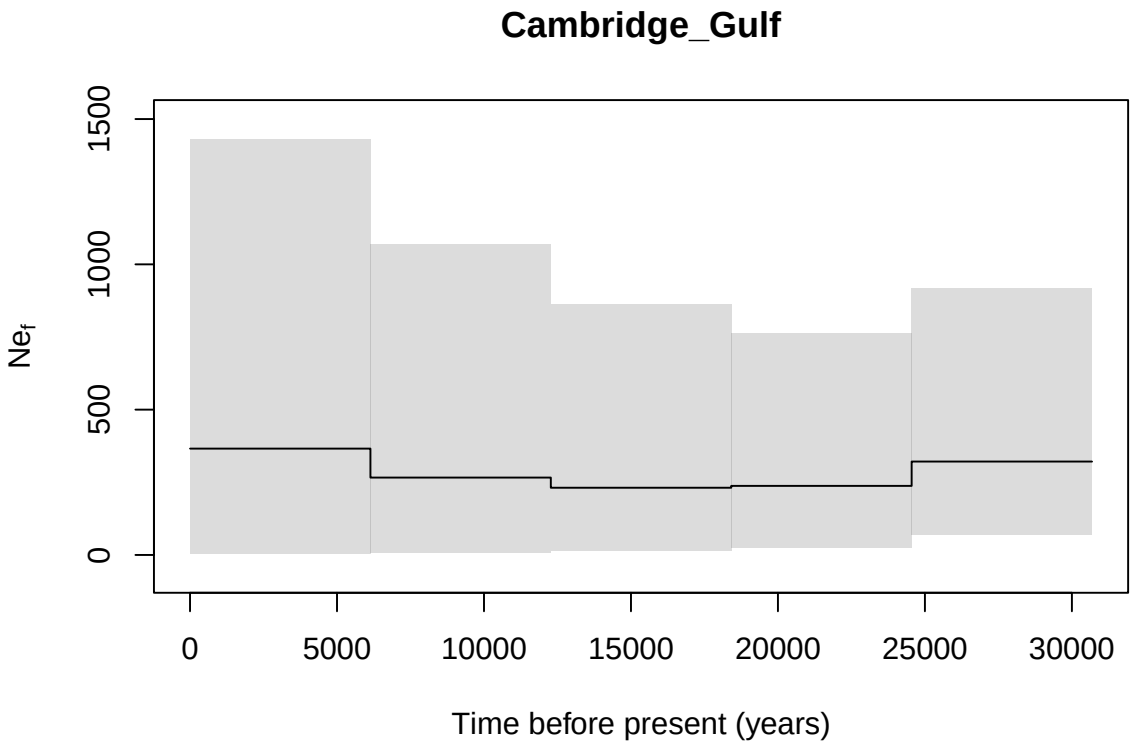


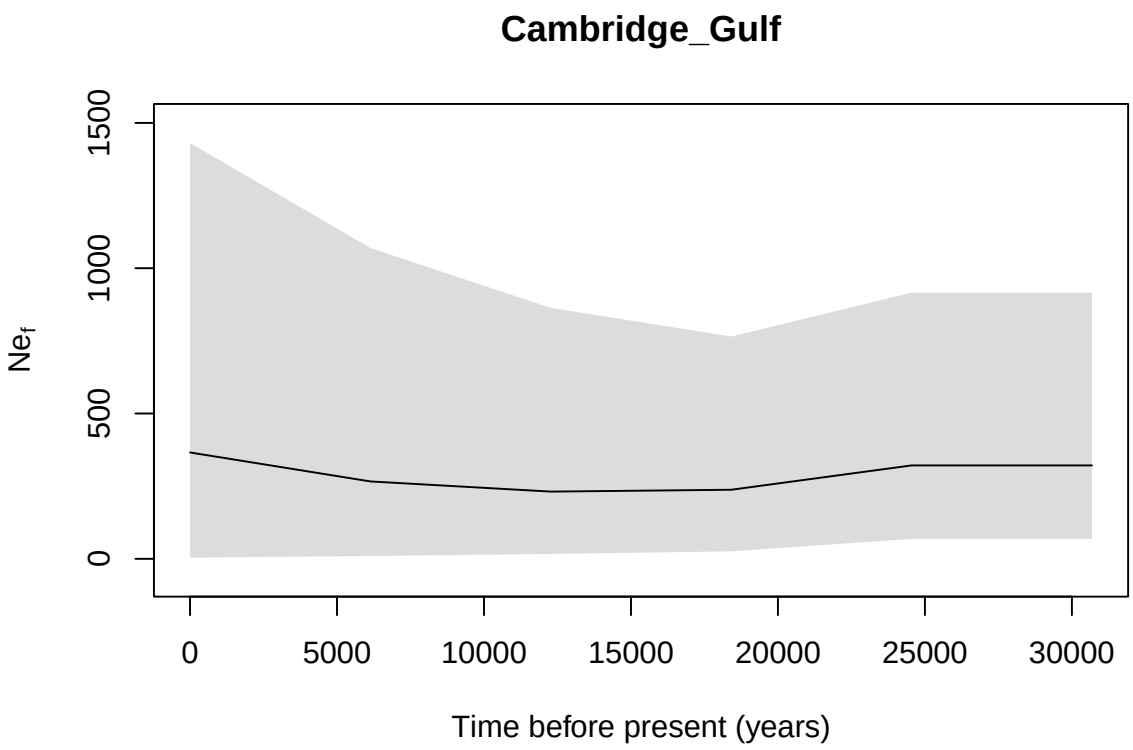




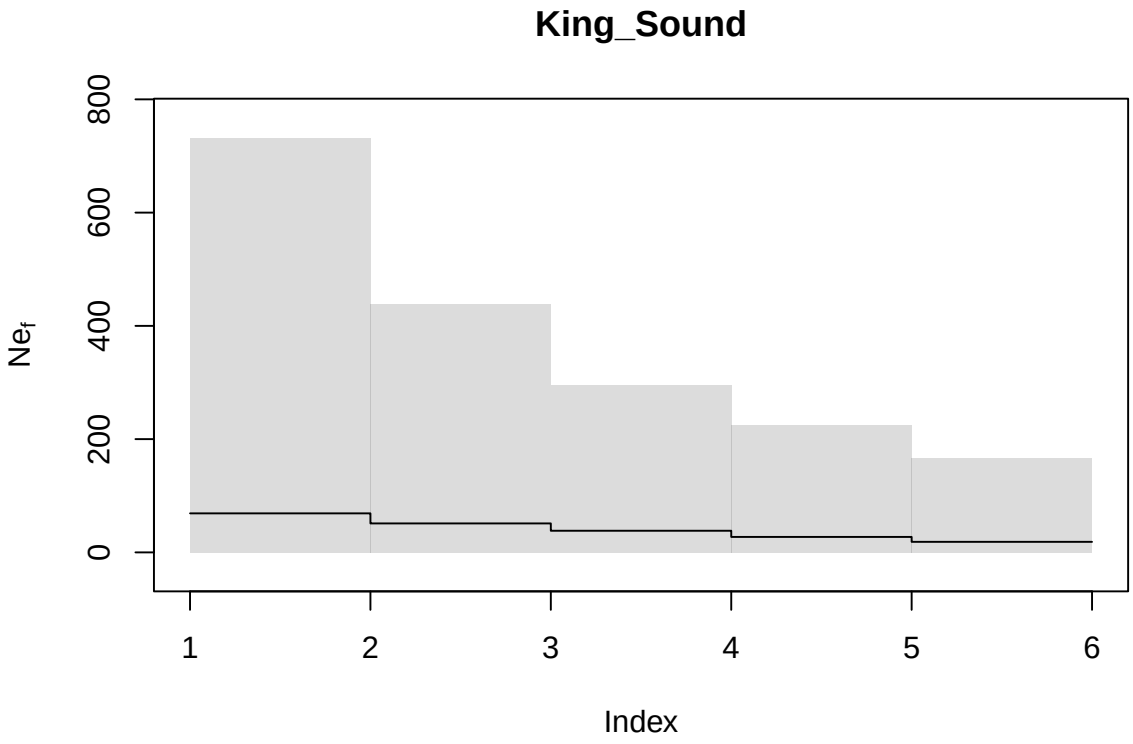

```
## [1] "Cambridge_Gulf"
## [1] "sample size = 30"
```

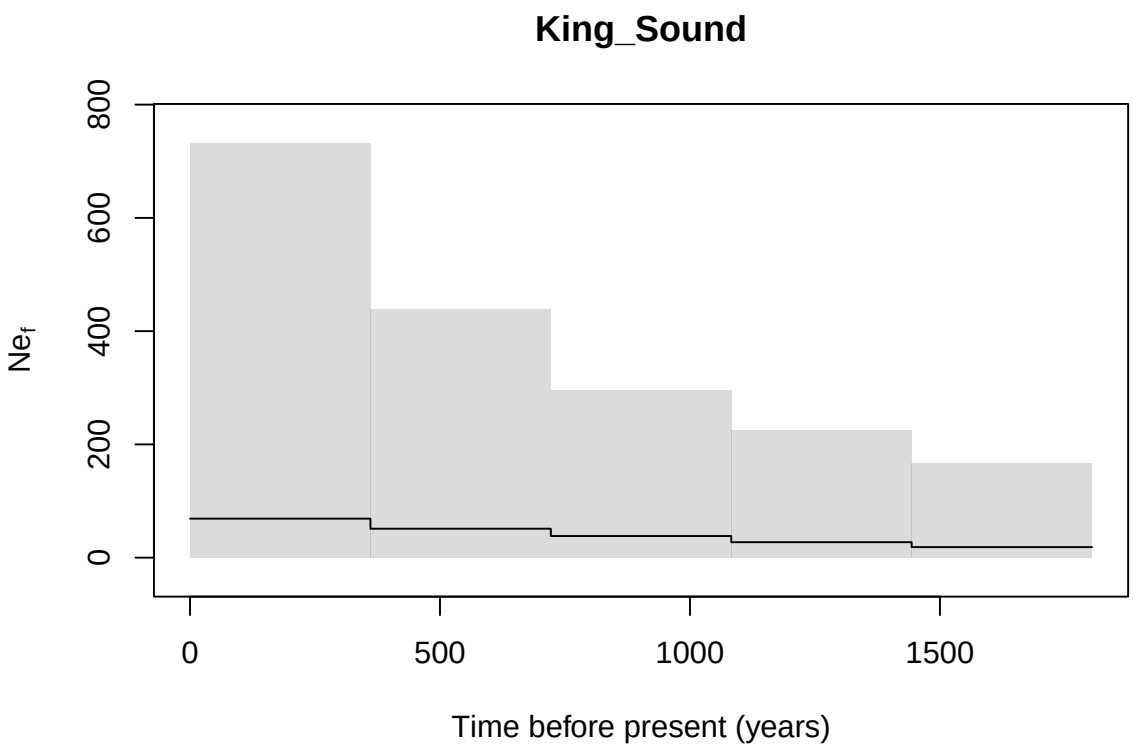

Cambridge_Gulf
Ne_f
0
500
1000
1500
1
2
3
4
5
6
Index

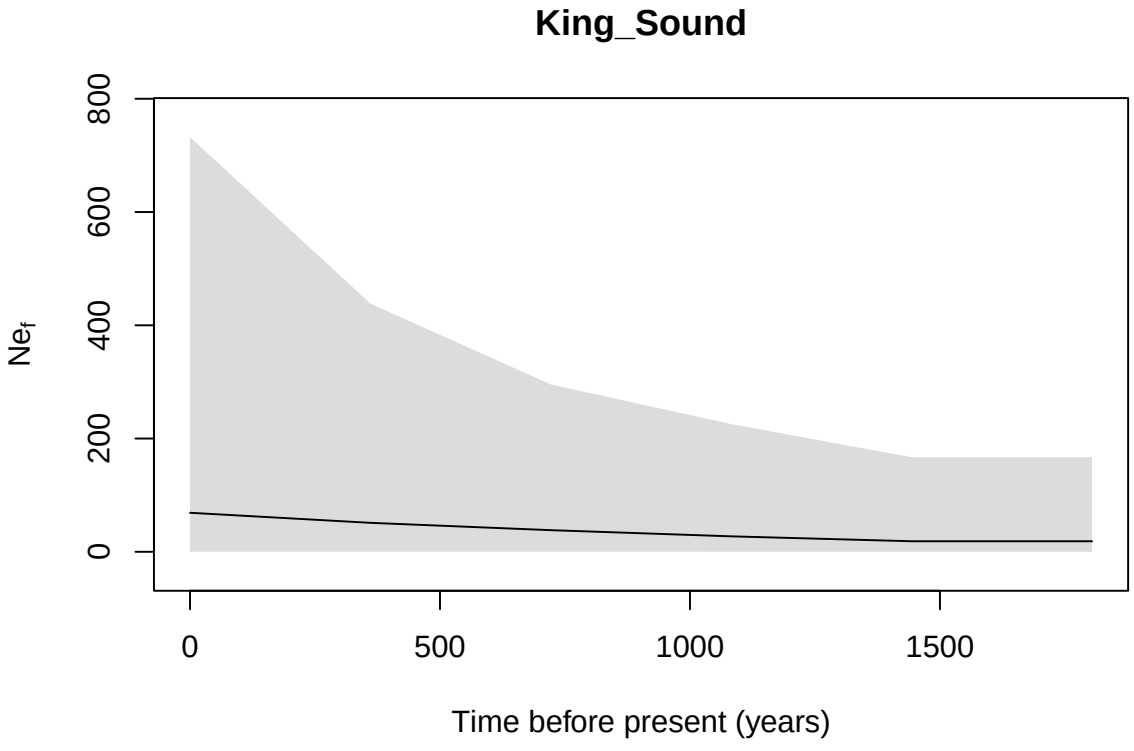

Cambridge_Gulf
Ne_f
0
500
1000
1500
0
5000
10000
15000
20000
25000
30000
Time before present (years)




```
## [1] "King_Sound"
## [1] "sample size = 14"
```








```
bsp.df$pop <- factor(bsp.df$pop, levels = c("bsp_ALL_unique","King_Sound",
                                            "Cambridge_Gulf","Daly_River",
                                            "Van_Diemen_Gulf",
                                            "Papua_New_Guinea"))
# save(bsp.df, file = "Baysian_skyline_plot_combined_pop.Rdata")
```

```
print(load("Baysian_skyline_plot_combined_pop.Rdata"))
```

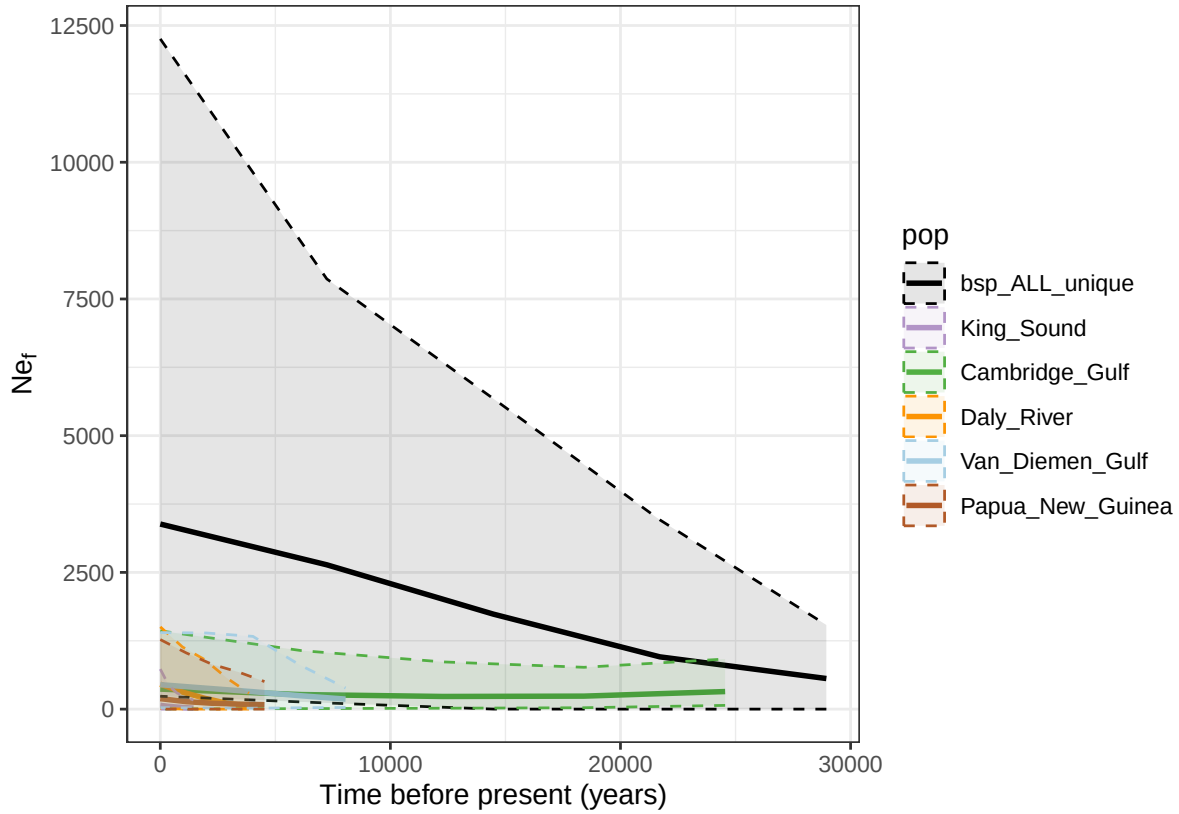
8.4.9.7 Combined BSP

```
## [1] "bsp.df"
```

```
colours.6.bsp <- c("black",colours.5)
names(colours.6.bsp) <- c("bsp_ALL_unique",gsub(pattern = " ", "_",
                                                x = names(colours.5)))

bsp <- ggplot2::ggplot(data = bsp.df, ggplot2::aes(x = time, y = med,
                                                   colour = pop)) +
  ggplot2::geom_line(size = 1) +
  ggplot2::geom_ribbon(aes(ymin = lower, ymax = upper, fill = pop),
                       linetype = 2, alpha = 0.1) +
  ggplot2::xlab("Time before present (years)") +
  ggplot2::ylab(expression("Ne"[f])) +
  ggplot2::scale_colour_manual(values = colours.6.bsp) +
  ggplot2::scale_fill_manual(values = colours.6.bsp) +
  ggplot2::theme_bw()
```

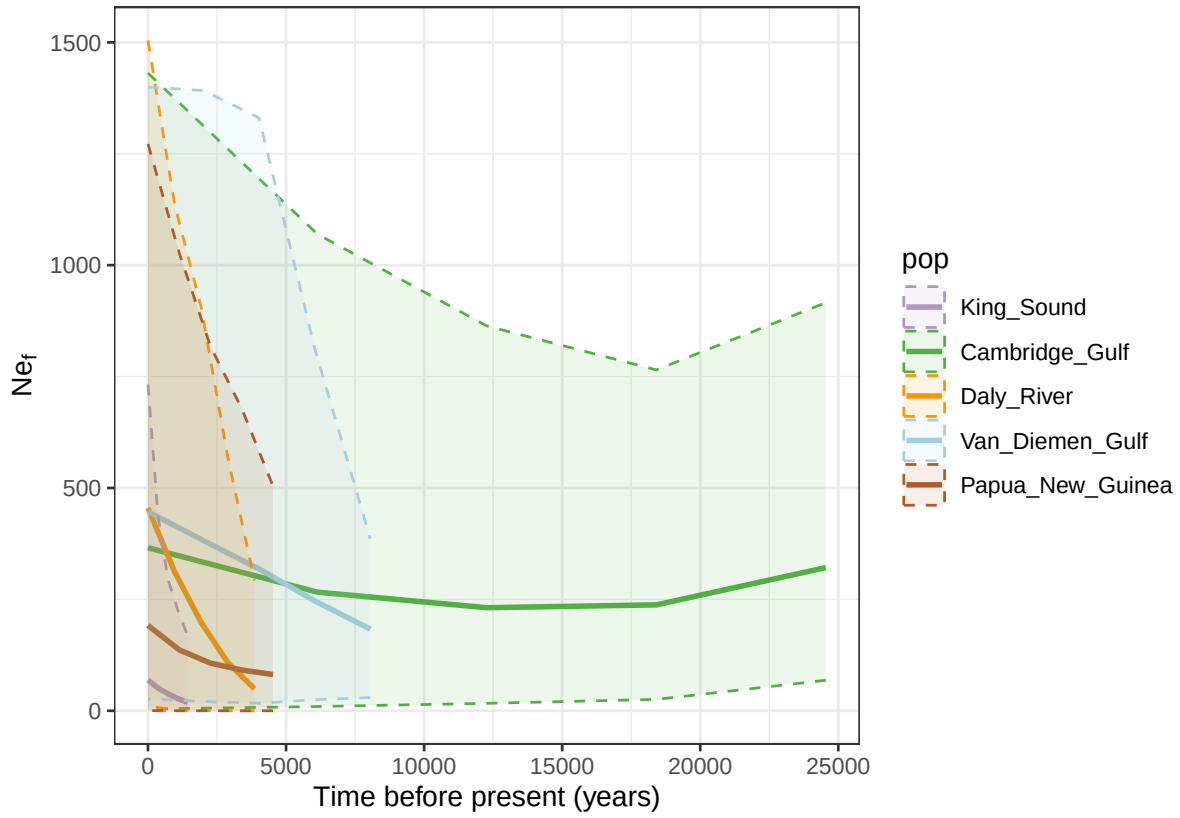
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
print(bsp)
```




```
ggplot2::ggsave(plot = bsp, filename = "Baysian_skyline_plot_combined_pop.png",
                device = "png", width = 30, height = 15, units = "cm")
```

```
bsp2 <- ggplot2::ggplot(data = bsp.df[bsp.df$pop != "bsp_ALL_unique",],
                        ggplot2::aes(x = time, y = med, colour = pop)) +
  ggplot2::geom_line(size = 1) +
  ggplot2::geom_ribbon(aes(ymin = lower, ymax = upper, fill = pop),
                       linetype = 2, alpha = 0.1) +
  ggplot2::xlab("Time before present (years)") +
  ggplot2::ylab(expression("Ne"[f])) +
  ggplot2::scale_colour_manual(values = colours.6.bsp) +
  ggplot2::scale_fill_manual(values = colours.6.bsp) +
  ggplot2::theme_bw()

print(bsp2)
```




```
ggplot2::ggsave(plot = bsp2,
                filename = "Baysian_skyline_plot_combined_pop_subset.png",
                device = "png", width = 30, height = 15, units = "cm")
```

8.4.10 BEAST - Bayesian skyline per River

```
for (river in unique(meta$River)) {
  print(river)
  id <- meta$id[meta$River == river ]
  ggar.dna.new <- ggar.ml[attributes(ggar.ml)$labels %in% id, ,drop = TRUE]
  pop_1 <- gsub(pattern = " ", "_", x = river)
  ape::write.FASTA(ggar.dna.new@dna$`6.ggar_Nucleotide_alignment_edited_HTPsites`,
                   file = paste0("Ggar_alignment_",pop_1, ".fasta"),
                   header = NULL, append = FALSE)
  fsta <- phylotools::read.fasta(file =  paste0("Ggar_alignment_",pop_1,
                                                ".fasta"), clean_name = FALSE)
  phylotools::dat2phylip(fsta, outfile = paste0("Ggar_alignment_",pop_1,
                                                ".phy"))
}
```

8.4.10.1 Subset by fasta files

```
## [1] "Papua New Guinea"
## [1] "South Alligator River"
## [1] "Wildman River"
## [1] "East Alligator River"
## [1] "West Alligator River"
## [1] "Daly River"
## [1] "West Cambridge Gulf"
## [1] "King Sound"
## [1] "Adelaide River"
## [1] "Ord River"
## [1] "Sampan Creek"
```

```
for (river in unique(meta$River)) {
    pop_1 <- gsub(pattern = " ", "_", x = river)
    print(pop_1)
    fasta_filename <- paste0("Ggar_alignment_",pop_1, ".fasta")

      inference_model <-
        beautier::create_inference_model(
          site_model = beautier::create_site_model_hky( # based on IQtree
            gamma_site_model = beautier::create_gamma_site_model(
              gamma_cat_count = "4",
              gamma_shape = "1.0",
              prop_invariant = "0.0",
              gamma_shape_prior_distr = NA,
              freq_equilibrium = "estimated",
              freq_prior_uniform_distr_id = 1000
            ), freq_equilibrium = "estimated"
          ),
          tree_prior = beautier::create_tree_prior_cbs(
            group_sizes_dimension = 5,
            b_pop_sizes_param = beautier::create_b_pop_sizes_param(
              id = NA,
              upper = "50000.0",
              value = beautier::create_distr_log_normal(m = 6.91, s = 2.35,
                                                        value = 6.91, lower = 2.30,
                                                        upper = 10.81
                                                        )
              #curve(dlnorm(x, meanlog=10, sdlog=1), from=0, to=50000)
```

```
          ),
            pop_sizes_scaler_scale_factor = 0.5 #for mtDNA
          ),
          clock_model = beautier::create_strict_clock_model(
            clock_rate_param = beautier::create_clock_rate_param(
              value = 6.2e-9, #0.5-1% mutations per site per Ma (martin et al. 1992)
              #6.3e-3 mutations per site per Ma (2.0 × 10-3 Keeney and Heist 2006; 6.05 × 10-3 Nance
              #0.62% per site per million years (Galván-Tirado et al 2013)
              estimate = FALSE,
              id = NA
            ),
            clock_rate_distr = beautier::create_distr_log_normal(m = -18.88,
                                                                 s = 1.175,
                                                                 value = -18.88,
                                                                 lower = 1e-20,
                                                                 upper = 1),
            rate_scaler_factor = 1
          ),,
            mcmc = beautier::create_mcmc(
              chain_length = 1e+07,
              store_every = 5e+03,
              pre_burnin = 1e4,
              n_init_attempts = 3,
              tracelog = beautier::create_tracelog(filename =
                                                     paste0("tracelog_", pop_1, ".log")),
              screenlog = beautier::create_screenlog(filename =
                                                       paste0("screenlog_", pop_1, ".log")),
              treelog = beautier::create_treelog(filename =
                                                   paste0("treelog_", pop_1, ".log"))
            ),
            beauti_options = beautier::create_beauti_options()
          )


      beast2_input_file <- paste0("beast2_",pop_1,".xml")
      beautier::create_beast2_input_file_from_model(
        input_filename = fasta_filename,
        inference_model = inference_model,
        output_filename = beast2_input_file
      )


    if (beastier::is_beast2_installed()) {
      beastier::is_beast2_input_file(beast2_input_file,
                                     show_warnings = FALSE,
                                     verbose = FALSE)
    }
}
```

8.4.10.2 Create BEAUTI file

```
## [1] "Papua_New_Guinea"
## [1] "South_Alligator_River"
## [1] "Wildman_River"
## [1] "East_Alligator_River"
## [1] "West_Alligator_River"
## [1] "Daly_River"
## [1] "West_Cambridge_Gulf"
```

```
## [1] "King_Sound"
## [1] "Adelaide_River"
## [1] "Ord_River"
## [1] "Sampan_Creek"
```

```
for (river in unique(meta$River)) {
  pop_1 <- gsub(pattern = " ", "_", x = river)
  print(pop_1)

    if (beastier::is_beast2_installed()) {
      beast2_options <- beastier::create_beast2_options(
        input_filename = paste0("beast2_",pop_1,".xml"),
        output_state_filename = paste0("beast2_",pop_1,".xml.state"),
        n_threads = 4,
        verbose = TRUE
      )
      beastier::check_can_create_file(beast2_options$output_state_filename)
      beastier::check_can_create_treelog_file(beast2_options)
      run_beast2_from_options(
        beast2_options = beast2_options
      )
      testthat::expect_true(file.exists(beast2_options$output_state_filename))
    }

    if (beastier::is_beast2_installed()) {
      out <- babette::bbt_run_from_model(
        fasta_filename = fasta_filename,
        inference_model = inference_model,
        beast2_options = beast2_options)
    }
}
```

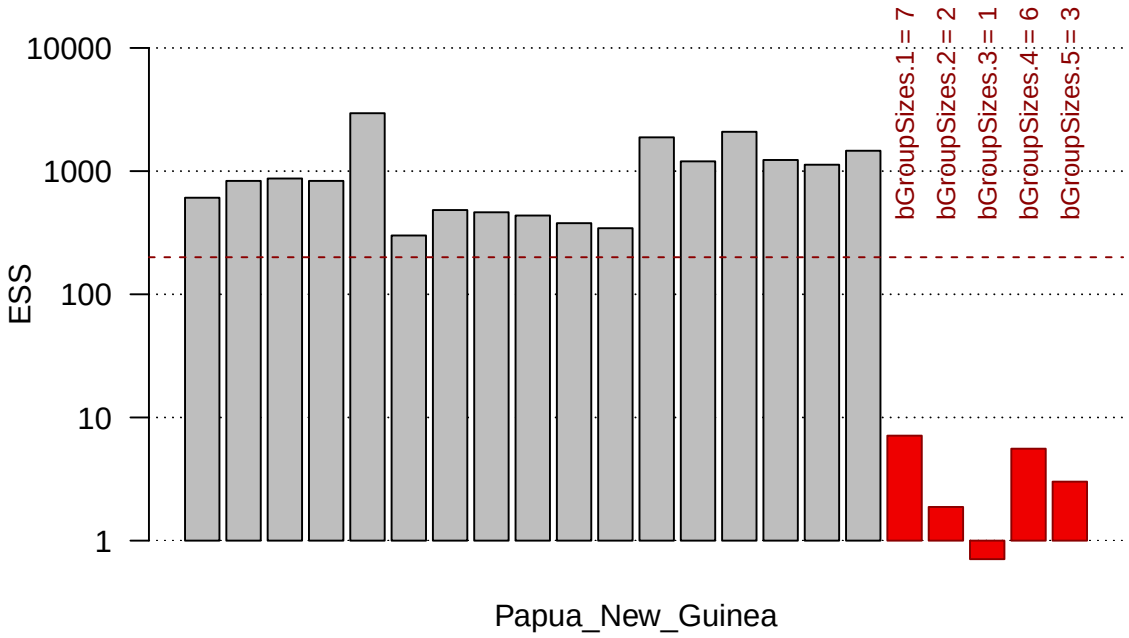
8.4.10.3 Run BEAST

```
river <- unique(meta$River)[2]
for (river in unique(meta$River)) {
  pop_1 <- gsub(pattern = " ", "_", x = river)
  print(pop_1)
  print(paste0("sample size = ", length(meta$River[meta$River == river])))

  folder_1 <- paste0("beast_bspe_", pop_1, "/")
  tracelog <- paste0(folder_1, "tracelog_", pop_1, ".log")
  estimates <- tracerer::parse_beast_tracelog_file(tracelog_filename = tracelog)
  estimates_burn <- tracerer::remove_burn_ins(estimates, burn_in_fraction = 0.1)
  esses <- tracerer::calc_esses(estimates_burn, sample_interval = 1000)
  esses <- t(esses)
  colnames(esses) <- c("ESS")
  knitr::kable(esses)

  bdsky_trace   <- beastio::readLog(tracelog, burnin = 0.1)
  beastio::checkESS(bdsky_trace)
  beastio::checkESS(
    bdsky_trace,
    cutoff = 200,
    plot = TRUE,
    log = 'y',
```

```
    ylim = c(1, 10000),
    title = pop_1,
    plot.grid = TRUE
  )

  sum_stats <- tracerer::calc_summary_stats(estimates_burn, sample_interval = 5000)
  knitr::kable(sum_stats) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")


  ggplot2::ggplot(data = estimates_burn, ggplot2::aes(x = Sample)) +
    ggplot2::geom_line(ggplot2::aes(y = posterior)) +
    ggplot2::ggtitle(paste0(pop_1, " :Posterior"))


  ggplot2::ggplot(data = estimates_burn, ggplot2::aes(x = posterior)) +
    ggplot2::geom_histogram(binwidth = 0.21) +
    ggplot2::ggtitle(paste0(pop_1, ": Posterior distribution")) +
    ggplot2::scale_x_continuous() +
    ggplot2::theme_bw()


  ggplot2::ggplot(data = estimates_burn, ggplot2::aes(x = Sample, y = posterior)) +
    ggplot2::geom_line() +
    ggplot2::ggtitle(paste0(pop_1, ": Trace plot")) +
    ggplot2::theme_bw()


  ggplot2::ggplot(data = estimates_burn, ggplot2::aes(x = TreeHeight)) +
    ggplot2::geom_histogram() +
    ggplot2::ggtitle(paste0(pop_1, ": TreeHeight")) +
    ggplot2::theme_bw()

}
```

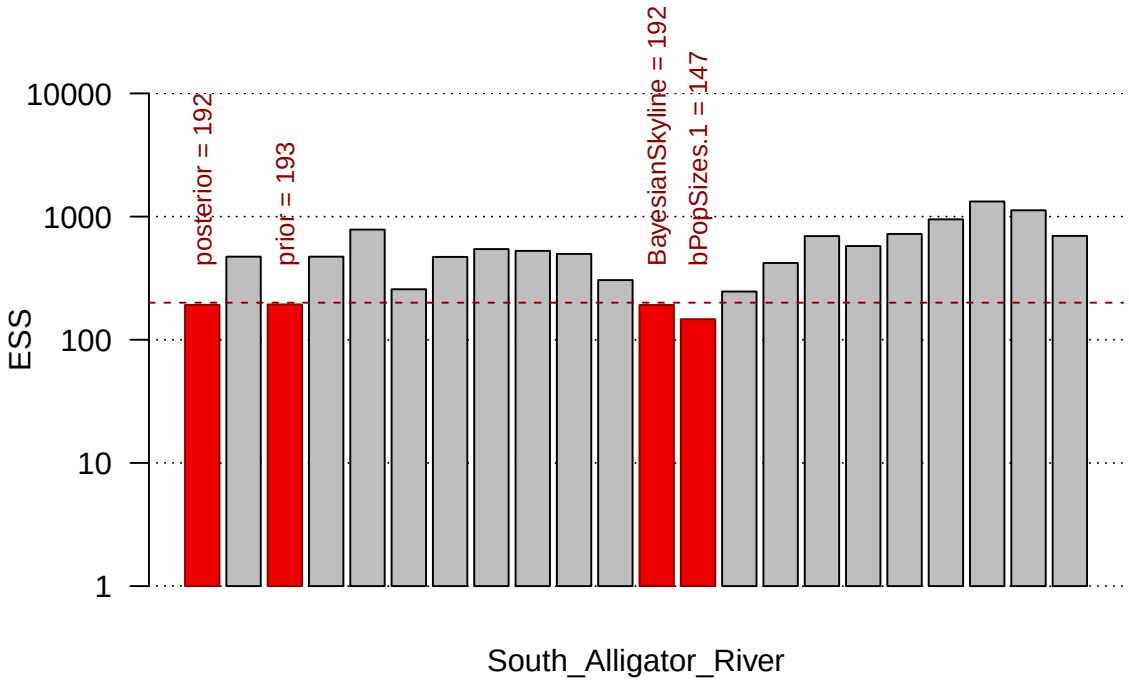
8.4.10.4 Summary stats

```
## [1] "Papua_New_Guinea"
## [1] "sample size = 6"
```



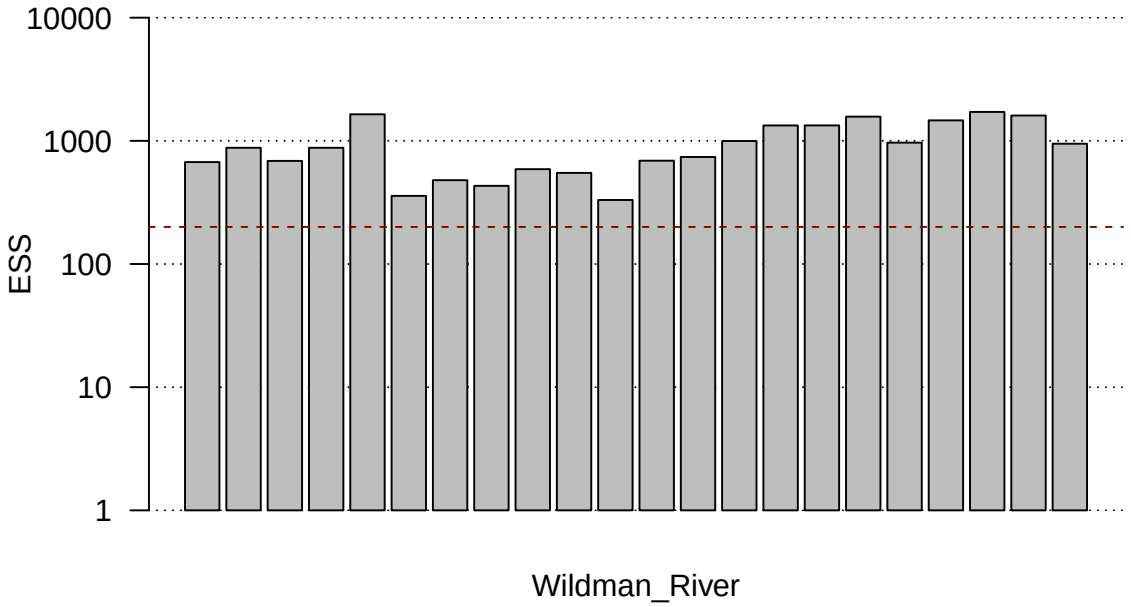

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "South_Alligator_River"
## [1] "sample size = 101"
```



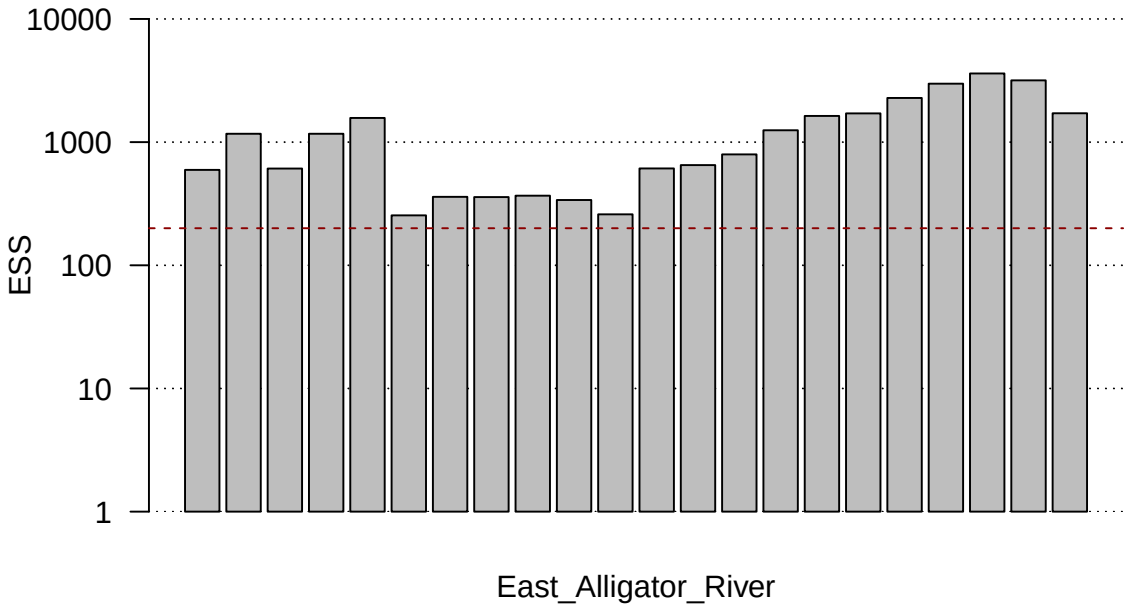
```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "Wildman_River"
## [1] "sample size = 45"
```



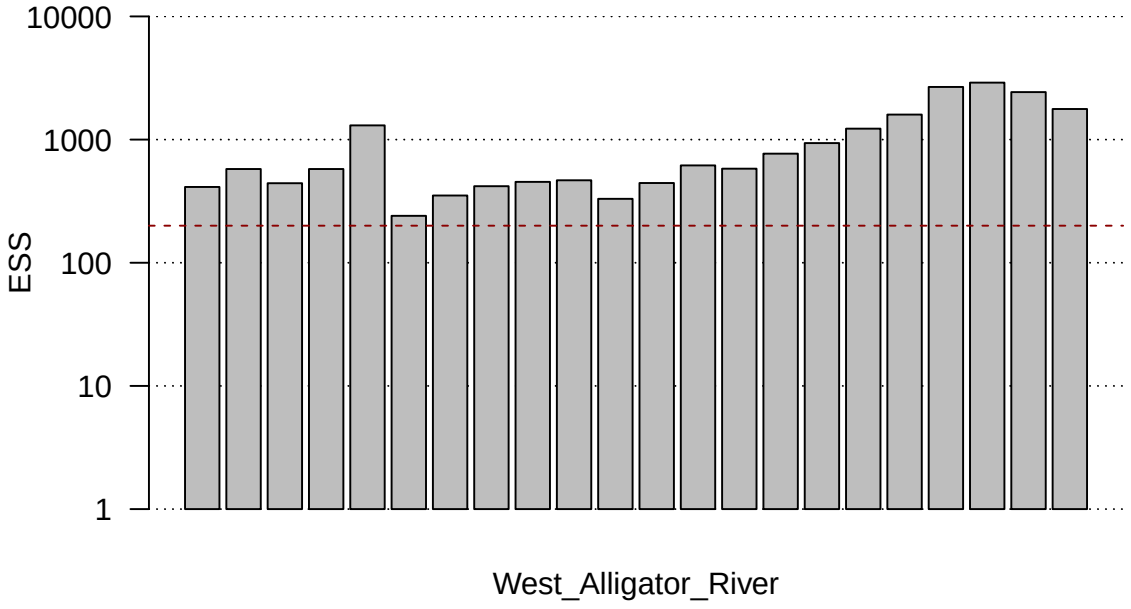

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "East_Alligator_River"
## [1] "sample size = 61"
```



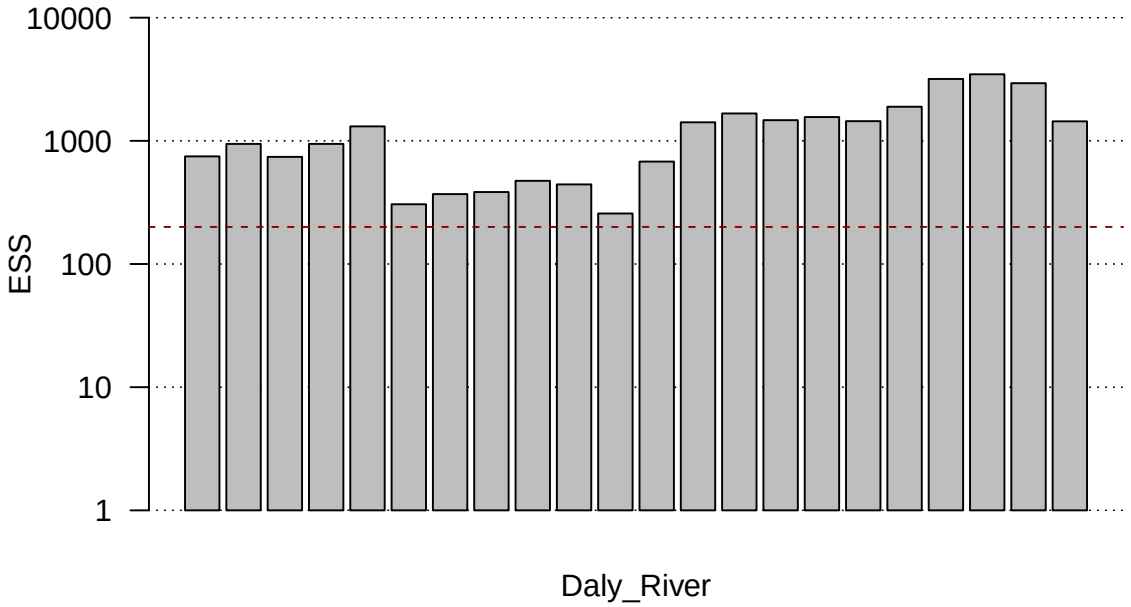

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "West_Alligator_River"
## [1] "sample size = 41"
```



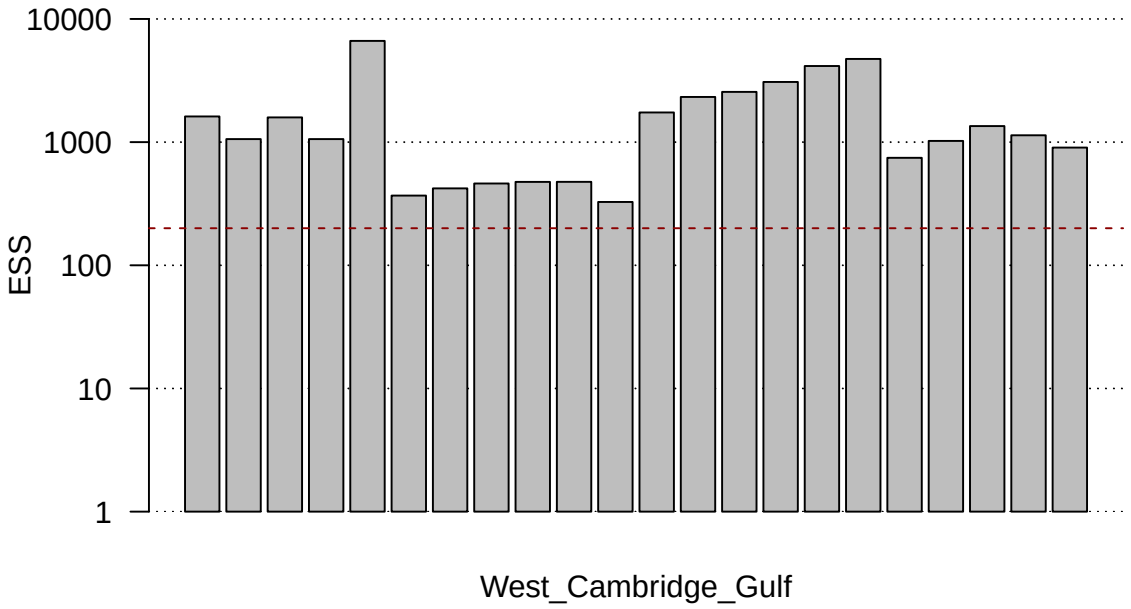
```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "Daly_River"
## [1] "sample size = 30"
```



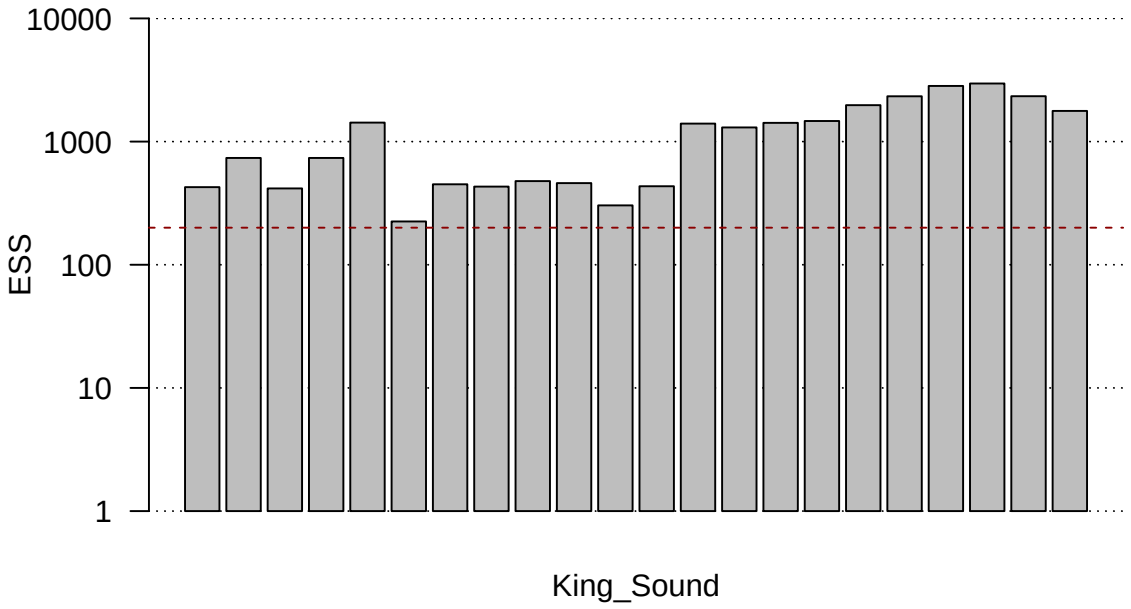

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "West_Cambridge_Gulf"
## [1] "sample size = 15"
```



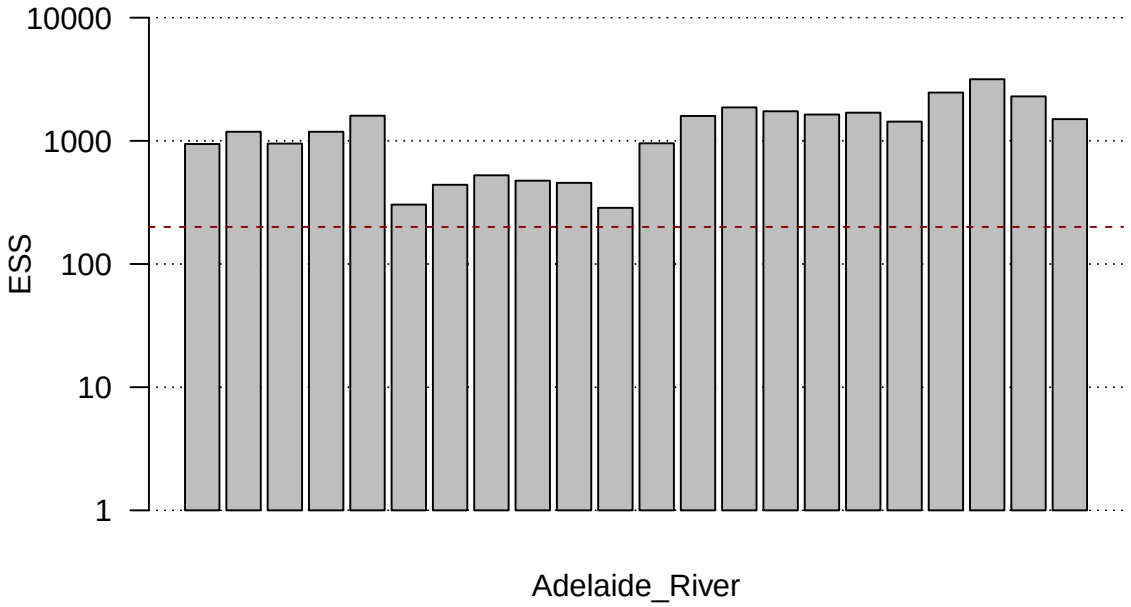

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "King_Sound"
## [1] "sample size = 14"
```



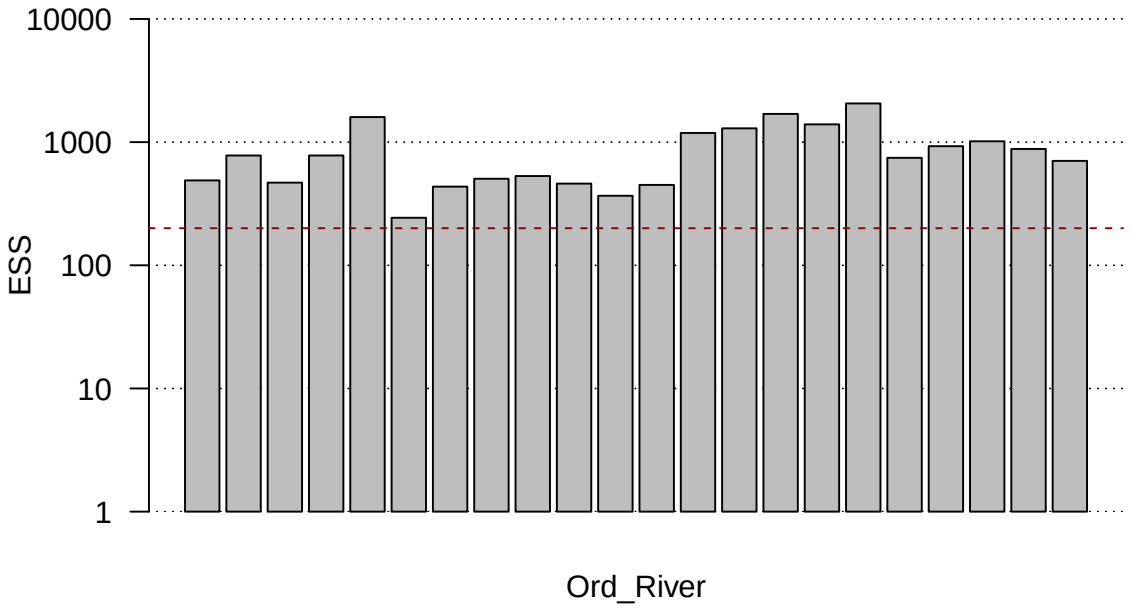
```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "Adelaide_River"
## [1] "sample size = 25"
```



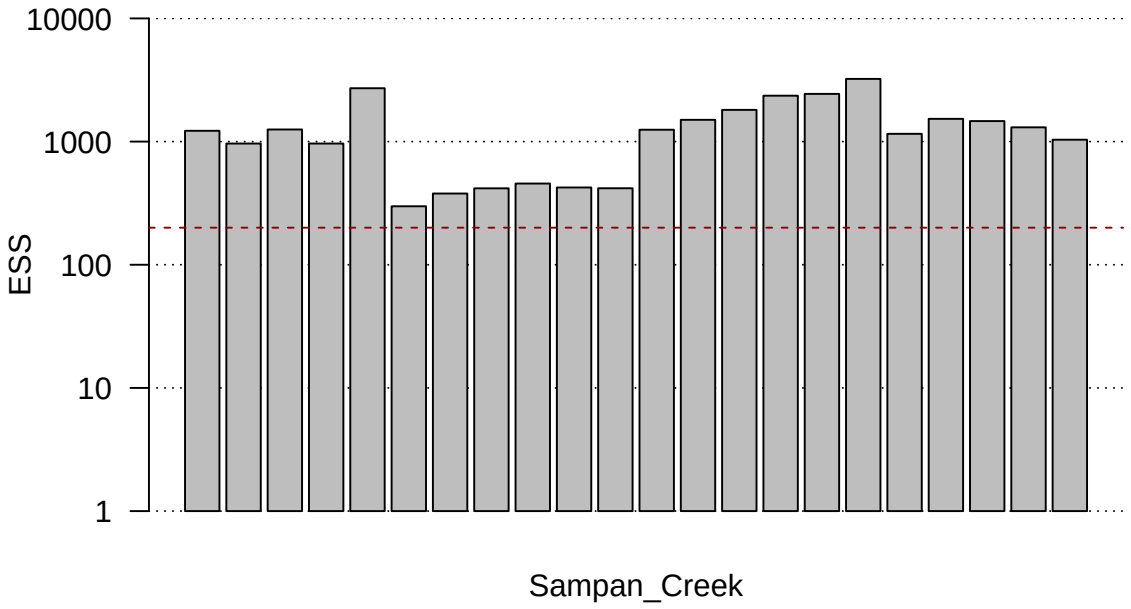

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "Ord_River"
## [1] "sample size = 15"
```




```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
## [1] "Sampan_Creek"
## [1] "sample size = 30"
```



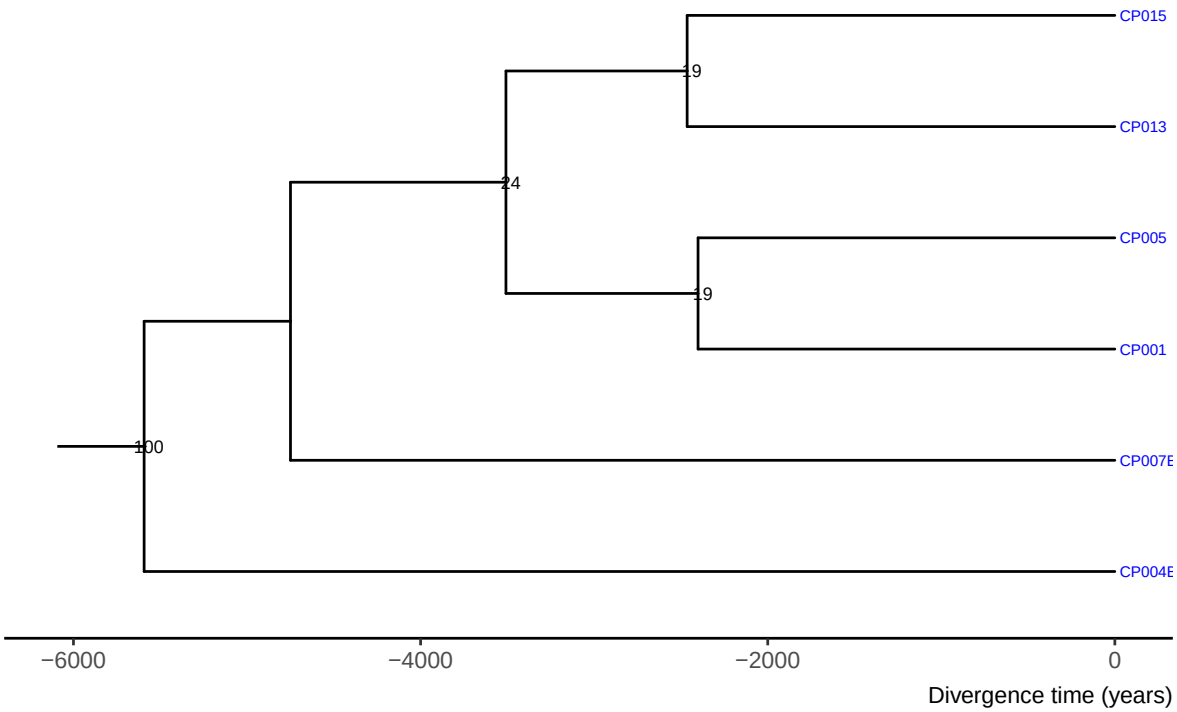
```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

```
for (river in unique(meta$River)) {
  pop_1 <- gsub(pattern = " ", "_", x = river)
  print(pop_1)

  folder_1 <- paste0("beast_bspe_", pop_1, "/")
  treelog <- paste0(folder_1, "treelog_", pop_1, ".log")

  trees <- tracerer::parse_beast_trees(filename = treelog)
  treefile <- paste0(folder_1, "Result_trees.tre") #with treeAnnotator
  # phyloch::treeannotator(input = treelog)

  ggar.tree <- ape::read.nexus(treefile)
  boot_support <- ape::prop.clades(ggar.tree, trees)
  boot_support <- boot_support / 100
  boot_support <- round(boot_support, 0)

  ggar.tree$root.edge <- 500


  p1 <- ggtree::ggtree(tr = ggar.tree, color = "black")
  num_tips <- length(p1$data$label[p1$data$isTip == TRUE])
  num_nodes <- length(p1$data$label[p1$data$isTip == FALSE])
  nodes <- paste0("t", 1:num_nodes)
  p1$data$label[p1$data$isTip == FALSE] <- nodes
  bs <- data.frame(label = p1$data$label,
                   bootstrap = c(rep(NA, num_tips), boot_support))
  p1 <- p1 %<+% bs

  p1 <- p1 +
    ggtree::geom_tiplab(size = 2, color = 'blue') +
    ggtree::geom_rootedge() +
    ggtree::geom_nodelab(aes(label = bootstrap), size = 2.5, ) +
    ggtree::theme_tree2() +
    ggplot2::ggtitle(pop_1) +
    ggplot2::labs(caption = "Divergence time (years)")
  p2 <- ggtree::revts(p1)
  print(p2)

  ggplot2::ggsave(
    plot = p2,
    filename = paste0("BEAST_bsp_", pop_1, "_ggtree.png"),
    device = "png",
    width = 30,
    height = 15,
    units = "cm"
  )
}
```

8.4.10.5 Plot Tree

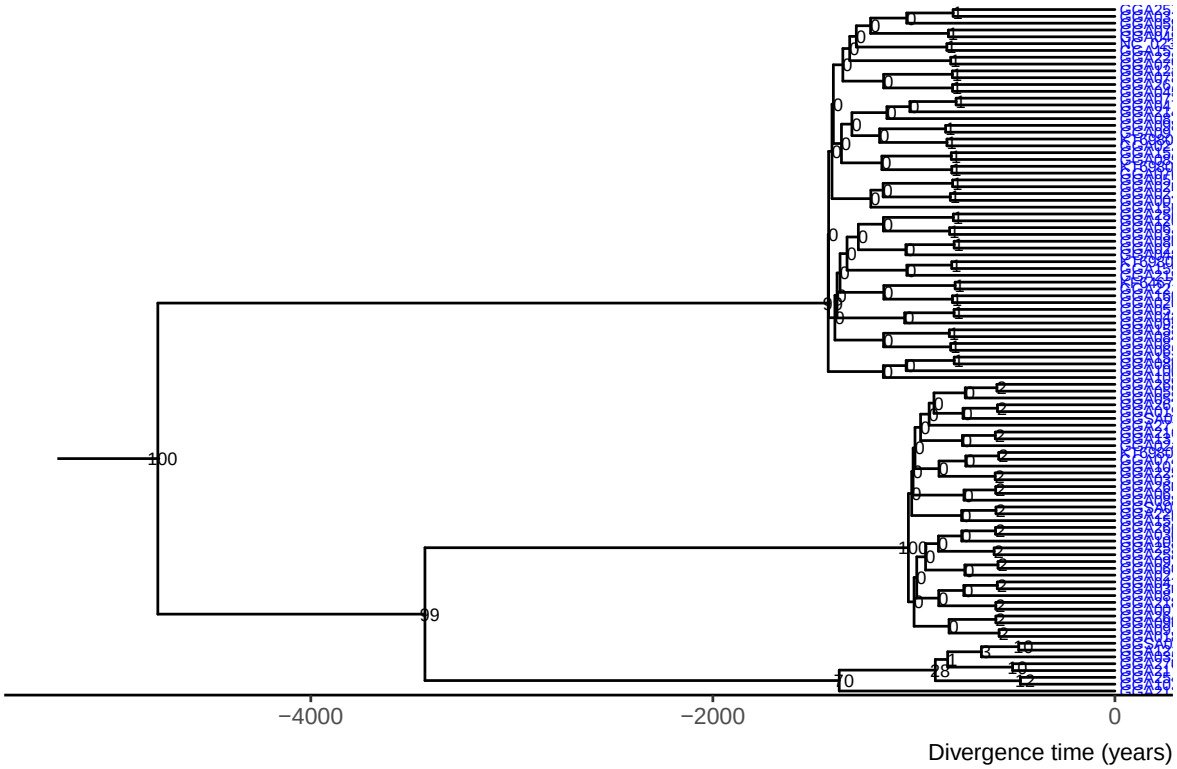
```
## [1] "Papua_New_Guinea"
```

Papua_New_Guinea



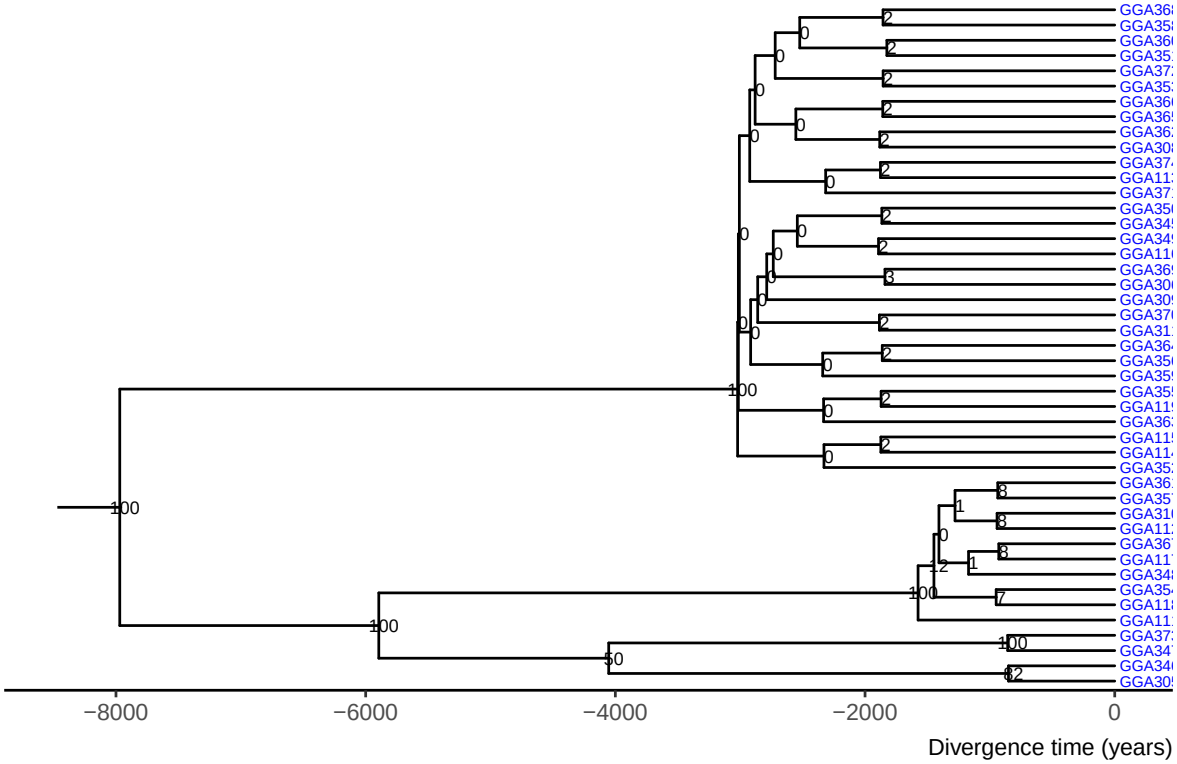

```
## [1] "South_Alligator_River"
```

South_Alligator_River



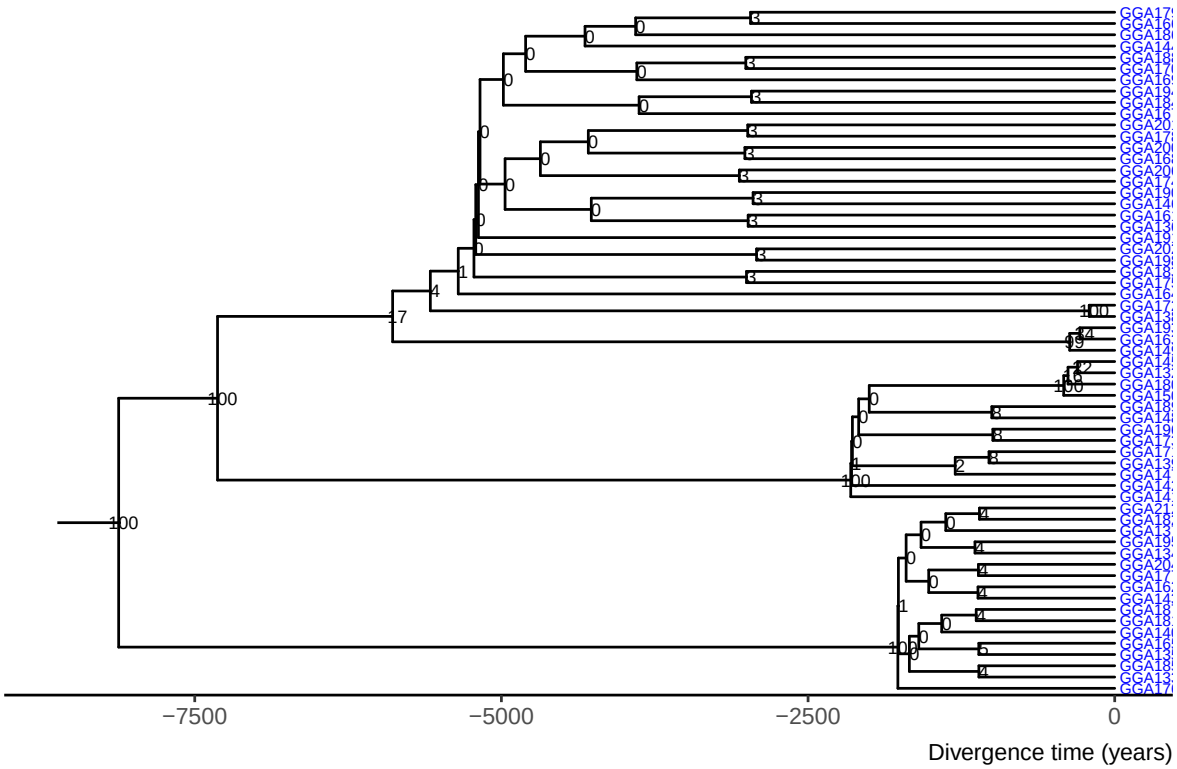

```
## [1] "Wildman_River"
```

Wildman_River



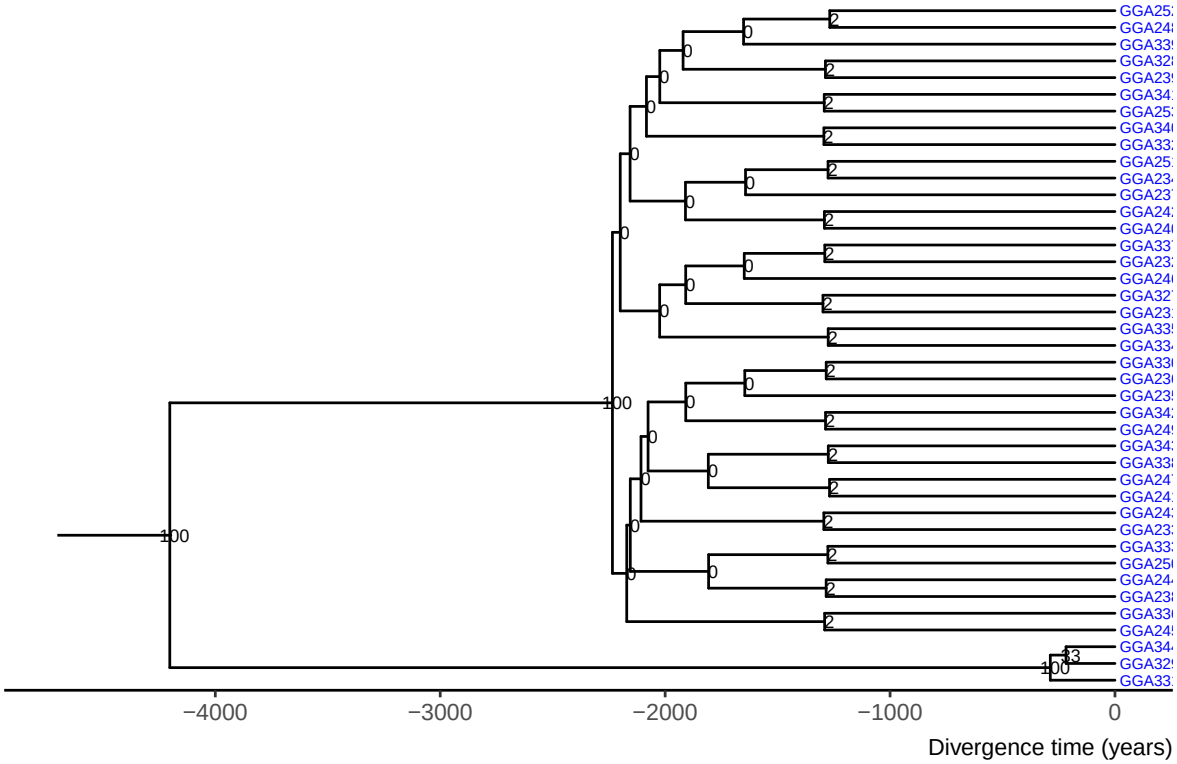

```
## [1] "East_Alligator_River"
```

East_Alligator_River



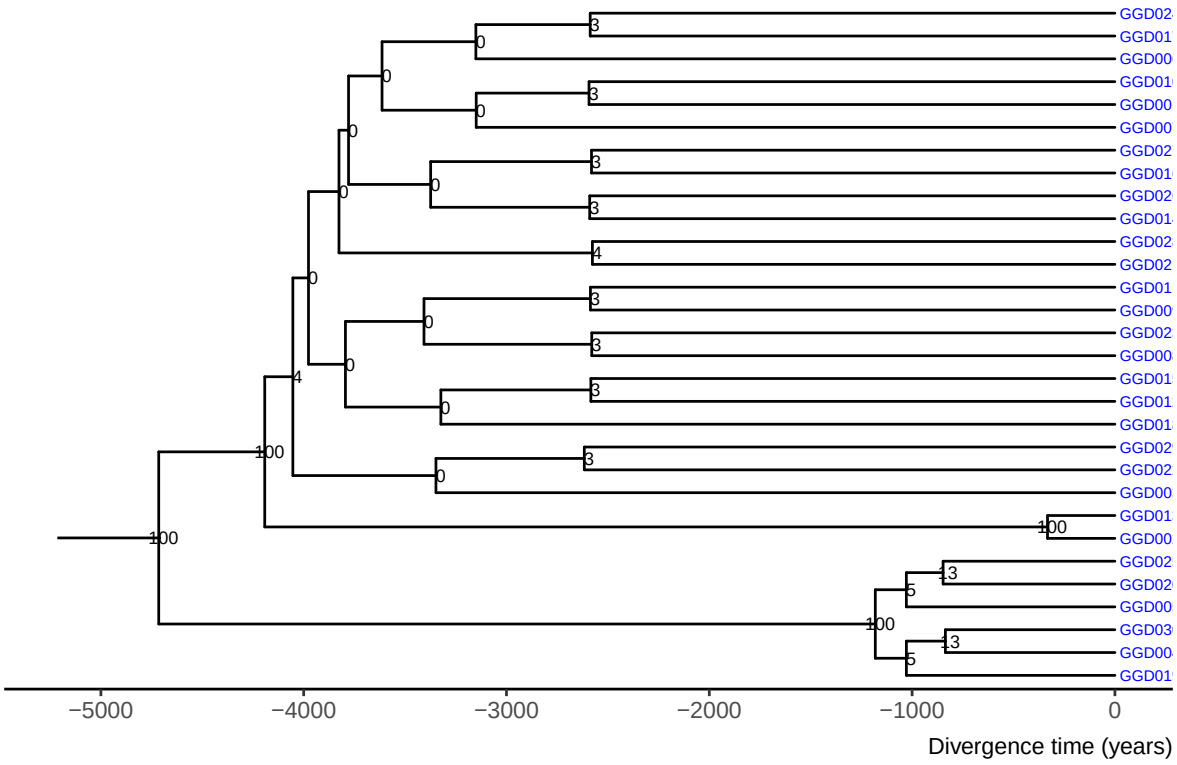

```
## [1] "West_Alligator_River"
```

West_Alligator_River



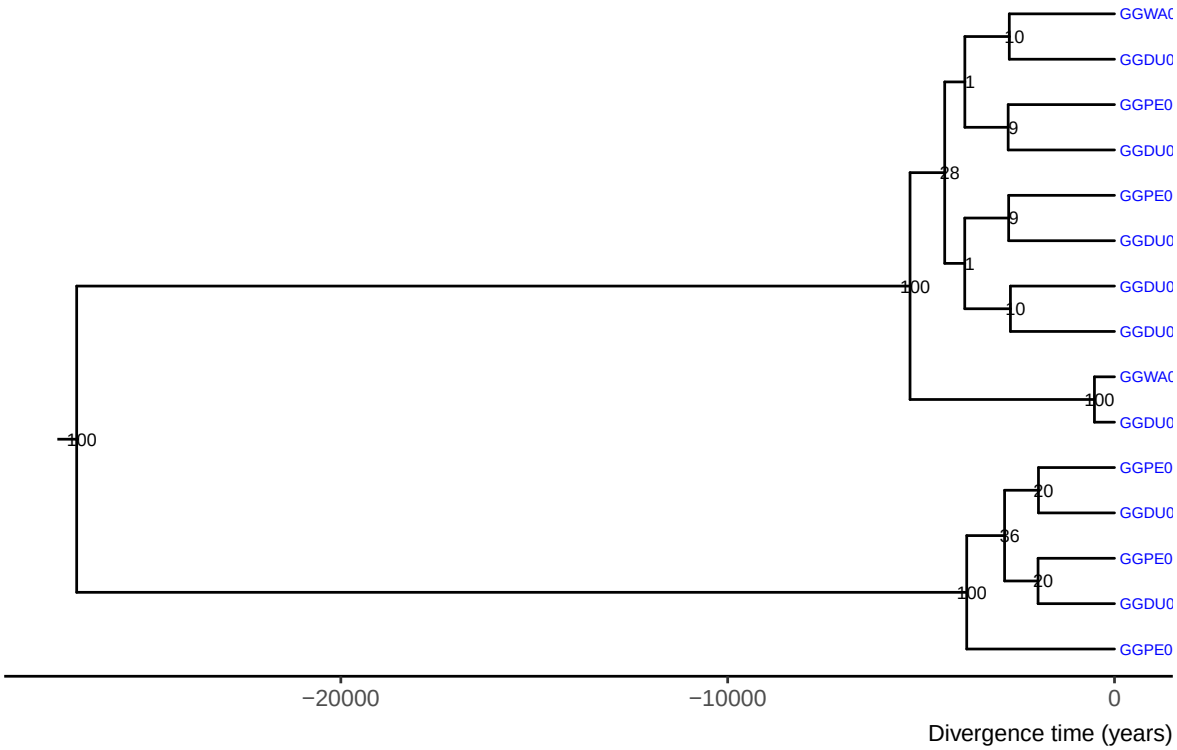

```
## [1] "Daly_River"
```

Daly_River



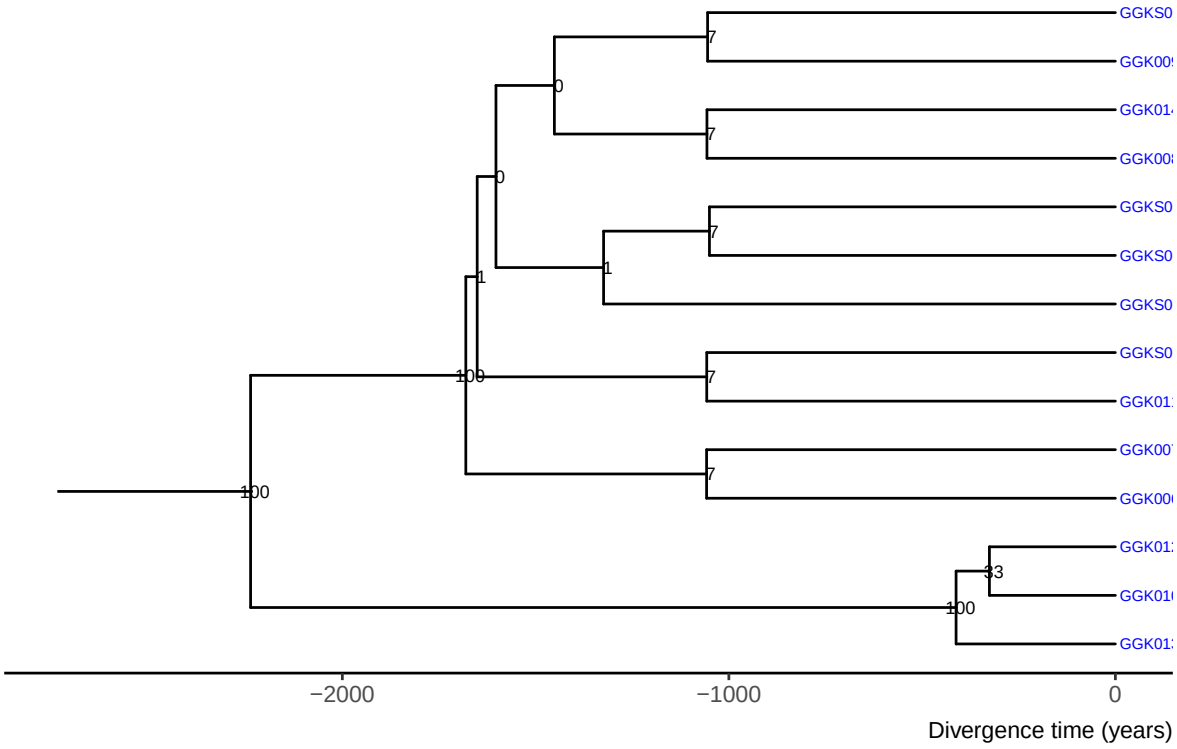

```
## [1] "West_Cambridge_Gulf"
```

West_Cambridge_Gulf



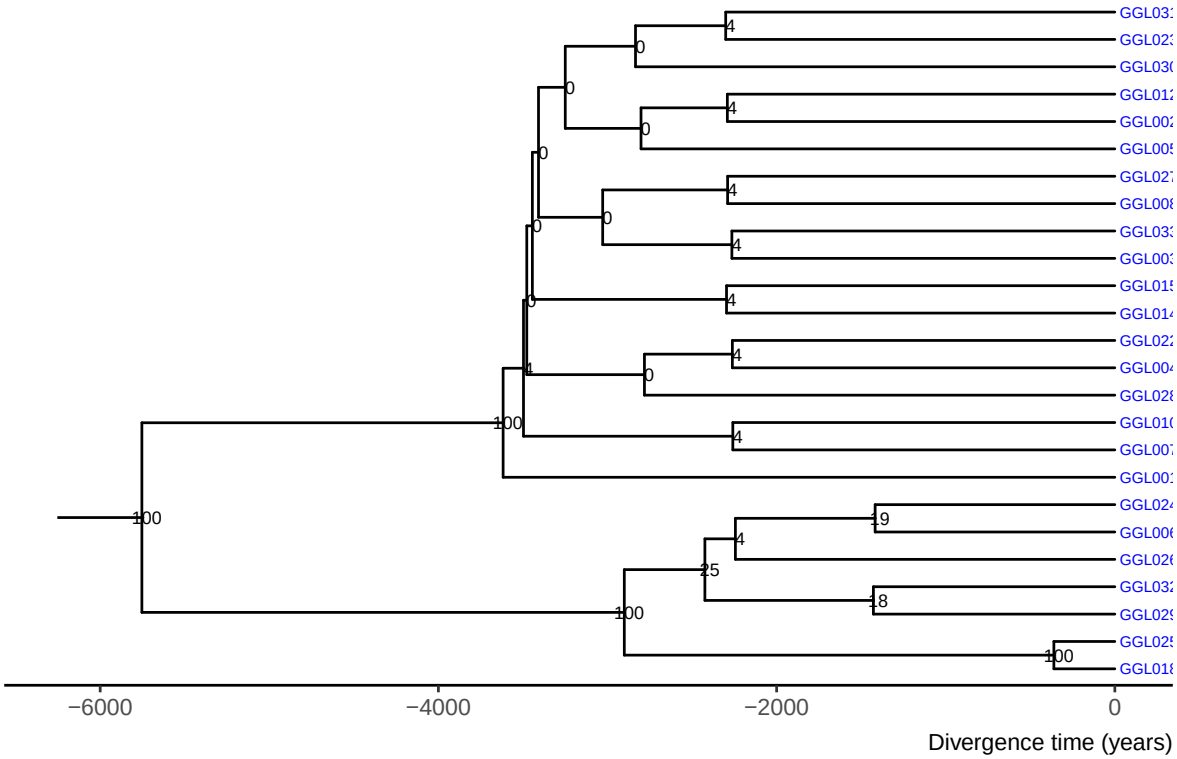

```
## [1] "King_Sound"
```

King_Sound



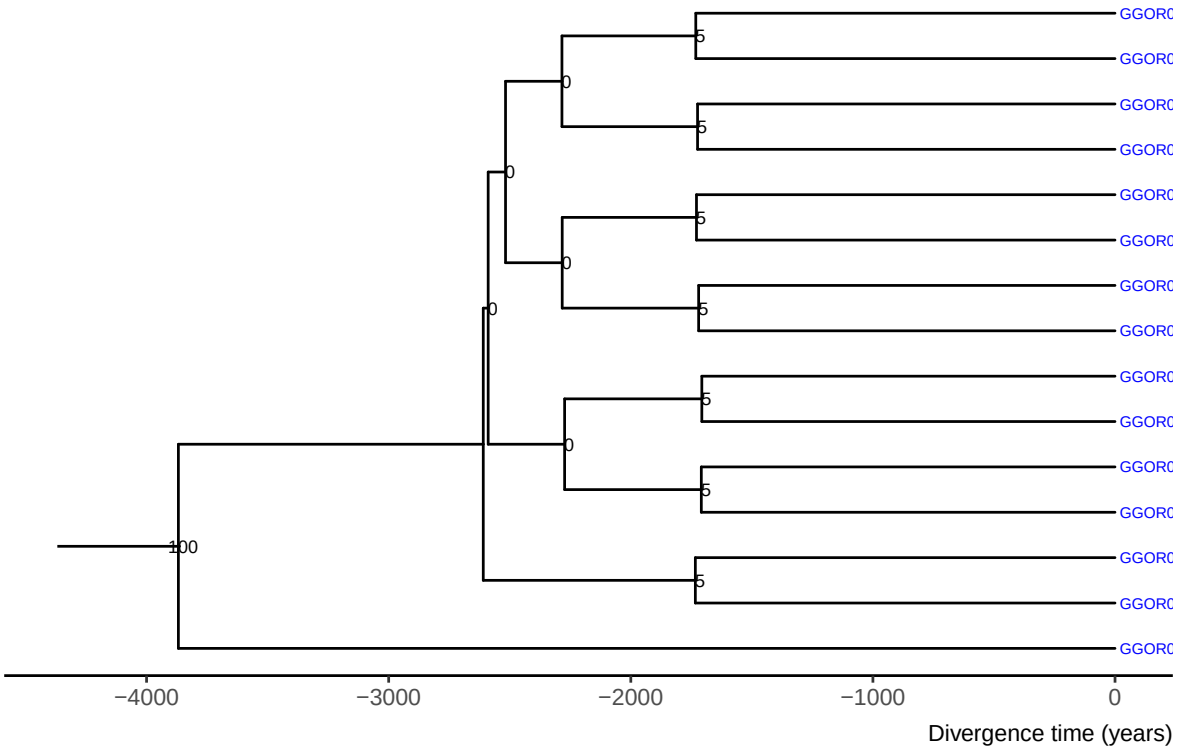

```
## [1] "Adelaide_River"
```

Adelaide_River



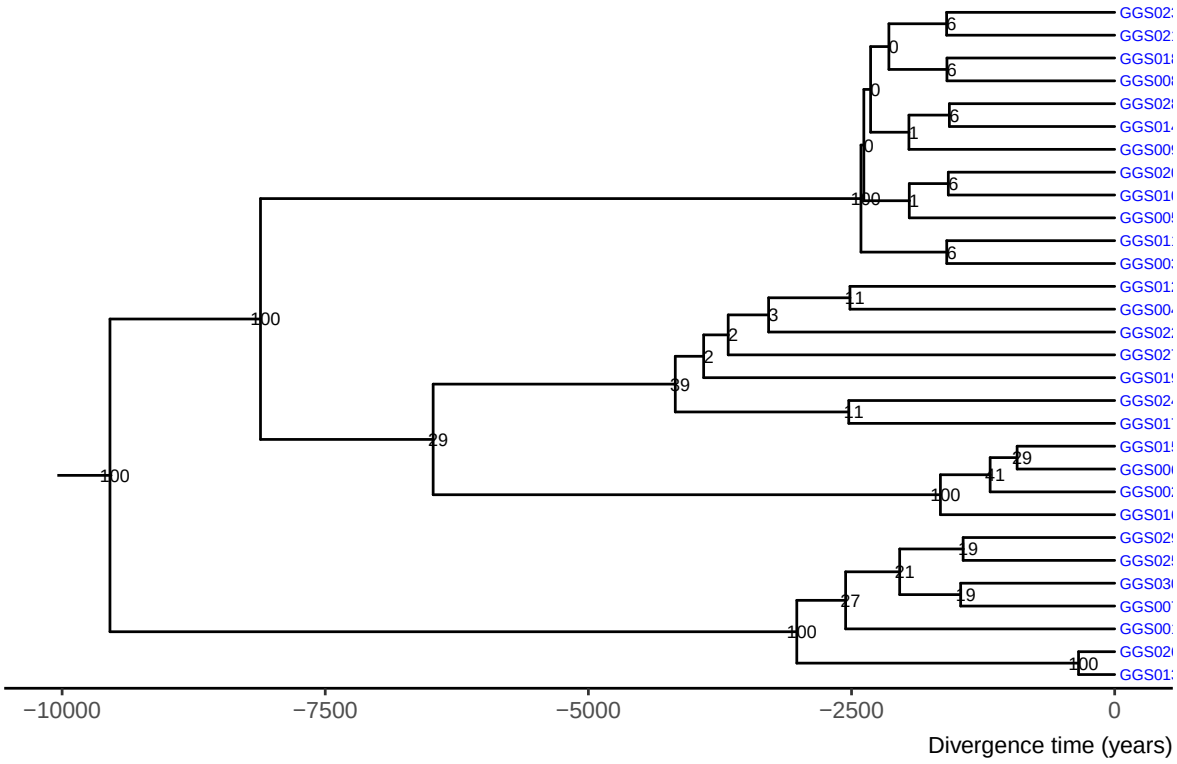

```
## [1] "Ord_River"
```

Ord_River




```
## [1] "Sampan_Creek"
```

Sampan_Creek




```
print(load("Baysian_skyline_plot_combined.Rdata"))
```

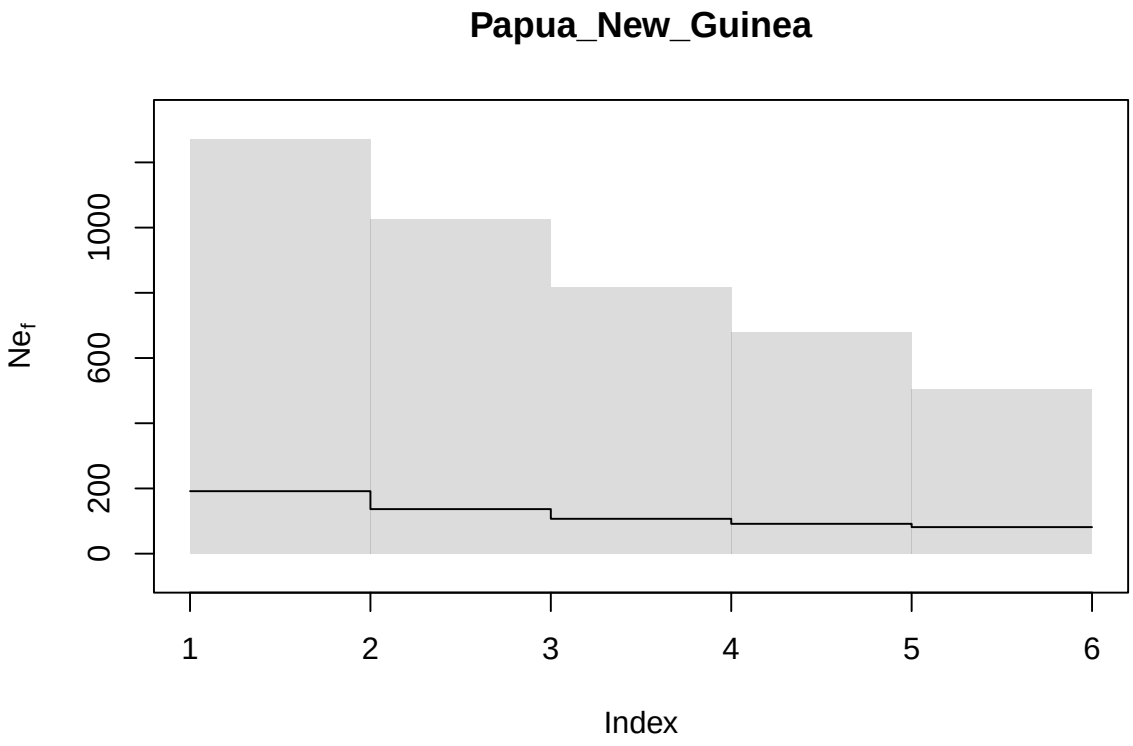
8.4.10.6 Plot BSP

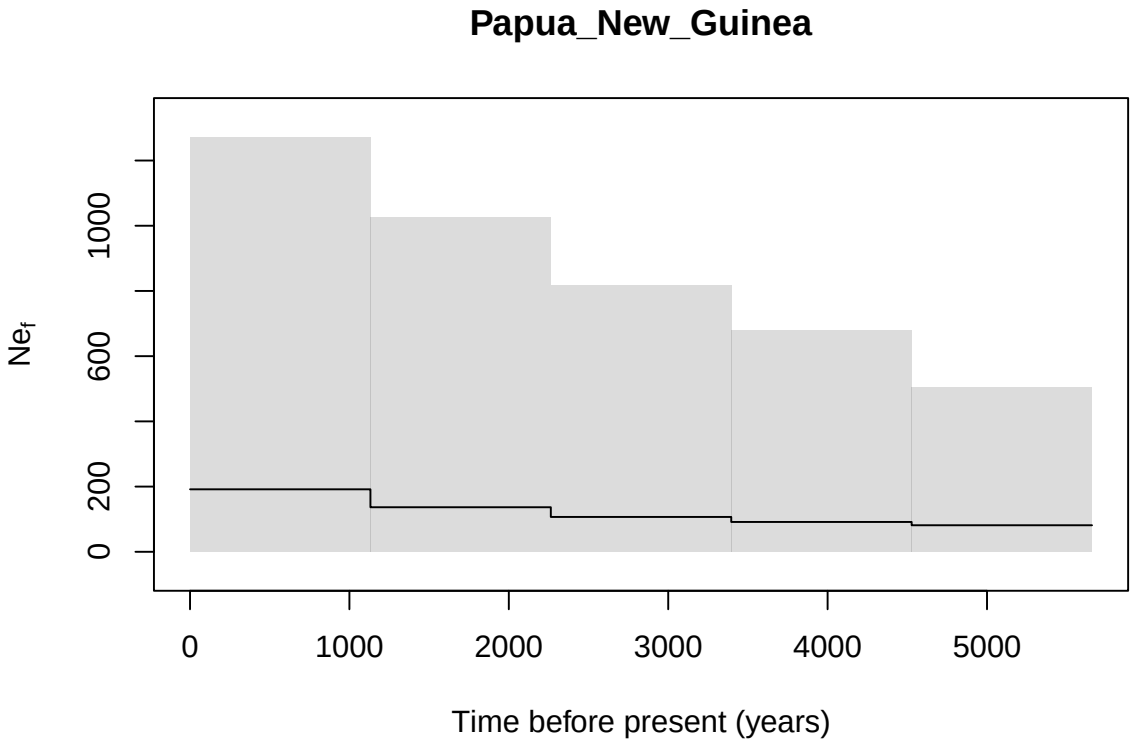
```
## [1] "bsp.df"
```

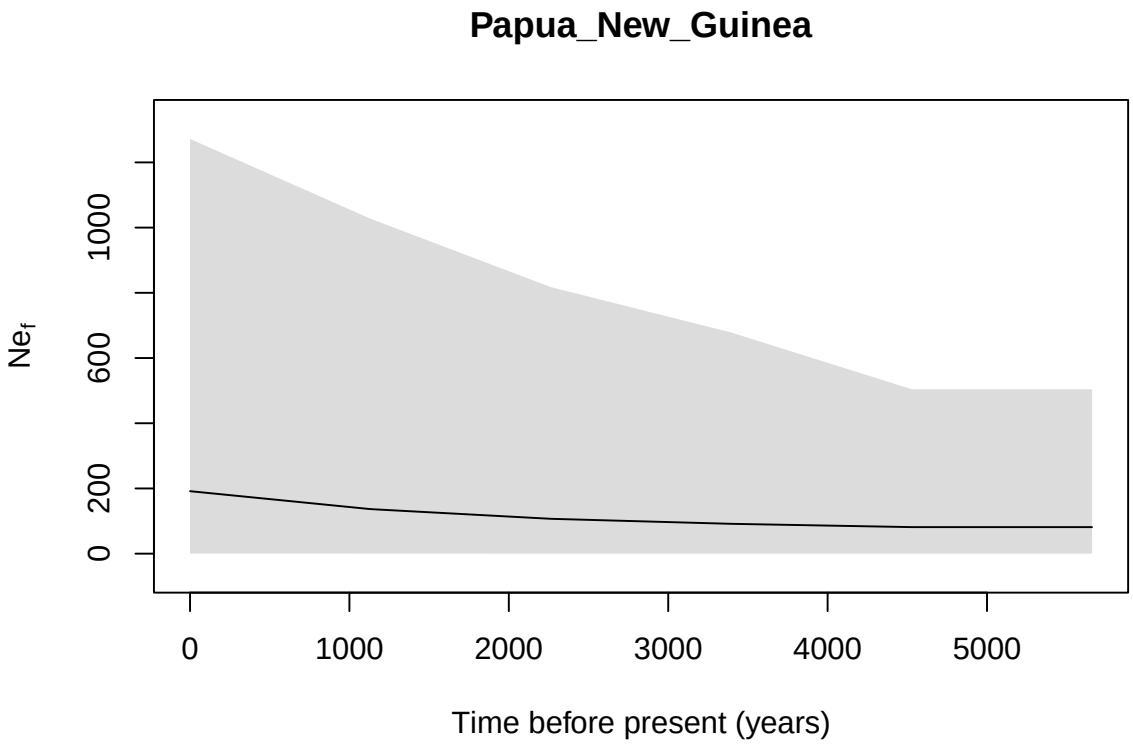
```
for (river in unique(meta$River)) {
  pop_1 <- gsub(pattern = " ", "_", x = river)
  print(pop_1)
  print(paste0("sample size = ", length(meta$River[meta$River == river])))
  folder_1 <- paste0("beast_bspe_",pop_1,"/")
  bdsky_trace   <- beastio::readLog(paste0(folder_1,"tracelog_",pop_1,".log"), burnin = 0.1)
  summary(bdsky_trace)

  Ne_sky <- beastio::getLogFileSubset(bdsky_trace, "bPopSizes")
  Ne_hpd <- t(beastio::getHPDMedian(Ne_sky))

  generation_time <- 27
  # gen_per_Ma <- 1e+06 / generation_time #37037.04 generations in 1 million year
  gen_per_Ma <- 1 / generation_time #0.03703704 generations in 1  year
  Ne_hpd_gen <- Ne_hpd*gen_per_Ma
  tryCatch({ #Error in clip(xlims[1], xlims[2], ylims[1], ylims[2])
    bdskytools::plotSkyline(1:5, Ne_hpd_gen, type = 'step',  main = pop_1,
                            ylab =  expression("Ne"[f]), new = TRUE)
  }, error=function(e){})
  bdskytools::plotSkyline(1:5, Ne_hpd_gen, type = 'step',
                          ylab =  expression("Ne"[f]), new = FALSE)

  delta_hpd <- beastio::getHPDMedian(bdsky_trace[, "TreeHeight"]) #0.027243441 Ma or  27243.44 yr
  # med_age <- delta_hpd[2] * 1e6
  med_age <- delta_hpd[2]
```

```
  time.seq <- seq(from = 0, to = med_age, length.out = 5)
  # times.mrca <- 15e6 - time.seq
  tryCatch({
    bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'step',  main = pop_1,
                            ylab =  expression("Ne"[f]),
                            xlab = "Time before present (years)", new = TRUE)
  }, error=function(e){})
  bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'step',
                          ylab =  expression("Ne"[f]),
                          xlab = "Time before present (years)", new = FALSE)
  dev.print(
    device = png,
    file = paste0("Bayesian_Skyline_plot_stepwise_", pop_1, ".png"),
    res = 300,
    width = 30,
    height = 15,
    units = "cm"
  )
  tryCatch({
    bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'smooth', main = pop_1,
                            ylab = expression("Ne"[f]),
                            xlab = "Time before present (years)", new = TRUE)
  }, error=function(e){})
  bdskytools::plotSkyline(time.seq, Ne_hpd_gen, type = 'smooth',
                          ylab = expression("Ne"[f]),
                          xlab = "Time before present (years)", new = FALSE)
  dev.print(
    device = png,
    file = paste0("Bayesian_Skyline_plot_smooth_", pop_1, ".png"),
    res = 300,
    width = 30,
    height = 15,
    units = "cm"
  )
  bsp.df.pop <- data.frame(time = time.seq, lower = Ne_hpd_gen[1,],
                           med = Ne_hpd_gen[2,], upper = Ne_hpd_gen[3,],
                           pop = pop_1)
  rownames(bsp.df.pop) <- paste0(rownames(bsp.df.pop), "_",bsp.df.pop$pop)
  bsp.df <- rbind(bsp.df, bsp.df.pop)
}
```

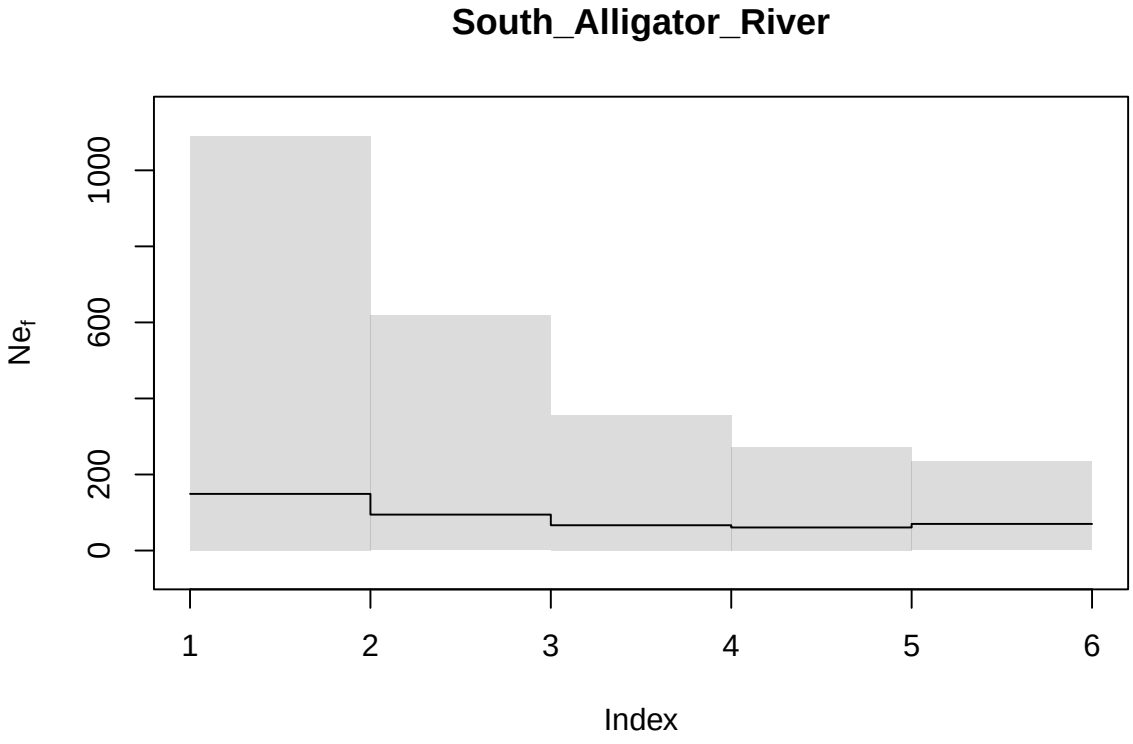
```
## [1] "Papua_New_Guinea"
## [1] "sample size = 6"
```

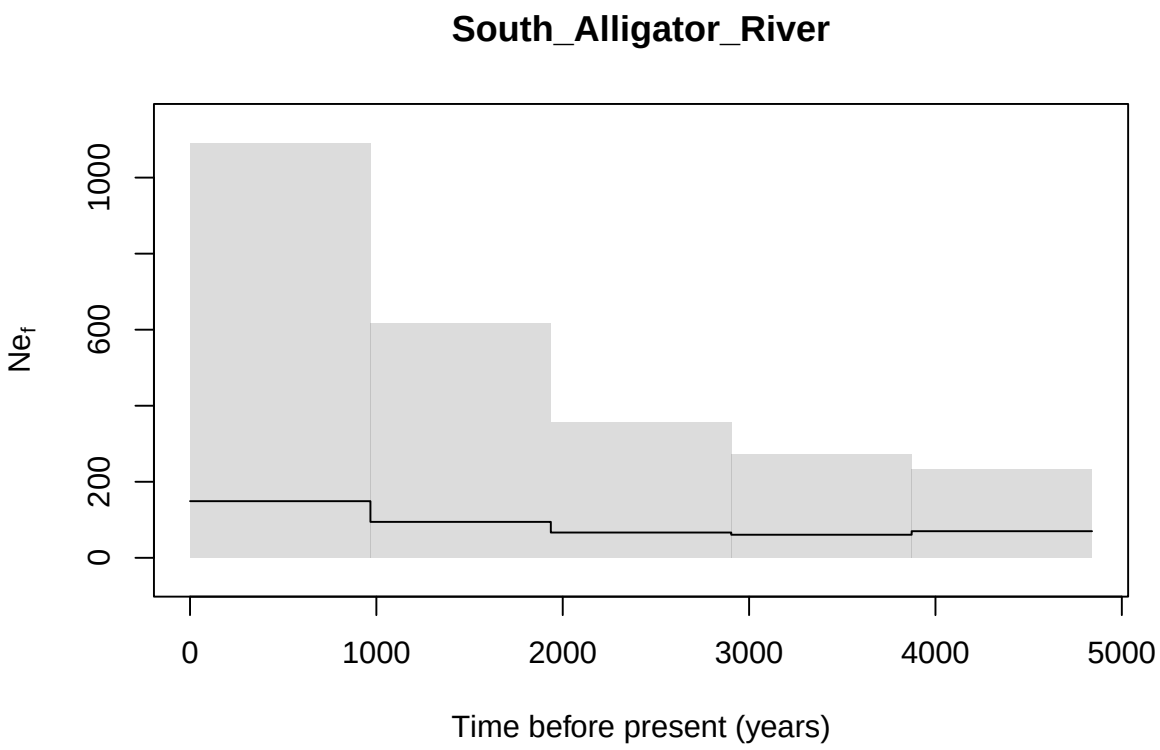

Papua_New_Guinea
Ne_f
0
200
600
1000
1
2
3
4
5
6
Index

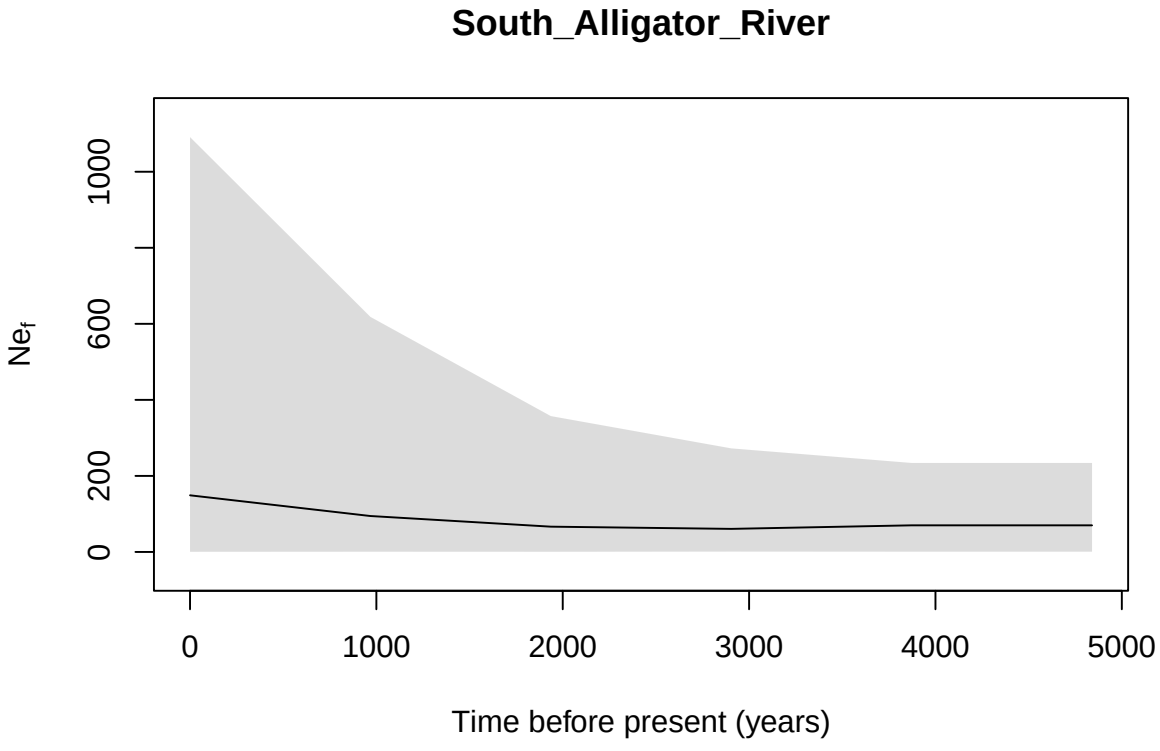

Papua_New_Guinea
Ne_f
0
200
600
1000
0
1000
2000
3000
4000
5000
Time before present (years)



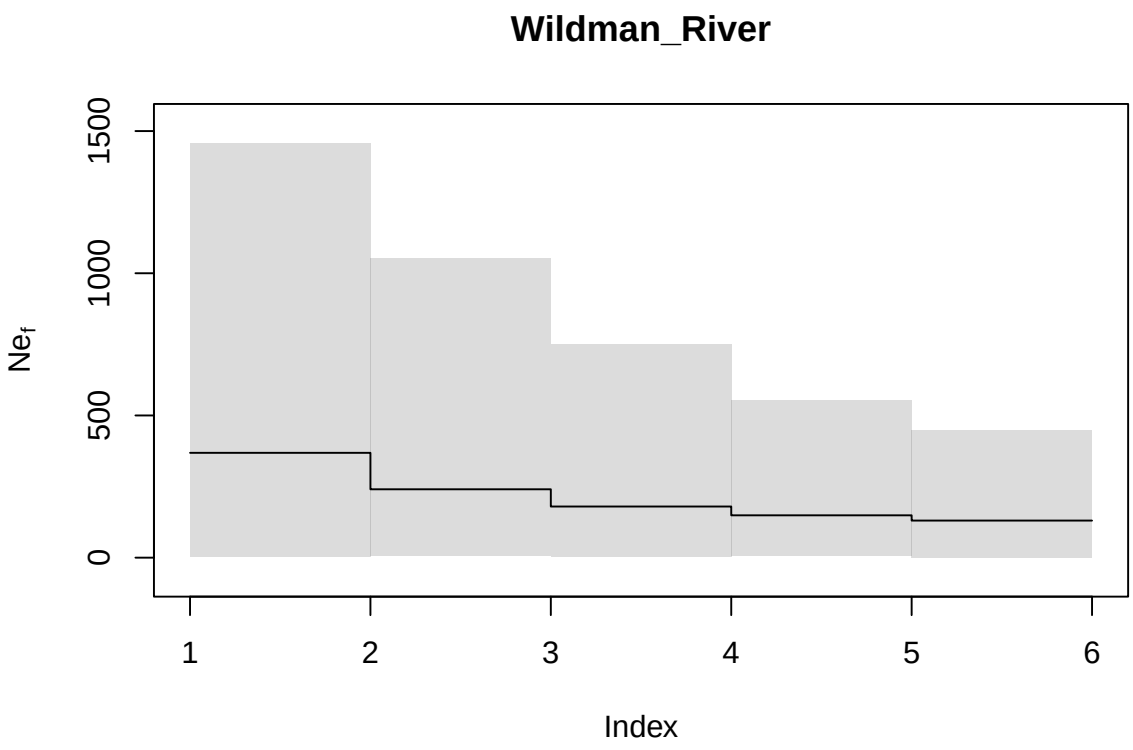

```
## [1] "South_Alligator_River"
## [1] "sample size = 101"
```

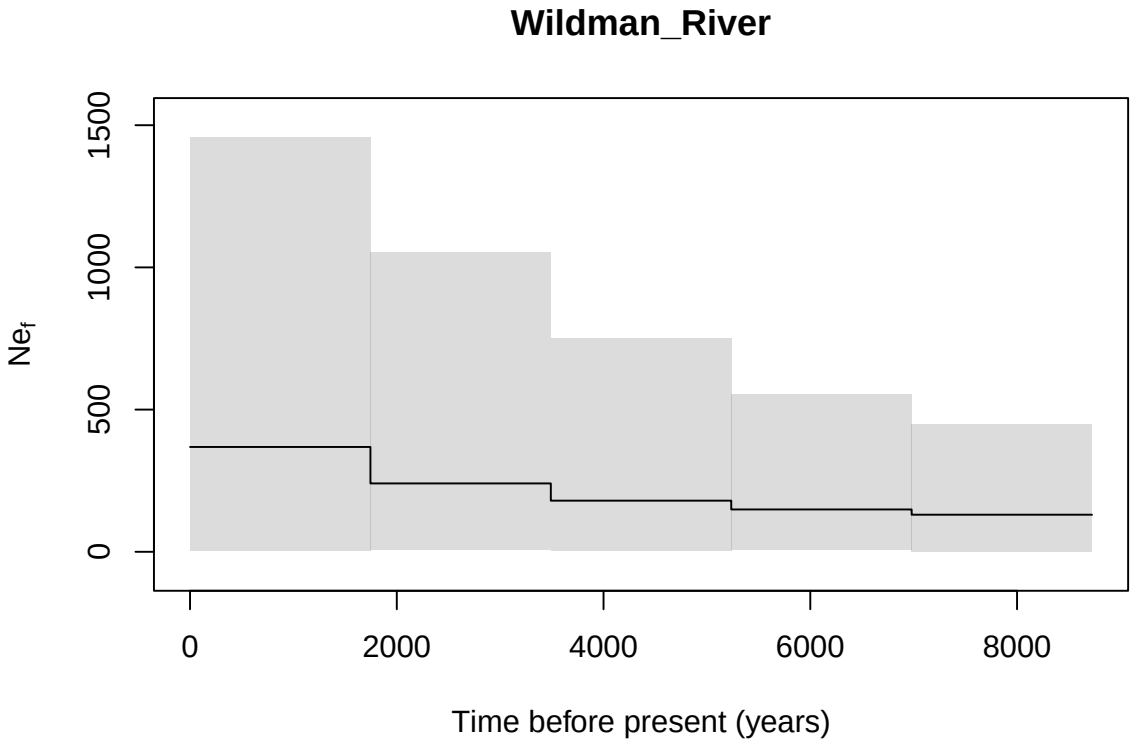


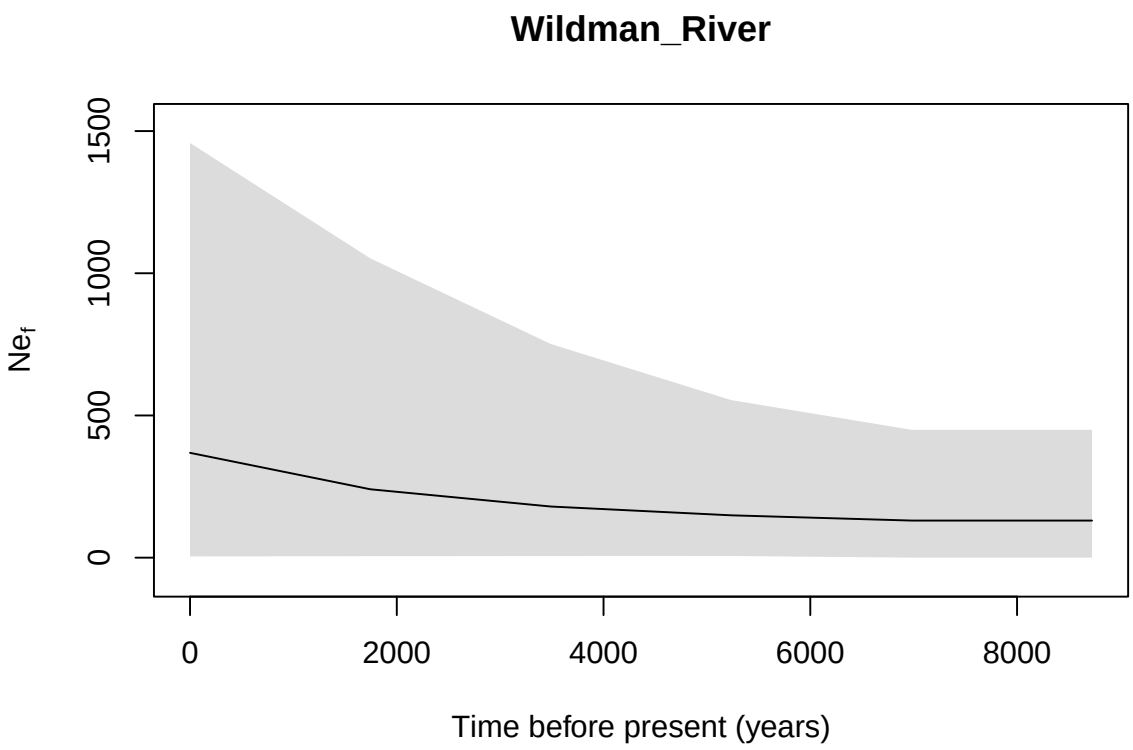




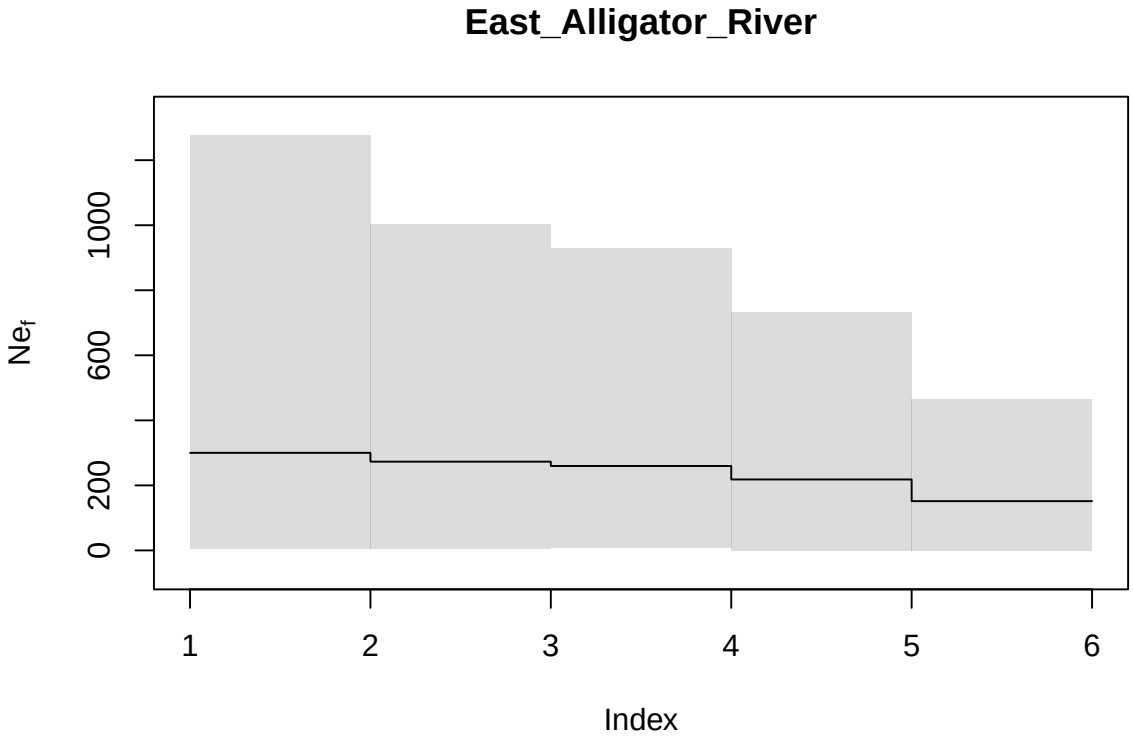

```
## [1] "Wildman_River"
## [1] "sample size = 45"
```

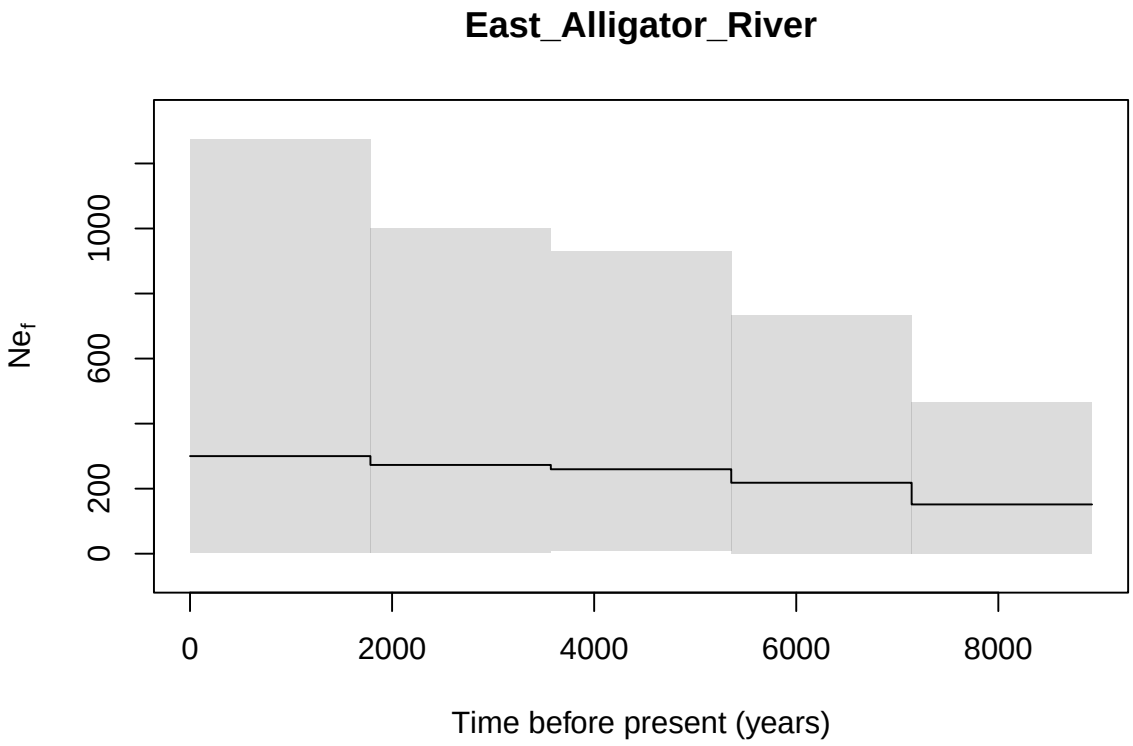

Wildman_River
Ne_f
0
500
1000
1500
1
2
3
4
5
6
Index

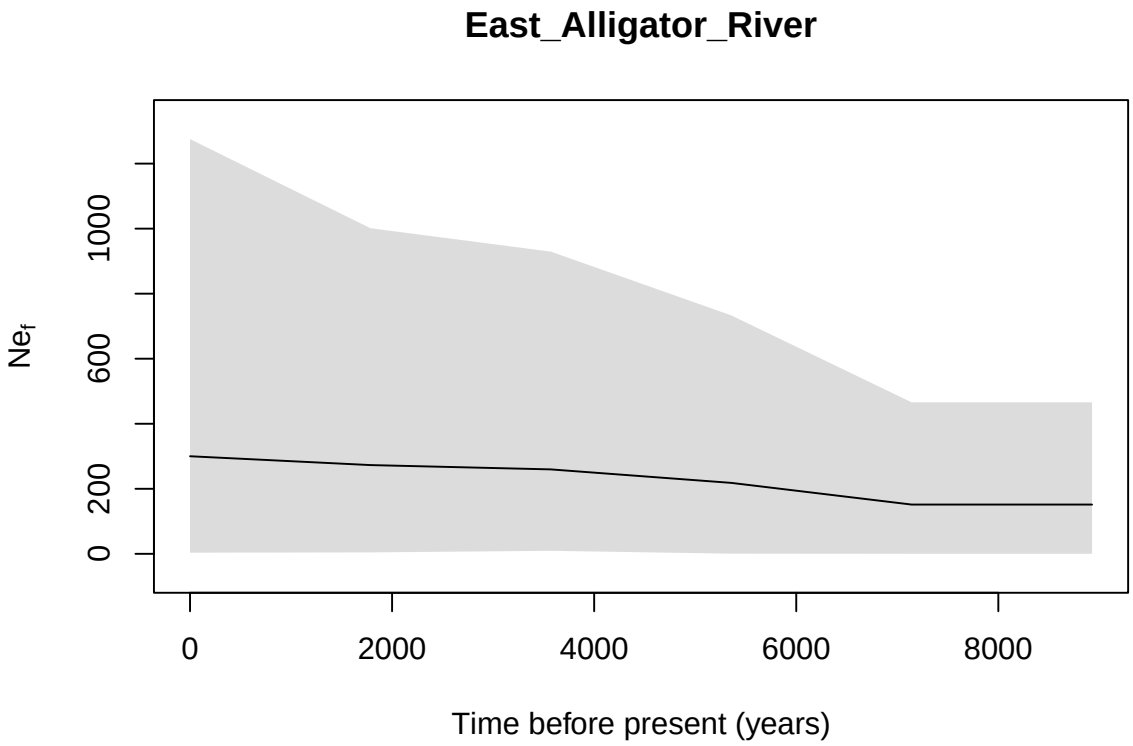

Wildman_River
Ne_f
0
500
1000
1500
0
2000
4000
6000
8000
Time before present (years)



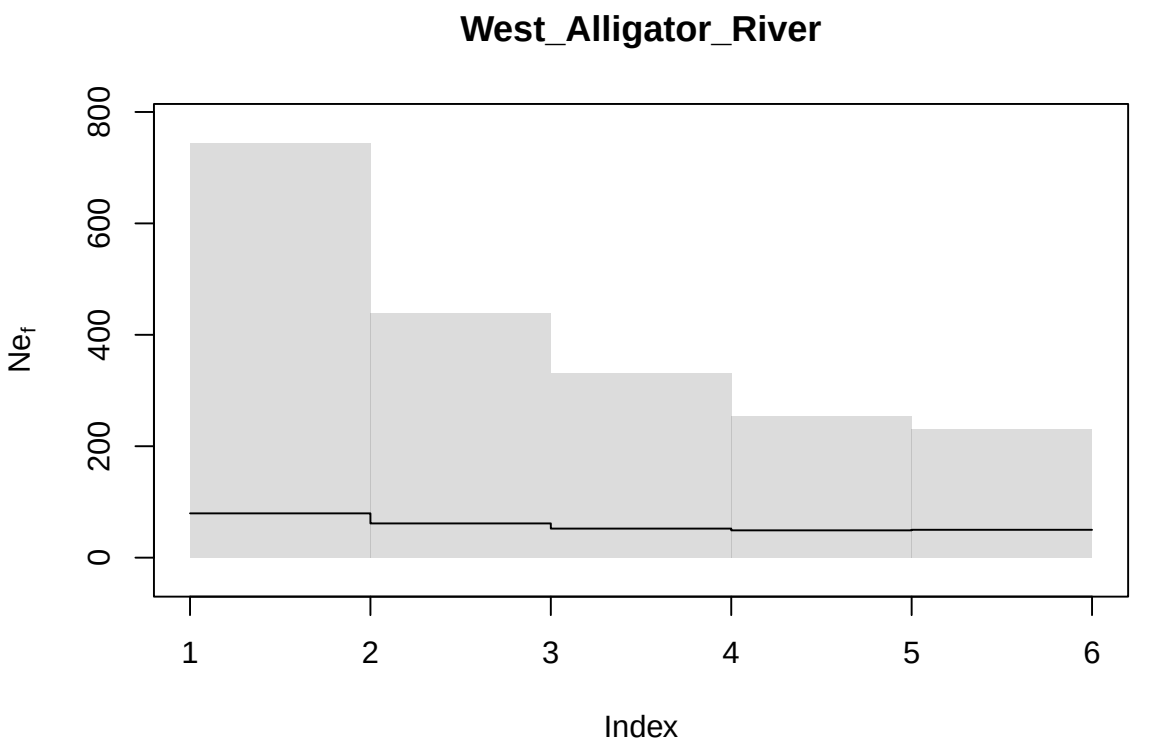

```
## [1] "East_Alligator_River"
## [1] "sample size = 61"
```

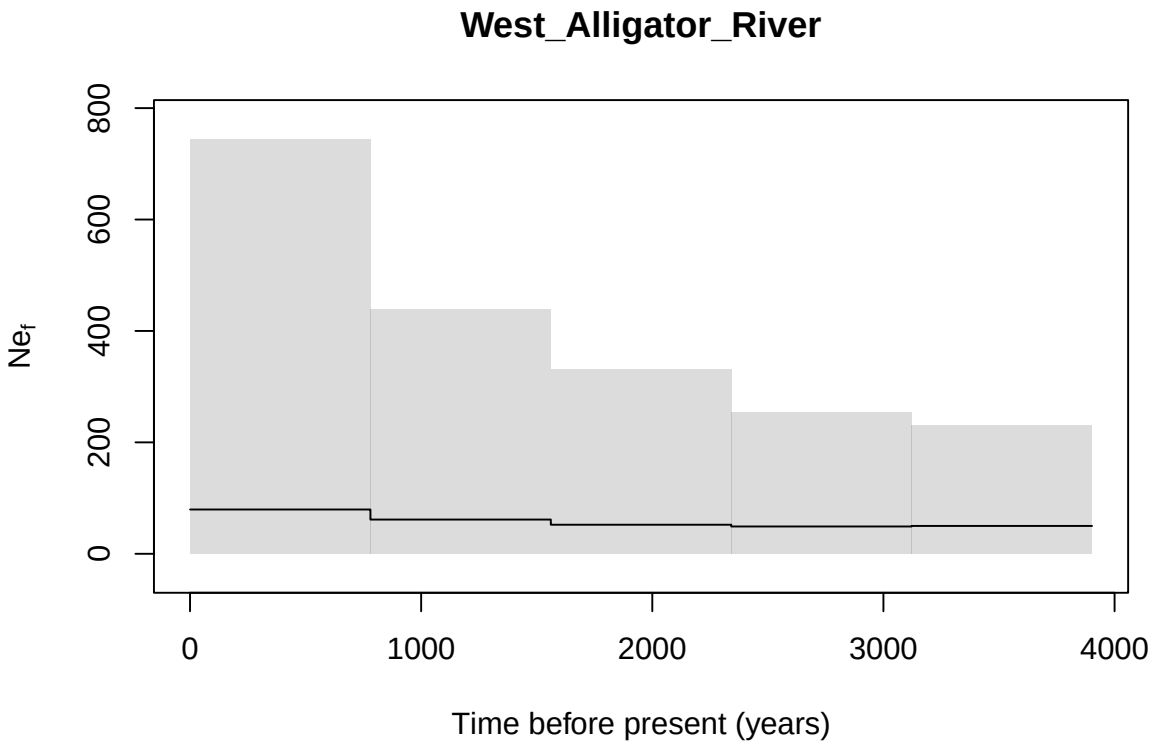


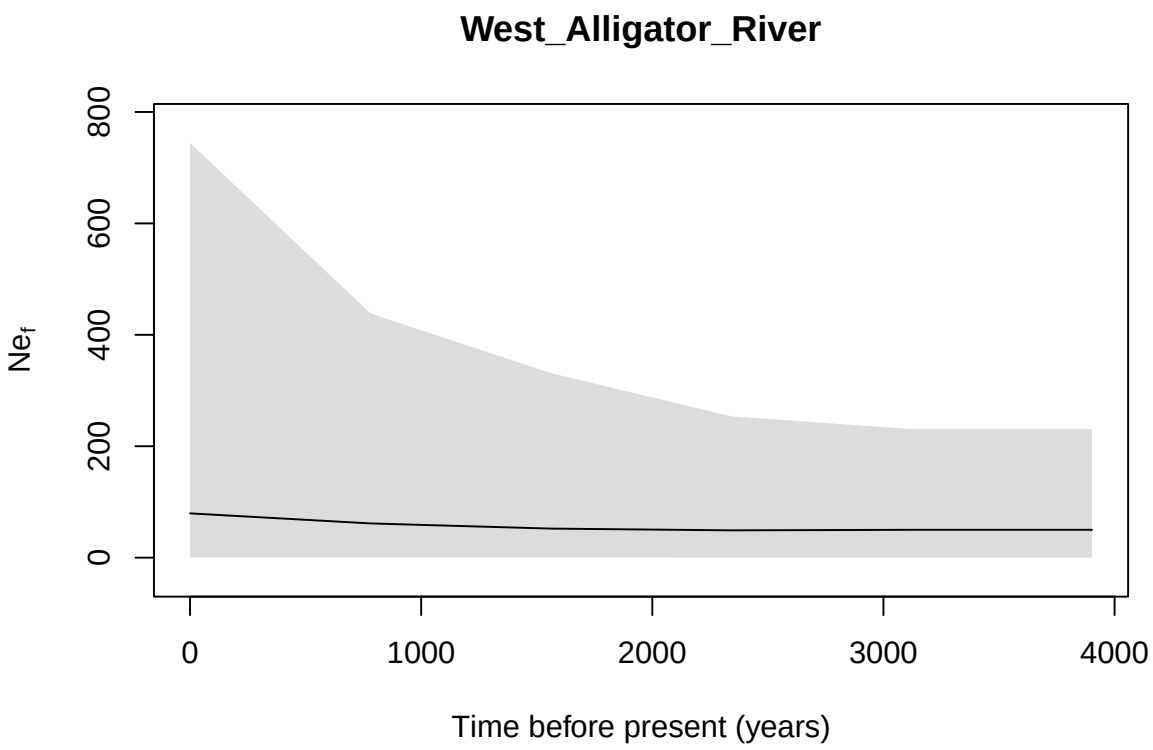




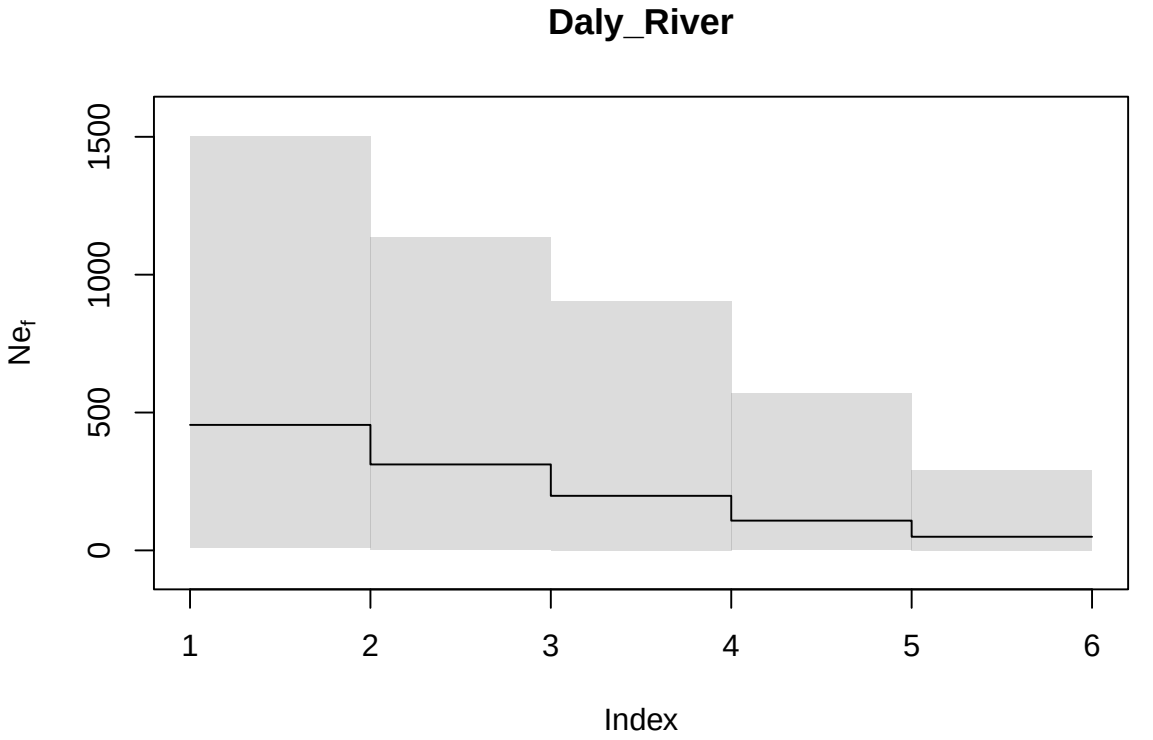

```
## [1] "West_Alligator_River"
## [1] "sample size = 41"
```

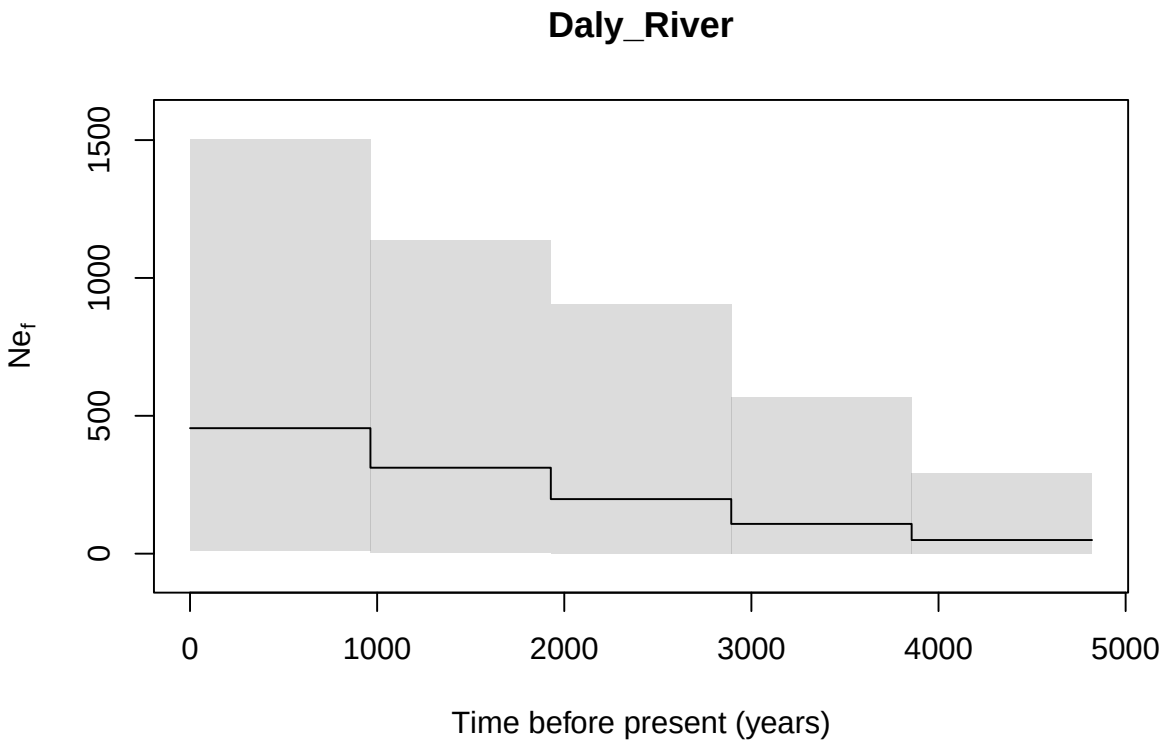

West_Alligator_River
Ne_f
0
200
400
600
800
1
2
3
4
5
6
Index

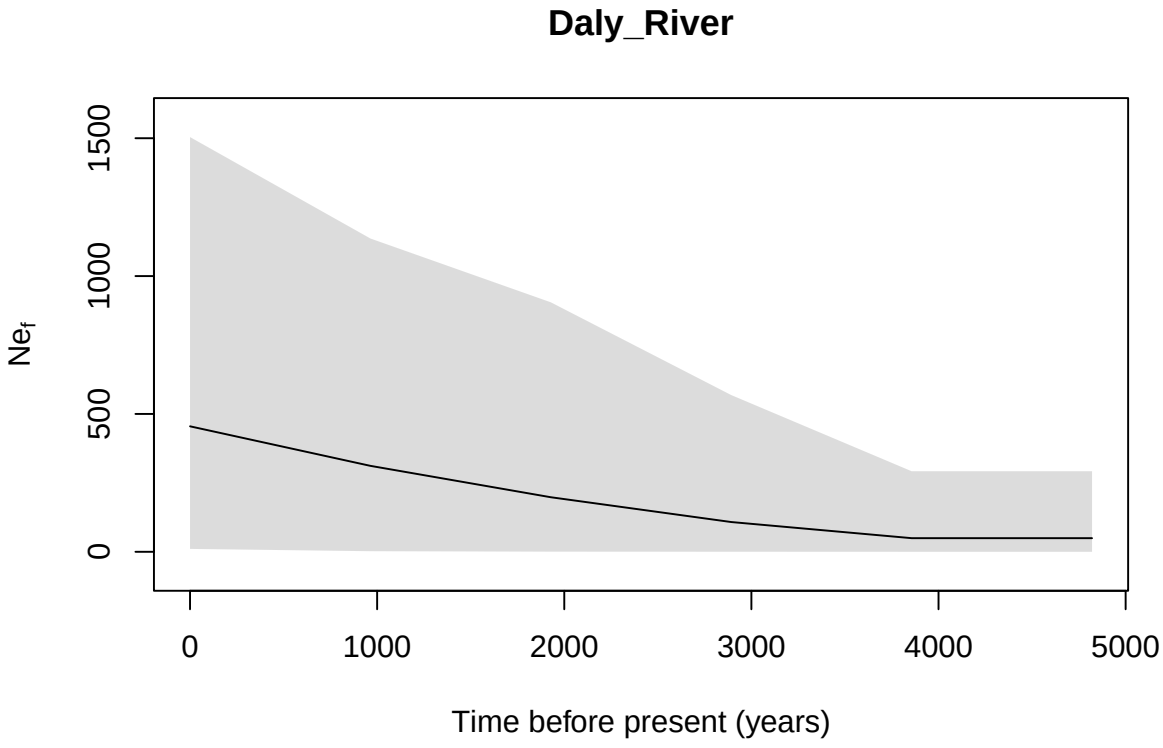

West_Alligator_River
Ne_f
0
200
400
600
800
0
1000
2000
3000
4000
Time before present (years)



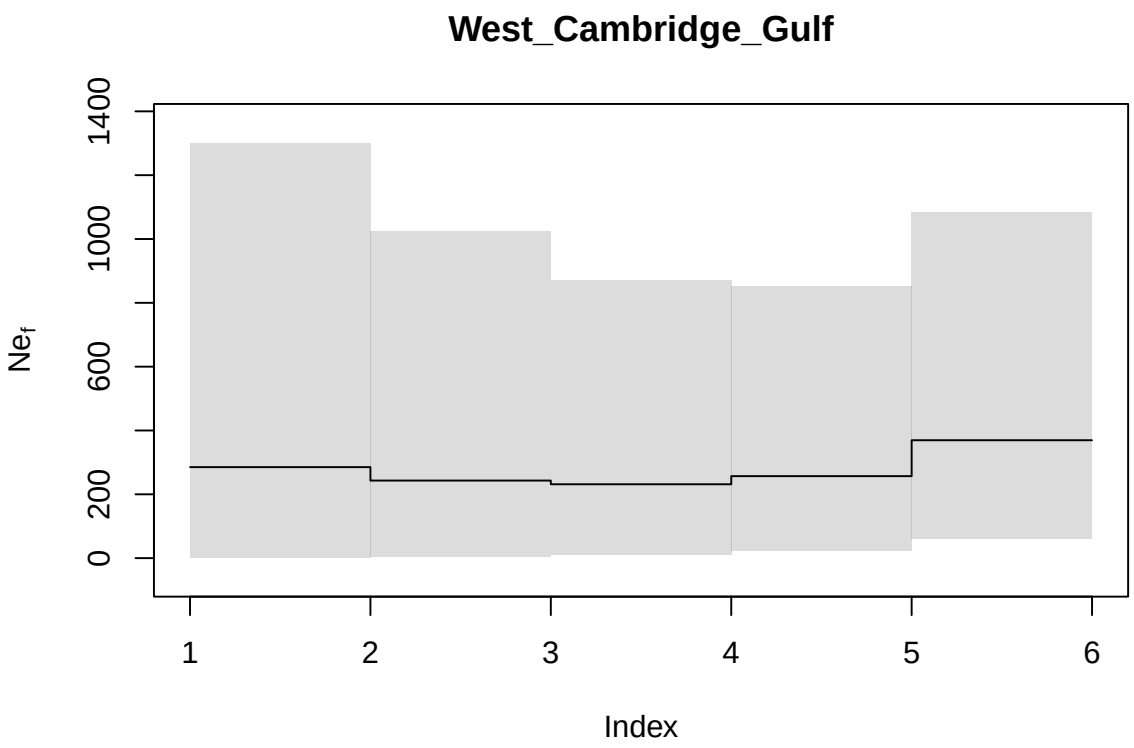

```
## [1] "Daly_River"
## [1] "sample size = 30"
```

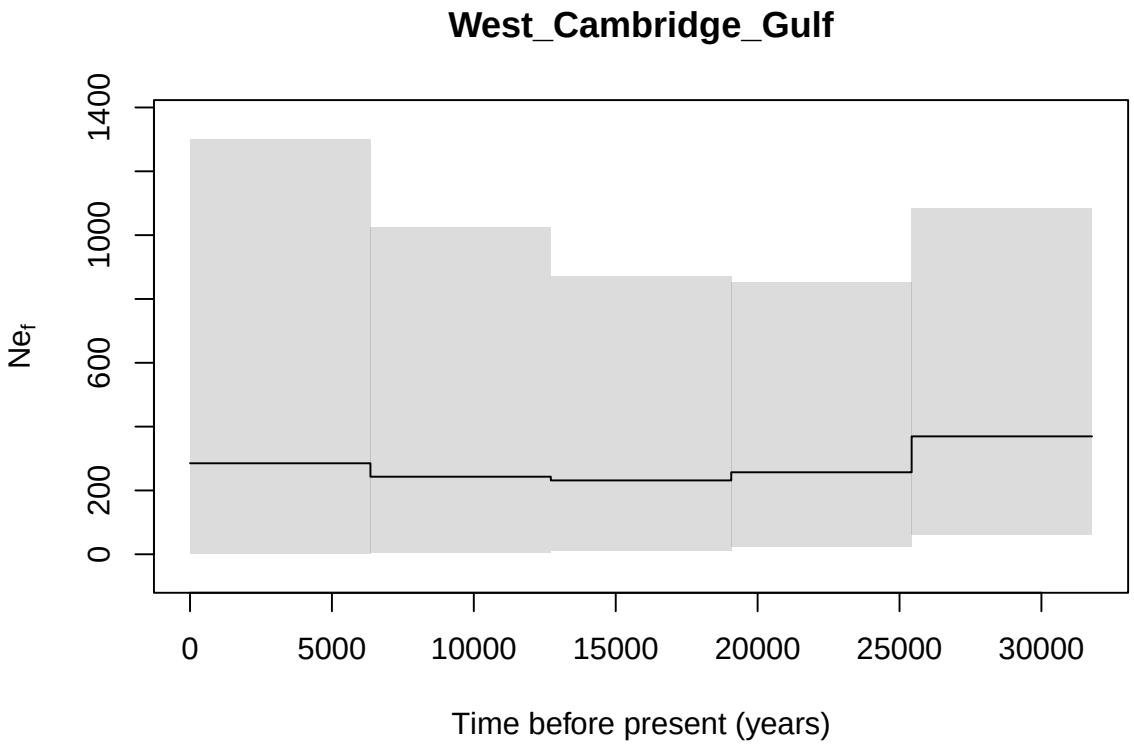


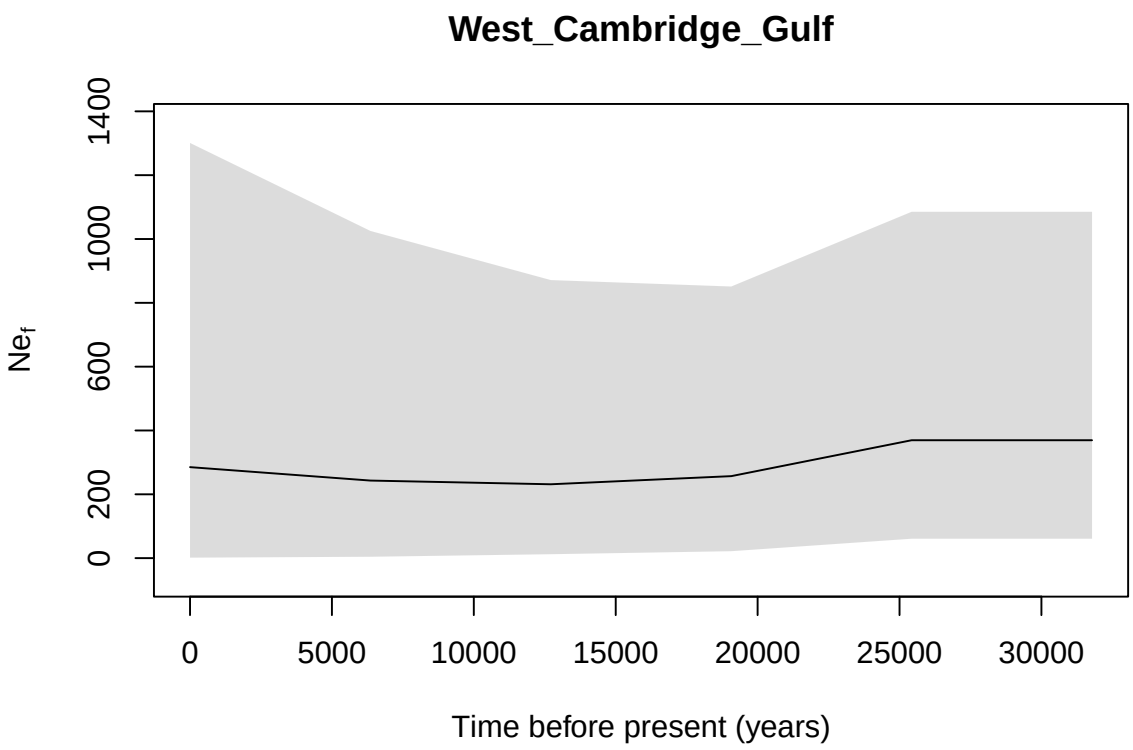




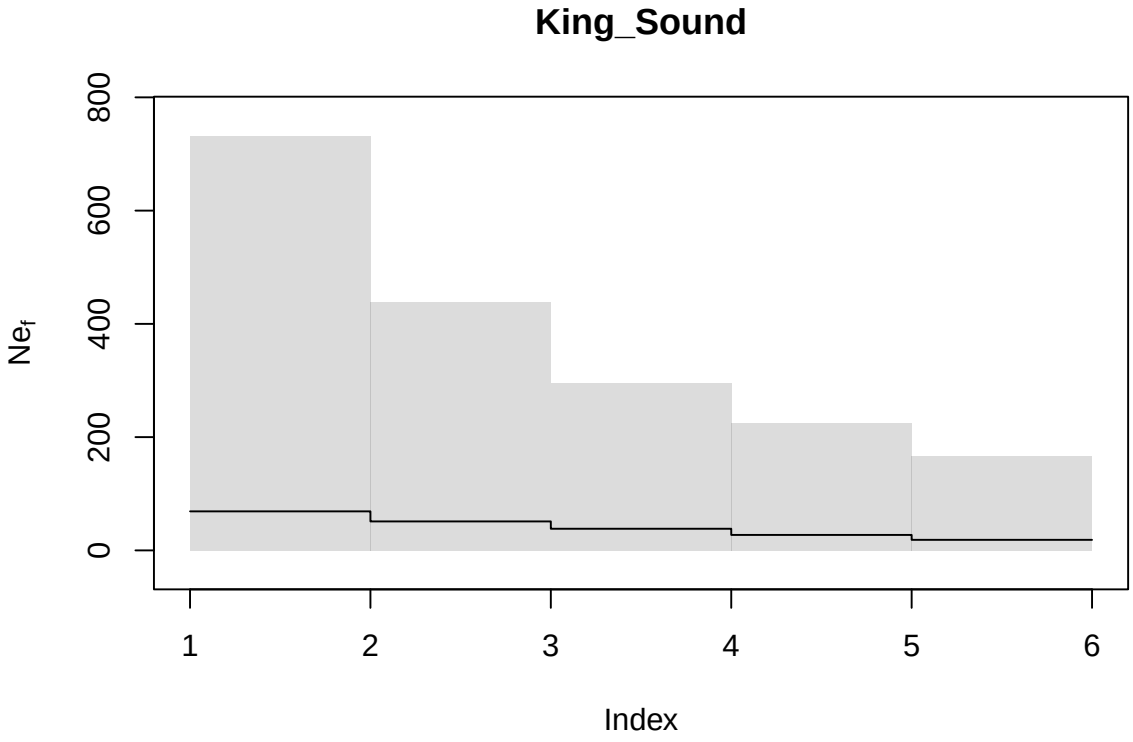

```
## [1] "West_Cambridge_Gulf"
## [1] "sample size = 15"
```

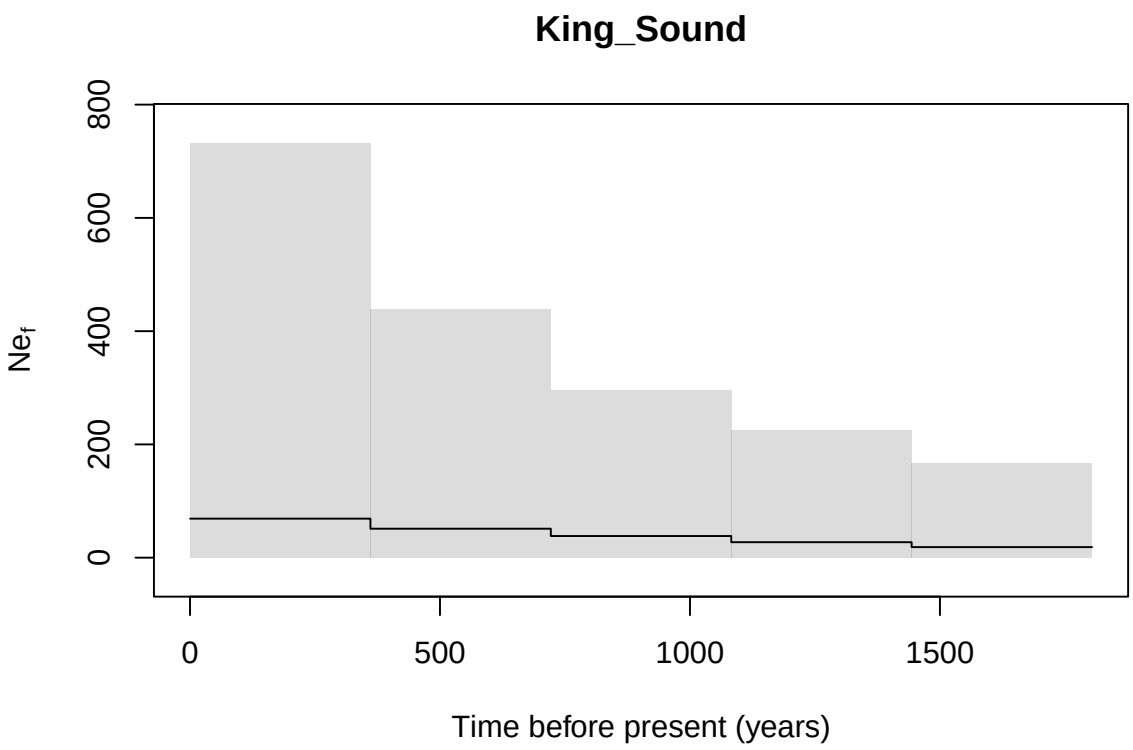

West_Cambridge_Gulf
Ne_f
0
200
600
1000
1400
1
2
3
4
5
6
Index

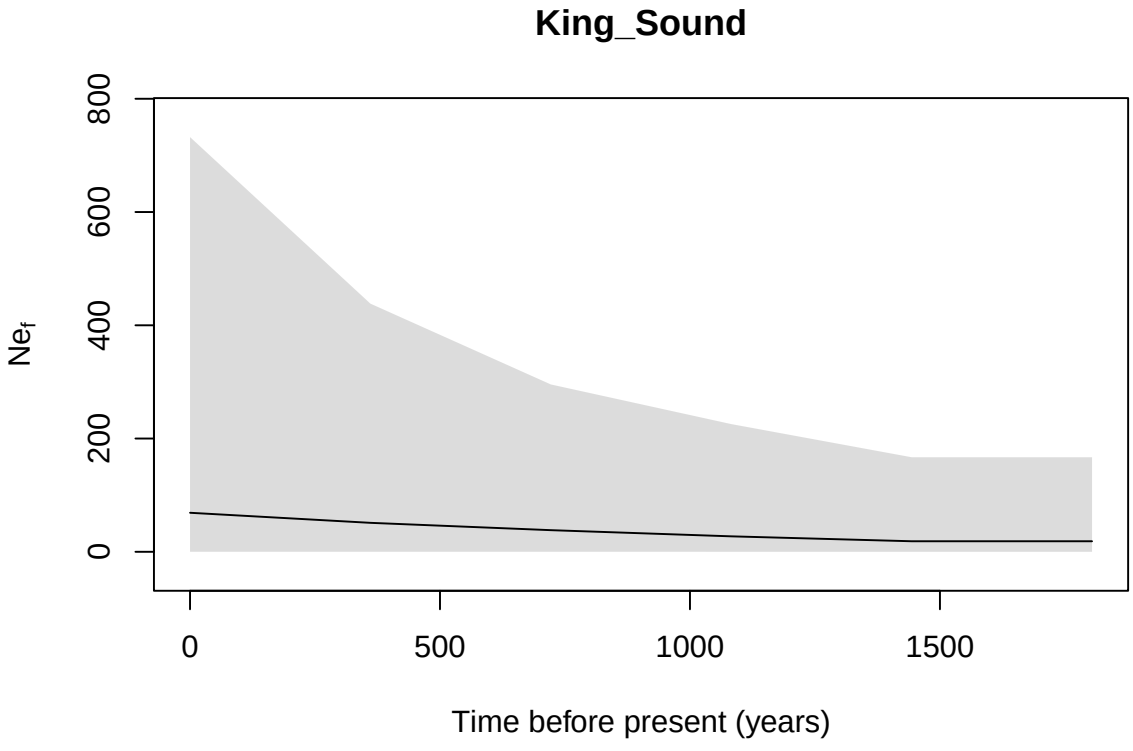

West_Cambridge_Gulf
Ne_f
0
200
600
1000
1400
0
5000
10000
15000
20000
25000
30000
Time before present (years)



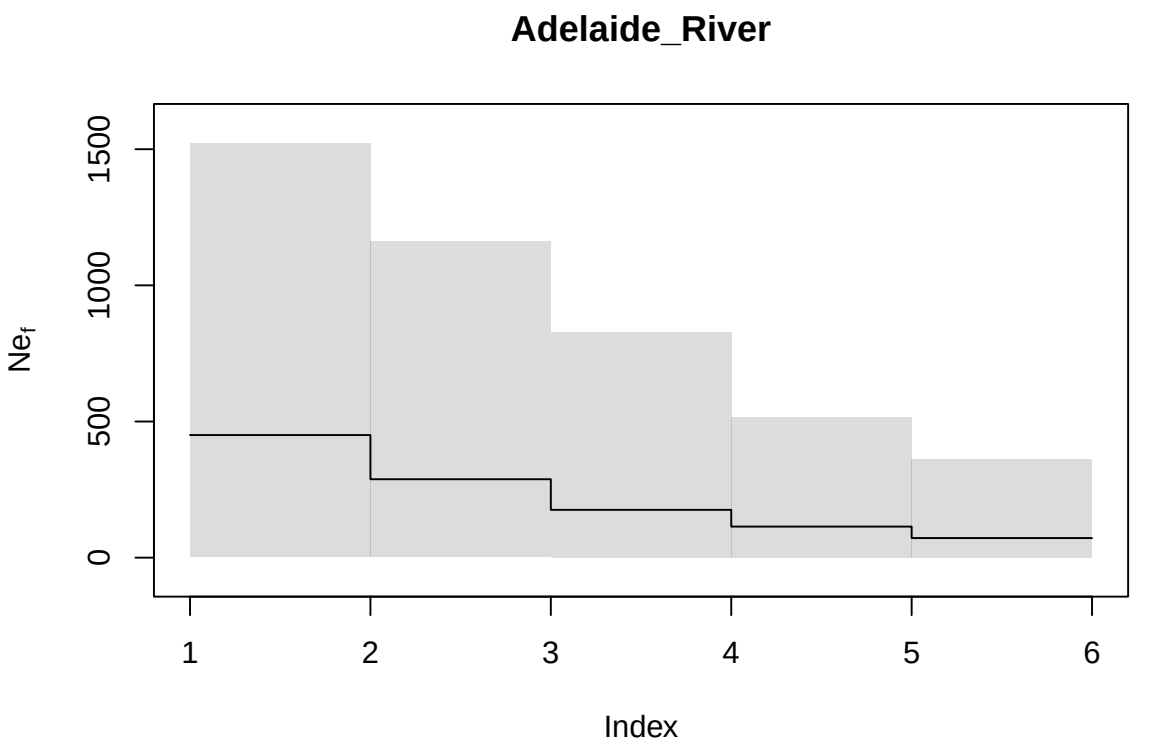

```
## [1] "King_Sound"
## [1] "sample size = 14"
```

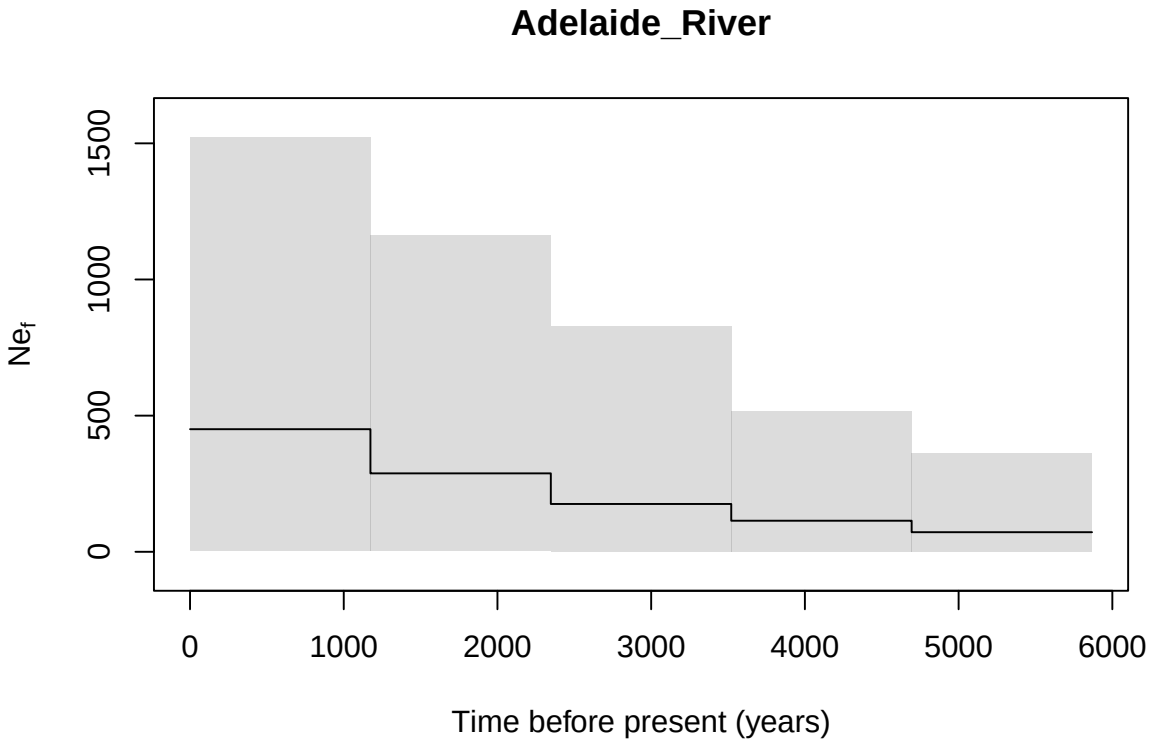


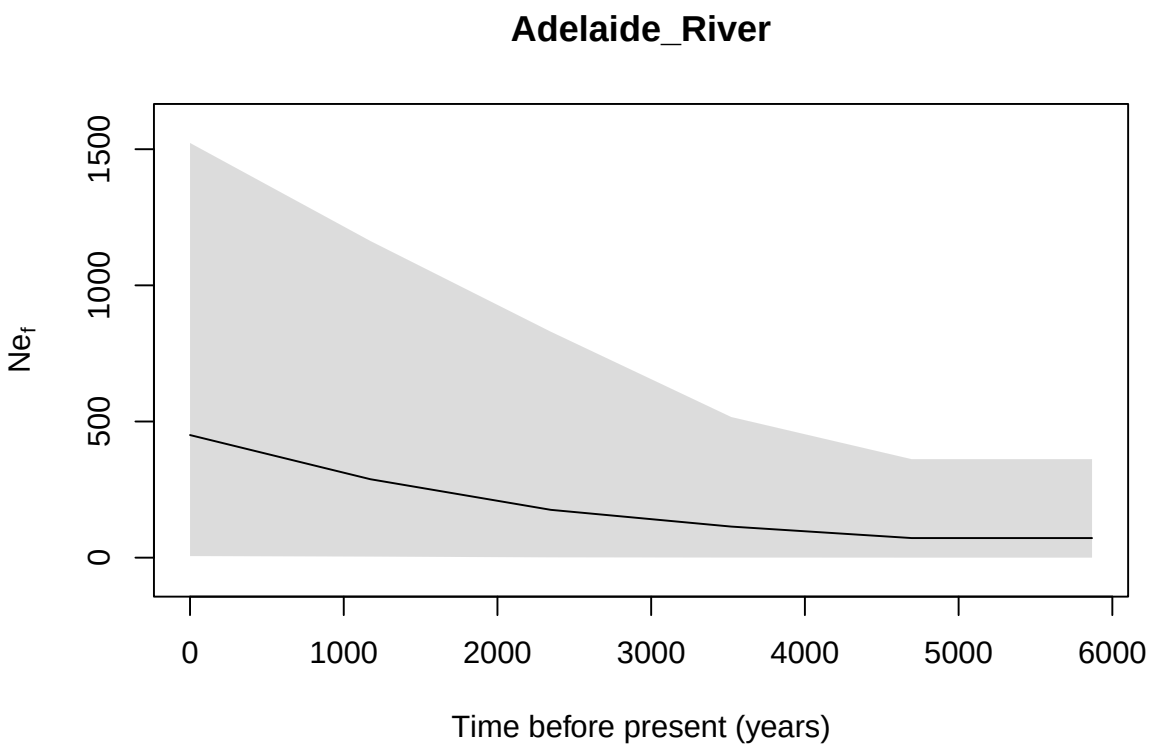




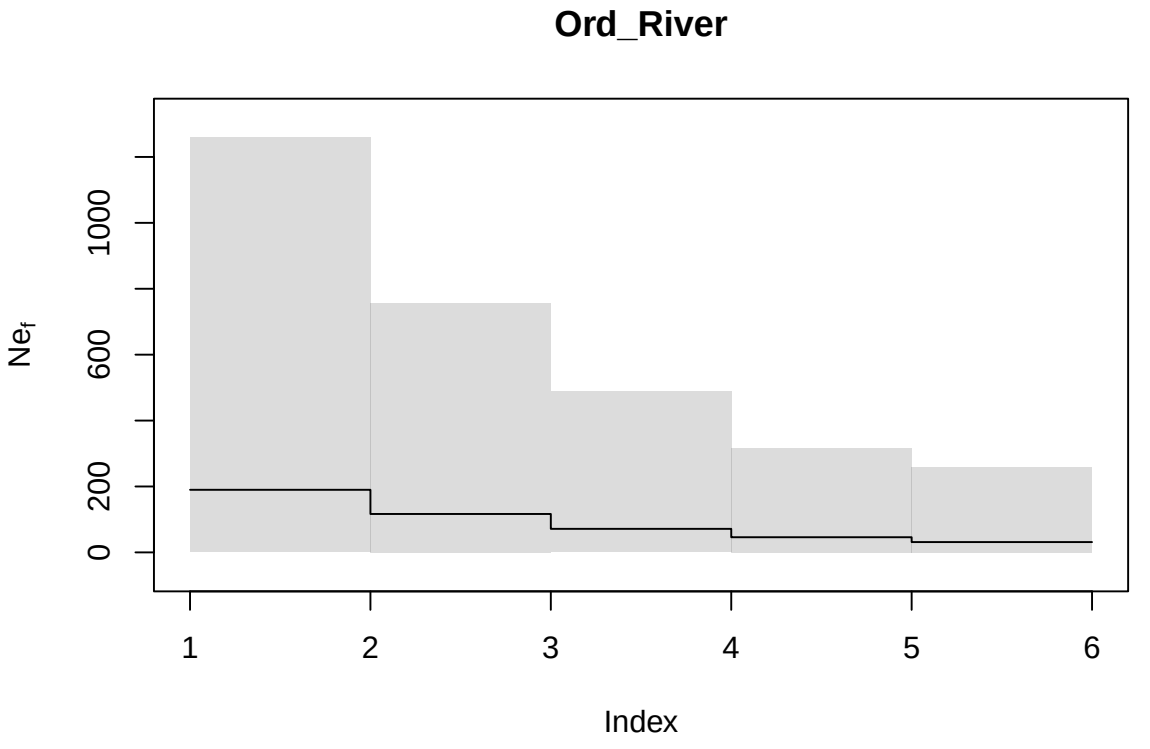

```
## [1] "Adelaide_River"
## [1] "sample size = 25"
```

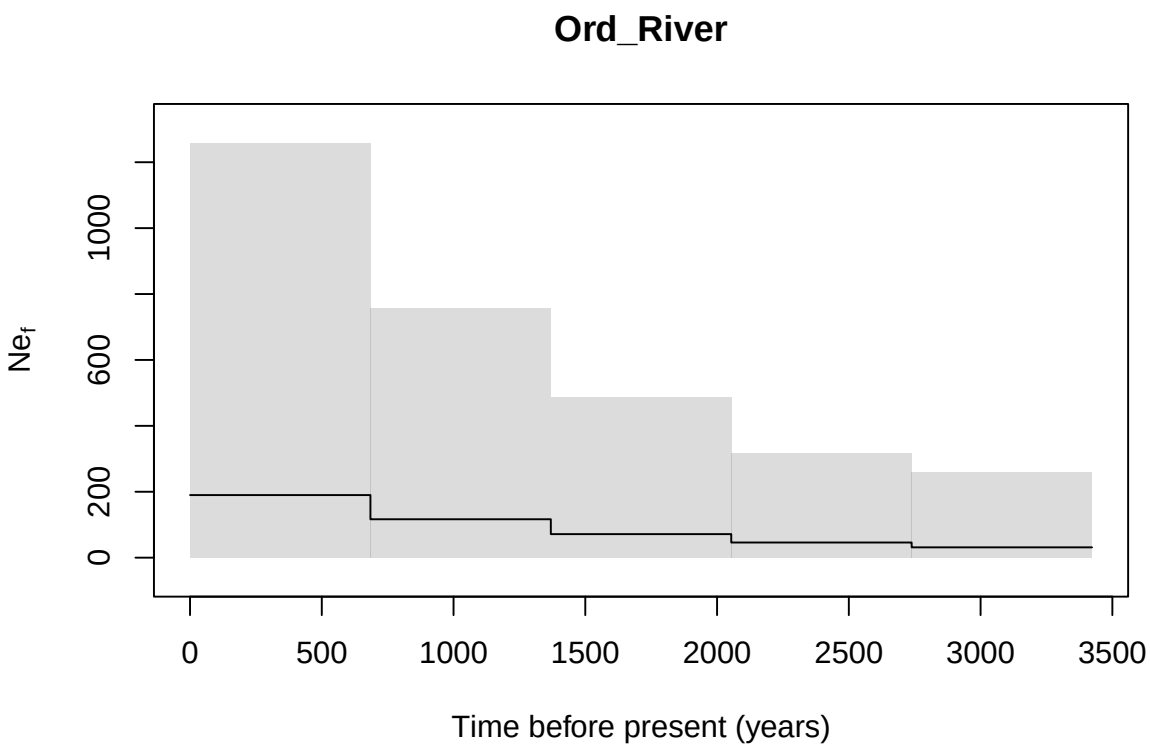

Adelaide_River
Ne_f
0
500
1000
1500
1
2
3
4
5
6
Index

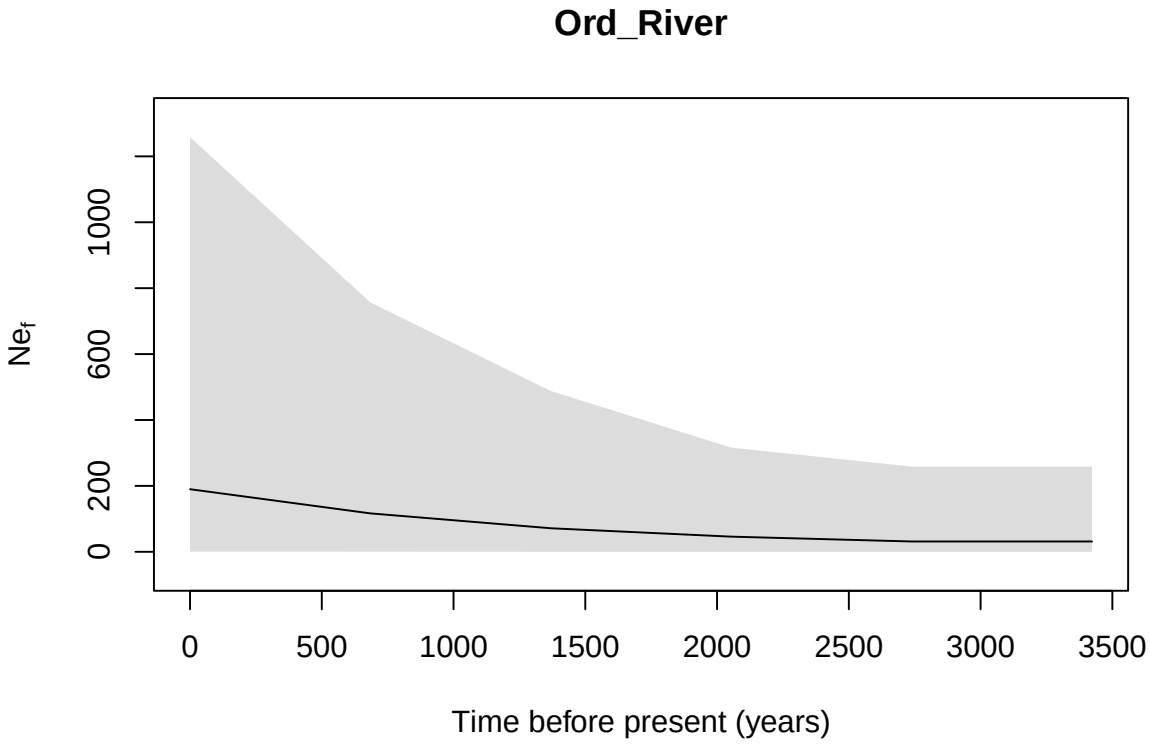

Adelaide_River
Ne_f
0
500
1000
1500
0
1000
2000
3000
4000
5000
6000
Time before present (years)



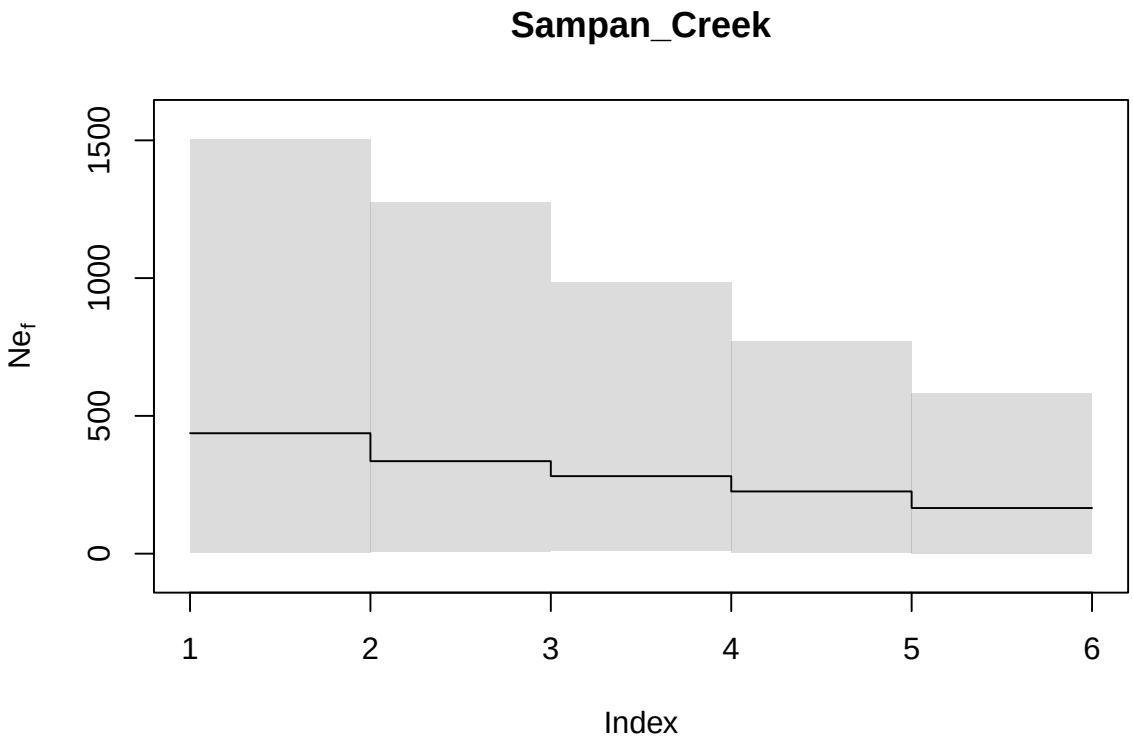

```
## [1] "Ord_River"
## [1] "sample size = 15"
```

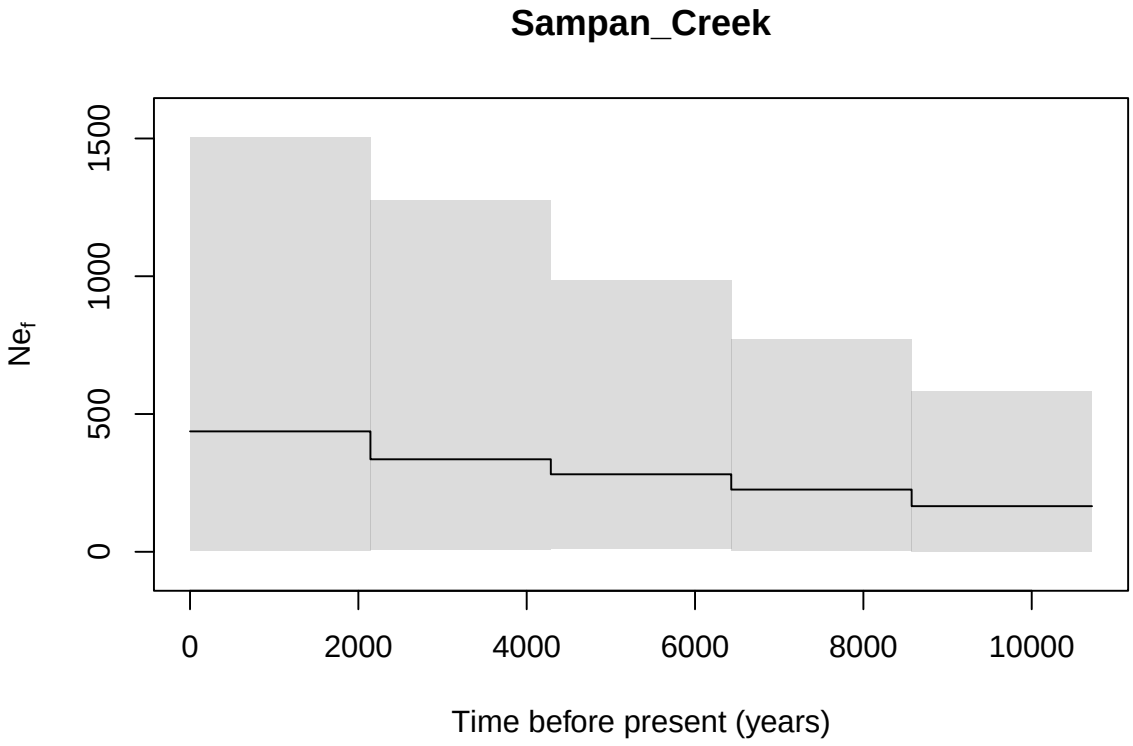


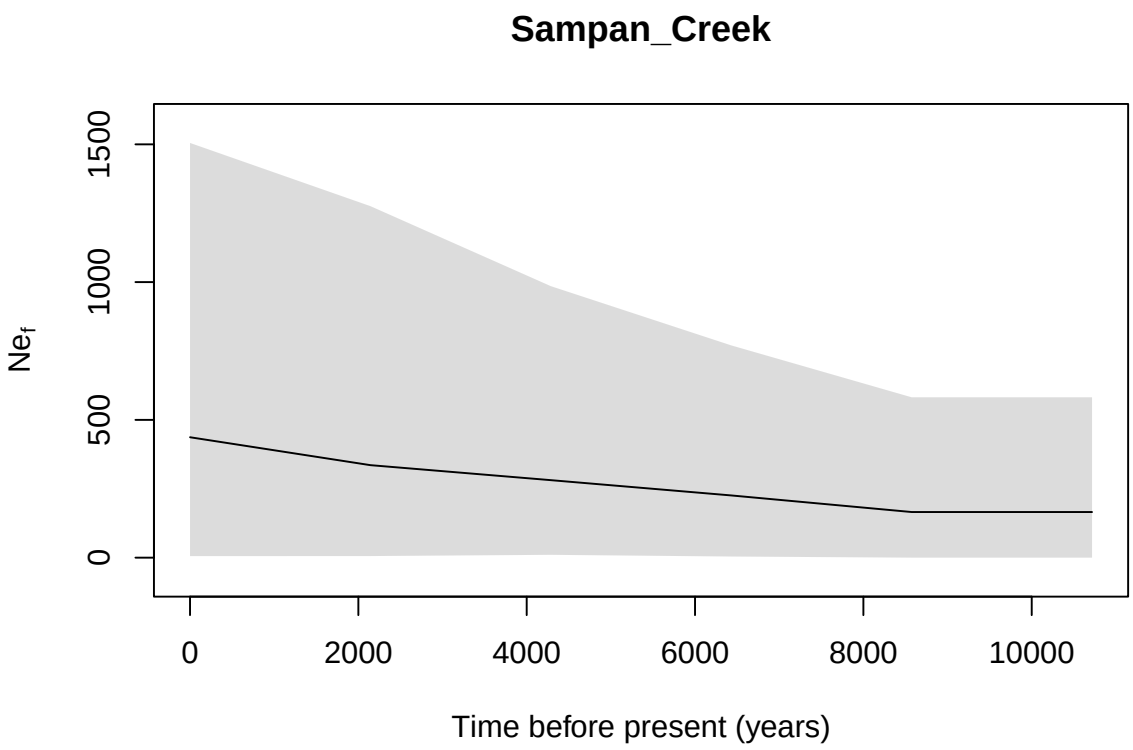





```
## [1] "Sampan_Creek"
## [1] "sample size = 30"
```


Sampan_Creek
Ne_f
0
500
1000
1500
1
2
3
4
5
6
Index


Sampan_Creek
Ne_f
0
500
1000
1500
0
2000
4000
6000
8000
10000
Time before present (years)



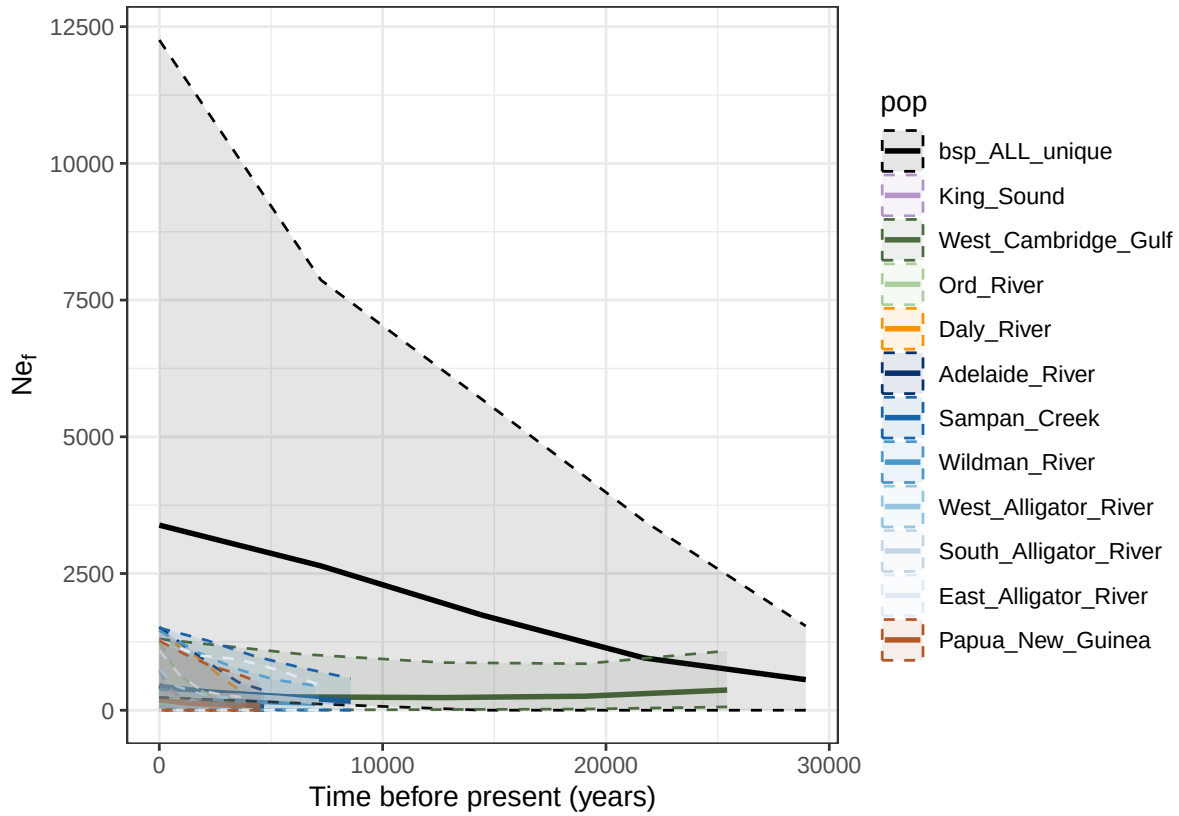

```
bsp.df$pop <- factor(bsp.df$pop, levels = c("bsp_ALL_unique","King_Sound",
                                            "West_Cambridge_Gulf", "Ord_River",
                                            "Daly_River", "Adelaide_River",
                                            "Sampan_Creek","Wildman_River",
                                            "West_Alligator_River",
                                            "South_Alligator_River",
                                            "East_Alligator_River",
                                            "Papua_New_Guinea"))
save(bsp.df, file = "Baysian_skyline_plot_combined_river.Rdata")
```

```
print(load("Baysian_skyline_plot_combined_river.Rdata"))
```

8.4.10.7 Combined BSP

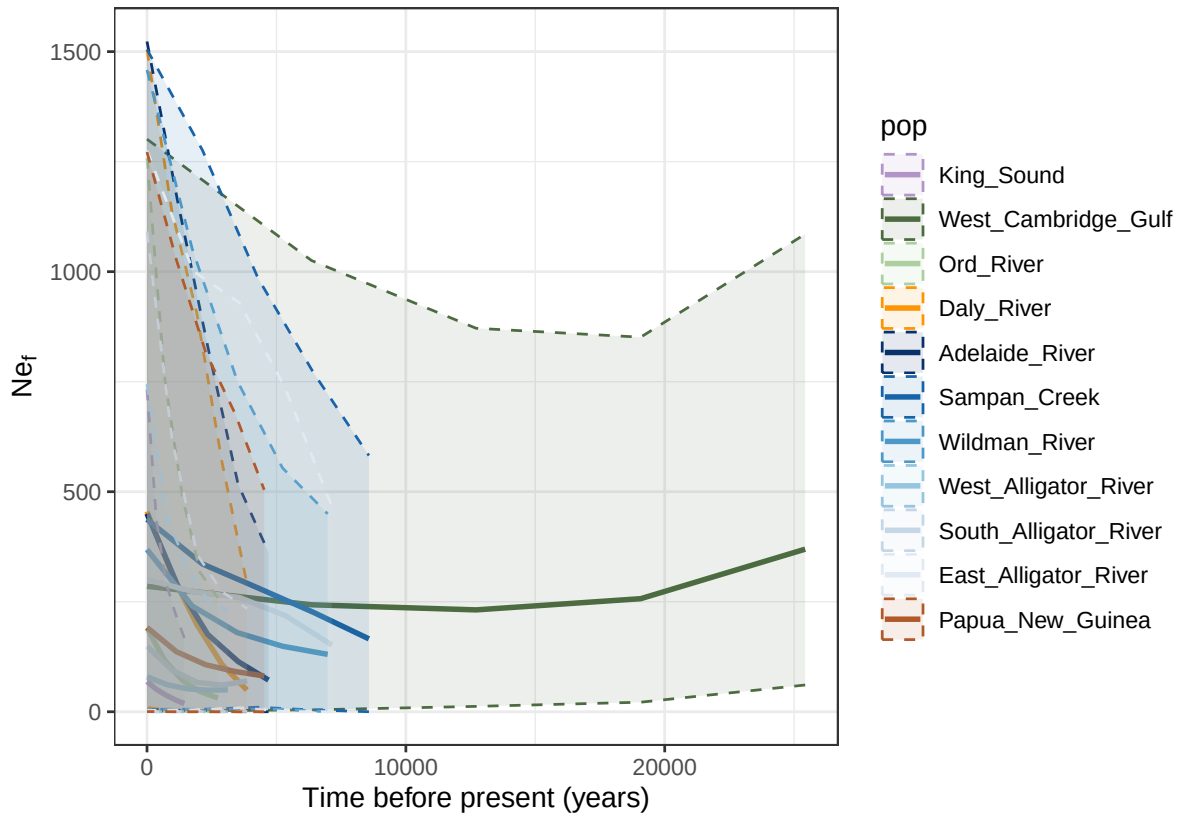
```
## [1] "bsp.df"
```

```
colours.12.bsp <- c("black",colours.11)
names(colours.12.bsp) <- c("bsp_ALL_unique",gsub(pattern = " ", "_",
                                                 x = names(colours.11)))

bsp <- ggplot2::ggplot(data = bsp.df, ggplot2::aes(x = time, y = med,
                                                   colour = pop)) +
  ggplot2::geom_line(size = 1) +
  ggplot2::geom_ribbon(aes(ymin = lower, ymax = upper, fill = pop),
                       linetype = 2, alpha = 0.1) +
  ggplot2::xlab("Time before present (years)") +
  ggplot2::ylab(expression("Ne"[f])) +
  ggplot2::scale_colour_manual(values = colours.12.bsp) +
  ggplot2::scale_fill_manual(values = colours.12.bsp) +
  ggplot2::theme_bw()

print(bsp)
```




```
ggplot2::ggsave(plot = bsp, filename = "Baysian_skyline_plot_combined_river.png",
                device = "png", width = 30, height = 15, units = "cm")


bsp2 <- ggplot2::ggplot(data = bsp.df[bsp.df$pop != "bsp_ALL_unique",],
                        ggplot2::aes(x = time, y = med, colour = pop)) +
  ggplot2::geom_line(size = 1) +
  ggplot2::geom_ribbon(aes(ymin = lower, ymax = upper, fill = pop),
```

```
                        linetype = 2, alpha = 0.1) +
  ggplot2::xlab("Time before present (years)") +
  ggplot2::ylab(expression("Ne"[f])) +
  ggplot2::scale_colour_manual(values = colours.12.bsp) +
  ggplot2::scale_fill_manual(values = colours.12.bsp) +
  ggplot2::theme_bw()

print(bsp2)
```




```
ggplot2::ggsave(plot = bsp2,
                filename = "Baysian_skyline_plot_combined_river_subset.png",
                device = "png", width = 30, height = 15, units = "cm")
```

---

9 Nuclear SNP data

9.1 DArTcap SNPs

9.1.1 Read SNP data - genlight format

```
dartfile <- "../3.Data/Glyphis garricki_SNP_data.csv"
lastmetric <- "RepAvg"

gl <- dartR.base::gl.read.dart(filename = dartfile,
                          covfilename = metafile, topskip = NULL,
                          lastmetric = lastmetric, probar = F, verbose = 0)
```

```
## Ids for individual metadata does not match the number of
ids in the SNP data file. Maybe this is fine if a subset
matches.
## ind.metafile ids not matched were:
## [1] "BA001" "BA002" "BA003" "CP001" "CP002" "CP003"
## [7] "CP004A" "CP004B" "CP005" "CP006A" "CP006B" "CP007A"
## [13] "CP007B" "CP008A" "CP008B" "CP009A" "CP009B"
"CP010A"
## [19] "CP010B" "CP011" "CP012A" "CP012B" "CP013" "CP014"
## [25] "CP015" "CP016" "CP017" "CP018" "CP019" "GGA004"
## [31] "GGA025" "GGA072" "GGA192" "GGA223" "GGA227"
"GGA273"
## [37] "GGA274" "GGA275" "GGA276" "GGA277" "GGA278"
"GGA279"
## [43] "GGA280" "GGA281" "GGA282" "GGA283" "GGA284"
"GGA285"
## [49] "GGA286" "GGA287" "GGA288" "GGA289" "GGA290"
"GGA291"
## [55] "GGA292" "GGA293" "GGA294" "GGA295" "GGA296"
"GGA297"
## [61] "GGA298" "GGA299" "GGA300" "GGA301" "GGA302"
"GGA303"
## [67] "GGA312" "GGA313" "GGA314" "GGA315" "GGA316"
"GGA317"
## [73] "GGA318" "GGA319" "GGA320" "GGA321" "GGA322"
"GGA323"
## [79] "GGA324" "GGA325" "GGA326" "GGL034" "GGL035"
"GGL036"
## [85] "GGL037" "GGL038" "GGL039" "GGL040" "GGL041"
"GGL042"
## [91] "GGL043" "GGL044" "GGL045" "GGL046" "GGL047"
"GGL048"
## [97] "GGL049" "GGSA012" "KF646786" "KT698042" "KT698044"
"KT698053"
## [103] "KT698059" "NC_023361"
## Starting gl.compliance.check
## Processing genlight object with SNP data
## Checking coding of SNPs
## SNP data scored NA, 0, 1 or 2 confirmed
## Checking for population assignments
## Population assignments confirmed
## Checking locus metrics and flags
## Recalculating locus metrics
## Checking for monomorphic loci
## Dataset contains monomorphic loci
## Checking for loci with all missing data
```

```
## No loci with all missing data detected
## Checking whether individual names are unique.
## Checking for individual metrics
## Individual metrics confirmed
## Spelling of coordinates checked and changed if necessary
to
## lat/lon
## Completed: gl.compliance.check
```

```
pop.levels <- c("King Sound", "West Cambridge Gulf", "Ord River", "Daly River",
                "Adelaide River", "Sampan Creek", "Wildman River",
                "West Alligator River", "South Alligator River",
                "East Alligator River","Papua New Guinea")
gl$pop <- factor(x = gl$other$ind.metrics$River, levels = pop.levels)
gl <- gl[order(gl$pop, gl$ind.names)]
```

9.1.2 Subset data

Subset so the data only includes the exact individuals that were sequenced for the mitogenome; except for PNG, where additional individuals were included.

```
gl <- gl[adegenet::indNames(gl) %in% adegenet::indNames(ggar.gi) |
           gl$pop == "Papua New Guinea",]
dim(gl)
```

```
## [1]  375 9111
```

9.1.3 Filter SNP data

See Feutry et al. 2020 for more details.

```
gl$other$loc.metrics[["SumCount"]] <- rowSums(gl$other$loc.metrics
                                              [,c("AvgCountRef","AvgCountSnp")])
names(gl$other$loc.metrics)[names(gl$other$loc.metrics) == 'clone'] <- 'CloneID'

gl <- gl.filter.secondaries.FDD(gl)

gl <- gl.filter.missing.data.FDD(gl, loc.lb = 0.5, loc.hb = 0.85 ,
                                 ind.lb = 0.5, ind.hb = 0.84 ,iterations = 100)
gl.report.het.FDD(gl) #not necessary

gl <- dartR::gl.filter.monomorphs(gl, verbose = 0)

gl <- dartR::gl.filter.reproducibility(gl, threshold = 0.98, verbose  = 0)

gl <- gl.filter.maf.FDD(gl, threshold = 0.01)

gl <- gl.filter.lowcount.FDD(gl,threshold = 10)

gl <- gl.filter.highcount.FDD(gl,threshold = 300)

gl <- gl.filter.loc.het.FDD(gl,threshold = 0.6)

sex.markers <- c(8076504, 4237270, 4248435, 8076068, 100006821,
100007336, 17682608, 17683020, 17683412, 17683498, 17683506,
4243144, 8076068, 8076504, 8126217, 8151800, 8153424, 8153554)
sum(gl$other$loc.metrics$CloneID %in% sex.markers)
gl <- gl[, !gl$other$loc.metrics$CloneID %in% sex.markers]
gl$other$loc.metrics <- gl$other$loc.metrics[!gl$other$loc.metrics$CloneID %in%
                                               sex.markers, ]
```

```
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))
Outliers <- OutFLANK::OutFLANK(FstDataFrame, NumberOfSamples = length(levels(gl@pop)),
                               qthreshold = 0.01, LeftTrimFraction = 0.01,
                               RightTrimFraction = 0.01,Hmin = 0.001)
Outliers$numberLowFstOutliers
Outliers$numberHighFstOutliers
OutflankNames <- Outliers$results$LocusName[Outliers$results$OutlierFlag ==
TRUE]

gl$other$loc.metrics <- gl$other$loc.metrics[!gl$loc.names %in% OutflankNames, ]
gl <- gl[, !gl$loc.names %in% OutflankNames]

gl$other$loc.metrics <- gl$other$loc.metrics[!gl$loc.names %in%
                                            c("17683723-27-C/T", "17683724-35-T/C"),]
gl <-  gl[,!gl$loc.names %in% c("17683723-27-C/T", "17683724-35-T/C")]

dim(gl)
dim(gl$other$loc.metrics)
summary(gl$pop)

ggar.SNP.gl <- gl
ggar.SNP.gi <- dartR::gl2gi(ggar.SNP.gl)
ggar.snp.gt <- strataG::genind2gtypes(ggar.SNP.gi)

schemes <- data.frame(
  id = ggar.SNP.gl$other$ind.metrics$id,
  pop = as.character(ggar.SNP.gl$other$ind.metrics$pop),
  river = as.character(ggar.SNP.gl$other$ind.metrics$River),
  stringsAsFactors = FALSE)
setSchemes(ggar.snp.gt) <- schemes
```

9.2 Save data

```
save(ggar.bin, ggar.dna, ggar.gi, ggar.gt, ggar.hap, ggar.h, ggar.SNP.gi,
     ggar.SNP.gl,ggar.snp.gt,ggar.SNP.gl.X, file = "6.ggar.Rdata")
```

9.3 Load data

```
load("6.ggar.Rdata")

print(ggar.SNP.gl)
```

```
## /// GENLIGHT OBJECT /////////
##
## // 373 genotypes, 1,721 binary SNPs, size: 6.4 Mb
## 1828 (0.28 %) missing data
##
## // Basic content
## @gen: list of 373 SNPbin
## @ploidy: ploidy of each individual (range: 2-2)
##
```

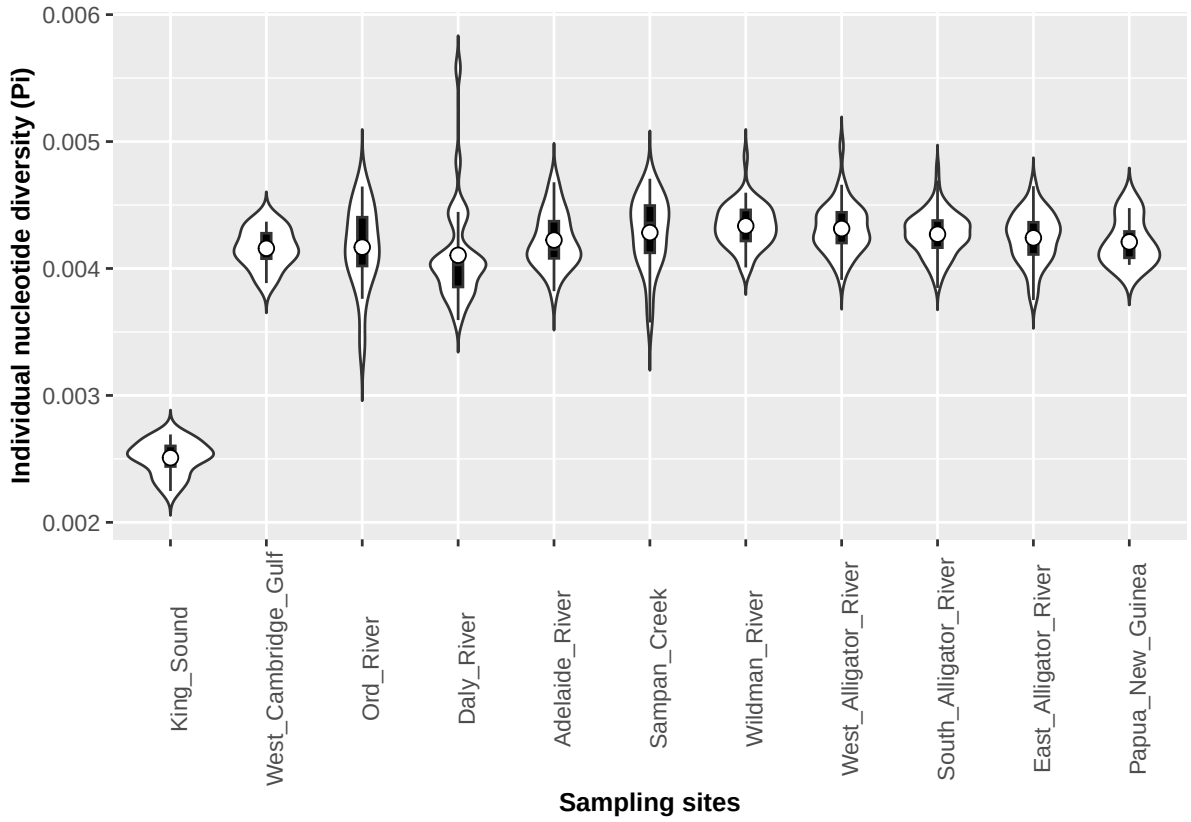
```
## // Optional content
## @ind.names: 373 individual labels
## @loc.names: 1721 locus labels
## @loc.all: 1721 alleles
## @position: integer storing positions of the SNPs
## @pop: population of each individual (group size range:
4-296)
## @other: a list containing: loc.metrics latlong
ind.metrics loc.metrics.flags verbose history
```

9.4 Genetic diversity

9.4.1 Nucleotide diversity per river

```
ggar.SNP.gl$pop <- factor(ggar.SNP.gl@other$ind.metrics$River,
                          levels = unique(ggar.SNP.gl@other$ind.metrics$River))

ggar.SNP.tidy <- radiator::tidy_genlight(ggar.SNP.gl)
ggar.SNP.tidy$POP_ID <- ggar.SNP.tidy$STRATA
pi.SNP.river <- radiator::pi(ggar.SNP.tidy, read.length = 69)
pi.SNP.river$boxplot.pi$theme$axis.text.x <- ggplot2::element_text(angle = 90)

diversity.SNP.river <- dartR.base::utils.basic.stats(ggar.SNP.gl)

save(divers.river, divers.region, pi.SNP.river, diversity.SNP.river,
     dA.gt.river,dA.gt.pop,
     file = "Genetic_diversity.Rdata")
```

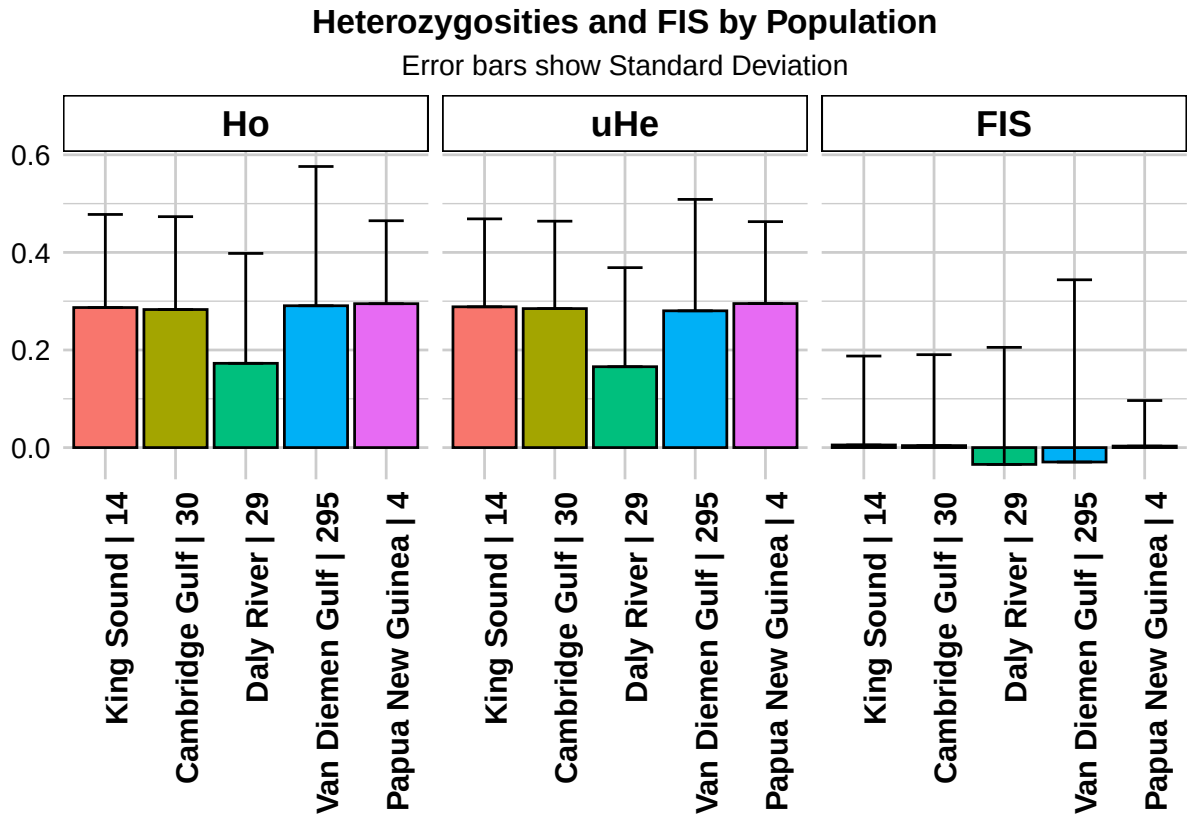
```
load("Genetic_diversity.Rdata")
print(pi.SNP.river$boxplot.pi)
```




```
hets <- dartR.base::gl.report.heterozygosity(ggar.SNP.gl, method = "pop")
```

```
## Starting ::
##  Starting dartR.base
##  Starting gl.report.heterozygosity
##   Processing genlight object with SNP data
##   Calculating Observed Heterozygosities, averaged across
##                     loci, for each population
##   Calculating Expected Heterozygosities
##
## Starting gl.colors
```

```
## Selected color type dis
## Completed: gl.colors
```




```
## pop n.Ind n.Loc n.Loc.adj polyLoc monoLoc
## King Sound King Sound 13.936665 1721 1 855 866
## Cambridge Gulf Cambridge Gulf 29.886113 1721 1 1531 190
## Daly River Daly River 28.893085 1721 1 1566 155
## Van Diemen Gulf Van Diemen Gulf 295.238815 1721 1 1710
11
## Papua New Guinea Papua New Guinea 3.983149 1721 1 1118
603
## all_NALoc Ho HoSD HoSE He HeSD
## King Sound 0 0.172692 0.225213 0.005429 0.159818
0.195731
## Cambridge Gulf 0 0.286975 0.190804 0.004599 0.283676
0.177184
## Daly River 0 0.282947 0.190315 0.004588 0.279900
0.176062
## Van Diemen Gulf 0 0.294924 0.169976 0.004097 0.294655
0.167589
## Papua New Guinea 0 0.290674 0.285384 0.006879 0.245095
0.199691
## HeSE uHe uHeSD uHeSE FIS FISSD FISSE
## King Sound 0.004718 0.165765 0.203015 0.004894 -0.034454
0.239850 NA
## Cambridge Gulf 0.004271 0.288503 0.180199 0.004344
0.005351 0.182296 NA
## Daly River 0.004244 0.284829 0.179163 0.004319 0.004056
0.186361 NA
## Van Diemen Gulf 0.004040 0.295155 0.167873 0.004047
0.003088 0.093341 NA
## Papua New Guinea 0.004814 0.280278 0.228357 0.005505
```

```
-0.029547 0.373426 NA
## Completed: ::
## Completed: dartR.base
## Completed: gl.report.heterozygosity
```

```
colMeans(diversity.SNP.river$Ho, na.rm = TRUE) #observed heterozygosity
```

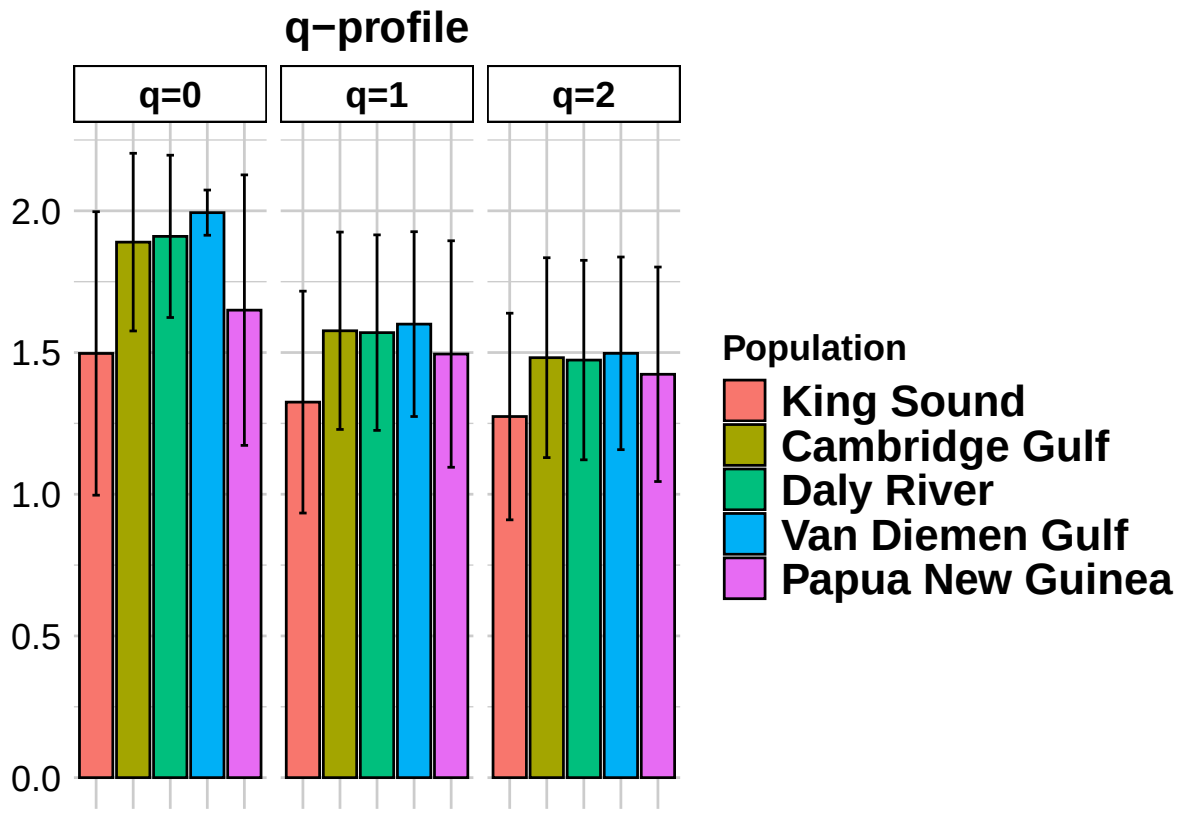
```
## King Sound Cambridge Gulf Daly River Van Diemen Gulf
## 0.1726898 0.2869719 0.2829447 0.2949234
## Papua New Guinea
## 0.2906739
```

```
colMeans(diversity.SNP.river$Fis, na.rm = TRUE) #inbreeding coefficient
```

```
## King Sound Cambridge Gulf Daly River Van Diemen Gulf
## -0.038220117 0.004862704 0.003490230 0.003077018
## Papua New Guinea
## -0.060902952
```

```
dartR.base::gl.report.diversity(ggar.SNP.gl, table = "N", verbose = 0, pbar = FALSE)
```

```
##   Processing genlight object with SNP data
```

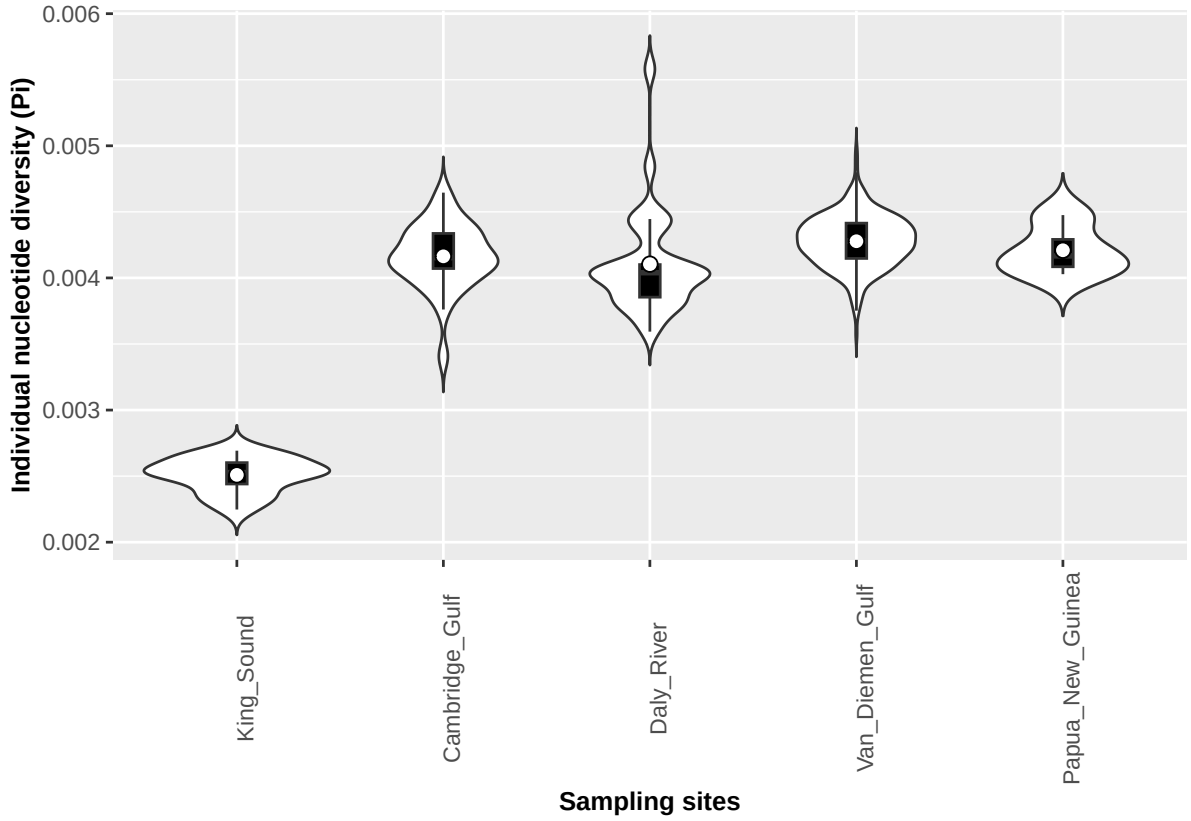


9.4.2 Nucleotide diversity per region

```
ggar.SNP.gl$pop <- ggar.SNP.gl@other$ind.metrics$pop
ggar.SNP.tidy <- radiator::tidy_genlight(ggar.SNP.gl)
ggar.SNP.tidy$POP_ID <- ggar.SNP.tidy$STRATA
pi.SNP.region <- radiator::pi(ggar.SNP.tidy, read.length = 69)
pi.SNP.region$boxplot.pi$theme$axis.text.x <- ggplot2::element_text(angle = 90)

diversity.SNP.region <- dartR.base::utils.basic.stats(ggar.SNP.gl)
```

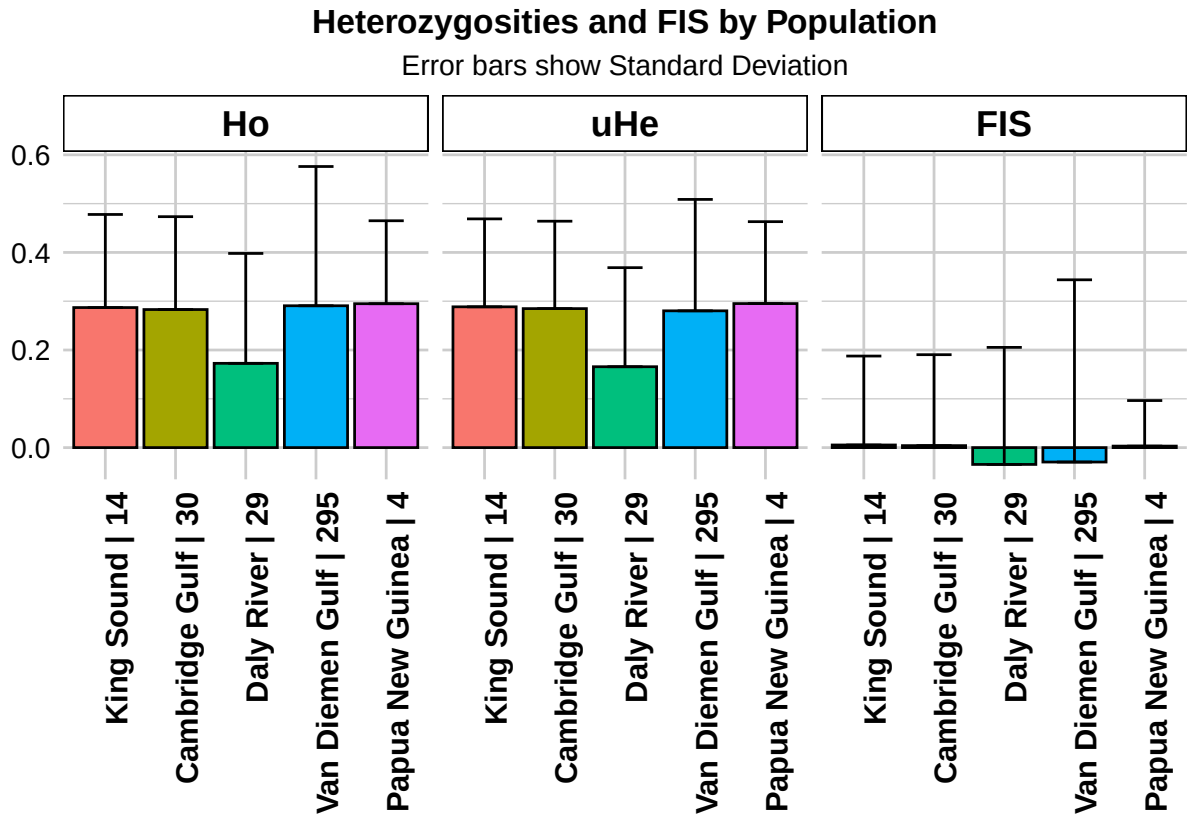
```
save(divers.river, divers.region, pi.SNP.river, diversity.SNP.river,
     pi.SNP.region,diversity.SNP.region, dA.gt.river,dA.gt.pop,
     file = "Genetic_diversity.Rdata")
```

```
load("Genetic_diversity.Rdata")
print(pi.SNP.region$boxplot.pi)
```




```
hets <- dartR.base::gl.report.heterozygosity(ggar.SNP.gl, method = "pop")
```

```
## Starting ::
##  Starting dartR.base
##  Starting gl.report.heterozygosity
##   Processing genlight object with SNP data
##   Calculating Observed Heterozygosities, averaged across
##                       loci, for each population
##   Calculating Expected Heterozygosities
##
## Starting gl.colors
## Selected color type dis
## Completed: gl.colors
```




```
## pop n.Ind n.Loc n.Loc.adj polyLoc monoLoc
## King Sound King Sound 13.936665 1721 1 855 866
## Cambridge Gulf Cambridge Gulf 29.886113 1721 1 1531 190
## Daly River Daly River 28.893085 1721 1 1566 155
## Van Diemen Gulf Van Diemen Gulf 295.238815 1721 1 1710
11
## Papua New Guinea Papua New Guinea 3.983149 1721 1 1118
603
## all_NALoc Ho HoSD HoSE He HeSD
## King Sound 0 0.172692 0.225213 0.005429 0.159818
0.195731
## Cambridge Gulf 0 0.286975 0.190804 0.004599 0.283676
0.177184
## Daly River 0 0.282947 0.190315 0.004588 0.279900
0.176062
## Van Diemen Gulf 0 0.294924 0.169976 0.004097 0.294655
0.167589
## Papua New Guinea 0 0.290674 0.285384 0.006879 0.245095
0.199691
## HeSE uHe uHeSD uHeSE FIS FISSD FISSE
## King Sound 0.004718 0.165765 0.203015 0.004894 -0.034454
0.239850 NA
## Cambridge Gulf 0.004271 0.288503 0.180199 0.004344
0.005351 0.182296 NA
## Daly River 0.004244 0.284829 0.179163 0.004319 0.004056
0.186361 NA
## Van Diemen Gulf 0.004040 0.295155 0.167873 0.004047
0.003088 0.093341 NA
## Papua New Guinea 0.004814 0.280278 0.228357 0.005505
-0.029547 0.373426 NA
## Completed: ::
```

```
## Completed: dartR.base
## Completed: gl.report.heterozygosity
```

```
colMeans(diversity.SNP.region$Ho, na.rm = TRUE) #observed heterozygosity
```

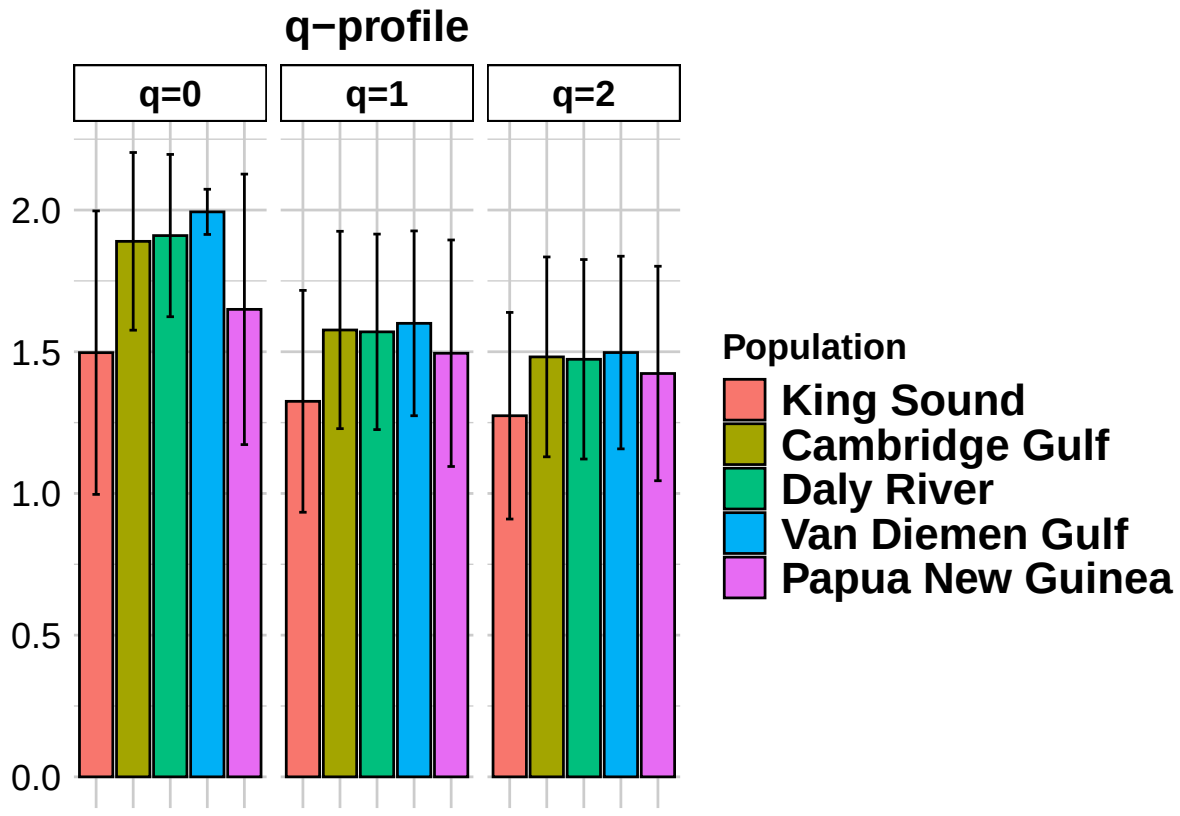
```
## King Sound Cambridge Gulf Daly River Van Diemen Gulf
## 0.1726898 0.2869719 0.2829447 0.2949234
## Papua New Guinea
## 0.2906739
```

```
colMeans(diversity.SNP.region$Fis, na.rm = TRUE) #inbreeding coefficient
```

```
## King Sound Cambridge Gulf Daly River Van Diemen Gulf
## -0.038220117 0.004862704 0.003490230 0.003077018
## Papua New Guinea
## -0.060902952
```

```
dartR.base::gl.report.diversity(ggar.SNP.gl, table = "N", verbose = 0, pbar = FALSE)
```

```
##   Processing genlight object with SNP data
```



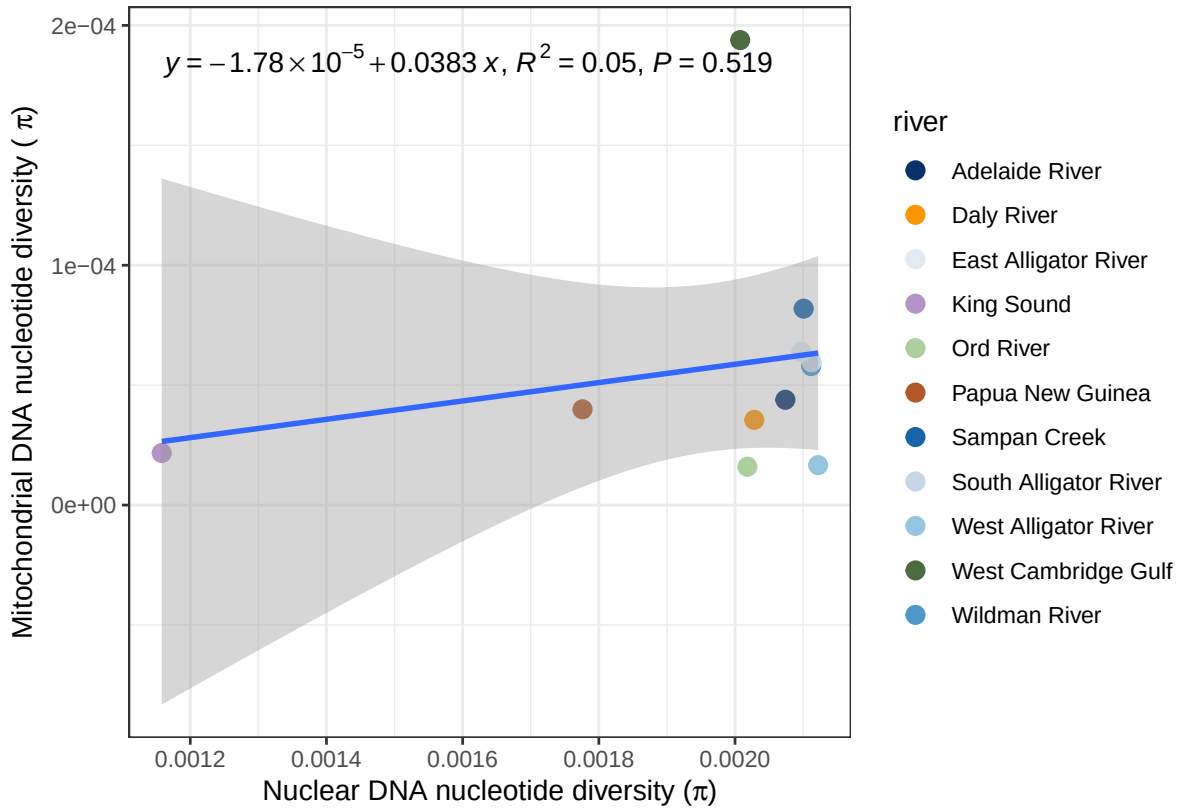
9.4.3 Compare mtDNA vs nuDNA nucleotide diversity per river

```
load("Genetic_diversity.Rdata")

mt <- data.frame(river = divers.river$River, pi_mtDNA = divers.river$pi.pegas,
                 marker = "mtDNA")
SNP <- data.frame(river = pi.SNP.river$pi.populations$POP_ID[-12],
                  pi_nuDNA = pi.SNP.river$pi.populations$PI_NEI[-12],
                  marker = "SNP")
plot.data <- data.frame(mt,SNP)

pi.plot <- ggplot2::ggplot(data = plot.data, ggplot2::aes(x = pi_nuDNA, y = pi_mtDNA)) +
```

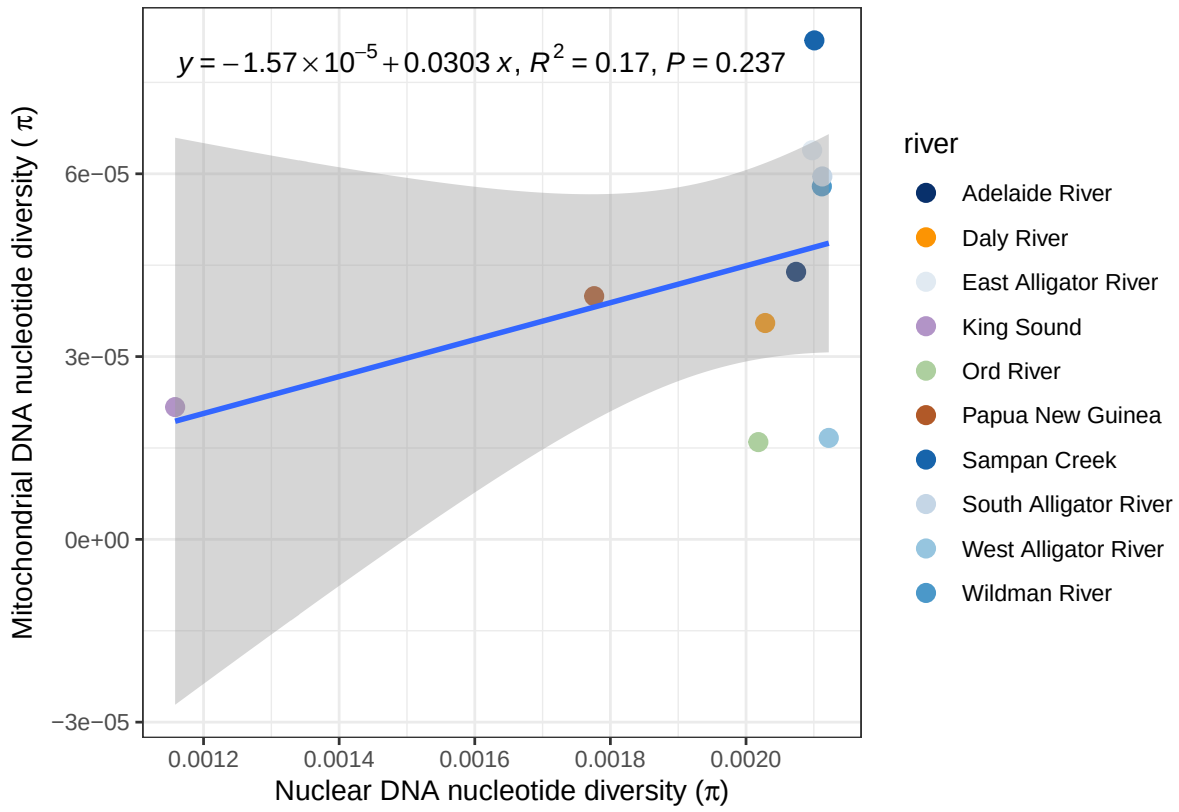
```
  ggplot2::geom_point(ggplot2::aes(colour = river),size = 3) +
  ggpmisc::stat_poly_line() +
  ggpmisc::stat_poly_eq(ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::xlab(expression(paste("Nuclear DNA nucleotide diversity (", pi, ")"))) +
  ggplot2::ylab(expression(paste("Mitochondrial DNA nucleotide diversity ( ", pi, ")"))) +
  ggplot2::scale_colour_manual(values = colours.11) +
  ggplot2::theme_bw()
```

```
## Registered S3 methods overwritten by 'ggpp':
##   method                  from
##   heightDetails.titleGrob ggplot2
##   widthDetails.titleGrob  ggplot2
```

```
print(pi.plot)
```

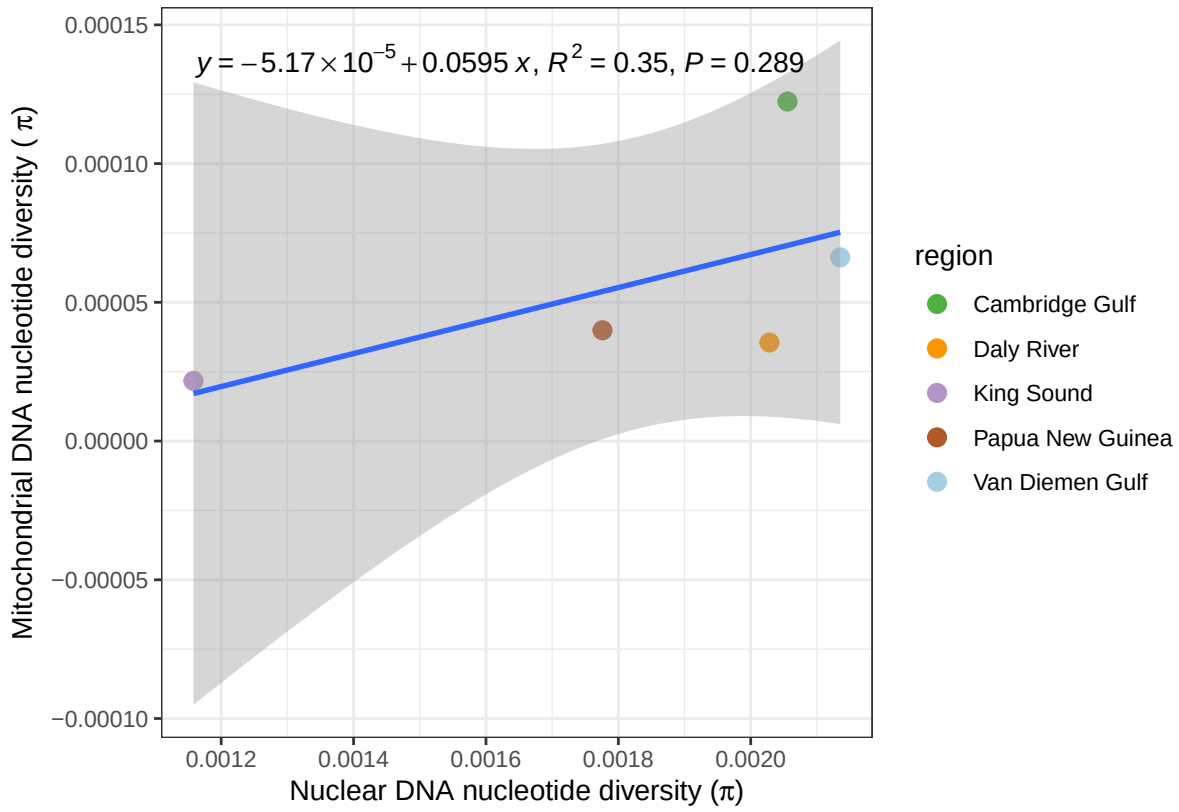



```
### WITHOUT WEST CAMBRIDGE GULF
plot.data2 <- plot.data[plot.data$river != "West Cambridge Gulf",]

pi.plot <- ggplot2::ggplot(data = plot.data2, ggplot2::aes(x = pi_nuDNA, y = pi_mtDNA)) +
  ggplot2::geom_point(ggplot2::aes(colour = river),size = 3) +
  ggpmisc::stat_poly_line() +
  ggpmisc::stat_poly_eq(ggpmisc::use_label(c("eq", "R2","p"))) +
  ggplot2::xlab(expression(paste("Nuclear DNA nucleotide diversity (", pi, ")"))) +
  ggplot2::ylab(expression(paste("Mitochondrial DNA nucleotide diversity ( ", pi, ")"))) +
  ggplot2::scale_colour_manual(values = colours.11) +
  ggplot2::theme_bw()
print(pi.plot)
```



9.4.4 Compare mtDNA vs nuDNA nucleotide diversity per region

```
load("Genetic_diversity.Rdata")

mt <- data.frame(region = divers.region$Region, pi_mtDNA = divers.region$pi.pegas,
                 marker = "mtDNA")
SNP <- data.frame(region = pi.SNP.region$pi.populations$POP_ID[-6],
                  pi_nuDNA = pi.SNP.region$pi.populations$PI_NEI[-6],
                  marker = "SNP")
plot.data <- data.frame(mt,SNP)

pi.plot <- ggplot2::ggplot(data = plot.data, ggplot2::aes(x = pi_nuDNA,
                                                          y = pi_mtDNA)) +
  ggplot2::geom_point(ggplot2::aes(colour = region),size = 3) +
  ggpmisc::stat_poly_line() +
  ggpmisc::stat_poly_eq(ggpmisc::use_label(c("eq", "R2","p"))) +
  ggplot2::xlab(expression(paste("Nuclear DNA nucleotide diversity (", pi, ")"))) +
  ggplot2::ylab(expression(paste("Mitochondrial DNA nucleotide diversity ( ", pi, ")"))) +
  ggplot2::scale_colour_manual(values = colours.5) +
  ggplot2::theme_bw()
print(pi.plot)
```




```
### WITHOUT CAMBRIDGE GULF
plot.data2 <- plot.data[plot.data$region != "Cambridge Gulf",]


pi.plot <- ggplot2::ggplot(data = plot.data2, ggplot2::aes(x = pi_nuDNA,
                                                           y = pi_mtDNA)) +
  ggplot2::geom_point(ggplot2::aes(colour = region),size = 3) +
  ggpmisc::stat_poly_line() +
  ggpmisc::stat_poly_eq(ggpmisc::use_label(c("eq", "R2","p"))) +
  ggplot2::xlab(expression(paste("Nuclear DNA nucleotide diversity (", pi, ")"))) +
  ggplot2::ylab(expression(paste("Mitochondrial DNA nucleotide diversity ( ", pi, ")"))) +
  ggplot2::scale_colour_manual(values = colours.5) +
  ggplot2::theme_bw()
print(pi.plot)
```

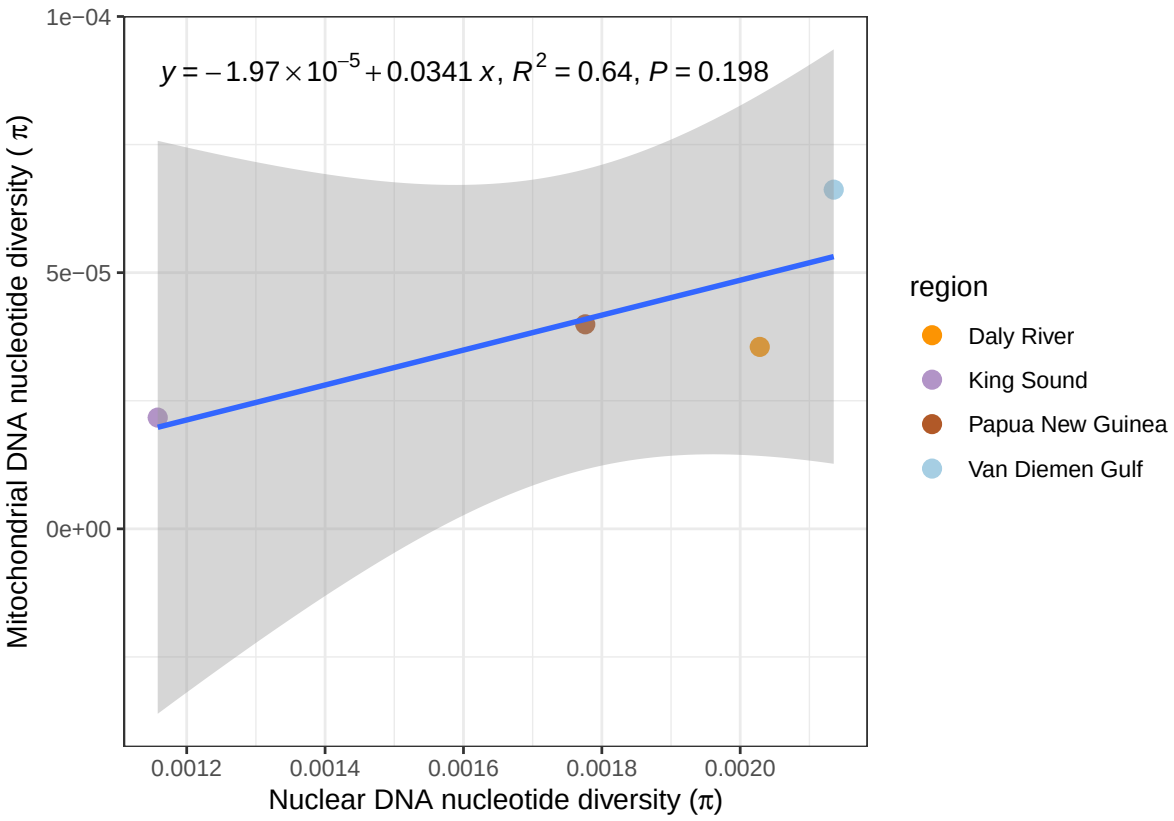

$y = -1.97 \times 10^{-5} + 0.0341\,x$, $R^2 = 0.64$, $P = 0.198$
Mitochondrial DNA nucleotide diversity (π)
1e−04
5e−05
0e+00
0.0012
0.0014
0.0016
0.0018
0.0020
Nuclear DNA nucleotide diversity (π)
region
Daly River
King Sound
Papua New Guinea
Van Diemen Gulf

Table 35: Overall differentiation between rivers for nuDNA

	estimate	p.val
Chi2	94044.2196	1e-04
D	0.0336	1e-04
Fis	-0.0061	1e+00
Fst	0.0693	1e-04
F'st	0.0978	1e-04
Gst	0.1200	1e-04
G'st	0.1712	1e-04
G"st	0.1810	1e-04

9.5 Genetic differentiation

9.5.1 Fst per river

```
strata <- as.character(ggar.snp.gt@schemes$river)
names(strata) <- as.character(ggar.snp.gt@schemes$id)
strataG::setStrata(ggar.snp.gt) <- strata

popstr.snp.river <-
  strataG::popStructTest(
    ggar.snp.gt,
    nrep = 10000,
    stats = "all",
    type = "both",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores()-1,
    write.output = TRUE
  )
```

```
knitr::kable(
  popstr.snp.river$overall$result,
  digits = 4,
  format = "latex",
  caption = "Overall differentiation between rivers for nuDNA"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
knitr::kable(
  popstr.snp.river$pairwise$result[order(popstr.snp.river$pairwise$result$Fst,
                                         decreasing = TRUE), -c(1:10)],
  format = "latex",
  digits = 4,
  caption = "Pairwise differentiation between rivers for nuDNA"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")
```

9.5.2 Fst per region

```
strata <- as.character(ggar.snp.gt@schemes$pop)
names(strata) <- as.character(ggar.snp.gt@schemes$id)
strataG::setStrata(ggar.snp.gt) <- strata

popstr.snp.pop <-
```

Table 36: Pairwise differentiation between rivers for nuDNA

	Fst	Fst.p.val	F'st	F'st.p.val	Gst	Gst.p.val	G'st	G'st.p.val	G"st	G"st.p.val
King Sound (14) v. Papua New Guinea (4)	0.3945	0.0003	0.4906	0.0003	0.2627	0.0003	0.3929	0.0003	0.5193	0.0003
King Sound (14) v. Ord River (15)	0.3023	0.0001	0.3933	0.0001	0.1888	0.0001	0.2971	0.0001	0.4087	0.0001
King Sound (14) v. West Cambridge Gulf (15)	0.2969	0.0001	0.3857	0.0001	0.1849	0.0001	0.2904	0.0001	0.4011	0.0001
Daly River (29) v. King Sound (14)	0.2898	0.0001	0.3831	0.0001	0.1868	0.0001	0.2947	0.0001	0.4057	0.0001
Adelaide River (24) v. King Sound (14)	0.2868	0.0001	0.3795	0.0001	0.1821	0.0001	0.2895	0.0001	0.3990	0.0001
King Sound (14) v. Sampan Creek (30)	0.2740	0.0001	0.3657	0.0001	0.1765	0.0001	0.2816	0.0001	0.3894	0.0001
King Sound (14) v. West Alligator River (41)	0.2712	0.0001	0.3661	0.0001	0.1782	0.0001	0.2851	0.0001	0.3932	0.0001
King Sound (14) v. Wildman River (45)	0.2696	0.0001	0.3637	0.0001	0.1775	0.0001	0.2837	0.0001	0.3917	0.0001
East Alligator River (61) v. King Sound (14)	0.2680	0.0001	0.3623	0.0001	0.1792	0.0001	0.2857	0.0001	0.3942	0.0001
King Sound (14) v. South Alligator River (95)	0.2593	0.0001	0.3540	0.0001	0.1762	0.0001	0.2815	0.0001	0.3891	0.0001
Daly River (29) v. Papua New Guinea (4)	0.1796	0.0001	0.2510	0.0001	0.1100	0.0001	0.1941	0.0001	0.2740	0.0001
Adelaide River (24) v. Papua New Guinea (4)	0.1737	0.0002	0.2448	0.0002	0.1075	0.0002	0.1914	0.0002	0.2699	0.0002
Papua New Guinea (4) v. West Alligator River (41)	0.1730	0.0001	0.2463	0.0001	0.1036	0.0001	0.1863	0.0001	0.2627	0.0001
Papua New Guinea (4) v. South Alligator River (95)	0.1723	0.0001	0.2440	0.0001	0.0990	0.0001	0.1781	0.0001	0.2522	0.0001
East Alligator River (61) v. Papua New Guinea (4)	0.1717	0.0001	0.2421	0.0001	0.1005	0.0001	0.1801	0.0001	0.2550	0.0001
Papua New Guinea (4) v. Wildman River (45)	0.1705	0.0001	0.2418	0.0001	0.1009	0.0001	0.1812	0.0001	0.2562	0.0001
Papua New Guinea (4) v. Sampan Creek (30)	0.1696	0.0001	0.2404	0.0001	0.1038	0.0001	0.1857	0.0001	0.2623	0.0001
Papua New Guinea (4) v. West Cambridge Gulf (15)	0.1556	0.0003	0.2185	0.0003	0.1052	0.0003	0.1838	0.0003	0.2615	0.0003
Ord River (15) v. Papua New Guinea (4)	0.1536	0.0004	0.2155	0.0004	0.1054	0.0004	0.1842	0.0004	0.2620	0.0004
Ord River (15) v. South Alligator River (95)	0.1291	0.0001	0.1829	0.0001	0.0696	0.0001	0.1267	0.0001	0.1835	0.0001
Ord River (15) v. Sampan Creek (30)	0.1283	0.0001	0.1818	0.0001	0.0704	0.0001	0.1281	0.0001	0.1855	0.0001
East Alligator River (61) v. Ord River (15)	0.1278	0.0001	0.1805	0.0001	0.0691	0.0001	0.1254	0.0001	0.1819	0.0001
Adelaide River (24) v. Ord River (15)	0.1277	0.0001	0.1806	0.0001	0.0703	0.0001	0.1275	0.0001	0.1848	0.0001
Ord River (15) v. Wildman River (45)	0.1264	0.0001	0.1792	0.0001	0.0686	0.0001	0.1249	0.0001	0.1811	0.0001
Ord River (15) v. West Alligator River (41)	0.1259	0.0001	0.1792	0.0001	0.0686	0.0001	0.1252	0.0001	0.1813	0.0001
East Alligator River (61) v. West Cambridge Gulf (15)	0.1225	0.0001	0.1729	0.0001	0.0662	0.0001	0.1199	0.0001	0.1746	0.0001
South Alligator River (95) v. West Cambridge Gulf (15)	0.1224	0.0001	0.1731	0.0001	0.0660	0.0001	0.1198	0.0001	0.1743	0.0001
Sampan Creek (30) v. West Cambridge Gulf (15)	0.1219	0.0001	0.1724	0.0001	0.0671	0.0001	0.1218	0.0001	0.1770	0.0001
Adelaide River (24) v. West Cambridge Gulf (15)	0.1217	0.0001	0.1718	0.0001	0.0666	0.0001	0.1207	0.0001	0.1756	0.0001
West Alligator River (41) v. West Cambridge Gulf (15)	0.1214	0.0001	0.1721	0.0001	0.0662	0.0001	0.1206	0.0001	0.1753	0.0001
West Cambridge Gulf (15) v. Wildman River (45)	0.1199	0.0001	0.1699	0.0001	0.0649	0.0001	0.1180	0.0001	0.1718	0.0001
Daly River (29) v. Ord River (15)	0.0964	0.0001	0.1354	0.0001	0.0532	0.0001	0.0956	0.0001	0.1413	0.0001
Daly River (29) v. West Cambridge Gulf (15)	0.0930	0.0001	0.1305	0.0001	0.0514	0.0001	0.0923	0.0001	0.1367	0.0001
Adelaide River (24) v. Daly River (29)	0.0912	0.0001	0.1286	0.0001	0.0488	0.0001	0.0883	0.0001	0.1308	0.0001
Daly River (29) v. East Alligator River (61)	0.0895	0.0001	0.1264	0.0001	0.0475	0.0001	0.0859	0.0001	0.1273	0.0001
Daly River (29) v. West Alligator River (41)	0.0888	0.0001	0.1258	0.0001	0.0473	0.0001	0.0860	0.0001	0.1273	0.0001
Daly River (29) v. South Alligator River (95)	0.0888	0.0001	0.1255	0.0001	0.0470	0.0001	0.0852	0.0001	0.1262	0.0001
Daly River (29) v. Sampan Creek (30)	0.0885	0.0001	0.1249	0.0001	0.0474	0.0001	0.0860	0.0001	0.1273	0.0001
Daly River (29) v. Wildman River (45)	0.0878	0.0001	0.1240	0.0001	0.0466	0.0001	0.0846	0.0001	0.1254	0.0001
Adelaide River (24) v. Wildman River (45)	0.0154	0.0001	0.0219	0.0001	0.0086	0.0001	0.0157	0.0001	0.0241	0.0001
Adelaide River (24) v. West Alligator River (41)	0.0151	0.0001	0.0215	0.0001	0.0086	0.0001	0.0158	0.0001	0.0242	0.0001
Adelaide River (24) v. South Alligator River (95)	0.0146	0.0001	0.0207	0.0001	0.0077	0.0001	0.0141	0.0001	0.0216	0.0001
Adelaide River (24) v. East Alligator River (61)	0.0138	0.0001	0.0196	0.0001	0.0076	0.0001	0.0139	0.0001	0.0214	0.0001
Adelaide River (24) v. Sampan Creek (30)	0.0126	0.0001	0.0180	0.0001	0.0079	0.0001	0.0144	0.0001	0.0221	0.0001
South Alligator River (95) v. Wildman River (45)	0.0076	0.0001	0.0108	0.0001	0.0040	0.0001	0.0074	0.0001	0.0113	0.0001
Ord River (15) v. West Cambridge Gulf (15)	0.0075	0.0074	0.0105	0.0074	0.0108	0.0075	0.0193	0.0074	0.0297	0.0074
West Alligator River (41) v. Wildman River (45)	0.0074	0.0001	0.0105	0.0001	0.0042	0.0001	0.0077	0.0001	0.0118	0.0001
East Alligator River (61) v. Wildman River (45)	0.0061	0.0001	0.0087	0.0001	0.0034	0.0001	0.0063	0.0001	0.0097	0.0001
Sampan Creek (30) v. Wildman River (45)	0.0059	0.0002	0.0084	0.0002	0.0037	0.0002	0.0067	0.0002	0.0103	0.0002
South Alligator River (95) v. West Alligator River (41)	0.0050	0.0001	0.0071	0.0001	0.0027	0.0001	0.0050	0.0001	0.0077	0.0001
East Alligator River (61) v. West Alligator River (41)	0.0041	0.0001	0.0059	0.0001	0.0024	0.0001	0.0045	0.0001	0.0069	0.0001
Sampan Creek (30) v. West Alligator River (41)	0.0037	0.0001	0.0053	0.0001	0.0026	0.0001	0.0048	0.0001	0.0074	0.0001
Sampan Creek (30) v. South Alligator River (95)	0.0024	0.0030	0.0034	0.0030	0.0016	0.0034	0.0028	0.0034	0.0044	0.0034
East Alligator River (61) v. South Alligator River (95)	0.0023	0.0001	0.0033	0.0001	0.0013	0.0001	0.0024	0.0001	0.0038	0.0001
East Alligator River (61) v. Sampan Creek (30)	0.0012	0.0497	0.0017	0.0497	0.0012	0.0532	0.0021	0.0532	0.0033	0.0532

Table 37: Overall differentiation between regions for nuDNA

	estimate	p.val
Chi2	79340.2793	1e-04
D	0.0500	1e-04
Fis	-0.0102	1e+00
Fst	0.1426	1e-04
F'st	0.2007	1e-04
Gst	0.1729	1e-04
G'st	0.2500	1e-04
G"st	0.2811	1e-04

Table 38: Pairwise differentiation between regions for nuDNA

	Fst	Fst.p.val	F'st	F'st.p.val	Gst	Gst.p.val	G'st	G'st.p.val	G"st	G"st.p.val
King Sound (14) v. Papua New Guinea (4)	0.3945	6e-04	0.4906	6e-04	0.2627	6e-04	0.3929	6e-04	0.5193	6e-04
Daly River (29) v. King Sound (14)	0.2898	1e-04	0.3831	1e-04	0.1868	1e-04	0.2947	1e-04	0.4057	1e-04
Cambridge Gulf (30) v. King Sound (14)	0.2778	1e-04	0.3686	1e-04	0.1799	1e-04	0.2847	1e-04	0.3937	1e-04
King Sound (14) v. Van Diemen Gulf (296)	0.2496	1e-04	0.3451	1e-04	0.1740	1e-04	0.2788	1e-04	0.3857	1e-04
Daly River (29) v. Papua New Guinea (4)	0.1796	1e-04	0.2510	1e-04	0.1100	1e-04	0.1941	1e-04	0.2740	1e-04
Papua New Guinea (4) v. Van Diemen Gulf (296)	0.1684	1e-04	0.2392	1e-04	0.0956	1e-04	0.1727	1e-04	0.2449	1e-04
Cambridge Gulf (30) v. Papua New Guinea (4)	0.1517	1e-04	0.2130	1e-04	0.0965	1e-04	0.1705	1e-04	0.2435	1e-04
Cambridge Gulf (30) v. Van Diemen Gulf (296)	0.1206	1e-04	0.1714	1e-04	0.0647	1e-04	0.1181	1e-04	0.1717	1e-04
Cambridge Gulf (30) v. Daly River (29)	0.0928	1e-04	0.1305	1e-04	0.0499	1e-04	0.0899	1e-04	0.1332	1e-04
Daly River (29) v. Van Diemen Gulf (296)	0.0856	1e-04	0.1216	1e-04	0.0455	1e-04	0.0826	1e-04	0.1225	1e-04

```
  strataG::popStructTest(
    ggar.snp.gt,
    nrep = 10000,
    stats = "all",
    type = "both",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores()-1,
    write.output = TRUE
  )
```

```
knitr::kable(
  popstr.snp.pop$overall$result,
  digits = 4,
  format = "latex",
  caption = "Overall differentiation between regions for nuDNA"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
knitr::kable(
  popstr.snp.pop$pairwise$result[order(popstr.snp.pop$pairwise$result$Fst,
                                       decreasing = TRUE), -c(1:10)],
  format = "latex",
  digits = 4,
  caption = "Pairwise differentiation between regions for nuDNA"
  ) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")
```

```
m1 <- popstr.mt.river$pairwise$result[order(
  match(popstr.mt.river$pairwise$result$strata.1,pop.levels1)),
        c(1,2,9)]
Gdf <- igraph::graph.data.frame(m1,directed=FALSE)
```

```
m1.1 <- igraph::as_adjacency_matrix(Gdf,names=TRUE,sparse=FALSE,attr="PHIst",
                                    type='lower')

m1 <- popstr.mt.river$pairwise$result[order(
  match(popstr.mt.river$pairwise$result$strata.1,pop.levels1)),c(1,2,10)]
Gdf <- igraph::graph.data.frame(m1, directed = FALSE)
m1.2 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "PHIst.p.val", type = 'lower')
m1$PHIst.p.adj <- p.adjust(m1$PHIst.p.val, "bonferroni")
Gdf <- igraph::graph.data.frame(m1, directed = FALSE)
m1.3 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "PHIst.p.adj", type = 'lower')

m2 <- popstr.snp.river$pairwise$result[order(
  match(popstr.snp.river$pairwise$result$strata.1,pop.levels1)),c(1,2,11)]
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.1 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst", type = 'lower')

m2 <- popstr.snp.river$pairwise$result[order(
  match(popstr.snp.river$pairwise$result$strata.1,pop.levels1)),c(1,2,12)]
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.2 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst.p.val", type = 'lower')
m2$Fst.p.adj <- p.adjust(m2$Fst.p.val, "bonferroni")
Gdf <- igraph::graph.data.frame(m2, directed = FALSE)
m2.3 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst.p.adj", type = 'lower')

m3 <- popstr.mt.pop$pairwise$result[order(
  match(popstr.mt.pop$pairwise$result$strata.1,pop.levels2)),c(1,2,9)]
Gdf <- igraph::graph.data.frame(m3, directed = FALSE)
m3.1 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "PHIst", type = 'lower')

m3 <- popstr.mt.pop$pairwise$result[order(
  match(popstr.mt.pop$pairwise$result$strata.1,pop.levels2)),c(1,2,10)]
Gdf <- igraph::graph.data.frame(m3, directed = FALSE)
m3.2 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "PHIst.p.val", type = 'lower')
m3$PHIst.p.adj <- p.adjust(m3$PHIst.p.val, "bonferroni")
Gdf <- igraph::graph.data.frame(m3, directed = FALSE)
m3.3 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "PHIst.p.adj", type = 'lower')

m4 <- popstr.snp.pop$pairwise$result[order(
  match(popstr.snp.pop$pairwise$result$strata.1,pop.levels2)),c(1,2,11)]
Gdf <- igraph::graph.data.frame(m4, directed = FALSE)
m4.1 <- igraph::as_adjacency_matrix(Gdf,names = TRUE, sparse = FALSE,
                                    attr = "Fst", type = 'lower')

m4 <- popstr.snp.pop$pairwise$result[order(
  match(popstr.snp.pop$pairwise$result$strata.1,pop.levels2)),c(1,2,12)]
Gdf <- igraph::graph.data.frame(m4, directed = FALSE)
m4.2 <- igraph::as_adjacency_matrix(Gdf,names = TRUE, sparse = FALSE,
                                    attr = "Fst.p.val", type = 'lower')
m4$Fst.p.adj <- p.adjust(m4$Fst.p.val, "bonferroni")
```

```
Gdf <- igraph::graph.data.frame(m4, directed = FALSE)
m4.3 <- igraph::as_adjacency_matrix(Gdf, names = TRUE, sparse = FALSE,
                                    attr = "Fst.p.adj", type = 'lower')

save(popstr.mt.river, popstr.mt.pop, popstr.snp.river,popstr.snp.pop,
     file = "6.Ggar_Fst_objects.Rdata")
```

9.5.2.1 Re-order data and add Bonferroni corrections

9.5.3 Compare PHIst/Fst per river

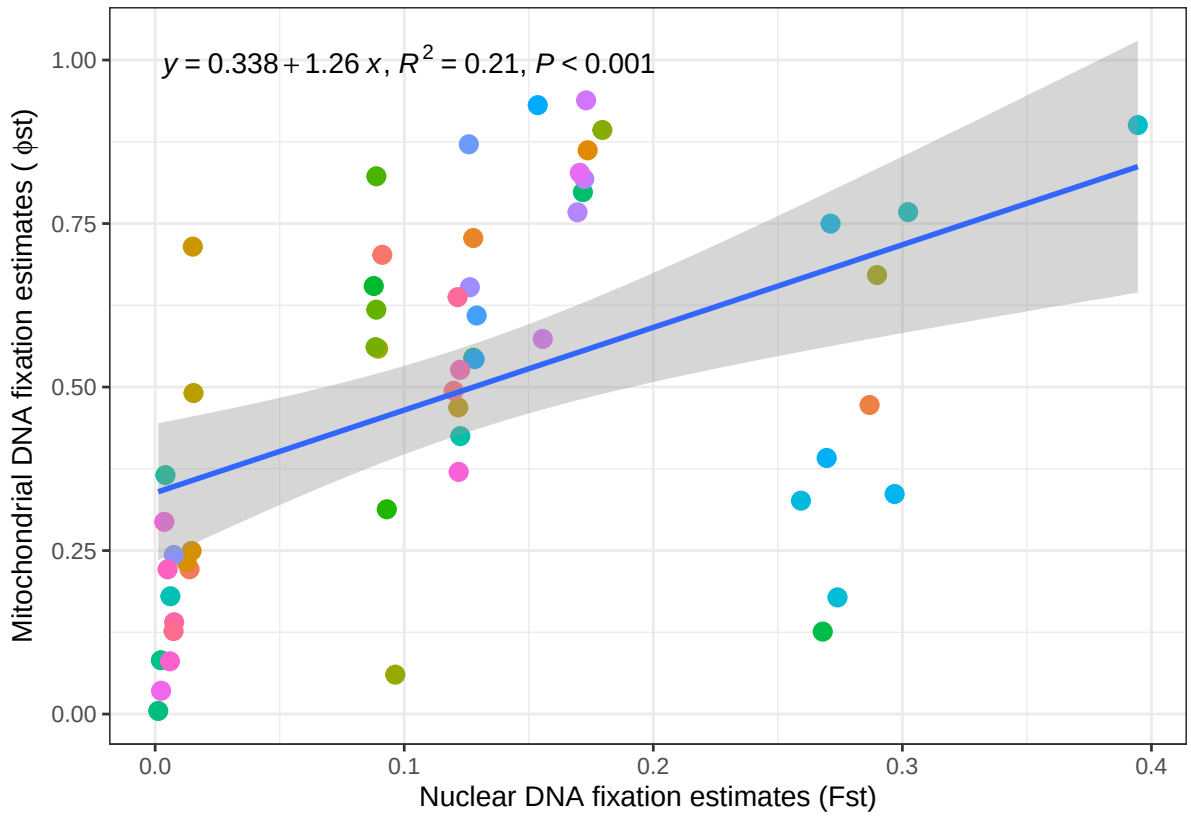
```
load("6.Ggar_Fst_objects.Rdata")
```

```
diff.river <- data.frame(Compared = rownames(popstr.mt.river$pairwise$result),
                         mtDNA.PHIst = popstr.mt.river$pairwise$result$PHIst,
                         mtDNA.PHIst.pval = popstr.mt.river$pairwise$result$PHIst.p.val,
                         nuDNA.Fst = popstr.snp.river$pairwise$result$Fst,
                         nuDNA.Fst.pval = popstr.snp.river$pairwise$result$Fst.p.val)
knitr::kable(diff.river,
             digits = 4,format = "latex",
             caption = "PHIst and Fst comparison between rivers") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")
```

```
fix.plot <- ggplot2::ggplot(data = diff.river, ggplot2::aes(x = nuDNA.Fst,
                                                            y = mtDNA.PHIst)) +
  ggplot2::geom_point(ggplot2::aes(colour = Compared),size = 3) +
  ggpmisc::stat_poly_line() +
  ggpmisc::stat_poly_eq(ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::xlab("Nuclear DNA fixation estimates (Fst)") +
  ggplot2::ylab(expression(paste("Mitochondrial DNA fixation estimates ( ", phi, "st)"))) +
  ggplot2::theme_bw() +
  ggplot2::theme(legend.position = "none", legend.title = ggplot2::element_blank())
print(fix.plot)
```

Table 39: PHIst and Fst comparison between rivers

Compared	mtDNA.PHIst	mtDNA.PHIst.pval	nuDNA.Fst	nuDNA.Fst.pval
Adelaide River (25) v. Daly River (30)	0.7022	0.0001	0.0912	0.0001
Adelaide River (25) v. East Alligator River (61)	0.2215	0.0001	0.0138	0.0001
Adelaide River (25) v. King Sound (14)	0.4726	0.0001	0.2868	0.0001
Adelaide River (25) v. Ord River (15)	0.7279	0.0001	0.1277	0.0001
Adelaide River (25) v. Papua New Guinea (6)	0.8620	0.0001	0.1737	0.0002
Adelaide River (25) v. Sampan Creek (30)	0.2325	0.0002	0.0126	0.0001
Adelaide River (25) v. South Alligator River (101)	0.2494	0.0003	0.0146	0.0001
Adelaide River (25) v. West Alligator River (41)	0.7147	0.0001	0.0151	0.0001
Adelaide River (25) v. West Cambridge Gulf (15)	0.4688	0.0001	0.1217	0.0001
Adelaide River (25) v. Wildman River (45)	0.4909	0.0001	0.0154	0.0001
Daly River (30) v. East Alligator River (61)	0.5587	0.0001	0.0895	0.0001
Daly River (30) v. King Sound (14)	0.6712	0.0001	0.2898	0.0001
Daly River (30) v. Ord River (15)	0.0603	0.0394	0.0964	0.0001
Daly River (30) v. Papua New Guinea (6)	0.8931	0.0001	0.1796	0.0001
Daly River (30) v. Sampan Creek (30)	0.5610	0.0001	0.0885	0.0001
Daly River (30) v. South Alligator River (101)	0.6183	0.0001	0.0888	0.0001
Daly River (30) v. West Alligator River (41)	0.8223	0.0001	0.0888	0.0001
Daly River (30) v. West Cambridge Gulf (15)	0.3131	0.0001	0.0930	0.0001
Daly River (30) v. Wildman River (45)	0.6545	0.0001	0.0878	0.0001
East Alligator River (61) v. King Sound (14)	0.1260	0.0042	0.2680	0.0001
East Alligator River (61) v. Ord River (15)	0.5448	0.0001	0.1278	0.0001
East Alligator River (61) v. Papua New Guinea (6)	0.7980	0.0001	0.1717	0.0001
East Alligator River (61) v. Sampan Creek (30)	0.0047	0.2881	0.0012	0.0497
East Alligator River (61) v. South Alligator River (101)	0.0823	0.0005	0.0023	0.0001
East Alligator River (61) v. West Alligator River (41)	0.3653	0.0001	0.0041	0.0001
East Alligator River (61) v. West Cambridge Gulf (15)	0.4250	0.0001	0.1225	0.0001
East Alligator River (61) v. Wildman River (45)	0.1801	0.0001	0.0061	0.0001
King Sound (14) v. Ord River (15)	0.7676	0.0001	0.3023	0.0001
King Sound (14) v. Papua New Guinea (6)	0.9006	0.0001	0.3945	0.0003
King Sound (14) v. Sampan Creek (30)	0.1784	0.0007	0.2740	0.0001
King Sound (14) v. South Alligator River (101)	0.3263	0.0002	0.2593	0.0001
King Sound (14) v. West Alligator River (41)	0.7499	0.0001	0.2712	0.0001
King Sound (14) v. West Cambridge Gulf (15)	0.3365	0.0001	0.2969	0.0001
King Sound (14) v. Wildman River (45)	0.3916	0.0001	0.2696	0.0001
Ord River (15) v. Papua New Guinea (6)	0.9311	0.0001	0.1536	0.0004
Ord River (15) v. Sampan Creek (30)	0.5426	0.0001	0.1283	0.0001
Ord River (15) v. South Alligator River (101)	0.6094	0.0001	0.1291	0.0001
Ord River (15) v. West Alligator River (41)	0.8712	0.0001	0.1259	0.0001
Ord River (15) v. West Cambridge Gulf (15)	0.2432	0.0271	0.0075	0.0074
Ord River (15) v. Wildman River (45)	0.6527	0.0001	0.1264	0.0001
Papua New Guinea (6) v. Sampan Creek (30)	0.7672	0.0001	0.1696	0.0001
Papua New Guinea (6) v. South Alligator River (101)	0.8182	0.0001	0.1723	0.0001
Papua New Guinea (6) v. West Alligator River (41)	0.9384	0.0001	0.1730	0.0001
Papua New Guinea (6) v. West Cambridge Gulf (15)	0.5736	0.0001	0.1556	0.0003
Papua New Guinea (6) v. Wildman River (45)	0.8275	0.0001	0.1705	0.0001
Sampan Creek (30) v. South Alligator River (101)	0.0357	0.0754	0.0024	0.0030
Sampan Creek (30) v. West Alligator River (41)	0.2939	0.0001	0.0037	0.0001
Sampan Creek (30) v. West Cambridge Gulf (15)	0.3703	0.0001	0.1219	0.0001
Sampan Creek (30) v. Wildman River (45)	0.0807	0.0130	0.0059	0.0002
South Alligator River (101) v. West Alligator River (41)	0.2213	0.0001	0.0050	0.0001
South Alligator River (101) v. West Cambridge Gulf (15)	0.5264	0.0001	0.1224	0.0001
South Alligator River (101) v. Wildman River (45)	0.1404	0.0004	0.0076	0.0001
West Alligator River (41) v. West Cambridge Gulf (15)	0.6375	0.0001	0.1214	0.0001
West Alligator River (41) v. Wildman River (45)	0.1269	0.0016	0.0074	0.0001
West Cambridge Gulf (15) v. Wildman River (45)	0.4943	0.0001	0.1199	0.0001




```
coord <- readr::read_csv("River_coord.csv")
```

```
## Rows: 11 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): River
## dbl (2): lat, lon
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

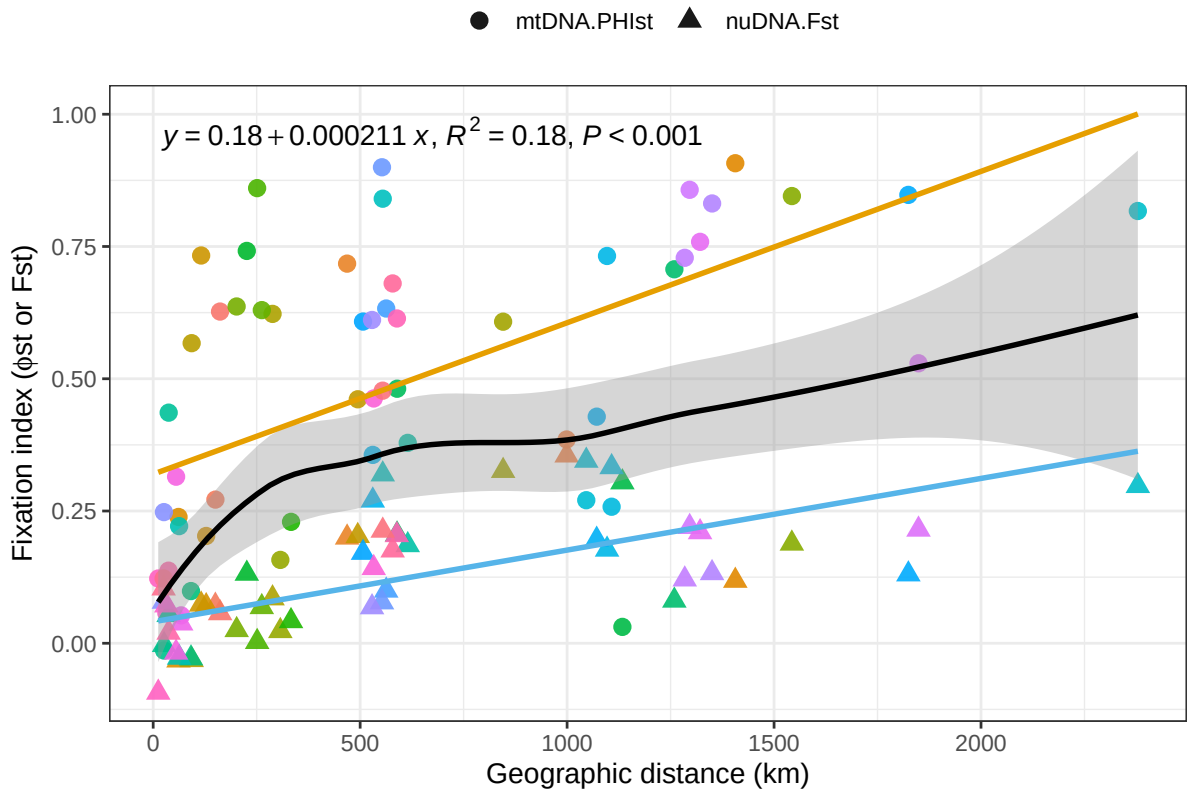
```
geo.dist <- geodist::geodist(coord[,2:3])/1000
```

```
## Maximum distance is > 100km. The 'cheap' measure is inaccurate over such
## large distances, you'd likely be better using a different 'measure',
## one of 'haversine', 'vincenty', or 'geodesic'.
```

```
geo.dist <- as.matrix(geo.dist)
dimnames(geo.dist) <- list(coord$River, coord$River)
xy <- t(combn(colnames(geo.dist), 2))
distance <- data.frame(xy, dist = geo.dist[xy])

mtch.names1 <- paste0(pmin(popstr.mt.river$pairwise$result$strata.1,
                           popstr.mt.river$pairwise$result$strata.2),
                      "_",
                      pmax(popstr.mt.river$pairwise$result$strata.1,
                           popstr.mt.river$pairwise$result$strata.2))
mtch.names2 <- paste0(pmin(distance$X1,distance$X2), "_", pmax(distance$X1,distance$X2))

distance <- distance[order(match(mtch.names2, mtch.names1)),]
diff.river2 <- cbind(diff.river, distance)

diff.river3 <- tidyr::gather(diff.river2, "Marker", "fixation_index",
```

```
                                mtDNA.PHIst, nuDNA.Fst)

plot <- ggplot2::ggplot(diff.river3,ggplot2::aes(x = dist, y = fixation_index)) +
  ggplot2::geom_jitter(ggplot2::aes(x = dist, y = fixation_index,
                                    colour = Compared , shape = Marker),
                       width = 0.1, height = 0.1, size = 3, alpha = 0.9) +
  ggplot2::geom_smooth(ggplot2::aes(x = dist, y = fixation_index),
                       method = 'loess', show.legend = FALSE, colour = "black") +
  ggpmisc::stat_poly_eq(ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::geom_smooth(data = subset(diff.river3, subset = Marker == "mtDNA.PHIst"),
                       ggplot2::aes(x = dist, y = fixation_index),se = FALSE,
                       method = 'lm', formula = y~x, colour = "#E69F00") +
  ggplot2::geom_smooth(data = subset(diff.river3, subset = Marker == "nuDNA.Fst"),
                       ggplot2::aes(x = dist, y = fixation_index),se = FALSE,
                       method = 'lm', formula = y~x, colour = "#56B4E9") +
  ggplot2::guides(colour = "none") +
  ggplot2::xlab("Geographic distance (km)") +
  ggplot2::ylab(expression(paste("Fixation index (", phi,"st or Fst)"))) +
  ggplot2::theme_bw() +
  ggplot2::theme(legend.position = "top",
                 legend.title = ggplot2::element_blank())
print(plot)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



9.5.4 Compare PHIst/Fst per region

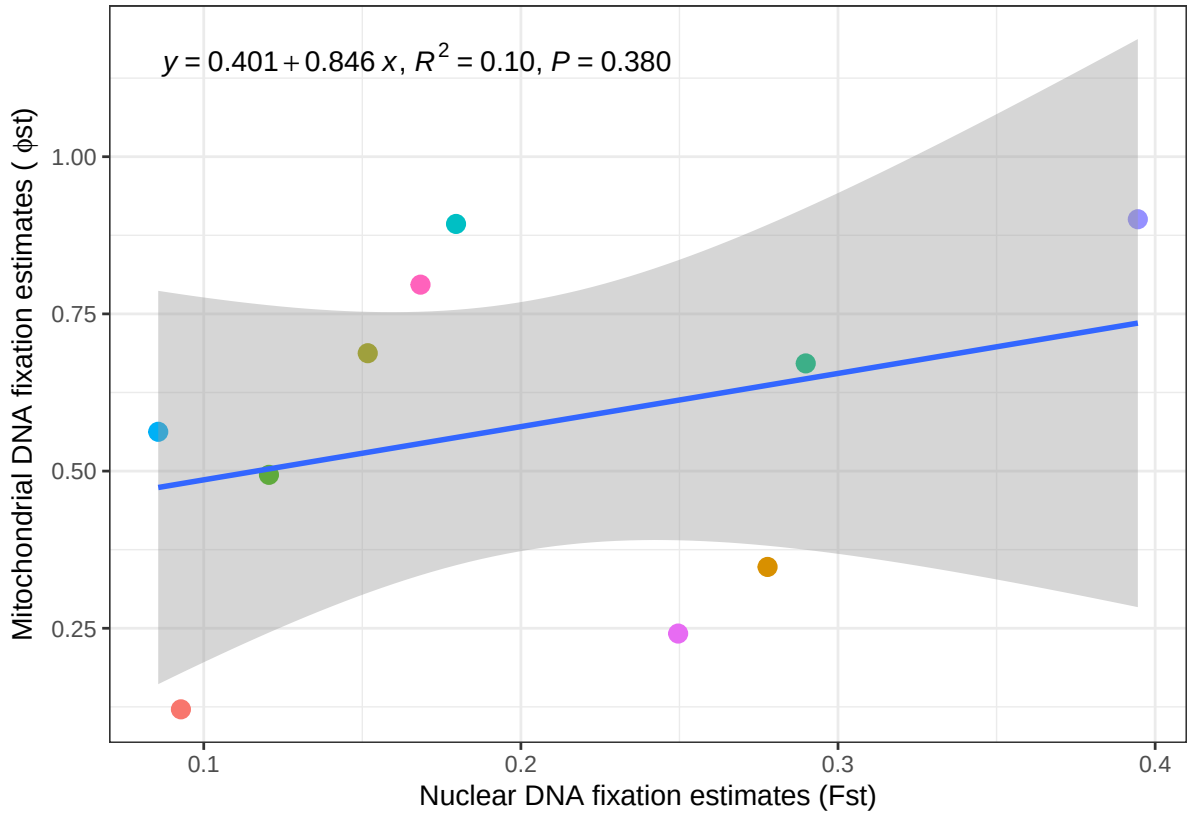
```
diff.region <- data.frame(Compared = rownames(popstr.mt.pop$pairwise$result),
                          mtDNA.PHIst = popstr.mt.pop$pairwise$result$PHIst,
                          mtDNA.PHIst.pval = popstr.mt.pop$pairwise$result$PHIst.p.val,
                          nuDNA.Fst = popstr.snp.pop$pairwise$result$Fst,
```

Table 40: PHIst and Fst comparison between regions

Compared	mtDNA.PHIst	mtDNA.PHIst.pval	nuDNA.Fst	nuDNA.Fst.pval
Cambridge Gulf (30) v. Daly River (30)	0.1211	4e-04	0.0928	1e-04
Cambridge Gulf (30) v. King Sound (14)	0.3475	1e-04	0.2778	1e-04
Cambridge Gulf (30) v. Papua New Guinea (6)	0.6875	1e-04	0.1517	1e-04
Cambridge Gulf (30) v. Van Diemen Gulf (303)	0.4940	1e-04	0.1206	1e-04
Daly River (30) v. King Sound (14)	0.6712	1e-04	0.2898	1e-04
Daly River (30) v. Papua New Guinea (6)	0.8931	1e-04	0.1796	1e-04
Daly River (30) v. Van Diemen Gulf (303)	0.5625	1e-04	0.0856	1e-04
King Sound (14) v. Papua New Guinea (6)	0.9006	1e-04	0.3945	6e-04
King Sound (14) v. Van Diemen Gulf (303)	0.2416	1e-04	0.2496	1e-04
Papua New Guinea (6) v. Van Diemen Gulf (303)	0.7965	1e-04	0.1684	1e-04

```
                        nuDNA.Fst.pval = popstr.snp.pop$pairwise$result$Fst.p.val)
knitr::kable(diff.region,
             digits = 4,format = "latex",
             caption = "PHIst and Fst comparison between regions") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")
```

```
fix.plot <- ggplot2::ggplot(data = diff.region, ggplot2::aes(x = nuDNA.Fst,
                                                             y = mtDNA.PHIst)) +
  ggplot2::geom_point(ggplot2::aes(colour = Compared),size = 3) +
  ggpmisc::stat_poly_line() +
  ggpmisc::stat_poly_eq(ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::xlab("Nuclear DNA fixation estimates (Fst)") +
  ggplot2::ylab(expression(paste("Mitochondrial DNA fixation estimates ( ", phi, "st)"))) +
  ggplot2::theme_bw() +
  ggplot2::theme(legend.position = "none", legend.title = ggplot2::element_blank())
print(fix.plot)
```



```
coord <- readr::read_csv("Region_coord.csv")
```

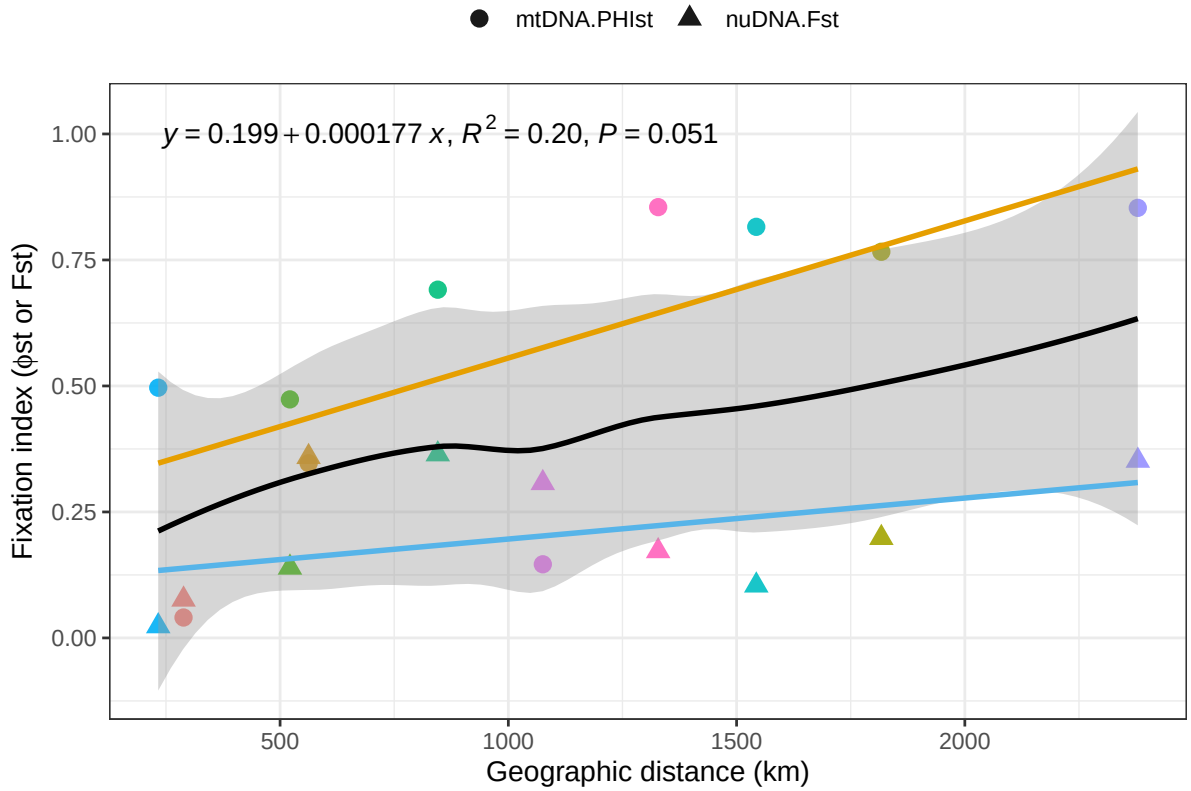
```
## Rows: 5 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Region
## dbl (2): lat, lon
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
geo.dist <- geodist::geodist(coord[,2:3])/1000
```

```
## Maximum distance is > 100km. The 'cheap' measure is inaccurate over such
## large distances, you'd likely be better using a different 'measure',
## one of 'haversine', 'vincenty', or 'geodesic'.
```

```
geo.dist <- as.matrix(geo.dist)
dimnames(geo.dist) <- list(coord$Region, coord$Region)
xy <- t(combn(colnames(geo.dist), 2))
distance <- data.frame(xy, dist = geo.dist[xy])

mtch.names1 <- paste0(pmin(popstr.mt.pop$pairwise$result$strata.1,
                           popstr.mt.pop$pairwise$result$strata.2),
                      "_",
                      pmax(popstr.mt.pop$pairwise$result$strata.1,
                           popstr.mt.pop$pairwise$result$strata.2))
mtch.names2 <- paste0(pmin(distance$X1,distance$X2), "_", pmax(distance$X1,distance$X2))

distance <- distance[order(match(mtch.names2, mtch.names1)),]
diff.region2 <- cbind(diff.region, distance)

diff.region3 <- tidyr::gather(diff.region2, "Marker", "fixation_index",
                              mtDNA.PHIst, nuDNA.Fst)

plot <- ggplot2::ggplot(diff.region3, ggplot2::aes(x = dist, y = fixation_index)) +
  ggplot2::geom_jitter(ggplot2::aes(x = dist, y = fixation_index,
                                    colour = Compared, shape = Marker),
                       width = 0.1, height = 0.1, size = 3, alpha = 0.9) +
  ggplot2::geom_smooth(ggplot2::aes(x = dist, y = fixation_index),
                       method = 'loess', show.legend = FALSE, colour = "black") +
  ggpmisc::stat_poly_eq(ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::geom_smooth(data = subset(diff.region3, subset = Marker == "mtDNA.PHIst"),
                       ggplot2::aes(x = dist, y = fixation_index),se = FALSE,
                       method = 'lm', formula = y~x, colour = "#E69F00") +
  ggplot2::geom_smooth(data = subset(diff.region3, subset = Marker == "nuDNA.Fst"),
                       ggplot2::aes(x = dist, y = fixation_index),se = FALSE,
                       method = 'lm', formula = y~x, colour = "#56B4E9") +
  ggplot2::guides(colour = "none") +
  ggplot2::xlab("Geographic distance (km)") +
  ggplot2::ylab(expression(paste("Fixation index (", phi,"st or Fst)"))) +
  ggplot2::theme_bw() +
  ggplot2::theme(legend.position = "top",
                 legend.title = ggplot2::element_blank())
print(plot)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


mtDNA.PHIst
nuDNA.Fst
y = 0.199 + 0.000177 x, R2 = 0.20, P = 0.051
Fixation index (φst or Fst)
1.00
0.75
0.50
0.25
0.00
500
1000
1500
2000
Geographic distance (km)

9.6 Assignment

Using the package assignPOP to estimate the assignment success of the DArTcap data

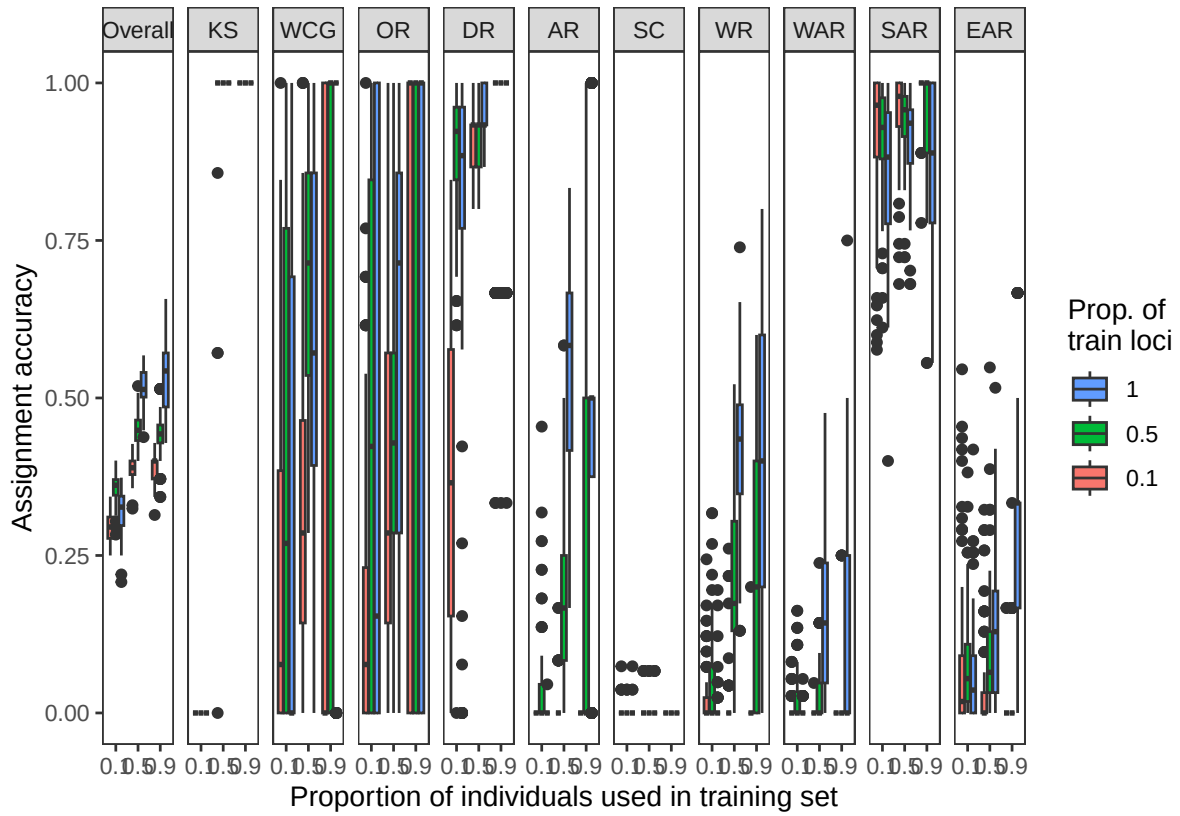
9.6.1 Per River

```
gl2gpop(ggar.SNP.gl[!ggar.SNP.gl$pop =="Papua New Guinea",],
        filename = "GG_assign_river_genepop.txt")
gpop <- assignPOP::read.Genepop(
  'GG_assign_river_genepop.txt',
  pop.names = shortnames[-11],
  haploid = FALSE,
  pos = 1
  )

assignPOP::assign.MC(
  x = gpop, train.inds = c(0.1, 0.5, 0.9),train.loci = c(0.1, 0.5, 1),
  loci.sample = 'fst',iterations = 100, dir = 'AssignPOP_MC_River/',
  scaled = FALSE,
  pca.method = 'mixed', pca.PCs = 'kaiser-guttman',pca.loadings = F,
  model = 'svm', svm.kernel = 'linear',svm.cost = 1, ntree = 50,
  multiprocess = TRUE, skipQ = FALSE)
```

```
df1 <- assignPOP::accuracy.MC(dir = "AssignPOP_MC_River/")
```

```
##
## Correct assignment rates were estimated!!
## A total of 900 assignment tests for 10 pops.
## Results were also saved in a
'Rate_of_900_tests_10_pops.txt' file in the directory.
```

```
MC.assign <- assignPOP::accuracy.plot(df1, pop = c("all",shortnames[-11]))
print(MC.assign)
```

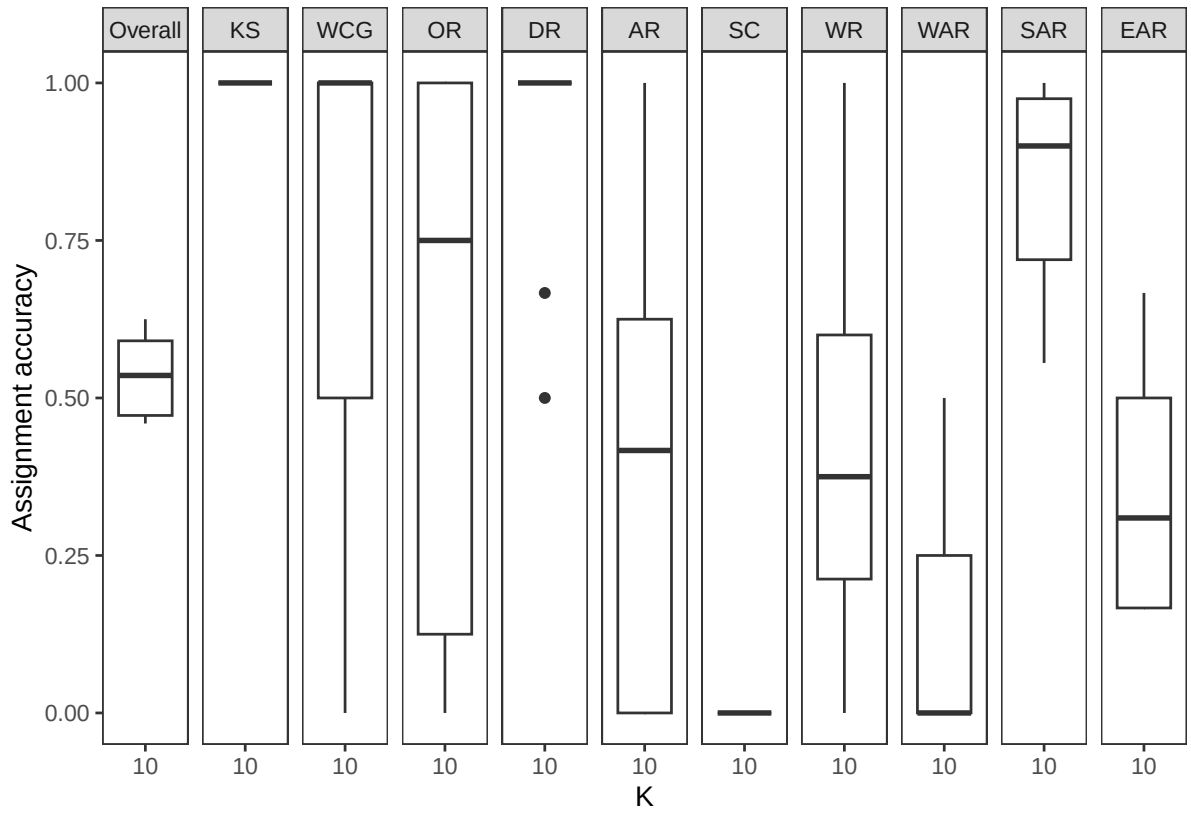



```
assignPOP::assign.kfold(
  x = gpop, k.fold = 10, train.loci = 1,loci.sample = 'fst',
  dir = 'AssignPOP_Kfold_River/', scaled = FALSE, pca.method = 'mixed',
  pca.PCs = 'kaiser-guttman', pca.loadings = F, model = 'svm',
  svm.kernel = 'linear',svm.cost = 1, ntree = 50, multiprocess = TRUE,
  skipQ =FALSE)
```

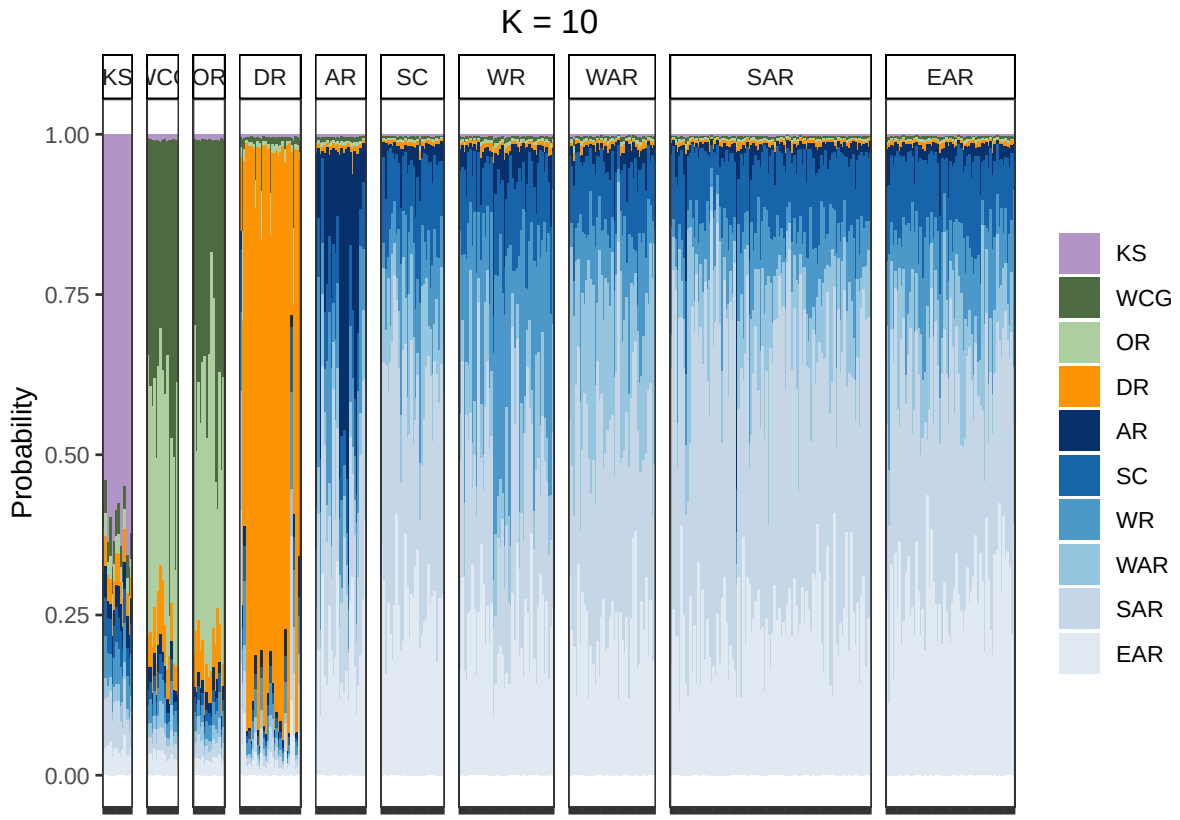
```
df2 <- assignPOP::accuracy.kfold(dir = "AssignPOP_Kfold_River/")
```

```
##
## Correct assignment rates were estimated!!
## A total of 10 assignment tests for 10 pops.
## Results were also saved in a
'Rate_of_10_tests_10_pops.txt' file in the directory.
```

```
kfold.assign <- assignPOP::accuracy.plot(df2, pop = c("all",shortnames[-11]))
print(kfold.assign)
```




```
col <- colours.11[-11]
names(col) <- shortnames[-11]
memb.plot <- assignPOP::membership.plot(dir = "AssignPOP_Kfold_River/",
                            style = 1,
                            non.genetic = TRUE)
memb.plot$data$origin.pop <- factor(memb.plot$data$origin.pop,
                                    levels = shortnames[-11])
memb.plot + ggplot2::scale_fill_manual(values = col) +
  ggplot2::theme(axis.text.x = ggplot2::element_blank())
```




```
df <- data.frame(
 ind = as.character(memb.plot$data$Ind.ID)[
    as.character(memb.plot$data$fold_n) == "fold_10"],
  pop = as.character(memb.plot$data$origin.pop)[
    as.character(memb.plot$data$fold_n) == "fold_10"],
  memb = as.character(memb.plot$data$pred.pop)[
    as.character(memb.plot$data$fold_n) == "fold_10"], stringsAsFactors = FALSE)

summary(df$pop[df$pop != df$memb])
```

```
##    Length     Class      Mode
##       150 character character
```

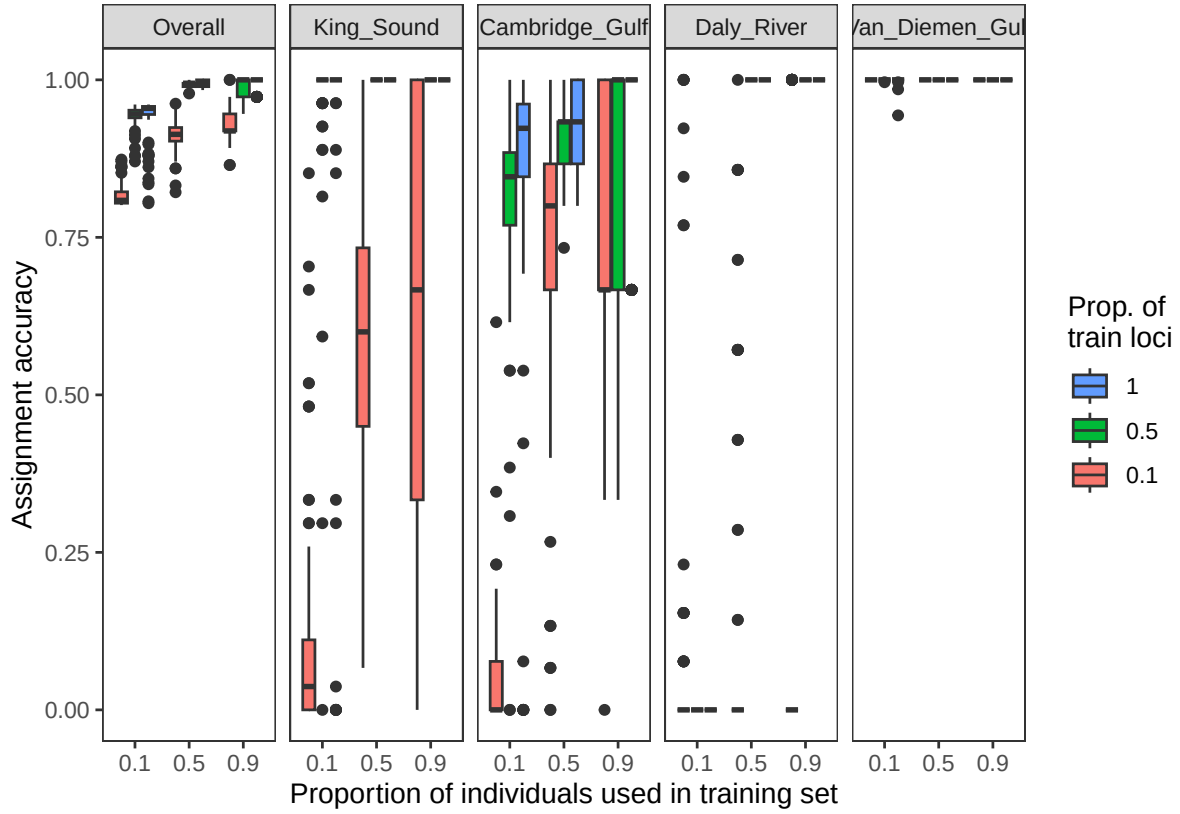
9.6.2 Per region

```
ggar.SNP.gl$pop <- ggar.SNP.gl$other$ind.metrics$pop
gl2gpop(ggar.SNP.gl[!ggar.SNP.gl$pop =="Papua New Guinea",],
        filename = "GG_assign_pop_genepop.txt")
gpop <- assignPOP::read.Genepop(
  'GG_assign_pop_genepop.txt',
  pop.names = c("King_Sound", "Cambridge_Gulf", "Daly_River", "Van_Diemen_Gulf"),
  haploid = FALSE,
  pos = 1
  )

assignPOP::assign.MC(
  x = gpop, train.inds = c(0.1, 0.5, 0.9),train.loci = c(0.1, 0.5, 1),
  loci.sample = 'fst',iterations = 100, dir = 'AssignPOP_MC_Regions/', scaled = FALSE,
  pca.method = 'mixed', pca.PCs = 'kaiser-guttman',pca.loadings = F,
  model = 'svm', svm.kernel = 'linear',svm.cost = 1, ntree = 50,
  multiprocess = TRUE, skipQ = FALSE)
```

```
df1 <- assignPOP::accuracy.MC(dir = "AssignPOP_MC_Regions/")
```

```
##
## Correct assignment rates were estimated!!
## A total of 900 assignment tests for 4 pops.
## Results were also saved in a
'Rate_of_900_tests_4_pops.txt' file in the directory.
```

```
MC.assign <- assignPOP::accuracy.plot(df1, pop = c("all","King_Sound",
                                                   "Cambridge_Gulf", "Daly_River",
                                                   "Van_Diemen_Gulf"))
print(MC.assign)
```

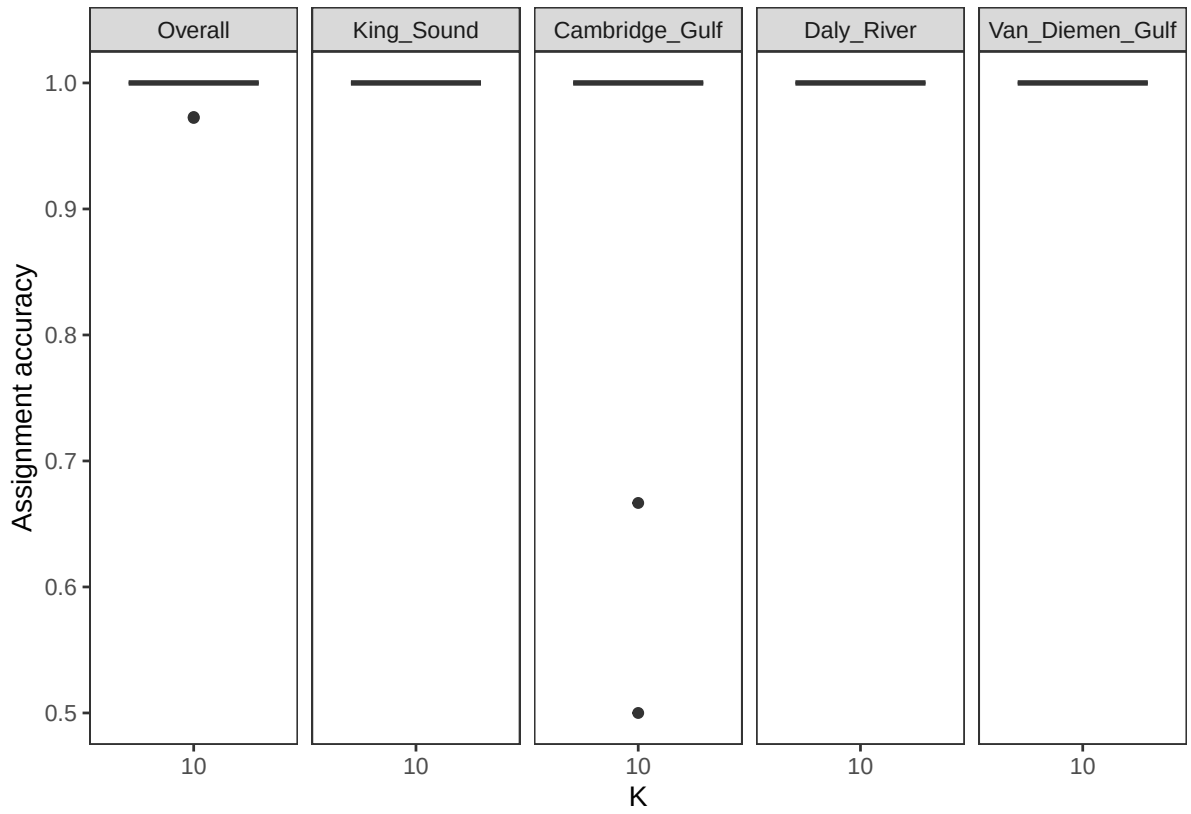



```
assignPOP::assign.kfold(
  x = gpop, k.fold = 10, train.loci =  1,loci.sample = 'fst',
  dir = 'AssignPOP_Kfold_Region/', scaled = FALSE, pca.method = 'mixed',
  pca.PCs = 'kaiser-guttman', pca.loadings = F, model = 'svm',
  svm.kernel = 'linear',svm.cost = 1, ntree = 50, multiprocess = TRUE,
  skipQ =FALSE)
```

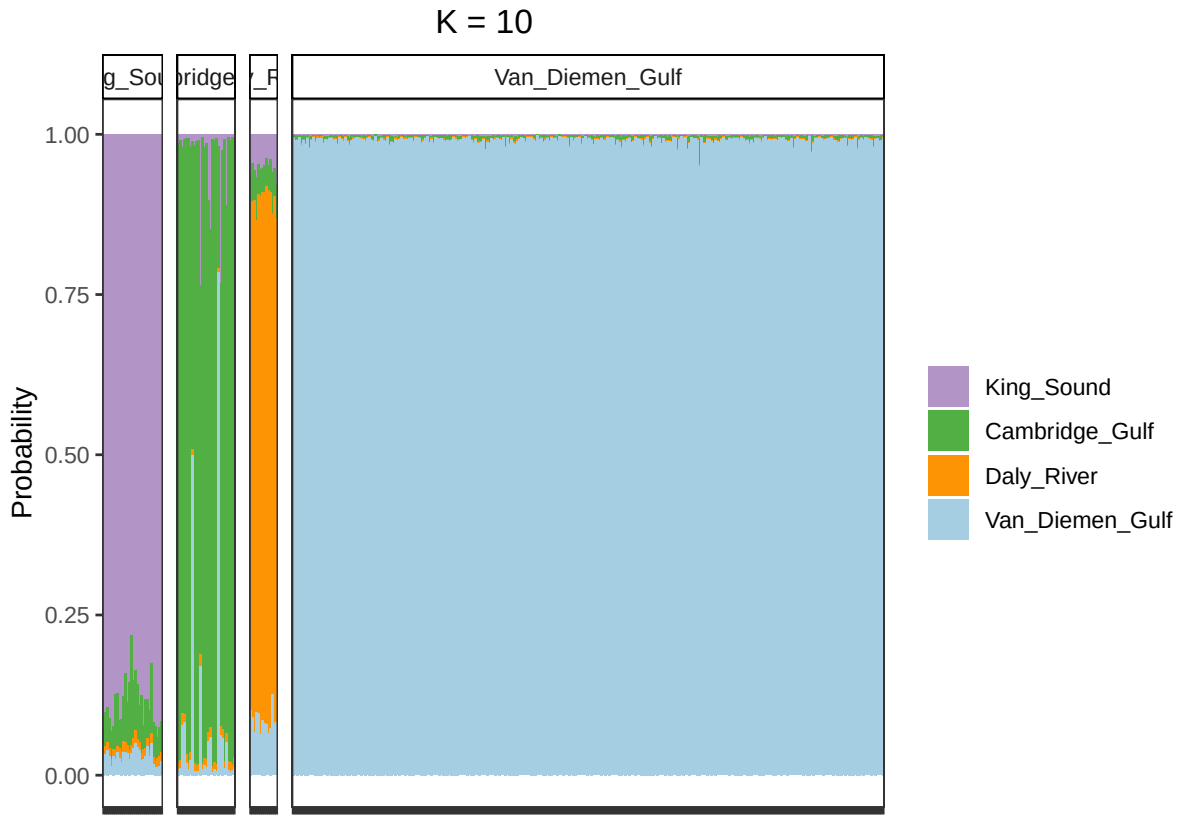
```
df2 <- assignPOP::accuracy.kfold(dir = "AssignPOP_Kfold_Region/")
```

```
##
## Correct assignment rates were estimated!!
## A total of 10 assignment tests for 4 pops.
## Results were also saved in a
'Rate_of_10_tests_4_pops.txt' file in the directory.
```

```
kfold.assign <- assignPOP::accuracy.plot(df2, pop = c("all","King_Sound",
                                                      "Cambridge_Gulf", "Daly_River",
                                                      "Van_Diemen_Gulf"))
print(kfold.assign)
```




```
col <- colours.5[-5]
names(col) <- c("King_Sound","Cambridge_Gulf", "Daly_River", "Van_Diemen_Gulf")
memb.plot <- assignPOP::membership.plot(dir = "AssignPOP_Kfold_Region/",
                            style = 1,
                            non.genetic = TRUE)
memb.plot$data$origin.pop <- factor(memb.plot$data$origin.pop,
                                    levels = c("King_Sound","Cambridge_Gulf",
                                               "Daly_River", "Van_Diemen_Gulf"))
memb.plot + ggplot2::scale_fill_manual(values = col) +
  ggplot2::theme(axis.text.x = ggplot2::element_blank())
```




```
df <- data.frame(
  ind = as.character(memb.plot$data$Ind.ID)[
    as.character(memb.plot$data$fold_n) == "fold_10"],
  pop = as.character(memb.plot$data$origin.pop)[
    as.character(memb.plot$data$fold_n) == "fold_10"],
  memb = as.character(memb.plot$data$pred.pop)[
    as.character(memb.plot$data$fold_n) == "fold_10"], stringsAsFactors = FALSE)

summary(df$pop[df$pop != df$memb])
```

```
##    Length     Class      Mode
##         0 character character
```

---

10 Kinship analysis

Kinship relations were determined in Feutry et al. 2020. Here, we assign haplotypes to all kin pairs.

Briefly, each individual's genotype was compared against each other to calculate the likelihood that two sharks were related following Bravington et al. (2016). This likelihood was also used to control for false positive and false negative detections, which is essential when estimating connectivity based on kinship. Kinship was only investigated in the region with the largest number of samples (i.e. VDG).

10.1 Print family groups

- 4 POPs, 34 FSPs and 139 HSPs
- 73 family groups:
  - 43 pairwise relationships
  - 1 POP-FSP group
  - 1 POP-HSP group
  - 13 FSP-HSP groups
  - 15 only HSP groups

```
source("pw_export_function.R")
pwdata <- read.csv("../3.Data/Ggar_pairwise relationship.csv")

paste0("Number of unique individuals that have identified kinship: ",
       length(unique(c(pwdata$iname, pwdata$jname))))
```

```
## [1] "Number of unique individuals that have identified
kinship: 208"
```

```
PWchain <- chain_pairwise(pwdata)

PLOD <- list()
REL <- list()
for (i in 1:length(PWchain)) {
  PLOD[[i]] <- chain_with_plods(achain = PWchain[[i]],
                                all_plods = pwdata,
                                digits = 1L,
                                character. = TRUE)
  name1 <- c()
  for (c in 1:ncol(PWchain[[i]])) {
    row1 <- as.integer(colnames(PWchain[[i]])[c])
    if (!sum(pwdata$i %in% row1) == 0) {
      name1 <- c(name1, as.character(unique(pwdata$iname[pwdata$i %in%
                                                         row1])))
    } else {
      name1 <- c(name1, as.character(unique(pwdata$jname[pwdata$j %in%
                                                         row1])))
    }
  }
  rownames(PLOD[[i]]) <- name1
  colnames(PLOD[[i]]) <- name1
  REL[[i]] <-  chain_with_rel(achain = PWchain[[i]],
                              all_rel = pwdata,
                              character. = TRUE)
  rownames(REL[[i]]) <- name1
  colnames(REL[[i]]) <- name1
}
print(REL)
```

```
## [[1]]
## GGA052 GGSA017
## GGA052 .  HSP
## GGSA017 HSP .
##
## [[2]]
## GGA054 GGA260
## GGA054 .  HSP
## GGA260 HSP .
##
## [[3]]
## GGA191 GGA206
## GGA191 .  HSP
## GGA206 HSP .
##
## [[4]]
## GGA194 GGA198
## GGA194 .  HSP
## GGA198 HSP .
##
## [[5]]
## GGA202 GGS019
## GGA202 .  HSP
## GGS019 HSP .
##
## [[6]]
## GGA204 GGA082
## GGA204 .  HSP
## GGA082 HSP .
##
## [[7]]
## GGA212 GGA182
## GGA212 .  FSP
## GGA182 FSP .
##
## [[8]]
## GGA335 GGL023
## GGA335 .  HSP
## GGL023 HSP .
##
## [[9]]
## GGA341 GGA239
## GGA341 .  FSP
## GGA239 FSP .
##
## [[10]]
## GGA342 GGA047
## GGA342 .  HSP
## GGA047 HSP .
##
## [[11]]
## GGA344 GGA331
## GGA344 .  HSP
## GGA331 HSP .
##
## [[12]]
## GGA371 GGA114
## GGA371 .  FSP
```

```
## GGA114 FSP .
##
## [[13]]
## GGL028 GGL018
## GGL028 .  HSP
## GGL018 HSP .
##
## [[14]]
## GGL031 GGA141
## GGL031 .  HSP
## GGA141 HSP .
##
## [[15]]
## GGL033 GGL002
## GGL033 .  FSP
## GGL002 FSP .
##
## [[16]]
## GGS025 GGA019
## GGS025 .  FSP
## GGA019 FSP .
##
## [[17]]
## GGSA015 GGA103
## GGSA015 .  HSP
## GGA103 HSP .
##
## [[18]]
## GGA116 GGA363
## GGA116 .  HSP
## GGA363 HSP .
##
## [[19]]
## GGA133 GGS018
## GGA133 .  HSP
## GGS018 HSP .
##
## [[20]]
## GGA148 GGA183
## GGA148 .  HSP
## GGA183 HSP .
##
## [[21]]
## GGA166 GGA168
## GGA166 .  HSP
## GGA168 HSP .
##
## [[22]]
## GGA243 GGA248
## GGA243 .  HSP
## GGA248 HSP .
##
## [[23]]
## GGA244 GGA253
## GGA244 .  HSP
## GGA253 HSP .
##
## [[24]]
```

```
## GGA247 GGA154
## GGA247 .  HSP
## GGA154 HSP .
##
## [[25]]
## GGA249 GGA250
## GGA249 .  FSP
## GGA250 FSP .
##
## [[26]]
## GGA356 GGA362
## GGA356 .  FSP
## GGA362 FSP .
##
## [[27]]
## GGS012 GGS004
## GGS012 .  FSP
## GGS004 FSP .
##
## [[28]]
## GGS017 GGA186
## GGS017 .  HSP
## GGA186 HSP .
##
## [[29]]
## GGA131 GGA267
## GGA131 .  FSP
## GGA267 FSP .
##
## [[30]]
## GGA214 GGA078
## GGA214 .  HSP
## GGA078 HSP .
##
## [[31]]
## GGA075 GGA108
## GGA075 .  HSP
## GGA108 HSP .
##
## [[32]]
## GGA087 GGA256
## GGA087 .  FSP
## GGA256 FSP .
##
## [[33]]
## GGA088 GGA347
## GGA088 .  POP
## GGA347 POP .
##
## [[34]]
## GGA254 GGA035
## GGA254 .  HSP
## GGA035 HSP .
##
## [[35]]
## GGA124 GGA213
## GGA124 .  HSP
## GGA213 HSP .
```

```
##
## [[36]]
## GGA080 GGA083
## GGA080 .  HSP
## GGA083 HSP .
##
## [[37]]
## GGA270 GGA001
## GGA270 .  HSP
## GGA001 HSP .
##
## [[38]]
## GGA085 GGA038
## GGA085 .  FSP
## GGA038 FSP .
##
## [[39]]
## GGL021 GGL008
## GGL021 .  HSP
## GGL008 HSP .
##
## [[40]]
## GGS021 GGS009
## GGS021 .  HSP
## GGS009 HSP .
##
## [[41]]
## GGA333 GGA235
## GGA333 .  HSP
## GGA235 HSP .
##
## [[42]]
## GGA334 GGA234
## GGA334 .  HSP
## GGA234 HSP .
##
## [[43]]
## GGA346 GGA348
## GGA346 .  HSP
## GGA348 HSP .
##
## [[44]]
## GGA055 GGS006 GGA164
## GGA055 .  .  POP
## GGS006 .  .  HSP
## GGA164 POP HSP .
##
## [[45]]
## GGA059 GGA023 GGA095
## GGA059 .  HSP HSP
## GGA023 HSP .  FSP
## GGA095 HSP FSP .
##
## [[46]]
## GGA061 GGA158 GGA048
## GGA061 .  HSP HSP
## GGA158 HSP .  FSP
## GGA048 HSP FSP .
```

```
##
## [[47]]
## GGA065 GGA327 GGA343
## GGA065 .   .   HSP
## GGA327 .   .   HSP
## GGA343 HSP HSP .
##
## [[48]]
## GGA340 GGA171 GGA330
## GGA340 .   .   HSP
## GGA171 .   .   HSP
## GGA330 HSP HSP .
##
## [[49]]
## GGA369 GGA372 GGA374
## GGA369 .   .   HSP
## GGA372 .   .   HSP
## GGA374 HSP HSP .
##
## [[50]]
## GGL026 GGA179 GGSA014
## GGL026 .   .   HSP
## GGA179 .   .   HSP
## GGSA014 HSP HSP .
##
## [[51]]
## GGL019 GGL007 GGL030
## GGL019 .   .   HSP
## GGL007 .   .   HSP
## GGL030 HSP HSP .
##
## [[52]]
## GGA030 GGA092 GGA258
## GGA030 .   HSP FSP
## GGA092 HSP .   HSP
## GGA258 FSP HSP .
##
## [[53]]
## GGA112 GGA367 GGA117
## GGA112 .   HSP FSP
## GGA367 HSP .   HSP
## GGA117 FSP HSP .
##
## [[54]]
## GGA147 GGA173 GGA139
## GGA147 .   HSP HSP
## GGA173 HSP .   HSP
## GGA139 HSP HSP .
##
## [[55]]
## GGA169 GGA242 GGA240
## GGA169 .   .   HSP
## GGA242 .   .   HSP
## GGA240 HSP HSP .
##
## [[56]]
## GGA049 GGA007 GGA091
## GGA049 .   .   HSP
```

```
## GGA007 .   .   HSP
## GGA091 HSP HSP .
##
## [[57]]
## GGA044 GGA175 GGA190
## GGA044 .   POP POP
## GGA175 POP .   FSP
## GGA190 POP FSP .
##
## [[58]]
## GGA066 GGA024 GGA027 GGA185
## GGA066 .   FSP FSP HSP
## GGA024 FSP .   FSP HSP
## GGA027 FSP FSP .   HSP
## GGA185 HSP HSP HSP .
##
## [[59]]
## GGA368 GGA113 GGA115 GGA119
## GGA368 .   .   HSP HSP
## GGA113 .   .   HSP HSP
## GGA115 HSP HSP .   FSP
## GGA119 HSP HSP FSP .
##
## [[60]]
## GGA218 GGA093 GGA018 GGA036
## GGA218 .   .   HSP HSP
## GGA093 .   .   HSP HSP
## GGA018 HSP HSP .   FSP
## GGA036 HSP HSP FSP .
##
## [[61]]
## GGA105 GGA259 GGA226 GGA033
## GGA105 .   .   .   HSP
## GGA259 .   .   HSP HSP
## GGA226 .   HSP .   HSP
## GGA033 HSP HSP HSP .
##
## [[62]]
## GGA174 GGL006 GGA111 GGS016
## GGA174 .   .   .   HSP
## GGL006 .   .   HSP .
## GGA111 .   HSP .   HSP
## GGS016 HSP .   HSP .
##
## [[63]]
## GGL015 GGL013 GGL003 GGL016
## GGL015 .   .   .   HSP
## GGL013 .   .   HSP HSP
## GGL003 .   HSP .   HSP
## GGL016 HSP HSP HSP .
##
## [[64]]
## GGA228 GGA084 GGA219 GGA160
## GGA228 .   .   HSP .
## GGA084 .   .   .   HSP
## GGA219 HSP .   .   HSP
## GGA160 .   HSP HSP .
##
```

```
## [[65]]
## GGA079 GGA121 GGA217 GGA101
## GGA079 .  .  .  HSP
## GGA121 .  .  HSP HSP
## GGA217 .  HSP .  HSP
## GGA101 HSP HSP HSP .
##
## [[66]]
## GGA264 GGA261 GGA063 GGA022 GGA155
## GGA264 .  .  HSP .  .
## GGA261 .  .  .  HSP HSP
## GGA063 HSP .  .  HSP HSP
## GGA022 .  HSP HSP .  FSP
## GGA155 .  HSP HSP FSP .
##
## [[67]]
## GGA201 GGA266 GGA336 GGA332 GGA003
## GGA201 .  .  .  HSP .
## GGA266 .  .  HSP .  .
## GGA336 .  HSP .  .  HSP
## GGA332 HSP .  .  .  HSP
## GGA003 .  .  HSP HSP .
##
## [[68]]
## GGA310 GGA357 GGA354 GGA304 GGA361
## GGA310 .  .  HSP HSP HSP
## GGA357 .  .  HSP HSP FSP
## GGA354 HSP HSP .  HSP HSP
## GGA304 HSP HSP HSP .  HSP
## GGA361 HSP FSP HSP HSP .
##
## [[69]]
## GGA026 GGA077 GGA156 GGA045 GGA021
## GGA026 .  .  FSP .  HSP
## GGA077 .  .  .  HSP HSP
## GGA156 FSP .  .  .  HSP
## GGA045 .  HSP .  .  HSP
## GGA021 HSP HSP HSP HSP .
##
## [[70]]
## GGA120 GGA076 GGA100 GGA257 GGA157
## GGA120 .  FSP FSP HSP FSP
## GGA076 FSP .  FSP HSP FSP
## GGA100 FSP FSP .  HSP FSP
## GGA257 HSP HSP HSP .  HSP
## GGA157 FSP FSP FSP HSP .
##
## [[71]]
## GGS026 GGA032 GGA150 GGA086 GGA225 GGA271 GGA096
## GGS026 .  .  HSP .  .  .  .
## GGA032 .  .  .  HSP .  .  .
## GGA150 HSP .  .  .  .  .  HSP
## GGA086 .  HSP .  .  .  HSP .
## GGA225 .  .  .  .  .  HSP HSP
## GGA271 .  .  .  HSP HSP .  HSP
## GGA096 .  .  HSP .  HSP HSP .
##
## [[72]]
```

```
## GGSA013 GGA043 GGA152 GGA216 GGA221 GGA020 GGA098
## GGSA013 .  HSP .  .  .  .  .
## GGA043 HSP .  .  HSP .  .  .
## GGA152 .  .  .  .  HSP HSP HSP
## GGA216 .  HSP .  .  .  HSP HSP
## GGA221 .  .  HSP .  .  HSP HSP
## GGA020 .  .  HSP HSP HSP .  FSP
## GGA098 .  .  HSP HSP HSP FSP .
##
## [[73]]
## GGA370 GGA358 GGA365 GGA366 GGA360 GGA364 GGA355 GGA353
GGA345
## GGA370 .  .  .  .  HSP .  .  .  .
## GGA358 .  .  .  .  .  .  .  HSP HSP
## GGA365 .  .  .  FSP .  .  .  .  HSP
## GGA366 .  .  FSP .  .  .  .  .  HSP
## GGA360 HSP .  .  .  .  .  HSP HSP .
## GGA364 .  .  .  .  .  .  FSP HSP HSP
## GGA355 .  .  .  .  HSP FSP .  HSP HSP
## GGA353 .  HSP .  .  HSP HSP HSP .  FSP
## GGA345 .  HSP HSP HSP .  HSP HSP FSP .
```

10.2 Add metadata to kinship data

10.2.1 Haplotypes

- 3 PO groups: each group have a different haplotype.
- 27 FS groups: each group shares a haplotypes: max 4 full sibling in the same group.
- 60 HS groups: 19 groups have different haplotypes: max 9 half siblings in the same group.

```
metafile <- "../3.Data/Glyphis garricki_Metadata2_ALL_10-05-2019.csv"
meta <- readr::read_csv(metafile)
```

```
## New names:
## Rows: 571 Columns: 34
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (24): SubsetID, id, pop, River, sample location, sex, tl_mm, date, Sampl... dbl
## (10): txtlat, txtlon, lat, lon, SampleLat, SampleLon, Conc. (ng/ul), 260...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `Mitogenome selected` -> `Mitogenome selected...28`
## * `Mitogenome selected` -> `Mitogenome selected...29`
```

```
meta %<>%  dplyr::rename(POP = `POP family(according to MVB)`,
                         FSP = `Full Sib family(according to MVB)`,
                         HSP = `Half Sib family(according to MVB)`)
meta["HT"] <- ""
haplist <- ggar.hap@haplist
HTs <- paste0("HT", sprintf('%0.2d', 1:ggar.hap@nhap))
for (i in 1:nrow(meta)) {
  for (j in 1:ggar.hap@nhap) {
    if (meta$id[i] %in% haplist[[j]]) {
      meta$HT[i] <- HTs[j]
    }
  }
}

POtab <- table(meta$POP, meta$HT)
```

Table 41: Number of individuals (per FS group) in each haplotype

	HT06	HT07	HT13	HT14	HT22	HT25
PO001	0	0	1	0	1	0
PO002	0	0	0	1	0	2
PO003	1	1	0	0	0	0

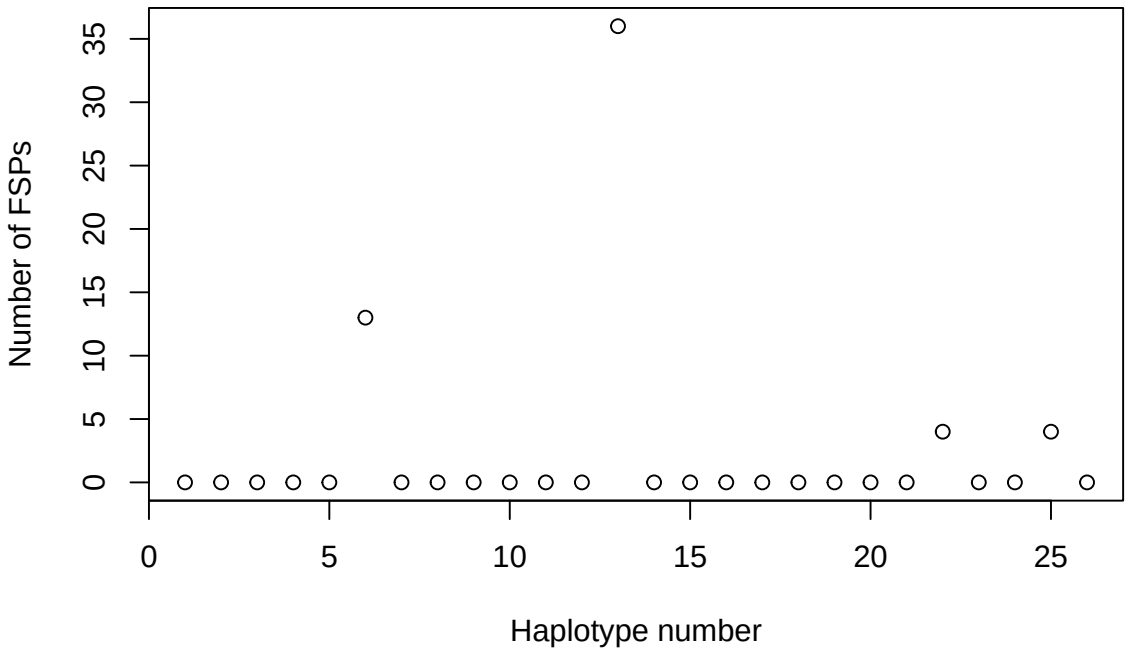
Table 42: Number of individuals (per FS group) in each haplotype

	HT06	HT13	HT22	HT25
FS001	2	0	0	0
FS002	3	0	0	0
FS003	0	2	0	0
FS004	0	2	0	0
FS005	0	2	0	0
FS006	0	4	0	0
FS007	0	2	0	0
FS008	0	2	0	0
FS009	2	0	0	0
FS010	0	2	0	0
FS011	2	0	0	0
FS012	0	2	0	0
FS013	0	2	0	0
FS014	0	2	0	0
FS015	0	2	0	0
FS016	2	0	0	0
FS017	0	2	0	0
FS018	0	2	0	0
FS019	0	0	0	2
FS020	0	0	2	0
FS021	0	0	2	0
FS022	0	2	0	0
FS023	0	2	0	0
FS024	0	2	0	0
FS025	2	0	0	0
FS026	0	0	0	2
FS027	0	2	0	0

```
FStab <- table(meta$FSP, meta$HT)
HStab <- table(meta$HSP, meta$HT)

PO0 <- colSums(POtab[,-1] != 0)
knitr::kable(POtab[,-1][, PO0 != 0], format = "latex",
             caption = "Number of individuals (per FS group) in each haplotype") %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
FS0 <- colSums(FStab[,-1] != 0)
knitr::kable(FStab[,-1][, FS0 != 0], format = "latex",
             caption = "Number of individuals (per FS group) in each haplotype") %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
nFS <- colSums(FStab[,-1])
plot(nFS, xlab = "Haplotype number", ylab = "Number of FSPs")
```

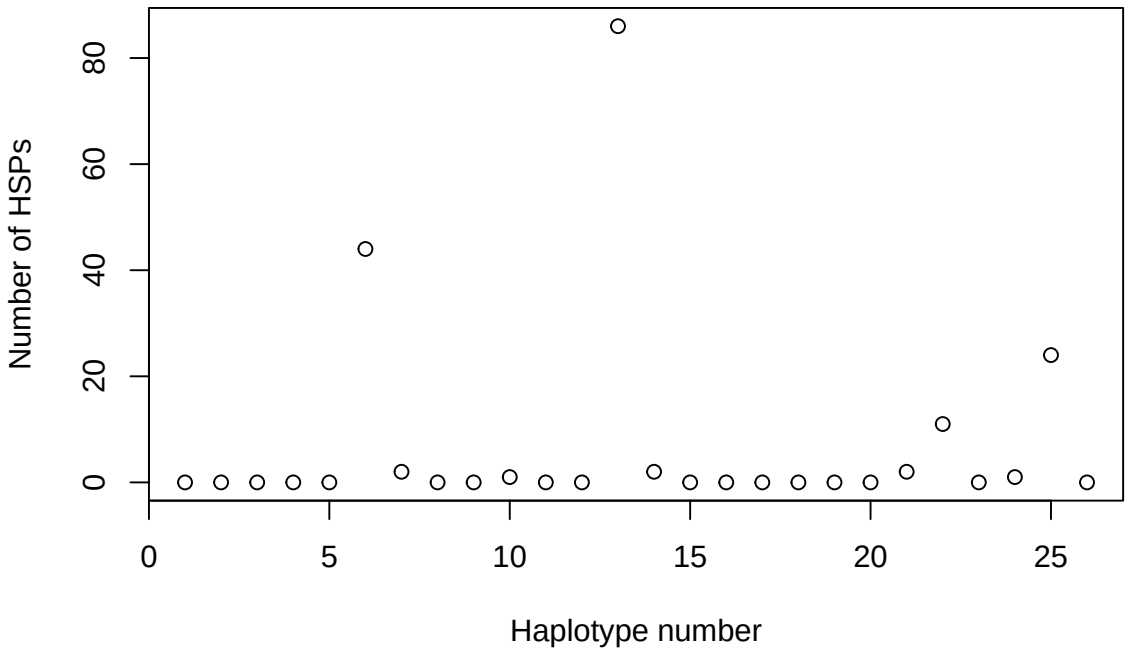



```
HS0 <- colSums(HStab[,-1] != 0)
HStab2 <- HStab[,-1][, HS0 != 0]
knitr::kable(HStab2, format = "latex",
             caption = "Number of individuals (per HS group) in each haplotype") %>%
  kableExtra::kable_styling(full_width = FALSE)
```

```
nHS <- colSums(HStab[,-1])
plot(nHS, xlab = "Haplotype number", ylab = "Number of HSPs")
```

Table 43: Number of individuals (per HS group) in each haplotype

	HT06	HT07	HT10	HT13	HT14	HT21	HT22	HT24	HT25
HS001	4	0	0	0	0	0	0	0	0
HS002	3	0	0	4	0	0	0	0	0
HS003	0	0	0	5	0	0	0	0	0
HS004	2	0	0	3	0	0	0	0	0
HS005	3	0	0	1	0	0	0	0	0
HS006	3	0	0	0	0	0	0	0	0
HS007	3	2	0	2	0	0	0	0	0
HS008	4	0	0	0	0	0	0	0	0
HS009	0	0	0	1	0	0	0	0	0
HS010	2	0	0	0	0	0	0	0	0
HS011	0	0	0	3	0	0	0	0	0
HS012	0	0	0	3	0	0	0	0	0
HS013	0	0	0	2	1	0	0	0	0
HS014	0	0	0	1	1	0	0	0	0
HS015	0	0	0	5	0	0	0	0	0
HS016	0	0	0	2	0	0	0	0	0
HS017	0	0	0	4	0	0	0	0	0
HS018	1	0	0	2	0	0	0	0	0
HS019	0	0	0	1	0	0	0	0	3
HS020	0	0	0	0	0	2	0	0	2
HS021	0	0	0	0	0	0	3	0	0
HS022	0	0	0	4	0	0	0	0	0
HS023	0	0	0	2	0	0	0	0	0
HS024	0	0	0	1	0	0	1	0	0
HS025	0	0	0	2	0	0	0	0	0
HS026	3	0	0	0	0	0	0	0	0
HS027	1	0	0	0	0	0	0	0	1
HS028	0	0	0	0	0	0	2	0	0
HS029	0	0	0	0	0	0	0	0	2
HS030	0	0	0	2	0	0	0	0	1
HS031	1	0	0	2	0	0	0	0	0
HS032	0	0	0	0	0	0	0	0	2
HS033	0	0	0	0	0	0	0	0	2
HS034	1	0	0	3	0	0	0	0	1
HS035	0	0	0	0	0	0	0	0	2
HS036	1	0	0	1	0	0	0	0	0
HS037	0	0	0	2	0	0	0	0	0
HS038	0	0	0	2	0	0	0	0	0
HS039	0	0	0	2	0	0	0	0	0
HS040	0	0	0	2	0	0	0	0	0
HS041	0	0	0	0	0	0	0	0	2
HS042	0	0	0	0	0	0	0	0	1
HS043	0	0	0	0	0	0	4	0	0
HS044	0	0	0	2	0	0	0	0	0
HS045	0	0	0	2	0	0	0	0	0
HS046	1	0	0	1	0	0	0	0	0
HS047	0	0	0	2	0	0	0	0	0
HS048	2	0	0	0	0	0	0	0	0
HS049	0	0	0	0	0	0	1	0	1
HS050	0	0	0	9	0	0	0	0	0
HS051	2	0	0	0	0	0	0	0	0
HS052	0	0	1	0	0	0	0	0	0
HS053	0	0	0	0	0	0	0	1	2
HS054	1	0	0	1	0	0	0	0	0
HS055	2	0	0	0	0	0	0	0	0
HS056	2	0	0	0	0	0	0	0	0
HS057	0	0	0	0	0	0	0	0	2
HS058	0	0	0	2	0	0	0	0	0
HS059	2	0	0	0	0	0	0	0	0




```
max(rowSums(HStab2))
```

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

```
x <- apply(HStab2, 1, function(c)plyr::count(c == 0))
HStab.multiple.HTs <- HStab2[sapply(x , `[[`, 2)[1,] > 1,]
knitr::kable(HStab.multiple.HTs, format = "latex",
             caption = "Number of individuals (per HS group) with differing haplotypes") %>%
  kableExtra::kable_styling(full_width = FALSE)
```

10.2.2 VBG growth curve

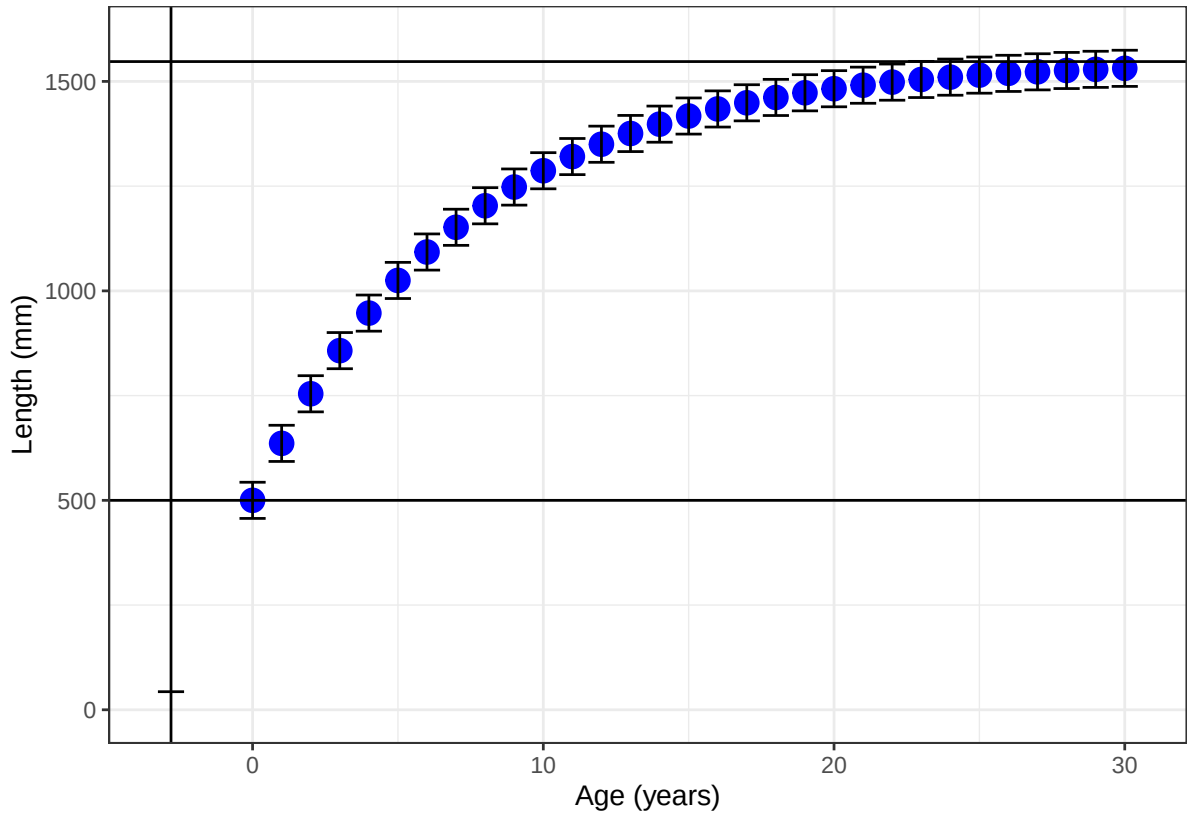
This curve was constructed from the 34 recaptured individuals (see Bravington et al. 2019)

```
k <- 0.1391
Linf <- 1547.3827
L0 <- 500
sd <- 43.1619
t0 <- log((L0 - Linf)/-Linf)*(1/k)
age <- c(t0, 0, seq(1,30))
length <- L0 + (Linf - L0) * (1 - exp(-k * age))
length.min <- length - sd
length.max <- length + sd

df <- data.frame(age,
                 length, length.min, length.max)

ggplot2::ggplot(df) +
  ggplot2::geom_point(ggplot2::aes(x = age, y = length),
                      size = 4, colour = "blue") +
  ggplot2::geom_errorbar(ggplot2::aes(x = age,ymax = length.min,
                                      ymin = length.max)) +
  ggplot2::geom_hline(yintercept = L0) +
```

Table 44: Number of individuals (per HS group) with differing haplotypes

	HT06	HT07	HT10	HT13	HT14	HT21	HT22	HT24	HT25
HS002	3	0	0	4	0	0	0	0	0
HS004	2	0	0	3	0	0	0	0	0
HS005	3	0	0	1	0	0	0	0	0
HS007	3	2	0	2	0	0	0	0	0
HS013	0	0	0	2	1	0	0	0	0
HS014	0	0	0	1	1	0	0	0	0
HS018	1	0	0	2	0	0	0	0	0
HS019	0	0	0	1	0	0	0	0	3
HS020	0	0	0	0	0	2	0	0	2
HS024	0	0	0	1	0	0	1	0	0
HS027	1	0	0	0	0	0	0	0	1
HS030	0	0	0	2	0	0	0	0	1
HS031	1	0	0	2	0	0	0	0	0
HS034	1	0	0	3	0	0	0	0	1
HS036	1	0	0	1	0	0	0	0	0
HS046	1	0	0	1	0	0	0	0	0
HS049	0	0	0	0	0	0	1	0	1
HS053	0	0	0	0	0	0	0	1	2
HS054	1	0	0	1	0	0	0	0	0

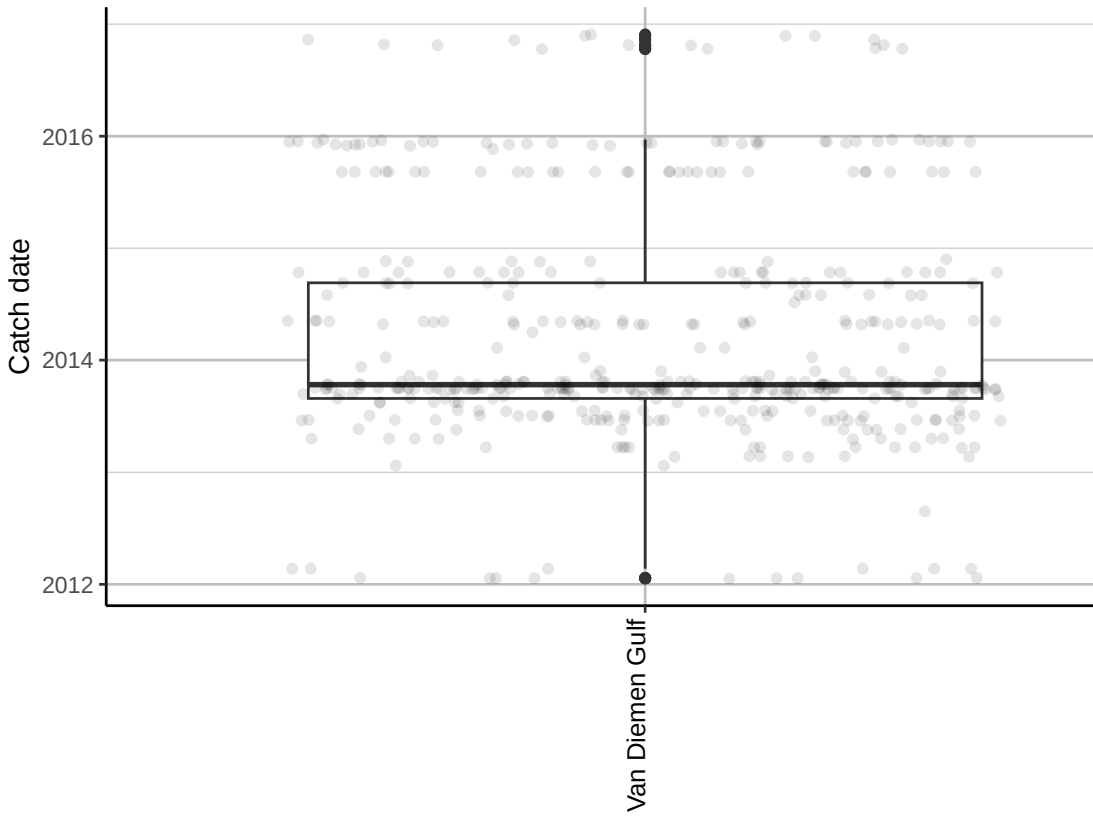
```
ggplot2::geom_hline(yintercept = Linf) +
ggplot2::geom_vline(xintercept = t0) +
ggplot2::ylim(0,1600) +
ggplot2::xlab("Age (years)") +  ggplot2::ylab("Length (mm)") +
ggplot2::theme_bw()
```



10.2.3 Add cohort data for all VDG samples

```
meta <- meta[meta$pop == "Van Diemen Gulf",]
meta <- meta[order(match(meta$id, adegenet::indNames(ggar.gi))),]

pop.levels1 <- c("King Sound", "West Cambridge Gulf", "Ord River",
                 "Daly River", "Adelaide River", "Sampan Creek", "Wildman River",
                 "West Alligator River", "South Alligator River",
                 "East Alligator River", "Papua New Guinea")
meta$River <- factor(x = meta$River, levels = pop.levels1)

pop.levels2 <- c("King Sound", "Cambridge Gulf", "Daly River",
                 "Van Diemen Gulf", "Papua New Guinea")
meta$pop <- factor(x = meta$pop, levels = pop.levels2)
meta$date <- as.Date(meta$date, "%d/%m/%Y")
```

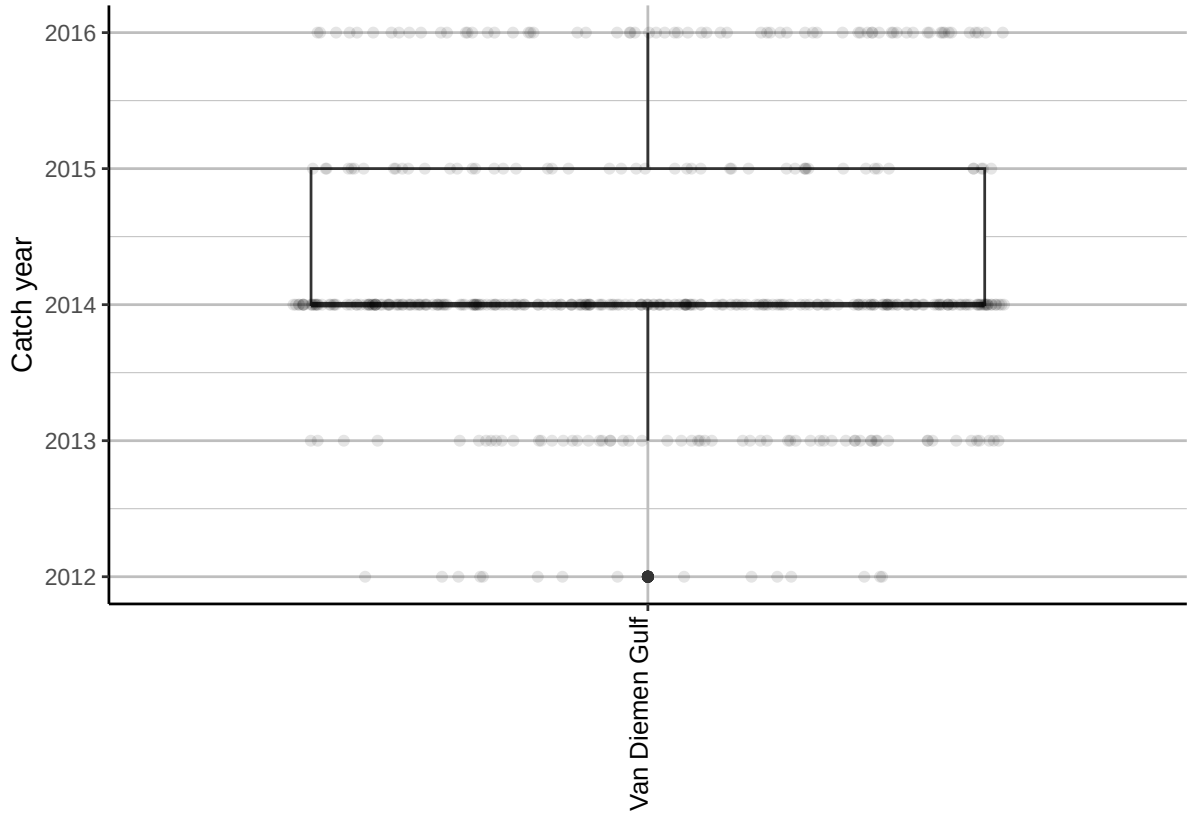
```
date.plot <- ggplot2::ggplot(meta, ggplot2::aes(x = pop, y = date)) +
  ggplot2::geom_boxplot() +
  ggplot2::geom_jitter(alpha = 0.1, height = 0) +
  ggplot2::labs(y = "Catch date") +
  ggplot2::theme(
    axis.title.x = ggplot2::element_blank(),
    axis.text.x = ggplot2::element_text(color = "black", size = 10, angle = 90,
                                        hjust = 1, vjust = 0),
    panel.background = ggplot2::element_rect(fill = "white"),
    panel.grid.major = ggplot2::element_line(size = 0.5, linetype = 'solid',
                                             colour = "grey"),
    panel.grid.minor = ggplot2::element_line(size = 0.1, linetype = 'solid',
                                             colour = "grey"),
    axis.line = ggplot2::element_line(colour = "black", size = 0.5,
                                      linetype = 1, lineend = "butt")
  )
print(date.plot)
```



10.2.3.1 Catch date

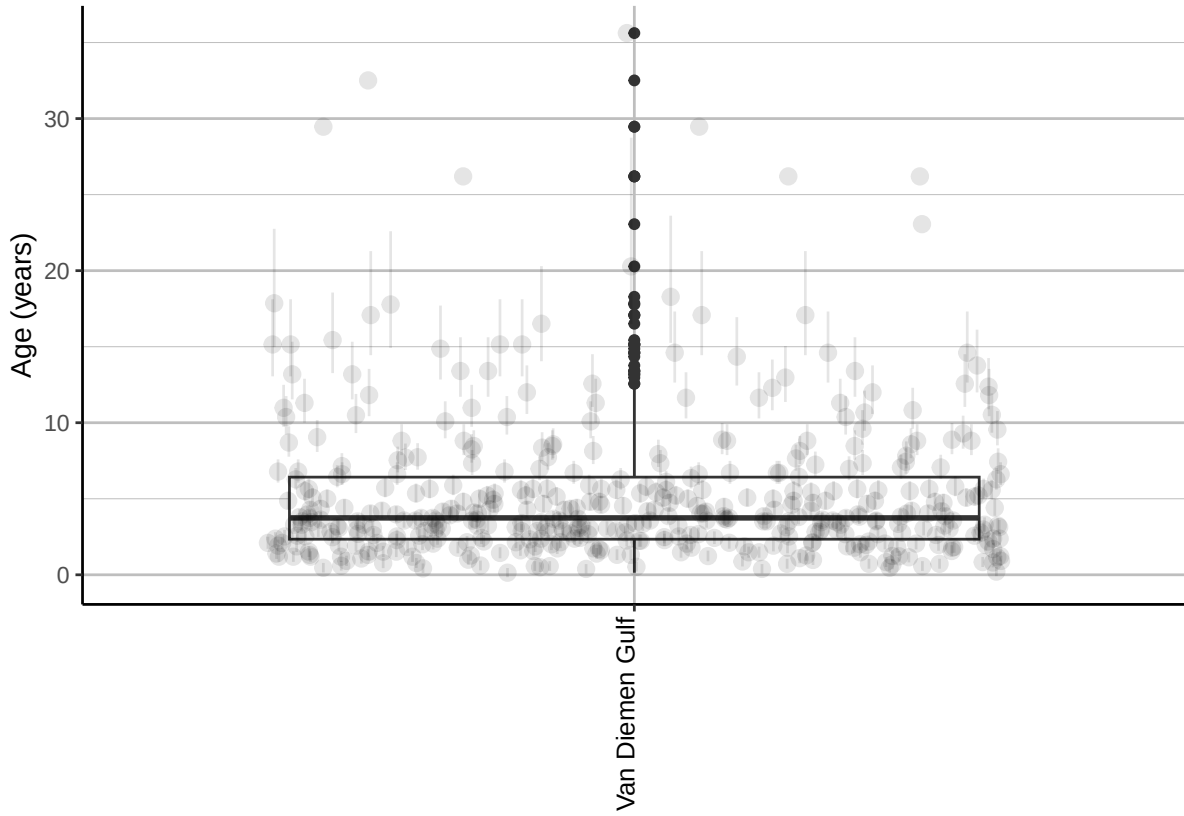
```
meta[["yearcollect"]] <- NA
meta$yearcollect[meta$date <= as.Date("2003-07-01")] <- "2003"
meta$yearcollect[meta$date >= as.Date("2003-07-01") &
                   meta$date < as.Date("2011-06-30")] <- "2010"
meta$yearcollect[meta$date >= as.Date("2011-07-01") &
                   meta$date < as.Date("2012-06-30")] <- "2012"
meta$yearcollect[meta$date >= as.Date("2012-07-01") &
                   meta$date < as.Date("2013-06-30")] <- "2013"
meta$yearcollect[meta$date >= as.Date("2013-07-01") &
                   meta$date < as.Date("2014-06-30")] <- "2014"
meta$yearcollect[meta$date >= as.Date("2014-07-01") &
                   meta$date < as.Date("2015-06-30")] <- "2015"
meta$yearcollect[meta$date >= as.Date("2015-07-01") &
                   meta$date < as.Date("2016-06-30")] <- "2016"
meta$yearcollect <- as.numeric(meta$yearcollect)

date2.plot <- ggplot2::ggplot(meta,
                              ggplot2::aes(x = pop, y = yearcollect)) +
  ggplot2::geom_boxplot() +
  ggplot2::geom_jitter(alpha = 0.1, height = 0) +
  ggplot2::labs(y = "Catch year") +
  ggplot2::theme(
    axis.title.x = ggplot2::element_blank(),
    axis.text.x = ggplot2::element_text(color = "black",
                                        size = 10, angle = 90,
                                        hjust = 1, vjust = 0),
    panel.background = ggplot2::element_rect(fill = "white"),
    panel.grid.major = ggplot2::element_line(size = 0.5, linetype = 'solid',
                                             colour = "grey"),
    panel.grid.minor = ggplot2::element_line(size = 0.1, linetype = 'solid',
                                             colour = "grey"),
```

```
    axis.line = ggplot2::element_line(colour = "black", size = 0.5,
                                      linetype = 1, lineend = "butt")
  )
print(date2.plot)
```




```
k <- 0.1391
Linf <- 1547.3827
L0 <- 500
sd <- 43.1619
meta[["Age"]] <- NA
if (class(meta$tl_mm) == "character") {
  meta$tl_mm <- as.numeric(meta$tl_mm)
}
meta$Age <- -log(1 - (meta$tl_mm - L0)/(Linf - L0))/k
meta[["Age_lower"]] <- NA
meta$Age_lower <- -log(1 - ((meta$tl_mm - 43.1619) - L0)/(Linf - L0))/k
meta[["Age_upper"]] <- NA
meta$Age_upper <- -log(1 - ((meta$tl_mm + 43.1619) - L0)/(Linf - L0))/k

age.plot <- ggplot2::ggplot(meta, ggplot2::aes(x = pop, y = Age)) +
  ggplot2::geom_boxplot() +
  ggplot2::geom_pointrange(ggplot2::aes(ymin = Age_lower, ymax = Age_upper),
                  position = ggplot2::position_jitter(height = 0), alpha = 0.1) +
  ggplot2::labs(y = "Age (years)") +
  ggplot2::theme(
    axis.title.x = ggplot2::element_blank(),
    axis.text.x = ggplot2::element_text(color = "black",
                                        size = 10, angle = 90,
                                        hjust = 1, vjust = 0),
```

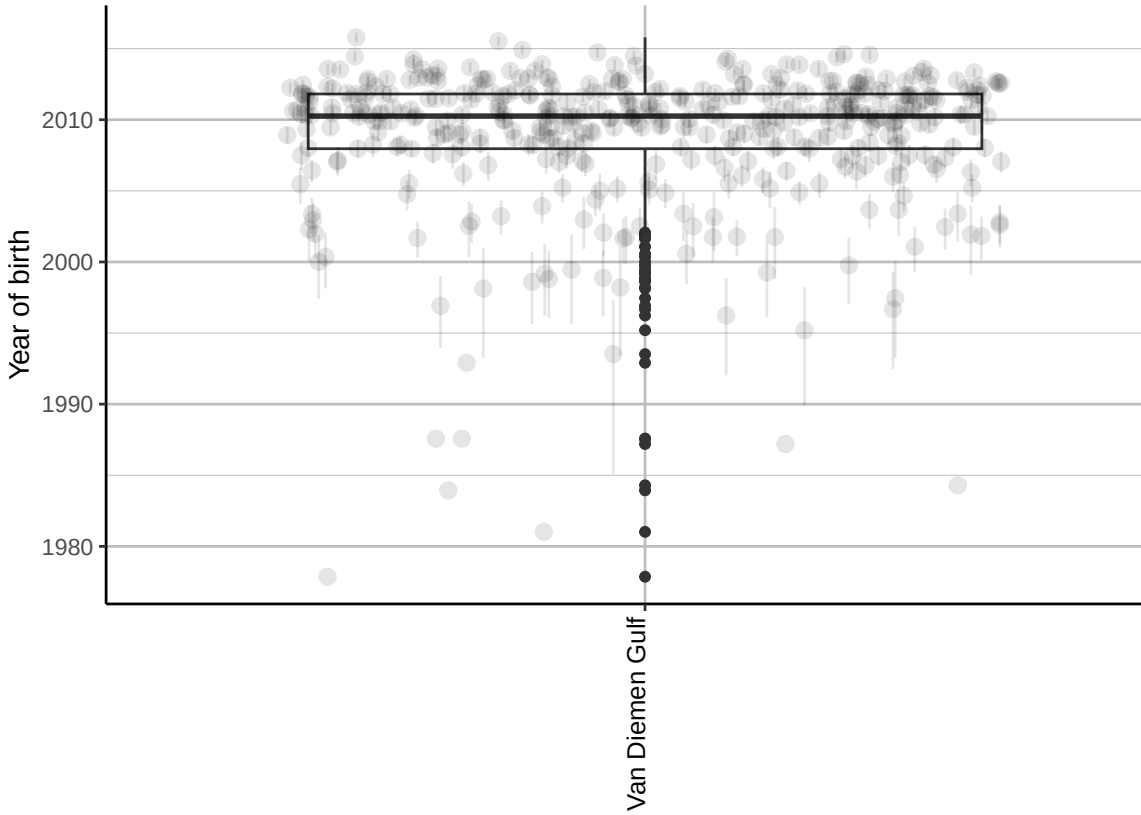
```
    panel.background = ggplot2::element_rect(fill = "white"),
    panel.grid.major = ggplot2::element_line(size = 0.5, linetype = 'solid',
                                             colour = "grey"),
    panel.grid.minor = ggplot2::element_line(size = 0.1, linetype = 'solid',
                                             colour = "grey"),
    axis.line = ggplot2::element_line(colour = "black", size = 0.5,
                                      linetype = 1, lineend = "butt")
  )
print(age.plot)
```



10.2.3.2 Age

```
meta[["Cohort"]] <- meta$date - meta$Age*365
meta[["Cohort_lower"]] <- meta$date - meta$Age_upper*365
meta[["Cohort_upper"]] <- meta$date - meta$Age_lower*365

cohort.plot <- ggplot2::ggplot(meta, ggplot2::aes(x = pop, y = Cohort)) +
  ggplot2::geom_boxplot() +
  ggplot2::geom_pointrange(ggplot2::aes(ymin = Cohort_lower, ymax = Cohort_upper),
                position = ggplot2::position_jitter(height = 0), alpha = 0.1) +
  ggplot2::labs(y = "Year of birth") +
  ggplot2::theme(
    axis.title.x = ggplot2::element_blank(),
        axis.text.x = ggplot2::element_text(color = "black",
                                            size = 10, angle = 90,
                                            hjust = 1, vjust = 0),
        panel.background = ggplot2::element_rect(fill = "white"),
        panel.grid.major = ggplot2::element_line(size = 0.5, linetype = 'solid',
                                                 colour = "grey"),
        panel.grid.minor = ggplot2::element_line(size = 0.1, linetype = 'solid',
                                                 colour = "grey"),
```

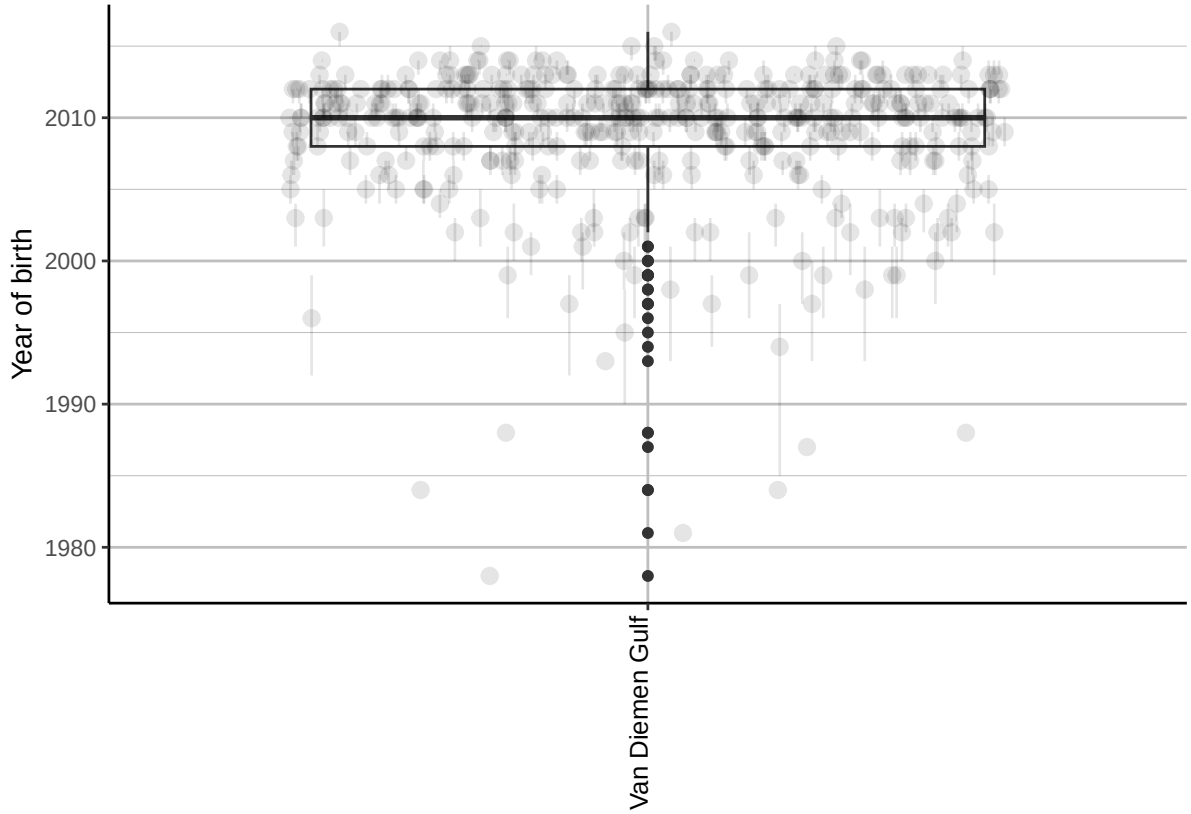
```
        axis.line = ggplot2::element_line(colour = "black", size = 0.5,
                                          linetype = 1, lineend = "butt")
  )
print(cohort.plot)
```



10.2.3.3 Cohort

```
years <- lubridate::round_date(meta$Cohort, unit = "year")
meta$Cohort2 <- as.integer(sub("-.*$", "", years))
years_lower <- lubridate::round_date(meta$Cohort_lower, unit = "year")
meta$Cohort2_lower <- as.integer(sub("-.*$", "", years_lower))
years_upper <- lubridate::round_date(meta$Cohort_upper, unit = "year")
meta$Cohort2_upper <- as.integer(sub("-.*$", "", years_upper))


cohort2.plot <- ggplot2::ggplot(meta, ggplot2::aes(x = pop, y = Cohort2)) +
  ggplot2::geom_boxplot() +
  ggplot2::geom_pointrange(ggplot2::aes(ymin = Cohort2_lower, ymax = Cohort2_upper),
                 position = ggplot2::position_jitter(height = 0),
                 alpha = 0.1) +
  ggplot2::labs(y = "Year of birth") +
  ggplot2::theme(
    axis.title.x = ggplot2::element_blank(),
        axis.text.x = ggplot2::element_text(color = "black",
                                            size = 10, angle = 90,
                                            hjust = 1, vjust = 0),
        panel.background = ggplot2::element_rect(fill = "white"),
        panel.grid.major = ggplot2::element_line(size = 0.5, linetype = 'solid',
                                                 colour = "grey"),
        panel.grid.minor = ggplot2::element_line(size = 0.1, linetype = 'solid',
                                                 colour = "grey"),
        axis.line = ggplot2::element_line(colour = "black", size = 0.5,
                                          linetype = 1, lineend = "butt")
```

```
  )
print(cohort2.plot)
```



10.2.4 Add haplotype and cohort to kin pairs

```
pwdata$DATE_i <- as.Date(pwdata$DATE_i, "%d/%m/%Y")
pwdata$DATE_j <- as.Date(pwdata$DATE_j, "%d/%m/%Y")

pwdata["HT_i"] <- ""
pwdata["HT_j"] <- ""
pwdata["HTfreq_i"] <- integer()
pwdata["HTfreq_j"] <- integer()
pwdata["Cohort_i_lower"] <- integer()
pwdata["Cohort_i"] <- integer()
pwdata["Cohort_i_upper"] <- integer()
pwdata["Cohort_j_lower"] <- integer()
pwdata["Cohort_j"] <- integer()
pwdata["Cohort_j_upper"] <- integer()


meta$id <- as.character(meta$id)
pwdata$iname <- as.character(pwdata$iname)
pwdata$jname <- as.character(pwdata$jname)

for (i in 1:nrow(pwdata)) {
  pwdata$HT_i[i] <- meta$HT[meta$id == pwdata$iname[i]]
  pwdata$HT_j[i] <- meta$HT[meta$id == pwdata$jname[i]]
  pwdata$HTfreq_i[i] <- meta$Htfreq_perRiver[meta$id == pwdata$iname[i]]
  pwdata$HTfreq_j[i] <- meta$Htfreq_perRiver[meta$id == pwdata$jname[i]]
  pwdata$Cohort_i_lower[i] <- as.integer(meta$Cohort2_lower[meta$id == pwdata$iname[i]])
```

```
  pwdata$Cohort_i[i] <- as.integer(meta$Cohort2[meta$id == pwdata$iname[i]])
  pwdata$Cohort_i_upper[i] <- as.integer(meta$Cohort2_upper[meta$id == pwdata$iname[i]])
  pwdata$Cohort_j_lower[i] <- as.integer(meta$Cohort2_lower[meta$id == pwdata$jname[i]])
  pwdata$Cohort_j[i] <- as.integer(meta$Cohort2[meta$id == pwdata$jname[i]])
  pwdata$Cohort_j_upper[i] <- as.integer(meta$Cohort2_upper[meta$id == pwdata$jname[i]])
}

pwdata$HT_i[pwdata$HT_i == ""] <- NA
pwdata$HT_j[pwdata$HT_j == ""] <- NA
```

10.2.5 Check if cohort of kin pairs overlap

```
rng1 = cbind(
  pmin(pwdata$Cohort_i_lower, pwdata$Cohort_i_upper),
  pmax(pwdata$Cohort_i_lower, pwdata$Cohort_i_upper),
  pmin(pwdata$Cohort_j_lower, pwdata$Cohort_j_upper),
  pmax(pwdata$Cohort_j_lower, pwdata$Cohort_j_upper))
olap1 = (rng1[,1] <= rng1[,4]) & (rng1[,2] >= rng1[,3])

sum(!olap1[pwdata$relation == "FSP"])
```

```
## [1] 7
```

```
## 7 FSP did not have overlapping cohort
## The lower and upper range are still not perfect since some FSPs appear cross cohort.


## additional 0.5 year range to account for out-of-season pupping
rng2 = cbind(
  pmin(pwdata$Cohort_i_lower - 0.5, pwdata$Cohort_i_upper - 0.5),
  pmax(pwdata$Cohort_i_lower + 0.5, pwdata$Cohort_i_upper + 0.5),
  pmin(pwdata$Cohort_j_lower - 0.5, pwdata$Cohort_j_upper - 0.5),
  pmax(pwdata$Cohort_j_lower + 0.5, pwdata$Cohort_j_upper + 0.5))
olap2 = (rng2[,1] <= rng2[,4]) & (rng2[,2] >= rng2[,3])
pwdata$Cohort_overlap_relaxed <- olap2
sum(!pwdata$Cohort_overlap_relaxed[pwdata$relation == "FSP"])
```

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

```
## 1 FSP did not have overlapping cohort

independent_catch <- pmax(pwdata$DATE_i, pwdata$DATE_j) -
  pmin(pwdata$DATE_i, pwdata$DATE_j) > 14
```

10.3 Philopatry and dispersal from kinship

```
data <- pwdata
kinNWdata <- data %>%
  dplyr::mutate(Cohort_gap = pmax(Cohort_i,Cohort_j) - pmin(Cohort_i,Cohort_j)) %>%
  dplyr::mutate(relation2 =
                  dplyr::case_when(relation == "HSP" & HT_i == HT_j ~ 'MHSP',
                                   relation == "HSP" & HT_i != HT_j ~ 'PHSP',
                                   relation == "HSP" &
                                     (is.na(HT_i) |is.na(HT_j)) ~ 'HSP')) %>%
  dplyr::select(iname, jname, relation, Cohort_gap)
network <- igraph::graph_from_data_frame(d = kinNWdata, directed = TRUE)


df <- data.frame(id = igraph::V(network)$name)
vertices <- dplyr::left_join(df, meta, by = "id") %>%
  dplyr::select(id,`sample location`, sex, HSP, yearcollect, Age,
                Cohort2, HT
                # , Htfreq_perRiver
                )
vertices$HT[vertices$HT == ""] <- NA
vertices <- as_data_frame(vertices, what = "vertices")
```

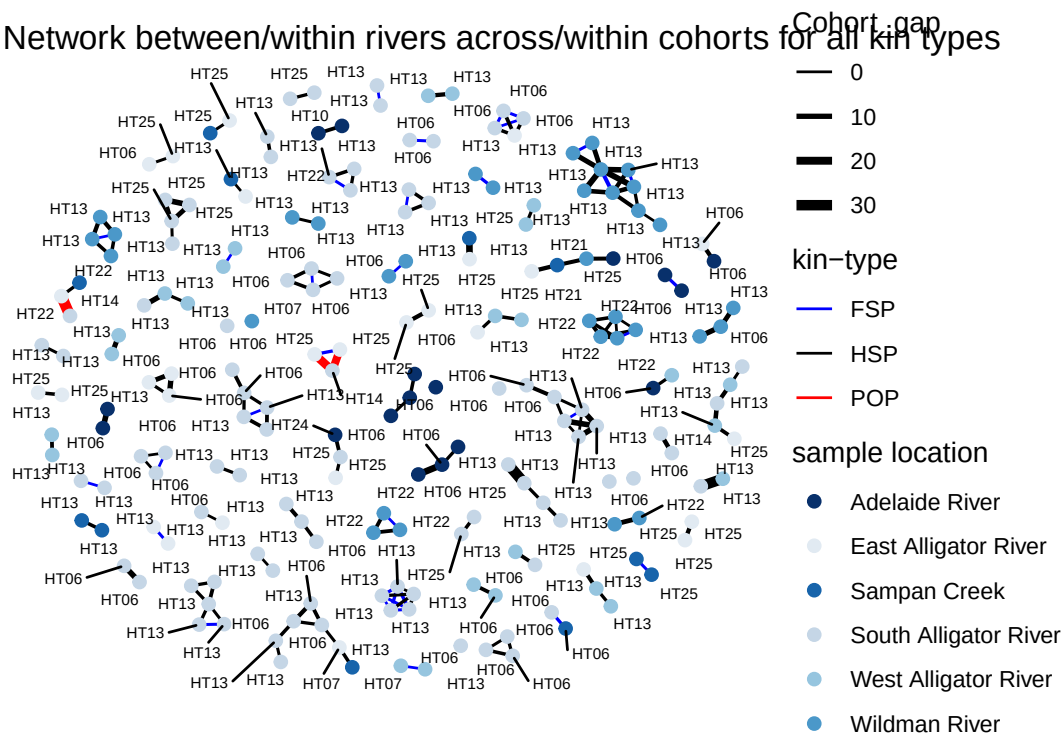
```
## Warning: `as_data_frame()` was deprecated in tibble 2.0.0.
## i Please use `as_tibble()` (with slightly different semantics) to convert to a
##   tibble, or `as.data.frame()` to convert to a data frame.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
network <- igraph::graph_from_data_frame(d = kinNWdata, directed = TRUE,
                                         vertices = vertices )

layout <- ggraph::create_layout(network, layout = 'igraph',
                                circular = FALSE, algorithm = 'fr')
# attributes(layout)

kin_network1 <- ggraph::ggraph(network, layout = layout) +
  ggraph::geom_edge_link(
    aes(width = Cohort_gap,
        edge_colour = factor(relation)),
    edge_alpha = 1) +
  ggraph::scale_edge_width(range = c(0.5, 2), name = "Cohort_gap") +
  ggraph::scale_edge_colour_manual(values = c("blue", "black", "red"),
                                   name = "kin-type",
                                   aesthetics = "edge_colour",
                                   na.value = "grey50") +
  ggraph::geom_node_point(aes(color = `sample location`),
                          size = 2) +
  ggraph::geom_node_text( aes(label = HT), repel = TRUE,
                          size = 2, color = "black") +
  ggplot2::ggtitle("Network between/within rivers across/within cohorts for all kin types") +
  ggplot2::scale_color_manual(values = colours.6, na.value = "grey50") +
  ggplot2::theme_void() +
  ggplot2::theme(
    legend.position = "right",
    plot.margin = unit(rep(1,4), "cm")
  )
```

```
## Warning: Ignoring `graph` as layout is already calculated
## i Pass the calculated layout to the `graph` argument to silence this warning
```

```
print(kin_network1)
```

```
## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_text_repel()`).
```




```
# ggplot2::ggsave(filename = "0.Kinship_network_all_relationships2.png",
#                 bg = "white", plot = kin_network1, device = "png",
#                 width = 40, height = 30, units = "cm")
```

```
data <- pwdata
NWdata <- data %>%
  dplyr::mutate(relation2 =
                  dplyr::case_when(relation == "HSP" & HT_i == HT_j ~ 'MHSP',
                                   relation == "HSP" & HT_i != HT_j ~ 'PHSP',
                                   relation == "HSP" &
                                     (is.na(HT_i) |is.na(HT_j)) ~ 'HSP',
                                   relation == "POP" ~ 'POP',
                                   relation == "FSP" ~ 'FSP')) %>%
  dplyr::mutate(River_year_i = paste0(RIVER_i, "_", Cohort_i )) %>%
  dplyr::mutate(River_year_j = paste0(RIVER_j, "_", Cohort_j )) %>%
  dplyr::select(iname, jname, River_year_i, River_year_j,
                RIVER_i, RIVER_j, relation, relation2,
                Cohort_i, Cohort_j ) %>%
  dplyr::filter(!is.na(Cohort_i)) %>%
```

```
  dplyr::filter(!is.na(Cohort_j))

group_data <- NWdata %>%
  dplyr::group_by( River_year_i, River_year_j, relation2) %>%
  dplyr::count()
nodes <- unique(c(group_data$River_year_i, group_data$River_year_j))

network <- igraph::graph_from_data_frame(d = group_data, directed = TRUE)

vertices <- meta %>%
  dplyr::select(River, Cohort2) %>%
  dplyr::filter(!is.na(Cohort2)) %>%
  dplyr::group_by(River, Cohort2) %>%
  dplyr::count()
vertices$River <- as.character(vertices$River)
vertices$River[vertices$River == "Adelaide River"] <- "AR"
vertices$River[vertices$River == "Sampan Creek"] <- "SC"
vertices$River[vertices$River == "Wildman River"] <- "WR"
vertices$River[vertices$River == "West Alligator River"] <- "WAR"
vertices$River[vertices$River == "South Alligator River"] <- "SAR"
vertices$River[vertices$River == "East Alligator River"] <- "EAR"
vertices$label <- paste0(vertices$River, "_", vertices$Cohort2)
vertices <- vertices[, c(4,1,2,3)]

vertices$River <- factor(vertices$River,
                         levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))

vertices <- as_data_frame(vertices, what = "vertices")
network <- igraph::graph_from_data_frame(d = group_data, directed = TRUE,
                                         vertices = vertices )

layout <- ggraph::create_layout(network, layout = 'igraph',
                                circular = FALSE, algorithm = 'fr')

layout <- ggraph::create_layout(network, layout = "auto")
```

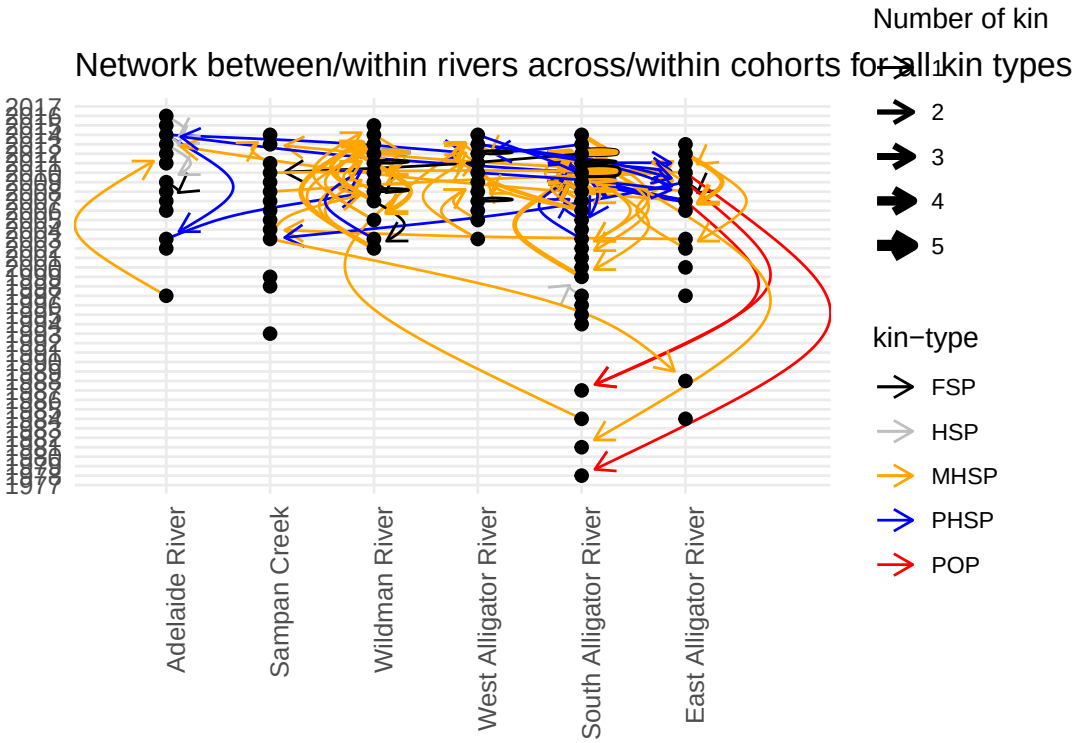
```
## Using "stress" as default layout
```

```
layout$y <- layout$Cohort2
layout$x <- factor(layout$River,
                   levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))


kin_network2 <- ggraph::ggraph(network,  layout = layout) +
  ggraph::geom_edge_arc(ggplot2::aes(width = n,
                            edge_colour = factor(relation2)),
                        arrow = arrow(length = unit(3, 'mm')),
                        end_cap = ggraph::circle(2, 'mm'),
                        alpha = 1, strength = 0.1) +
  ggraph::geom_edge_loop(ggplot2::aes(width = n, strength = 0.75,
                             edge_colour = factor(relation2)), position = "jitter") +
  ggraph::scale_edge_width(range = c(0.5, 2), name = "Number of kin") +
  ggraph::scale_edge_colour_manual(values =
                                     c("black", "grey", "orange","blue", "red"),
                                   name = "kin-type",
                                   aesthetics = "edge_colour",
                                   na.value = "grey50") +
```

```
  ggraph::geom_node_point(colour = "black",
                          size = 2) +
  ggplot2::scale_colour_manual(values = adegenet::funky(6), na.value = "grey50") +
  ggplot2::scale_y_continuous(limits = c(1978, 2016), n.breaks = 38, minor_breaks = NULL) +
  ggplot2::scale_x_discrete(
    labels = c("AR" = "Adelaide River",
               "SC" = "Sampan Creek",
               "WR" = "Wildman River",
               "WAR" = "West Alligator River",
               "SAR" = "South Alligator River",
               "EAR" = "East Alligator River")) +
  ggplot2::ggtitle("Network between/within rivers across/within cohorts for all kin types") +
  ggplot2::theme_void() +
  ggplot2::theme_minimal() +
  ggplot2::theme(
    legend.position = "right",
    plot.margin = unit(rep(1,4), "cm"),
    axis.text.y = ggplot2::element_text(size = 10, angle = 0, hjust = 0),
    axis.text.x = ggplot2::element_text(size = 10, angle = 90, hjust = 1),
    axis.title.x = ggplot2::element_blank(),
    axis.title.y = ggplot2::element_blank()
  )
```

```
## Warning: Ignoring `graph` as layout is already calculated
## i Pass the calculated layout to the `graph` argument to silence this warning
```

```
print(kin_network2)
```




```
ggplot2::ggsave(filename = "0.Kinship_network_all_relationships.png",
                bg = "white", plot = kin_network2, device = "png",
                width = 40, height = 30, units = "cm")
```

10.3.1 Direct contemporary dispersal

10.3.1.1 Parent-offspring pairs Adult dispersal was directly observed (i.e. by sampling the adult that moved) from parent-offspring pairs (POPs) that were distributed between sampling locations, where we assumed that the oldest individual (i.e. parent) dispersed.

Results

- 4 POPs
- All parents are male (fathers)
- All POPs have different haplotype
- All POPs have a different cohort
- All POPs were caught independently (> 14 days apart)
- All 4 POPs occurred across rivers. We assume the father moved, since the pup is < 1000 mm

```
sum(pwdata$Cohort_overlap_relaxed[pwdata$relation == "POP"],na.rm = TRUE) ## 0
```

```
## [1] 0
```

```
sum(pwdata$relation == "POP" & pwdata$HT_i == pwdata$HT_j,na.rm = TRUE) ## 0
```

```
## [1] 0
```

```
sum(pwdata$relation == "POP" & pwdata$RIVER_i != pwdata$RIVER_j,na.rm = TRUE) ## 4
```

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

```
sum(independent_catch[pwdata$relation == "POP"]) ## all are caught independently
```

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

```
knitr::kable(pwdata[pwdata$relation == "POP",-c(1:5,10,11,14,25)],
             format = "latex") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
935	1540	F	M	EAR	SAR	HT22	HT13	0.01639	0.51485	2009	2010	2010	NA	1978	1992
1155	1520	F	M	EAR	SAR	HT25	HT14	0.40984	0.02970	2006	2007	2007	NA	1987	1994
1025	1520	M	M	EAR	SAR	HT25	HT14	0.40984	0.02970	2008	2009	2009	NA	1987	1994
1580	735	M	M	WR	SAR	HT07	HT06	0.02222	0.37624	NA	NA	1983	2011	2012	2012

10.3.1.2 Full-sibling pairs The occurrence of cross-river full-sibling pairs (FSPs) was used to examine the incidence of juvenile dispersal. These are expected to be very low if dispersal only occurs in adults. We also assumed minimal juvenile dispersal based on the close association to nursery habitat, limited linear extent of river occupancy and the salinity preference of juvenile euryhaline sharks (Lyon et al. 2017; Pillans et al. 2020; Pillans et al. 2009).

Results

- 34 FSPs
- All FSPs have the same haplotype
- All FSP were juveniles (< 1410 mm)
- 10 FSPs were caught within 14 days (not independent)
- One FSP from Wildman River did not have overlapping cohorts. This was the FSP with the largest body size (1240-1385 mm TL). This could be explained by the inaccuracy of the growth function to assign large individuals to a cohort. Conversely, the small population size in Wildman River (see Bravington et al. 2019) could increase the probability that the same female and male mated in different years; thus yielding a cross-cohort FSP. Sperm storage is another likely explanation for this observation.
- One FSP occurred across river. We assume the largest individual moved (female = 1085 mm)
  - But 6 FSP had individuals > 1085 mm and did NOT move; hence, we assume juveniles (< 1410 mm) generally do not move between rivers

```
sum(pwdata$relation == "FSP" & pwdata$HT_i == pwdata$HT_j) ## all the same haplotypes
```

```
## [1] 34
```

```
sum(!independent_catch[pwdata$relation == "FSP"]) ## 10 FSPs caught within 14 days
```

```
## [1] 10
```

```
knitr::kable(pwdata[!pwdata$Cohort_overlap_relaxed & pwdata$relation == "FSP",
                    -c(1:5,10,11,14,25)],
             format = "latex", row.names = FALSE) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape() ## One FSP did NOT have overlapping cohorts
```

LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
1240	1385	F	F	WR	WR	HT13	HT13	0.68889	0.68889	2006	2007	2008	2000	2002	2004

```
knitr::kable(pwdata[pwdata$relation == "FSP" & pwdata$RIVER_i != pwdata$RIVER_j,
                    -c(1:5,10,11,14,25)],
             format = "latex") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape() ## 1 cross-river FSP
```

	LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
5	670	1085	M	F	SAR	SC	HT06	HT06	0.37624	0.13333	2012	2012	2012	2009	2010	2011

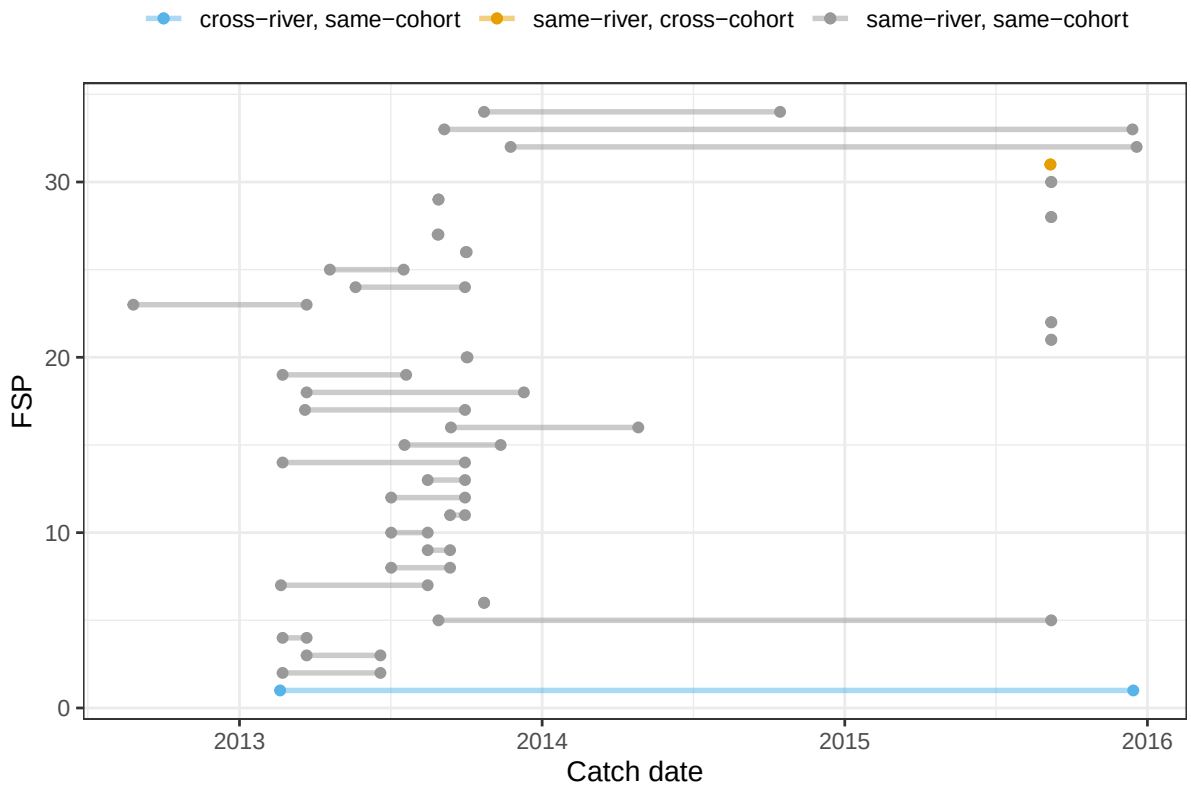
```r
sum(pwdata$relation == "FSP" & pwdata$RIVER_i == pwdata$RIVER_j &
      (pwdata$LEN_i > 1085 | pwdata$LEN_j > 1085))  ## 6 FSP larger than 1085 mm TL
```

```
## [1] 6
```

```r
knitr::kable(pwdata[pwdata$relation == "FSP",-c(1:5,10,11,14,25)],
             format = "latex", row.names = FALSE) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

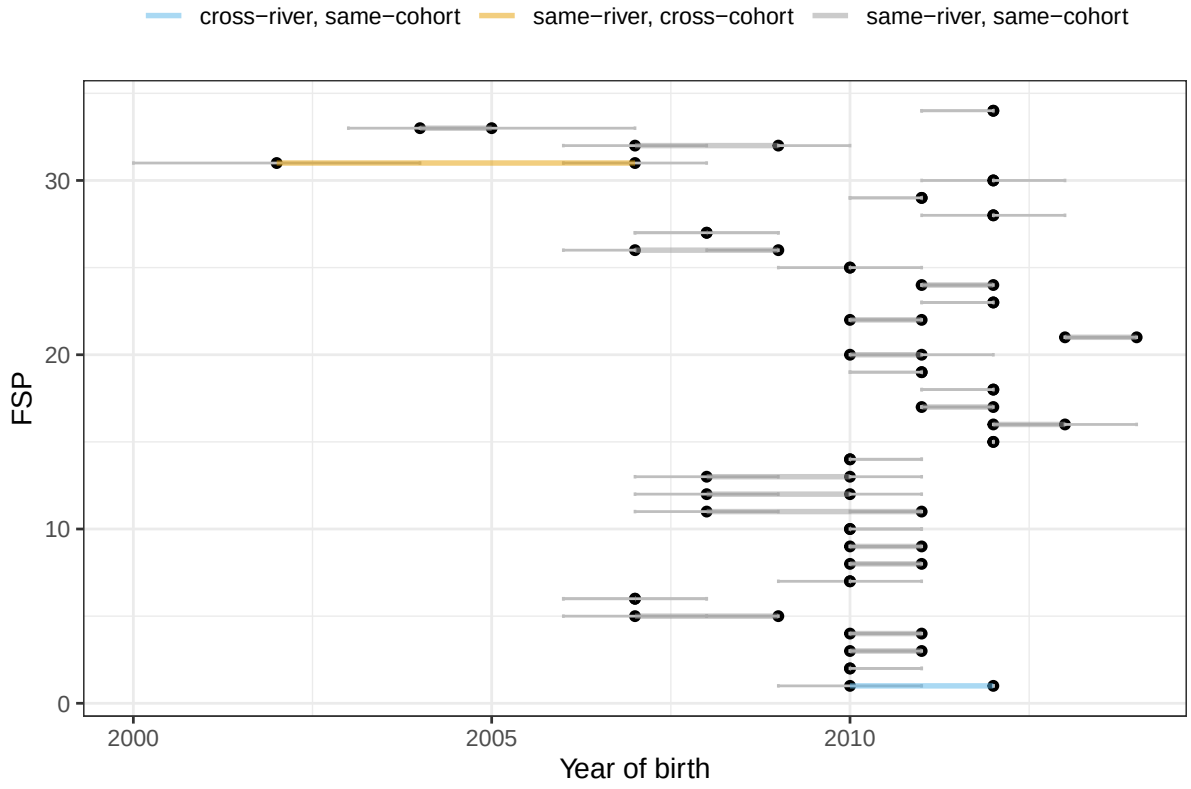
LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
670	1085	M	F	SAR	SC	HT06	HT06	0.37624	0.13333	2012	2012	2012	2009	2010	2011
825	870	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2010	2010	2011	2010	2010	2011
795	870	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2010	2011	2011	2010	2010	2011
795	825	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2010	2011	2011	2010	2010	2011
1005	1243	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2008	2009	2009	2006	2007	2008
1140	1140	M	M	WAR	WAR	HT13	HT13	0.92683	0.92683	2006	2007	2008	2006	2007	2008
925	837	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2009	2010	2010	2010	2010	2011
880	865	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2011	2011
880	865	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2011	2011
880	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2010	2011
1070	865	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2011	2011
1070	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2010	2011
1070	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2010	2011
895	827	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2010	2011
750	690	F	M	SAR	SAR	HT13	HT13	0.51485	0.51485	2012	2012	2012	2012	2012	2012
740	570	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2012	2012	2013	2013	2013	2014
825	695	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2011	2011	2012	2012
765	695	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2011	2012	2012
842	795	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2010	2011	2011
805	915	M	F	EAR	EAR	HT13	HT13	0.27869	0.27869	2011	2011	2012	2010	2010	2011
780	740	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2013	2013	2014	2013	2014	2014
1060	1010	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2010	2010	2011	2010	2011	2011
695	648	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2011	2012	2012
745	740	M	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2012	2011	2012	2012
905	885	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2009	2010	2010	2010	2010	2011
1025	1155	M	F	EAR	EAR	HT25	HT25	0.40984	0.40984	2008	2009	2009	2006	2007	2007
1065	1065	F	F	WR	WR	HT22	HT22	0.20000	0.20000	2007	2008	2009	2007	2008	2009
915	920	M	F	WR	WR	HT22	HT22	0.20000	0.20000	2012	2012	2013	2011	2012	2012
840	870	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2010	2011	2011	2010	2011	2011
900	950	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2012	2012	2013	2011	2012	2012
1240	1385	F	F	WR	WR	HT13	HT13	0.68889	0.68889	2006	2007	2008	2000	2002	2004
1000	1225	M	F	AR	AR	HT06	HT06	0.52000	0.52000	2009	2009	2010	2006	2007	2008
1260	1304	M	M	SC	SC	HT25	HT25	0.23333	0.23333	2003	2004	2005	2004	2005	2007
770	855	F	F	WAR	WAR	HT13	HT13	0.92683	0.92683	2011	2012	2012	2011	2012	2012

```
## catch date
dat <- pwdata[pwdata$relation == "FSP",]
FSP_legend <- rep("same-river, same-cohort", nrow(dat))
FSP_legend[!dat$Cohort_overlap_relaxed] <- "same-river, cross-cohort"
FSP_legend[dat$RIVER_i != dat$RIVER_j] <- "cross-river, same-cohort"

data <- data.frame(y = 1:nrow(dat),
                   x1 = pmin(dat$DATE_i, dat$DATE_j),
                   x2 = pmax(dat$DATE_i, dat$DATE_j),
                   FSP_legend = FSP_legend)
ggplot2::ggplot(data = data, aes(colour = FSP_legend)) +
  ggplot2::geom_point(ggplot2::aes(x = x1, y = y)) +
  ggplot2::geom_point(ggplot2::aes(x = x2, y = y)) +
  ggplot2::geom_segment(mapping = ggplot2::aes(x = x1, y = y, xend = x2, yend = y,
                                               colour = FSP_legend),
                        alpha = 0.5, size = 1, lineend = "butt",linejoin = "round") +
  ggplot2::scale_color_manual(values = c("#56B4E9","#E69F00","#999999")) +
  ggplot2::xlab("Catch date") +
  ggplot2::ylab("FSP") +
  ggplot2::theme_bw() +
  ggplot2::theme(legend.position = "top", legend.title = element_blank())
```




```
## cohort
data <- data.frame(y = 1:nrow(dat),
                   x1 = pmin(dat$Cohort_i, dat$Cohort_j),
                   x1_min = pmin(dat$Cohort_i_lower, dat$Cohort_j_lower),
                   x1_max = pmin(dat$Cohort_i_upper, dat$Cohort_j_upper),
                   x2 = pmax(dat$Cohort_i, dat$Cohort_j),
                   x2_min = pmax(dat$Cohort_i_lower, dat$Cohort_j_lower),
                   x2_max = pmax(dat$Cohort_i_upper, dat$Cohort_j_upper),
                   FSP_legend = FSP_legend)
```

```
ggplot2::ggplot(data = data, aes(olour = FSP_legend)) +
  ggplot2::geom_point(ggplot2::aes(x = x1, y = y)) +
  ggplot2::geom_errorbarh(ggplot2::aes(xmin = x1_min,xmax = x1_max, y = y,
                                       height = .2), colour = "grey") +
  ggplot2::geom_point(ggplot2::aes(x = x2, y = y)) +
  ggplot2::geom_errorbarh(ggplot2::aes(xmin = x2_min,xmax = x2_max, y = y,
                                       height = .2), colour = "grey") +
  ggplot2::geom_segment(mapping = ggplot2::aes(x = x1, y = y, xend = x2,
                                               yend = y,colour = FSP_legend),
                        alpha = 0.5, size = 1, lineend = "butt",linejoin = "round") +
  ggplot2::scale_color_manual(values = c("#56B4E9","#E69F00","#999999")) +
  ggplot2::xlab("Year of birth") +
  ggplot2::ylab("FSP") +
  ggplot2::theme_bw() +
  ggplot2::theme(legend.position = "top", legend.title = element_blank())
```



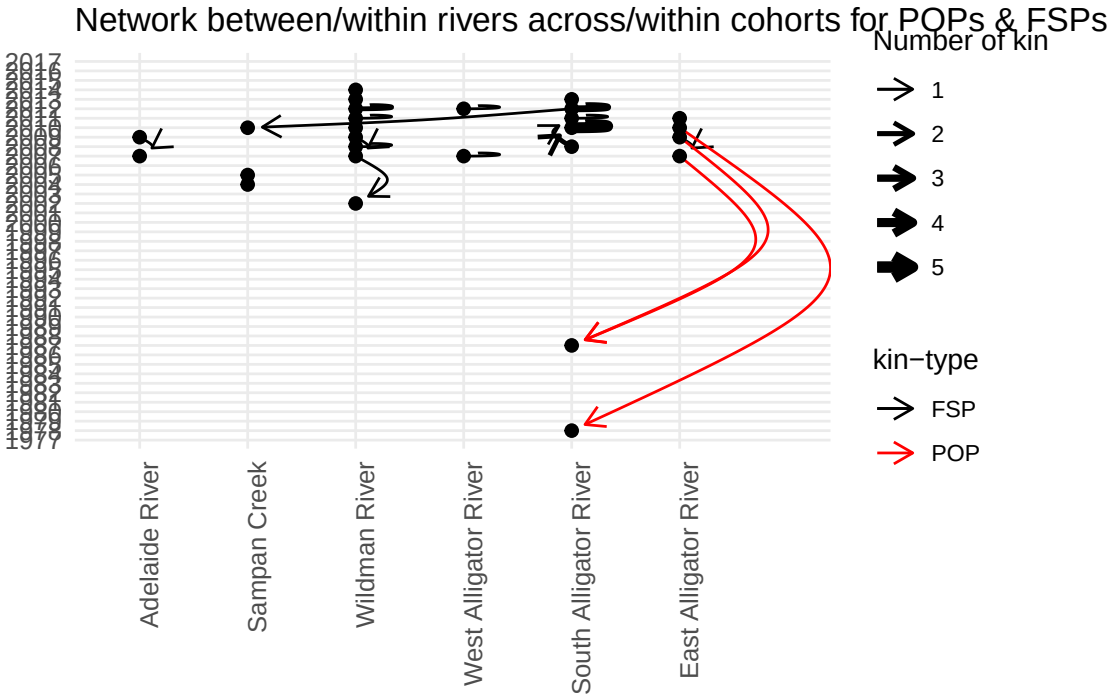

```
data <- pwdata
NWdata <- data %>%
  dplyr::filter(relation %in% c("POP", "FSP")) %>%
  dplyr::mutate(River_year_i = paste0(RIVER_i, "_", Cohort_i )) %>%
  dplyr::mutate(River_year_j = paste0(RIVER_j, "_", Cohort_j )) %>%
  dplyr::select(iname, jname, River_year_i, River_year_j,
                RIVER_i, RIVER_j, relation, relation,
                Cohort_i, Cohort_j ) %>%
  dplyr::filter(!is.na(Cohort_i)) %>%
  dplyr::filter(!is.na(Cohort_j))

group_data <- NWdata %>%
  dplyr::group_by( River_year_i, River_year_j, relation) %>%
  dplyr::count()
nodes <- unique(c(group_data$River_year_i, group_data$River_year_j))
```

```
network <- igraph::graph_from_data_frame(d = group_data, directed = TRUE)


pwPOP <- data %>%
  dplyr::filter(relation %in% c("POP", "FSP"))

vertices <- meta %>%
  dplyr::filter(id %in% c(pwPOP$iname,pwPOP$jname)) %>%
  dplyr::select(River, Cohort2) %>%
  dplyr::filter(!is.na(Cohort2)) %>%
  dplyr::filter(!is.na(Cohort2)) %>%
  dplyr::group_by(River, Cohort2) %>%
  dplyr::count()
vertices$River <- as.character(vertices$River)
vertices$River[vertices$River == "Adelaide River"] <- "AR"
vertices$River[vertices$River == "Sampan Creek"] <- "SC"
vertices$River[vertices$River == "Wildman River"] <- "WR"
vertices$River[vertices$River == "West Alligator River"] <- "WAR"
vertices$River[vertices$River == "South Alligator River"] <- "SAR"
vertices$River[vertices$River == "East Alligator River"] <- "EAR"
vertices$label <- paste0(vertices$River, "_", vertices$Cohort2)
vertices <- vertices[, c(4,1,2,3)]

vertices$River <- factor(vertices$River,
                         levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))

vertices <- as_data_frame(vertices, what = "vertices")
network <- igraph::graph_from_data_frame(d = group_data, directed = TRUE,
                                         vertices = vertices )


layout <- ggraph::create_layout(network, layout = 'igraph',
                                circular = FALSE, algorithm = 'fr')
layout$y <- layout$Cohort2
layout$x <- factor(layout$River,
                   levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))


kin_network3 <- ggraph::ggraph(network,  layout = layout) +
  ggraph::geom_edge_arc(aes(width = n,
                            edge_colour = factor(relation)),
                        arrow = arrow(length = unit(3, 'mm')),
                        end_cap = ggraph::circle(2, 'mm'),
                        alpha = 1, strength = 0.1) +
  ggraph::geom_edge_loop(aes(width = n, strength = 0.75,
                             edge_colour = factor(relation)),
                         end_cap = ggraph::circle(2, 'mm'),) +
  ggraph::scale_edge_width(range = c(0.5, 2), name = "Number of kin") +
  ggraph::scale_edge_colour_manual(values =
                                     c("black", "red"),
                                   name = "kin-type",
                                   aesthetics = "edge_colour",
                                   na.value = "grey50") +

  ggraph::geom_node_point(colour = "black",
                          size = 2) +
  # ggraph::geom_node_text( aes(label = Cohort2), repel = TRUE,
```

```
  #                           size = 5, color = "black") +
  # ggplot2::scale_colour_manual(values = colours.6, na.value = "grey50") +
  ggplot2::scale_y_continuous(limits = c(1978, 2016), n.breaks = 36,
                              minor_breaks = NULL) +
  ggplot2::scale_x_discrete(
    labels = c("AR" = "Adelaide River",
               "SC" = "Sampan Creek",
               "WR" = "Wildman River",
               "WAR" = "West Alligator River",
               "SAR" = "South Alligator River",
               "EAR" = "East Alligator River")) +
  ggplot2::ggtitle("Network between/within rivers across/within cohorts for POPs & FSPs") +
  ggplot2::theme_void() +
  ggplot2::theme_minimal() +
  ggplot2::theme(
    legend.position = "right",
    plot.margin = unit(rep(1,4), "cm"),
    axis.text.y = element_text(size = 10, angle = 0, hjust = 0),
    axis.text.x = element_text(size = 10, angle = 90, hjust = 1),
    axis.title.x = element_blank(),
    axis.title.y = element_blank()
  )
```

```
## Warning: Ignoring `graph` as layout is already calculated
## i Pass the calculated layout to the `graph` argument to silence this warning
```

```
print(kin_network3)
```




```
ggplot2::ggsave(filename = "0.Kinship_network_POP-FSP2.png",
                bg = "white", plot = kin_network3, device = "png",
                width = 40, height = 30, units = "cm")
```

10.3.2 Indirect contemporary dispersal

10.3.2.1 Half-sibling pairs Sex-specific adult dispersal was inferred with an indirect approach (i.e. without sampling adults) by comparing the mtDNA haplotypes of half-sibling pairs (HSPs). Cross-cohort (i.e. born in different years), cross-river HSPs inform if parents moved between locations between breeding seasons. By comparing the mtDNA haplotypes of each pair (h1==h2 or h1!=h2), we can infer which parent, the mother or the father, is more likely to have moved between sampling locations.

ASSUMPTIONS

- Juvenile can be assigned to their cohort (year of birth). We are relatively confident we can assign juveniles to the same cohort, since all, except one, FSP were assigned to the same cohort.
- Siblings with the same haplotype share the same mother. Given that we only have 26 haplotype and three of those occur at high frequency, this assumption is likely not true. This leads to over-estimating female dispersal, and under-estimating male dispersal events.
- All potential surrounding *G. garricki* habitats were sampled. If they were not we would be under-estimating dispersive breeding events. Most rivers, creeks and estuaries in Van Diemen Gulf were surveyed, thus have confidence this assumption is met in Van Diemen Gulf.

Results

- 139 HSPs
- 54 HSPs have the same cohort and 77 HSPs have a different cohort (8 NAs)
- 101 HSPs had the same haplotype and 26 had a different haplotype (12 NAs)
  - 57 HSPs with the same HT had a high probability to be paternally related:
    - 3 cross-cohort, cross-river HSPs
    - 27 cross-cohort, same-river HSPs
    - 25 same-cohort, same-river HSPs
    - 0 same-cohort, cross-river HSPs
    - 1 missing-cohort, cross-river HSPs
    - 1 missing-cohort, same-river HSPs
- 40 HSPs were caught within 14 days (not independent), but only 12 of those were from the same cohort

```
## Calculate probability of paternal and maternal relationship, based on HT frequency
pwdata_prob <- pwdata
pwdata_prob <- within( pwdata_prob, {
  Pr_if_mat <- ifelse( HT_i  == HT_j, pmin(pwdata_prob$HTfreq_i, pwdata_prob$HTfreq_j), 0)
  Pr_if_pat <- pwdata_prob$HTfreq_i * pwdata_prob$HTfreq_j
})
pwdata_prob <- pwdata_prob[order(pwdata_prob$Pr_if_pat, decreasing = TRUE),]

## cohort
sum(pwdata$Cohort_overlap_relaxed[pwdata$relation == "HSP"], na.rm = TRUE) ## 54
```

```
## [1] 54
```

```
sum(!pwdata$Cohort_overlap_relaxed[pwdata$relation == "HSP"], na.rm = TRUE) ## 77
```

```
## [1] 77
```

```
sum(is.na(pwdata$Cohort_overlap_relaxed[pwdata$relation == "HSP"])) ## 8 pairs with missing cohort i
```

```
## [1] 8
```

```
## Haplotype
sum(pwdata$relation == "HSP" & pwdata$HT_i == pwdata$HT_j, na.rm = TRUE) ## 101
```

```
## [1] 101
```

```
sum(pwdata$relation == "HSP" & pwdata$HT_i != pwdata$HT_j, na.rm = TRUE) ## 26
```

```
## [1] 26
```

```
sum(is.na(pwdata$relation == "HSP" & pwdata$HT_i == pwdata$HT_j)) ## 12 HSPs with missing haplotype
```

```
## [1] 12
```

```
sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
      pwdata_prob$HT_i == pwdata_prob$HT_j, na.rm = TRUE) ## 57 HSPs with higher probability to be p
```

```
## [1] 57
```

```
  sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        !pwdata_prob$Cohort_overlap_relaxed &
        pwdata_prob$HT_i == pwdata_prob$HT_j
      , na.rm = TRUE) ## 30 cross-cohort HSPs with higher probability to be paternally related
```

```
## [1] 30
```

```
  sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        !pwdata_prob$Cohort_overlap_relaxed &
        pwdata_prob$RIVER_i != pwdata_prob$RIVER_j &
        pwdata_prob$HT_i == pwdata_prob$HT_j
        , na.rm = TRUE) ## 3 cross-cohort, cross-river HSPs with higher probability to be paternally
```

```
## [1] 3
```

```
  sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        !pwdata_prob$Cohort_overlap_relaxed &
        pwdata_prob$RIVER_i == pwdata_prob$RIVER_j &
        pwdata_prob$HT_i == pwdata_prob$HT_j
      , na.rm = TRUE) ## 27 cross-cohort, same-river HSPs with higher probability to be paternally r
```

```
## [1] 27
```

```
   sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        pwdata_prob$Cohort_overlap_relaxed &
        pwdata_prob$RIVER_i == pwdata_prob$RIVER_j &
        pwdata_prob$HT_i == pwdata_prob$HT_j
      , na.rm = TRUE) ## 25 same-cohort, same-river HSPs with higher probability to be paternally re
```

```
## [1] 25
```

```
   sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        pwdata_prob$Cohort_overlap_relaxed &
        pwdata_prob$RIVER_i != pwdata_prob$RIVER_j &
        pwdata_prob$HT_i == pwdata_prob$HT_j
      , na.rm = TRUE) ## 0 same-cohort, cross-river HSPs with higher probability to be paternally re
```

```
## [1] 0
```

```
      sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        is.na(pwdata_prob$Cohort_overlap_relaxed) &
        pwdata_prob$RIVER_i != pwdata_prob$RIVER_j &
        pwdata_prob$HT_i == pwdata_prob$HT_j
      , na.rm = TRUE) ## 1 missing-cohort, cross-river HSPs with higher probability to be paternally
```

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

```
      sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        is.na(pwdata_prob$Cohort_overlap_relaxed) &
        pwdata_prob$RIVER_i == pwdata_prob$RIVER_j &
```

```
        pwdata_prob$HT_i == pwdata_prob$HT_j
      , na.rm = TRUE) ## 1 missing-cohort, same-river HSPs with higher probability to be paternally
```

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

```
## Independence
sum(!independent_catch[pwdata$relation == "HSP"]) ## 40 HSPs were within 14 days
```

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

```
sum(!independent_catch[pwdata$relation == "HSP" &
    pwdata$Cohort_overlap_relaxed], na.rm = TRUE) ## 12 of those had overlapping cohorts
```

```
## [1] 12
```

```
kableExtra::kbl(pwdata_prob[pwdata_prob$relation == "HSP",-c(1:5,10,11,14,25)],
             format = "latex", longtable = TRUE, row.names = FALSE) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            font_size = 7) %>%
  kableExtra::landscape()
```

LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
860	663	M	M	WAR	WAR	HT13	HT13	0.92683	0.92683	2011	2012	2012	2013	2014	2014
860	625	F	F	WAR	WAR	HT13	HT13	0.92683	0.92683	2010	2011	2011	2013	2013	2013
1025	1170	M	F	WAR	WAR	HT13	HT13	0.92683	0.92683	2009	2010	2010	2007	2007	2008
1330	1035	M	F	WAR	WAR	HT13	HT13	0.92683	0.92683	2001	2003	2004	2008	2009	2009
1320	775	F	F	WAR	WAR	HT13	HT13	0.92683	0.92683	2001	2003	2004	2011	2012	2012
1250	1070	M	M	WAR	WAR	HT13	HT13	0.92683	0.92683	2004	2005	2006	2008	2009	2010
615	925	F	M	WAR	WAR	HT13	HT13	0.92683	0.92683	2013	2013	2013	2011	2011	2012
695	855	F	F	AR	WAR	HT06	HT13	0.52000	0.92683	2014	2014	2015	2011	2012	2012
1070	782	M	M	SAR	WAR	HT13	HT13	0.51485	0.92683	2006	2006	2007	2012	2013	2013
1070	840	M	M	SAR	WAR	HT13	HT13	0.51485	0.92683	2006	2006	2007	2012	2012	2012
1210	900	F	F	SAR	WAR	HT13	HT13	0.51485	0.92683	2005	2006	2007	2010	2010	2011
1530	660	F	M	SAR	WAR	HT13	HT13	0.51485	0.92683	NA	1984	1993	2013	2014	2014
870	1230	M	F	WR	WR	HT13	HT13	0.68889	0.68889	2010	2011	2011	2004	2005	2006
840	1230	F	F	WR	WR	HT13	HT13	0.68889	0.68889	2010	2011	2011	2004	2005	2006
870	765	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2010	2011	2011	2013	2014	2014
840	765	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2010	2011	2011	2013	2014	2014
855	675	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2012	2013	2013	2012	2012	2013
1385	740	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2000	2002	2004	2013	2014	2014
1240	740	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2006	2007	2008	2013	2014	2014
1240	930	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2006	2007	2008	2011	2012	2012
1385	780	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2000	2002	2004	2013	2013	2014
1240	780	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2006	2007	2008	2013	2013	2014
1385	1060	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2000	2002	2004	2010	2010	2011
1385	1010	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2000	2002	2004	2010	2011	2011
930	740	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2011	2012	2012	2013	2014	2014
930	932	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2011	2012	2012	2011	2012	2012
1385	730	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2000	2002	2004	2014	2014	2014
1240	730	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2006	2007	2008	2014	2014	2014
1305	870	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2004	2005	2006	2012	2013	2013
1305	607	F	F	WR	WR	HT13	HT13	0.68889	0.68889	2004	2005	2006	2015	2015	2015
625	1055	F	M	WAR	EAR	HT13	HT25	0.92683	0.40984	2013	2013	2013	2008	2008	2009
782	1025	M	F	WAR	EAR	HT13	HT25	0.92683	0.40984	2012	2013	2013	2008	2009	2009
705	840	F	M	SAR	WAR	HT06	HT13	0.37624	0.92683	2012	2013	2013	2012	2012	2012
1450	970	M	M	AR	AR	HT06	HT06	0.52000	0.52000	1993	1997	2000	2011	2012	2012
1420	837	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	1996	1999	2001	2010	2010	2011
910	837	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2010	2011
1420	925	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	1996	1999	2001	2009	2010	2010
910	925	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2009	2010	2010
910	1420	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	1996	1999	2001
695	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2012	2012	2012	2013	2013
825	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2011	2012	2013	2013
840	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2012	2013	2013
1005	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2008	2009	2009	2012	2013	2013
840	1005	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2008	2009	2009
520	895	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2014	2014	2014	2010	2010	2011
520	827	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2014	2014	2014	2010	2010	2011

905	945	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2010	2009	2009	2010	
842	1030	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2008	2008	2009	
795	1030	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2008	2008	2009	
740	950	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2012	2012	2009	2009	2010	
745	950	M	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2012	2009	2009	2010	
1080	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2010	2011	
1080	865	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2011	2011	
1080	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2010	2011	
1070	1080	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2007	2008	2009	
960	950	M	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2009	2009	2010	2009	2009	2010	
750	1536	F	M	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2012	2012	NA	1981	1992	
910	670	F	M	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2012	2013	2013	
750	910	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2012	2012	2010	2010	2011	
600	1040	F	M	SAR	SAR	HT13	HT13	0.51485	0.51485	2012	2013	2013	2008	2008	2009	
795	585	M	M	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2012	2013	2013	2013	
880	837	F	F	SAR	SAR	HT06	HT13	0.37624	0.51485	2010	2011	2011	2010	2010	2011	
880	925	F	F	SAR	SAR	HT06	HT13	0.37624	0.51485	2010	2011	2011	2009	2010	2010	
827	700	F	M	SAR	SAR	HT13	HT06	0.51485	0.37624	2010	2010	2011	2012	2012	2012	
895	700	F	M	SAR	SAR	HT13	HT06	0.51485	0.37624	2010	2010	2011	2012	2012	2012	
685	905	M	F	SAR	SAR	HT06	HT13	0.37624	0.51485	2013	2013	2013	2010	2010	2010	
1165	1040	F	M	SAR	SAR	HT06	HT13	0.37624	0.51485	2004	2005	2006	2008	2008	2009	
1000	775	F	M	EAR	EAR	HT25	HT25	0.40984	0.40984	2008	2009	2010	2011	2012	2012	
745	1215	M	F	EAR	EAR	HT25	HT25	0.40984	0.40984	2011	2012	2012	2005	2006	2006	
780	1200	F	M	EAR	EAR	HT25	HT25	0.40984	0.40984	2011	2012	2012	2005	2006	2007	
850	795	M	M	SC	SC	HT13	HT13	0.40000	0.40000	2010	2011	2011	2013	2014	2014	
695	560	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2013	2013	2013	
1030	648	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2008	2009	2009	2011	2012	2012	
560	648	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2013	2013	2013	2011	2012	2012	
695	1030	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2008	2009	2009	
NA	1150	NA	F	SAR	SAR	HT06	HT06	0.37624	0.37624	NA	NA	NA	2006	2006	2007	
1150	880	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2006	2006	2007	2010	2011	2011	
1130	700	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2007	2008	2008	2012	2012	2012	
705	695	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2012	2012	2012	2011	2012	2012	
765	705	M	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2012	2012	2012	
1135	830	M	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2006	2007	2008	2010	2011	2011	
830	685	M	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2010	2011	2011	2013	2013	2013	
1135	685	M	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2006	2007	2008	2013	2013	2013	
1670	1160	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	NA	NA	NA	2005	2006	2007	
1030	1160	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2008	2009	2009	2005	2006	2007	
1140	1160	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2006	2007	2008	2005	2006	2007	
1030	1140	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2008	2009	2009	2006	2007	2008	
730	1550	M	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2012	2012	2013	NA	NA	1990	
735	1350	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	1999	2001	2002	
1025	1100	M	M	WAR	EAR	HT13	HT06	0.92683	0.14754	2009	2010	2010	2007	2008	2008	
1064	935	M	M	SC	EAR	HT13	HT13	0.40000	0.27869	2010	2010	2011	2009	2010	2010	
1120	795	M	F	EAR	SAR	HT13	HT06	0.27869	0.37624	2007	2007	2008	2010	2011	2011	
1120	825	M	F	EAR	SAR	HT13	HT06	0.27869	0.37624	2007	2007	2008	2010	2010	2011	

1120	870	M	F	EAR	SAR	HT13	HT06	0.27869	0.37624	2007	2007	2008	2010	2010	2011	
815	795	M	F	SAR	EAR	HT06	HT13	0.37624	0.27869	2010	2011	2011	2011	2011	2012	
1365	1520	M	M	SC	EAR	HT25	HT25	0.23333	0.40984	2001	2003	2005	NA	1988	1994	
1315	1358	F	M	EAR	SC	HT25	HT25	0.40984	0.23333	2001	2003	2004	2002	2004	2005	
1135	880	M	F	EAR	AR	HT06	HT06	0.14754	0.52000	2006	2007	2008	2012	2013	2013	
1030	1050	M	M	EAR	EAR	HT25	HT06	0.40984	0.14754	2008	2009	2009	2008	2008	2009	
700	1385	M	M	AR	AR	HT13	HT06	0.08000	0.52000	2014	2014	2015	2000	2003	2004	
905	1065	F	F	WR	WR	HT22	HT22	0.20000	0.20000	2012	2012	2013	2007	2008	2009	
1065	905	F	F	WR	WR	HT22	HT22	0.20000	0.20000	2007	2008	2009	2012	2012	2013	
660	1065	M	M	WR	WR	HT22	HT22	0.20000	0.20000	2014	2014	2015	2008	2009	2010	
915	1065	M	M	WR	WR	HT22	HT22	0.20000	0.20000	2012	2012	2013	2008	2009	2010	
920	660	F	M	WR	WR	HT22	HT22	0.20000	0.20000	2011	2012	2012	2014	2014	2015	
915	660	M	M	WR	WR	HT22	HT22	0.20000	0.20000	2012	2012	2013	2014	2014	2015	
825	935	M	F	SAR	SAR	HT13	HT25	0.51485	0.06931	2010	2011	2011	2009	2010	2010	
1070	895	F	F	EAR	SAR	HT25	HT25	0.40984	0.06931	2007	2008	2009	2009	2010	2010	
663	1270	M	M	WAR	SAR	HT13	HT14	0.92683	0.02970	2013	2014	2014	2003	2004	2005	
830	965	M	F	SAR	EAR	HT06	HT07	0.37624	0.06557	2010	2011	2011	2009	2010	2010	
685	915	M	M	EAR	EAR	HT06	HT06	0.14754	0.14754	2012	2012	2013	2010	2010	2011	
1330	915	F	M	EAR	EAR	HT06	HT06	0.14754	0.14754	2001	2002	2004	2010	2010	2011	
685	1330	M	F	EAR	EAR	HT06	HT06	0.14754	0.14754	2012	2012	2013	2001	2002	2004	
1290	910	F	M	SAR	SAR	HT14	HT13	0.02970	0.51485	2002	2003	2004	2009	2010	2010	
1010	1365	M	M	EAR	SC	HT25	HT21	0.40984	0.03333	2008	2009	2010	2001	2003	2005	
1380	970	M	F	WR	WR	HT22	HT25	0.20000	0.04444	2000	2003	2004	2011	2011	2012	
1015	666	M	M	WAR	WAR	HT06	HT06	0.07317	0.07317	2009	2010	2010	2013	2014	2014	
930	615	F	M	SAR	SAR	HT22	HT13	0.00990	0.51485	2009	2010	2010	2013	2013	2013	
625	935	M	F	SAR	SAR	HT25	HT25	0.06931	0.06931	2013	2013	2013	2009	2010	2010	
935	1340	F	M	SAR	SAR	HT25	HT25	0.06931	0.06931	2009	2010	2010	2000	2002	2003	
625	1340	M	M	SAR	SAR	HT25	HT25	0.06931	0.06931	2013	2013	2013	2000	2002	2003	
970	760	F	F	SAR	SAR	HT25	HT25	0.06931	0.06931	2008	2009	2009	2011	2012	2012	
965	810	F	M	EAR	SC	HT07	HT07	0.06557	0.06667	2009	2010	2010	2013	2013	2014	
1330	1055	M	F	AR	WR	HT25	HT21	0.16000	0.02222	2001	2003	2004	2008	2008	2009	
895	720	F	M	SAR	AR	HT25	HT24	0.06931	0.04000	2009	2010	2010	2014	2014	2015	
1070	935	M	F	SC	EAR	HT22	HT22	0.10000	0.01639	2007	2008	2009	2009	2010	2010	
1365	1055	M	F	SC	WR	HT21	HT21	0.03333	0.02222	2001	2003	2005	2008	2008	2009	
775	800	M	M	SAR	SAR	NA	HT13	NA	0.51485	2009	2010	2010	2011	2011	2011	
1420	1420	M	M	SAR	SAR	NA	HT25	NA	0.06931	1994	1997	1999	1996	1999	2001	
820	1065	F	M	WR	WR	NA	HT22	NA	0.20000	2012	2012	2012	2008	2009	2010	
820	660	F	M	WR	WR	NA	HT22	NA	0.20000	2012	2012	2012	2014	2014	2015	
820	920	F	F	WR	WR	NA	HT22	NA	0.20000	2012	2012	2012	2011	2012	2012	
915	820	M	F	WR	WR	HT22	NA	0.20000	NA	2012	2012	2013	2012	2012	2012	
555	675	F	M	AR	AR	NA	HT06	NA	0.52000	2015	2016	2016	2014	2015	2015	
1575	755	M	M	AR	AR	HT06	NA	0.52000	NA	NA	NA	1984	2012	2013	2013	
1575	555	M	F	AR	AR	HT06	NA	0.52000	NA	NA	NA	1984	2015	2016	2016	
555	755	F	M	AR	AR	NA	NA	NA	NA	2015	2016	2016	2012	2013	2013	
745	1120	M	F	AR	AR	HT10	NA	0.16000	NA	2013	2013	2013	2009	2009	2010	
735	970	F	M	AR	AR	NA	HT06	NA	0.52000	2014	2014	2014	2011	2012	2012	

10.3.2.1.1 Same-cohort, cross-river HSPs Results

- 8 same-cohort HSPs occur across rivers (4 have the same HT and 4 have a different HT).

Discussion

Four same-cohort, cross-river HSPs are paternally-related, indicating the fathers reproduced with multiple females within one breeding season. Since little is known about the breeding grounds of *G. garricki*, these paternally-related HSPs could suggest that the fathers moved between sampling locations within a season or that mothers from different sampling locations aggregated at the same breeding ground. The four same-cohort, cross-river HSPs with the same haplotype prove more difficult to explain, as it seems implausible that the mother would have moved during pupping. The simplest explanation is that they are actually paternally-related, given their high haplotype frequencies, or they are cross-cohort HSPs.

```
## same-cohort, cross-river
sum(pwdata$relation == "HSP" &
      pwdata$RIVER_i != pwdata$RIVER_j &
      pwdata$Cohort_overlap_relaxed, na.rm = TRUE) ## 8
```

```
## [1] 8
```

```
knitr::kable(pwdata[pwdata$relation == "HSP" &
         pwdata$RIVER_i != pwdata$RIVER_j &
         pwdata$Cohort_overlap_relaxed &
         !is.na(pwdata$Cohort_overlap_relaxed),-c(1:5,10,11,14,25)],
         format = "latex") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

	LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
68	830	965	M	F	SAR	EAR	HT06	HT07	0.37624	0.06557	2010	2011	2011	2009	2010	2010
112	1064	935	M	M	SC	EAR	HT13	HT13	0.40000	0.27869	2010	2010	2011	2009	2010	2010
117	1070	935	M	F	SC	EAR	HT22	HT22	0.10000	0.01639	2007	2008	2009	2009	2010	2010
121	1025	1100	M	M	WAR	EAR	HT13	HT06	0.92683	0.14754	2009	2010	2010	2007	2008	2008
125	705	840	F	M	SAR	WAR	HT06	HT13	0.37624	0.92683	2012	2013	2013	2012	2012	2012
130	815	795	M	F	SAR	EAR	HT06	HT13	0.37624	0.27869	2010	2011	2011	2011	2011	2012
167	1070	895	F	F	EAR	SAR	HT25	HT25	0.40984	0.06931	2007	2008	2009	2009	2010	2010
173	1315	1358	F	M	EAR	SC	HT25	HT25	0.40984	0.23333	2001	2003	2004	2002	2004	2005

```
  ## same HT
  sum(pwdata$relation == "HSP" &
        pwdata$RIVER_i != pwdata$RIVER_j &
        pwdata$Cohort_overlap_relaxed &
        pwdata$HT_i == pwdata$HT_j, na.rm = TRUE) ## 4
```

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

```
  ## Different HT
  sum(pwdata$relation == "HSP" &
        pwdata$RIVER_i != pwdata$RIVER_j &
        pwdata$Cohort_overlap_relaxed &
        pwdata$HT_i != pwdata$HT_j, na.rm = TRUE) ## 4
```

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

10.3.2.1.2 Cross-cohort, cross-river HSPs Results

- 16 cross-cohort HSPs occurred across river (6 same HT, 10 different HT).
  - *NOTE: 3 of the 6 HSPs with the same haplotypes had HTs (HT06, HT13 and HT25) with a high haplotype frequency in the population*

CONCLUSION

- male dispersal estimates
  - Observed: 10 out of 16 cross-cohort HSP with the different haplotypes (i.e. paternal HSPs) were sampled from different river (i.e. six cross-river). = 62.5%.
  - Maximum: Three of the six cross-river maternal HSPs had a haplotype with high frequencies in the river. Therefore, they could be paternal HSPs. Hence the maximum male dispersal is 13/19 = 68.4%.
  - Minimum: if we assume the 27 cross-cohort, same-river HSPs with high haplotype frequencies are actually paternal related, and all six cross-river HSP with same haplotype are maternally related, then the minimum male dispersal is 10/43 = 23.2%.
- female dispersal estimates
  - Observed: 6 out of 56 cross-cohort HSP with the same haplotypes (i.e. maternal HSPs) were sampled from different river (i.e. six cross-river). = 10.7%.
  - Maximum: if we assume the 27 cross-cohort, same-river HSPs with high haplotype frequencies are actually paternal related, and all six cross-river HSP with same haplotype are maternally related, then the maximum male dispersal is 6/29 = 20.7%.
  - Minimum: Three of the six cross-river maternal HSPs had a haplotype with high frequencies in the river. Therefore, they could be paternal HSPs. Hence the maximum male dispersal is 3/56 = 5.4%

```
## cross-cohort, cross-river HSP
sum(pwdata$relation == "HSP" &
      pwdata$RIVER_i != pwdata$RIVER_j &
      !pwdata$Cohort_overlap_relaxed, na.rm = TRUE) #16
```

```
## [1] 16
```

```
knitr::kable(pwdata[pwdata$relation == "HSP" &
      pwdata$RIVER_i != pwdata$RIVER_j &
      !pwdata$Cohort_overlap_relaxed &
      !is.na(pwdata$Cohort_overlap_relaxed),-c(1:5,10,11,14,25)],
      format = "latex", row.names = FALSE) %>%
  kableExtra::kable_styling(full_width = FALSE,
```

```
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
1120	795	M	F	EAR	SAR	HT13	HT06	0.27869	0.37624	2007	2007	2008	2010	2011	2011
1120	825	M	F	EAR	SAR	HT13	HT06	0.27869	0.37624	2007	2007	2008	2010	2010	2011
1120	870	M	F	EAR	SAR	HT13	HT06	0.27869	0.37624	2007	2007	2008	2010	2010	2011
965	810	F	M	EAR	SC	HT07	HT07	0.06557	0.06667	2009	2010	2010	2013	2013	2014
663	1270	M	M	WAR	SAR	HT13	HT14	0.92683	0.02970	2013	2014	2014	2003	2004	2005
1330	1055	M	F	AR	WR	HT25	HT21	0.16000	0.02222	2001	2003	2004	2008	2008	2009
1010	1365	M	M	EAR	SC	HT25	HT21	0.40984	0.03333	2008	2009	2010	2001	2003	2005
1365	1055	M	F	SC	WR	HT21	HT21	0.03333	0.02222	2001	2003	2005	2008	2008	2009
625	1055	F	M	WAR	EAR	HT13	HT25	0.92683	0.40984	2013	2013	2013	2008	2008	2009
782	1025	M	F	WAR	EAR	HT13	HT25	0.92683	0.40984	2012	2013	2013	2008	2009	2009
1070	782	M	M	SAR	WAR	HT13	HT13	0.51485	0.92683	2006	2006	2007	2012	2013	2013
1070	840	M	M	SAR	WAR	HT13	HT13	0.51485	0.92683	2006	2006	2007	2012	2012	2012
1210	900	F	F	SAR	WAR	HT13	HT13	0.51485	0.92683	2005	2006	2007	2010	2010	2011
695	855	F	F	AR	WAR	HT06	HT13	0.52000	0.92683	2014	2014	2015	2011	2012	2012
895	720	F	M	SAR	AR	HT25	HT24	0.06931	0.04000	2009	2010	2010	2014	2014	2015
1135	880	M	F	EAR	AR	HT06	HT06	0.14754	0.52000	2006	2007	2008	2012	2013	2013

```
  ## same HT
  sum(pwdata$relation == "HSP" &
        pwdata$RIVER_i != pwdata$RIVER_j &
        !pwdata$Cohort_overlap_relaxed &
        pwdata$HT_i == pwdata$HT_j, na.rm = TRUE) ## 6
```

```
## [1] 6
```

```
  ## different HT
  sum(pwdata$relation == "HSP" &
        pwdata$RIVER_i != pwdata$RIVER_j &
        !pwdata$Cohort_overlap_relaxed &
        pwdata$HT_i != pwdata$HT_j, na.rm = TRUE) ## 10
```

```
## [1] 10
```

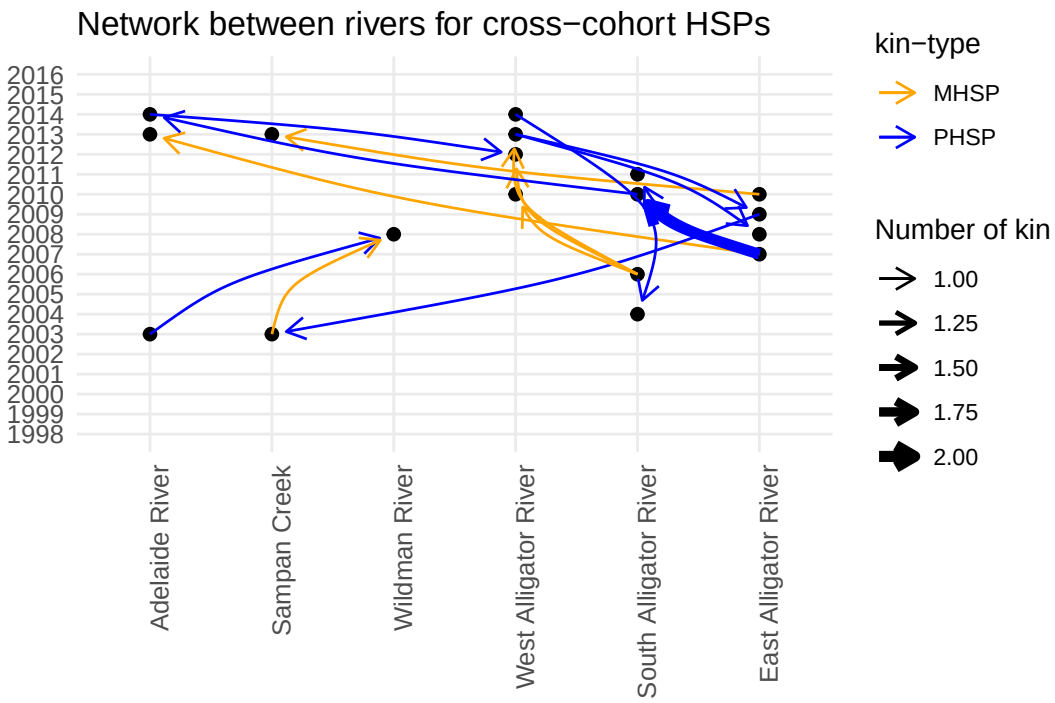
```
data <- pwdata
NWdata <- data %>%
  dplyr::filter(relation == "HSP") %>%
  dplyr::mutate(Cohort_gap = pmax(Cohort_i,Cohort_j) - pmin(Cohort_i,Cohort_j)) %>%
  dplyr::filter(!Cohort_overlap_relaxed) %>%
  dplyr::filter(RIVER_i != RIVER_j) %>%
  dplyr::mutate(relation2 =
                  dplyr::case_when(relation == "HSP" & HT_i == HT_j ~ 'MHSP',
                                   relation == "HSP" & HT_i != HT_j ~ 'PHSP',
                                   relation == "HSP" &
                                     (is.na(HT_i) |is.na(HT_j)) ~ 'HSP')) %>%
  dplyr::mutate(River_year_i = paste0(RIVER_i, "_", Cohort_i )) %>%
  dplyr::mutate(River_year_j = paste0(RIVER_j, "_", Cohort_j )) %>%
  dplyr::select(iname, jname, River_year_i, River_year_j,
                RIVER_i, RIVER_j, relation, relation2,
                Cohort_i, Cohort_j ) %>%
  dplyr::filter(!is.na(Cohort_i)) %>%
  dplyr::filter(!is.na(Cohort_j))

group_data <- NWdata %>%
  dplyr::group_by( River_year_i, River_year_j, relation2) %>%
  dplyr::count()
nodes <- unique(c(group_data$River_year_i, group_data$River_year_j))

pwPOP <- data %>%
  dplyr::filter(relation == "HSP") %>%
  dplyr::mutate(Cohort_gap = pmax(Cohort_i,Cohort_j) - pmin(Cohort_i,Cohort_j)) %>%
  dplyr::filter(!Cohort_overlap_relaxed) %>%
  dplyr::filter(RIVER_i != RIVER_j)

vertices <- meta %>%
  dplyr::filter(id %in% c(pwPOP$iname,pwPOP$jname)) %>%
  dplyr::select(River, Cohort2) %>%
  dplyr::filter(!is.na(Cohort2)) %>%
  dplyr::group_by(River, Cohort2) %>%
  dplyr::count()
vertices$River <- as.character(vertices$River)
vertices$River[vertices$River == "Adelaide River"] <- "AR"
vertices$River[vertices$River == "Sampan Creek"] <- "SC"
vertices$River[vertices$River == "Wildman River"] <- "WR"
vertices$River[vertices$River == "West Alligator River"] <- "WAR"
vertices$River[vertices$River == "South Alligator River"] <- "SAR"
vertices$River[vertices$River == "East Alligator River"] <- "EAR"
```

```
vertices$label <- paste0(vertices$River, "_", vertices$Cohort2)
vertices <- vertices[, c(4,1,2,3)]

vertices$River <- factor(vertices$River,
                         levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))
vertices <- as_data_frame(vertices, what = "vertices")
network <- igraph::graph_from_data_frame(d = group_data, directed = TRUE,
                                         vertices = vertices )

layout <- ggraph::create_layout(network, layout = 'igraph',
                                circular = FALSE, algorithm = 'fr')
layout$y <- layout$Cohort2
layout$River <- factor(layout$River,
                       levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))
layout$x <- factor(layout$River,
                   levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))


kin_network7 <- ggraph::ggraph(network, layout = layout) +
  ggraph::geom_node_point(colour = "black",
                          size = 2) +
  ggraph::geom_edge_arc(ggplot2::aes(width = n,
                             edge_colour = factor(relation2)),
                        arrow = arrow(length = unit(3, 'mm')),
                        end_cap = ggraph::circle(2, 'mm'),
                        alpha = 1, strength = 0.1
                        ) +
  ggraph::geom_edge_loop(aes(width = n, strength = 0.1,
                         edge_colour = factor(relation2)),
                         end_cap = ggraph::circle(2, 'mm')) +
  ggraph::scale_edge_width(range = c(0.5, 2), name = "Number of kin") +
  ggraph::scale_edge_colour_manual(values =
                                     c("orange","blue"),
                                   name = "kin-type",
                                   aesthetics = "edge_colour",
                                   na.value = "grey50") +

  ggplot2::scale_colour_manual(values = colours.6, na.value = "grey50") +
  ggplot2::scale_y_continuous(limits = c(1998, 2016), n.breaks = 18, minor_breaks = NULL) +
  ggplot2::scale_x_discrete(
    labels = c("AR" = "Adelaide River",
               "SC" = "Sampan Creek",
               "WR" = "Wildman River",
               "WAR" = "West Alligator River",
               "SAR" = "South Alligator River",
               "EAR" = "East Alligator River")) +
  ggplot2::ggtitle("Network between rivers for cross-cohort HSPs") +
  ggplot2::theme_void() +
  ggplot2::theme_minimal() +
  ggplot2::theme(
    legend.position = "right",
    plot.margin = unit(rep(1,4), "cm"),
    axis.text.y = element_text(size = 10, angle = 0, hjust = 0),
    axis.text.x = element_text(size = 10, angle = 90, hjust = 1),
    axis.title.x = element_blank(),
    axis.title.y = element_blank()
  )
```

```
## Warning: Ignoring `graph` as layout is already calculated
## i Pass the calculated layout to the `graph` argument to silence this warning
```

```
print(kin_network7)
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
```




```
ggplot2::ggsave(filename = "0.Kinship_network_HSP_between_river_between_cohort.png",
                bg = "white", plot = kin_network7, device = "png",
                width = 40, height = 30, units = "cm")
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
```

10.3.2.1.3 Cross-cohort, same-river HSPs Results

- 113 HSPs occurred in the same river, of which 61 HSPs were across cohort (50 same HT, 6 different HT, 7 NA).

```
## cross-cohort, same-river HSP
sum(pwdata$relation == "HSP" &
      pwdata$RIVER_i == pwdata$RIVER_j, na.rm = TRUE) ## 113
```

```
## [1] 113
```

```
sum(pwdata$relation == "HSP" &
      pwdata$RIVER_i == pwdata$RIVER_j &
      !pwdata$Cohort_overlap_relaxed, na.rm = TRUE) ## 61
```

```
## [1] 61
```

```
kableExtra::kbl(pwdata[pwdata$relation == "HSP" &
        pwdata$RIVER_i == pwdata$RIVER_j &
        !pwdata$Cohort_overlap_relaxed &
```

```
        !is.na(pwdata$Cohort_overlap_relaxed) &
        !(is.na(pwdata$HT_i)|is.na(pwdata$HT_j))
          ,-c(1:5,10,11,14,25)],
           format = "latex", longtable = TRUE,
      row.names = FALSE) %>%
kableExtra::kable_styling(full_width = FALSE,
                          font_size = 7) %>%
kableExtra::landscape()
```

LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
1030	648	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2008	2009	2009	2011	2012	2012
695	1030	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2008	2009	2009
1420	837	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	1996	1999	2001	2010	2010	2011
1420	925	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	1996	1999	2001	2009	2010	2010
910	1420	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	1996	1999	2001
1150	880	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2006	2006	2007	2010	2011	2011
1005	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2008	2009	2009	2012	2013	2013
520	895	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2014	2014	2014	2010	2010	2011
520	827	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2014	2014	2014	2010	2010	2011
1130	700	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2007	2008	2008	2012	2012	2012
685	905	M	F	SAR	SAR	HT06	HT13	0.37624	0.51485	2013	2013	2013	2010	2010	2010
1135	830	M	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2006	2007	2008	2010	2011	2011
830	685	M	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2010	2011	2011	2013	2013	2013
1135	685	M	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2006	2007	2008	2013	2013	2013
1290	910	F	M	SAR	SAR	HT14	HT13	0.02970	0.51485	2002	2003	2004	2009	2010	2010
1165	1040	F	M	SAR	SAR	HT06	HT13	0.37624	0.51485	2004	2005	2006	2008	2008	2009
600	1040	F	M	SAR	SAR	HT13	HT13	0.51485	0.51485	2012	2013	2013	2008	2008	2009
625	935	M	F	SAR	SAR	HT25	HT25	0.06931	0.06931	2013	2013	2013	2009	2010	2010
935	1340	F	M	SAR	SAR	HT25	HT25	0.06931	0.06931	2009	2010	2010	2000	2002	2003
625	1340	M	M	SAR	SAR	HT25	HT25	0.06931	0.06931	2013	2013	2013	2000	2002	2003
905	1065	F	F	WR	WR	HT22	HT22	0.20000	0.20000	2012	2012	2013	2007	2008	2009
1065	905	F	F	WR	WR	HT22	HT22	0.20000	0.20000	2007	2008	2009	2012	2012	2013
870	1230	M	F	WR	WR	HT13	HT13	0.68889	0.68889	2010	2011	2011	2004	2005	2006
840	1230	F	F	WR	WR	HT13	HT13	0.68889	0.68889	2010	2011	2011	2004	2005	2006
870	765	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2010	2011	2011	2013	2014	2014
840	765	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2010	2011	2011	2013	2014	2014
930	615	F	M	SAR	SAR	HT22	HT13	0.00990	0.51485	2009	2010	2010	2013	2013	2013
1330	915	F	M	EAR	EAR	HT06	HT06	0.14754	0.14754	2001	2002	2004	2010	2010	2011
685	1330	M	F	EAR	EAR	HT06	HT06	0.14754	0.14754	2012	2012	2013	2001	2002	2004
860	625	F	F	WAR	WAR	HT13	HT13	0.92683	0.92683	2010	2011	2011	2013	2013	2013
745	1215	M	F	EAR	EAR	HT25	HT25	0.40984	0.40984	2011	2012	2012	2005	2006	2006
780	1200	F	M	EAR	EAR	HT25	HT25	0.40984	0.40984	2011	2012	2012	2005	2006	2007
1330	1035	M	F	WAR	WAR	HT13	HT13	0.92683	0.92683	2001	2003	2004	2008	2009	2009
1320	775	F	F	WAR	WAR	HT13	HT13	0.92683	0.92683	2001	2003	2004	2011	2012	2012
970	760	F	F	SAR	SAR	HT25	HT25	0.06931	0.06931	2008	2009	2009	2011	2012	2012
660	1065	M	M	WR	WR	HT22	HT22	0.20000	0.20000	2014	2014	2015	2008	2009	2010
915	1065	M	M	WR	WR	HT22	HT22	0.20000	0.20000	2012	2012	2013	2008	2009	2010
920	660	F	M	WR	WR	HT22	HT22	0.20000	0.20000	2011	2012	2012	2014	2014	2015
1250	1070	M	M	WAR	WAR	HT13	HT13	0.92683	0.92683	2004	2005	2006	2008	2009	2010
1015	666	M	M	WAR	WAR	HT06	HT06	0.07317	0.07317	2009	2010	2010	2013	2014	2014
1380	970	M	F	WR	WR	HT22	HT25	0.20000	0.04444	2000	2003	2004	2011	2011	2012
1385	740	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2000	2002	2004	2013	2014	2014
1240	740	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2006	2007	2008	2013	2014	2014
1240	930	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2006	2007	2008	2011	2012	2012
1385	780	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2000	2002	2004	2013	2013	2014
1240	780	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2006	2007	2008	2013	2013	2014

1385	1060	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2000	2002	2004	2010	2010	2011
1385	1010	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2000	2002	2004	2010	2011	2011
1385	730	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2000	2002	2004	2014	2014	2014
1240	730	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2006	2007	2008	2014	2014	2014
700	1385	M	M	AR	AR	HT13	HT06	0.08000	0.52000	2014	2014	2015	2000	2003	2004
1450	970	M	M	AR	AR	HT06	HT06	0.52000	0.52000	1993	1997	2000	2011	2012	2012
850	795	M	M	SC	SC	HT13	HT13	0.40000	0.40000	2010	2011	2011	2013	2014	2014
735	1350	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	1999	2001	2002
1305	870	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2004	2005	2006	2012	2013	2013
1305	607	F	F	WR	WR	HT13	HT13	0.68889	0.68889	2004	2005	2006	2015	2015	2015

```
  ## same HT
  sum(pwdata$relation == "HSP" &
        pwdata$RIVER_i == pwdata$RIVER_j &
        !pwdata$Cohort_overlap_relaxed &
        pwdata$HT_i == pwdata$HT_j, na.rm = TRUE) ## 50
```

```
## [1] 50
```

```
  ## different HT
  sum(pwdata$relation == "HSP" &
        pwdata$RIVER_i == pwdata$RIVER_j &
        !pwdata$Cohort_overlap_relaxed &
        pwdata$HT_i != pwdata$HT_j, na.rm = TRUE) ## 6
```

```
## [1] 6
```

```
  ## missing data
  sum(is.na(pwdata$relation == "HSP" &
      pwdata$RIVER_i == pwdata$RIVER_j &
      !pwdata$Cohort_overlap_relaxed &
        pwdata$HT_i != pwdata$HT_j)) ## 7
```

```
## [1] 7
```

```
data <- pwdata
NWdata <- data %>%
  dplyr::filter(relation == "HSP") %>%
  dplyr::mutate(Cohort_gap = pmax(Cohort_i,Cohort_j) - pmin(Cohort_i,Cohort_j)) %>%
  dplyr::filter(!Cohort_overlap_relaxed) %>%
  dplyr::filter(RIVER_i == RIVER_j) %>%
  dplyr::mutate(relation2 =
                  dplyr::case_when(relation == "HSP" & HT_i == HT_j ~ 'MHSP',
                                   relation == "HSP" & HT_i != HT_j ~ 'PHSP',
                                   relation == "HSP" &
                                     (is.na(HT_i) |is.na(HT_j)) ~ 'HSP')) %>%
  dplyr::mutate(River_year_i = paste0(RIVER_i, "_", Cohort_i )) %>%
  dplyr::mutate(River_year_j = paste0(RIVER_j, "_", Cohort_j )) %>%
  dplyr::select(iname, jname, River_year_i, River_year_j,
                RIVER_i, RIVER_j, relation, relation2,
                Cohort_i, Cohort_j ) %>%
  dplyr::filter(!is.na(Cohort_i)) %>%
  dplyr::filter(!is.na(Cohort_j))

group_data <- NWdata %>%
  dplyr::group_by( River_year_i, River_year_j, relation2) %>%
  dplyr::count()
nodes <- unique(c(group_data$River_year_i, group_data$River_year_j))

pwPOP <- data %>%
  dplyr::filter(relation == "HSP") %>%
  dplyr::mutate(Cohort_gap = pmax(Cohort_i,Cohort_j) - pmin(Cohort_i,Cohort_j)) %>%
  dplyr::filter(!Cohort_overlap_relaxed) %>%
  dplyr::filter(RIVER_i == RIVER_j)

vertices <- meta %>%
  dplyr::filter(id %in% c(pwPOP$iname,pwPOP$jname)) %>%
  dplyr::select(River, Cohort2) %>%
  dplyr::filter(!is.na(Cohort2)) %>%
  dplyr::group_by(River, Cohort2) %>%
  dplyr::count()
```

```
vertices$River <- as.character(vertices$River)
vertices$River[vertices$River == "Adelaide River"] <- "AR"
vertices$River[vertices$River == "Sampan Creek"] <- "SC"
vertices$River[vertices$River == "Wildman River"] <- "WR"
vertices$River[vertices$River == "West Alligator River"] <- "WAR"
vertices$River[vertices$River == "South Alligator River"] <- "SAR"
vertices$River[vertices$River == "East Alligator River"] <- "EAR"
vertices$label <- paste0(vertices$River, "_", vertices$Cohort2)
vertices <- vertices[, c(4,1,2,3)]

vertices$River <- factor(vertices$River,
                         levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))
vertices <- as_data_frame(vertices, what = "vertices")
```

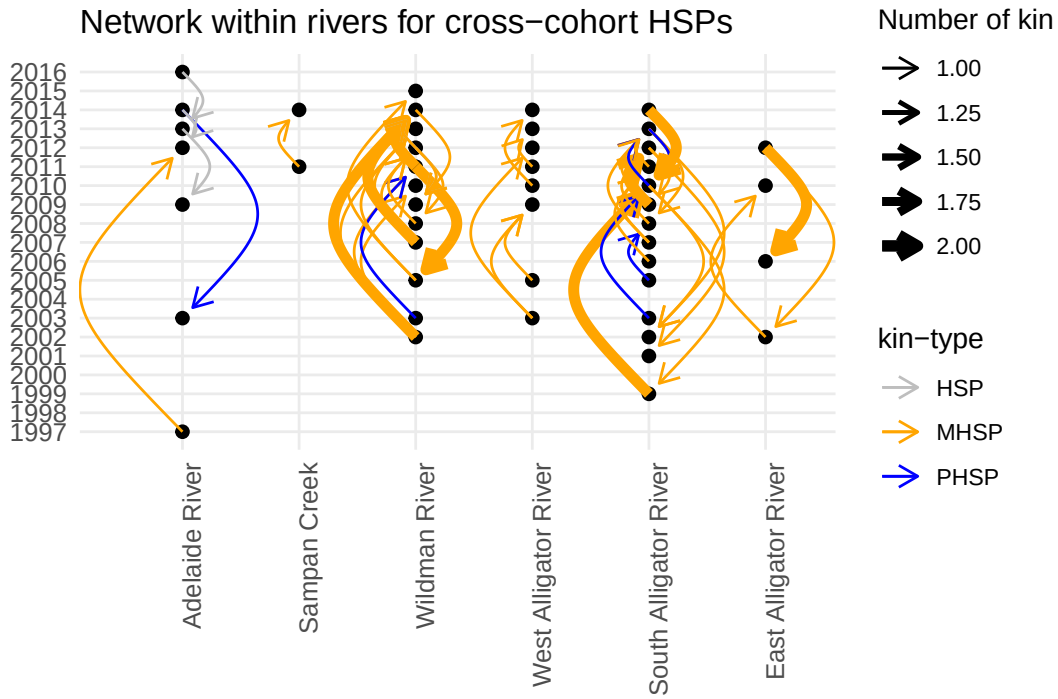
```
## Warning: `as_data_frame()` was deprecated in tibble 2.0.0.
## i Please use `as_tibble()` (with slightly different semantics) to convert to a
##   tibble, or `as.data.frame()` to convert to a data frame.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
network <- igraph::graph_from_data_frame(d = group_data, directed = TRUE,
                                         vertices = vertices )

layout <- ggraph::create_layout(network, layout = 'igraph',
                                circular = FALSE, algorithm = 'fr')
layout$y <- layout$Cohort2
layout$River <- factor(layout$River,
                       levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))
layout$x <- factor(layout$River,
                   levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))




kin_network8 <- ggraph::ggraph(network, layout = layout) +
  ggraph::geom_node_point(colour = "black",
                          size = 2) +
  ggraph::geom_edge_arc(aes(width = n,
                            edge_colour = factor(relation2)),
  arrow = arrow(length = unit(3, 'mm')),
  end_cap = ggraph::circle(2, 'mm'),
  alpha = 1, strength = 0.1
  ) +
  ggraph::geom_edge_loop(aes(width = n, strength = 0.1,
                         edge_colour = factor(relation2)),
                         end_cap = ggraph::circle(2, 'mm')) +
  ggraph::scale_edge_width(range = c(0.5, 2), name = "Number of kin") +
  ggraph::scale_edge_colour_manual(values =
                                     c("grey","orange","blue"),
                                   name = "kin-type",
                                   aesthetics = "edge_colour",
                                   na.value = "grey50") +

  ggplot2::scale_colour_manual(values = adegenet::funky(6), na.value = "grey50") +
  ggplot2::scale_y_continuous(limits = c(1997, 2016), n.breaks = 19, minor_breaks = NULL) +
  ggplot2::scale_x_discrete(
    labels = c("AR" = "Adelaide River",
```

```
                "SC" = "Sampan Creek",
                "WR" = "Wildman River",
                "WAR" = "West Alligator River",
                "SAR" = "South Alligator River",
                "EAR" = "East Alligator River")) +
  ggplot2::ggtitle("Network within rivers for cross-cohort HSPs") +
  ggplot2::theme_void() +
  ggplot2::theme_minimal() +
  ggplot2::theme(
    legend.position = "right",
    plot.margin = unit(rep(1,4), "cm"),
    axis.text.y = element_text(size = 10, angle = 0, hjust = 0),
    axis.text.x = element_text(size = 10, angle = 90, hjust = 1),
    axis.title.x = element_blank(),
    axis.title.y = element_blank()
  )
```

```
## Warning: Ignoring `graph` as layout is already calculated
## i Pass the calculated layout to the `graph` argument to silence this warning
```

```
print(kin_network8)
```




```
ggplot2::ggsave(filename = "0.Kinship_network_HSP_within_river_between_cohort.png",
                bg = "white", plot = kin_network8, device = "png",
                width = 40, height = 30, units = "cm")
```

10.3.3 Contemporary philopatry

Results

Cross-cohort, same-river HSPs reveal whether fathers or mothers are more likely to return to the same river between breeding seasons (i.e. philopatry).

- 72 cross-cohort HSPs with complete haplotype data
  - 16 HSPs have different haplotypes and are paternally related (6 same river, 10 cross river)
  - 56 HSP that are likely to be maternally related (50 same river, 6 cross river)
    - 27 of the 50 cross-cohort, same-river HSPs have haplotypes with high frequencies (<50%). This mean they could be paternally related
    - 3 of 6 cross-cohort, cross-river HSPs have haplotypes with high frequencies, and could be paternally related
- If there is no maternal philopatry, we would expect (H0) that the proportion of cross-cohort HSPs with a shared haplotype to be the same whether the two animals in the pair come from different rivers, or from the same river. If there is some maternal philopatry, we would expect a lower proportion of pairs with matching haplotypes across rivers.
  - A log-likelihood ratio (Delta) is calculated and tested against the null expectation (H0: equal distribution of maternal HSPs within and between rivers) using 10,000 permutations, following Feutry et al. (2017). This test took the mtDNA haplotype frequencies into account to calculate the likelihood maternally(or paternally)-related HSPs are more likely to occur in the same river. This likelihood approach is important when several haplotypes are common in the population.

CONCLUSION

- Female philopatry estimates
  - Observed: 50 out of 56 cross-cohort HSP with the same haplotype (i.e. maternal HSPs) were sampled from the same river (i.e. six cross-river). = 89.3%.
  - Maximum: Three of the six cross-river maternal HSPs had a matching haplotype that was rare (frequencies = 2-6%) in the river. Therefore, they must be maternal HSPs, while the remaining three could be paternal HSPs. Hence the maximum female philopatry is 50/53 = 94.3%.
  - Minimum: if we assume the 27 cross-cohort, same-river HSPs with high haplotype frequencies are actually paternal related, and all six cross-river HSP with same haplotype are maternally related, then the minimum female philopatry is 23/29 = 79.3%.
  - Null hypothesis: "no maternal philopatry": log-likelihood ratio (Delta) = 14.195, p = < 0.0001. Thus, we reject the null hypothesis and assume females are philopatric
- Male philopatry estimates
  - Observed: 6 out of 16 cross-cohort HSP with different haplotypes (i.e. paternal HSPs) were sampled from the same river (i.e. 10 cross-river). = 37.5%.
  - Maximum: if we assume the 27 cross-cohort, same-river HSPs with high haplotype frequencies are actually paternal related, and all six cross-river HSP with same haplotype are maternally related, then the maximum male philopatry is 33/43 = 76.7%.
  - Minimum: Three of the six cross-river maternal HSPs had a matching haplotype that was rare (frequencies = 2-6%) in the river. Therefore, they must be maternal HSPs, while the remaining three could be paternal HSPs. Here we assume that the 27 same-river HSPs with matching haplotype are maternal. Hence the minimum male philopatry is 6/19 = 31.5%.
  - Null hypothesis: "no paternal philopatry": log-likelihood ratio (Delta) = 0.008486497, p = 1.000. Thus, we cannot reject the null hypothesis.

```
sum(pwdata_prob$relation == "HSP" &
      !pwdata_prob$Cohort_overlap_relaxed &
```

```
      pwdata_prob$RIVER_i == pwdata_prob$RIVER_j &
      pwdata_prob$HT_i == pwdata_prob$HT_j
      , na.rm = TRUE) ## 50 cross-cohort, same-river maternal HSPs
```

10.3.3.1 Female philopatry

```
## [1] 50
```

```
  sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        !pwdata_prob$Cohort_overlap_relaxed &
        pwdata_prob$RIVER_i == pwdata_prob$RIVER_j &
        pwdata_prob$HT_i == pwdata_prob$HT_j
        , na.rm = TRUE) ## 27 cross-cohort, same-river HSPs with higher probability to be paternally
```

```
## [1] 27
```

```
sum(pwdata_prob$relation == "HSP" &
      !pwdata_prob$Cohort_overlap_relaxed &
      pwdata_prob$RIVER_i != pwdata_prob$RIVER_j &
      pwdata_prob$HT_i == pwdata_prob$HT_j
      , na.rm = TRUE) ## 6 cross-cohort, cross-river maternal HSPs
```

```
## [1] 6
```

```
  sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        !pwdata_prob$Cohort_overlap_relaxed &
        pwdata_prob$RIVER_i != pwdata_prob$RIVER_j &
        pwdata_prob$HT_i == pwdata_prob$HT_j
        , na.rm = TRUE) ## 3 cross-cohort, cross-river HSPs with higher probability to be paternally
```

```
## [1] 3
```

```
hapcount <- as.data.frame(table(meta$HT[meta$pop == "Van Diemen Gulf"]),
                          stringsAsFactors = FALSE)

colnames(hapcount) <- c("H", "n")
hapcount <- hapcount[!hapcount$H == "",]
hapfreq <- hapcount$n / sum( hapcount$n)
names( hapfreq) <- hapcount$H


haps <- pwdata[pwdata$relation == "HSP" & !pwdata$Cohort_overlap_relaxed &
                 !is.na(pwdata$Cohort_overlap_relaxed) &
                 !is.na(pwdata$HT_i) & !is.na(pwdata$HT_j),c(12,13,15,16,17,18)]

haps$samediff <- ifelse( haps$RIVER_i == haps$RIVER_j, "S", "D")
haps <- haps[order(haps$samediff),]
haps <- data.frame(samediff = haps$samediff, h1 = haps$HT_i, h2 = haps$HT_j,
                   h1_freq = haps$HTfreq_i, h2_freq = haps$HTfreq_j,
                   stringsAsFactors = FALSE)


haps <- within( haps, {
  Pr_if_mat <- ifelse( h1 == h2, pmin(haps$h1_freq, haps$h2_freq), 0)
  Pr_if_pat <- haps$h1_freq * haps$h2_freq
})


knitr::kable(haps, format = "latex") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")
```

samediff	h1	h2	h1_freq	h2_freq	Pr_if_pat	Pr_if_mat
D	HT13	HT06	0.27869	0.37624	0.1048543	0.00000
D	HT13	HT06	0.27869	0.37624	0.1048543	0.00000
D	HT13	HT06	0.27869	0.37624	0.1048543	0.00000
D	HT07	HT07	0.06557	0.06667	0.0043716	0.06557
D	HT13	HT14	0.92683	0.02970	0.0275269	0.00000
D	HT25	HT21	0.16000	0.02222	0.0035552	0.00000
D	HT25	HT21	0.40984	0.03333	0.0136600	0.00000
D	HT21	HT21	0.03333	0.02222	0.0007406	0.02222
D	HT13	HT25	0.92683	0.40984	0.3798520	0.00000
D	HT13	HT25	0.92683	0.40984	0.3798520	0.00000
D	HT13	HT13	0.51485	0.92683	0.4771784	0.51485
D	HT13	HT13	0.51485	0.92683	0.4771784	0.51485
D	HT13	HT13	0.51485	0.92683	0.4771784	0.51485
D	HT06	HT13	0.52000	0.92683	0.4819516	0.00000
D	HT25	HT24	0.06931	0.04000	0.0027724	0.00000
D	HT06	HT06	0.14754	0.52000	0.0767208	0.14754
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT06	HT13	0.37624	0.51485	0.1937072	0.00000
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT14	HT13	0.02970	0.51485	0.0152910	0.00000
S	HT06	HT13	0.37624	0.51485	0.1937072	0.00000
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT25	HT25	0.06931	0.06931	0.0048039	0.06931
S	HT25	HT25	0.06931	0.06931	0.0048039	0.06931
S	HT25	HT25	0.06931	0.06931	0.0048039	0.06931
S	HT22	HT22	0.20000	0.20000	0.0400000	0.20000
S	HT22	HT22	0.20000	0.20000	0.0400000	0.20000
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT22	HT13	0.00990	0.51485	0.0050970	0.00000
S	HT06	HT06	0.14754	0.14754	0.0217681	0.14754
S	HT06	HT06	0.14754	0.14754	0.0217681	0.14754
S	HT13	HT13	0.92683	0.92683	0.8590138	0.92683
S	HT25	HT25	0.40984	0.40984	0.1679688	0.40984
S	HT25	HT25	0.40984	0.40984	0.1679688	0.40984
S	HT13	HT13	0.92683	0.92683	0.8590138	0.92683
S	HT13	HT13	0.92683	0.92683	0.8590138	0.92683
S	HT25	HT25	0.06931	0.06931	0.0048039	0.06931
S	HT22	HT22	0.20000	0.20000	0.0400000	0.20000
S	HT22	HT22	0.20000	0.20000	0.0400000	0.20000
S	HT22	HT22	0.20000	0.20000	0.0400000	0.20000
S	HT13	HT13	0.92683	0.92683	0.8590138	0.92683
S	HT06	HT06	0.07317	0.07317	0.0053538	0.07317
S	HT22	HT25	0.20000	0.04444	0.0088880	0.00000
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889

```r
sum(haps$Pr_if_pat > haps$Pr_if_mat) ## 16
```

```
## [1] 16
```

```r
sum(haps$Pr_if_mat > haps$Pr_if_pat) ## 56
```

```
## [1] 56
```

```r
sum(haps$Pr_if_pat > haps$Pr_if_mat & haps$samediff == "D") ## 10
```

```
## [1] 10
```

```r
sum(haps$Pr_if_mat > haps$Pr_if_pat & haps$samediff == "D") ## 6
```

```
## [1] 6
```

```r
sum(haps$Pr_if_pat > haps$Pr_if_mat & haps$samediff == "S") ## 6
```

```
## [1] 6
```

```r
sum(haps$Pr_if_mat > haps$Pr_if_pat & haps$samediff == "S") ## 50
```

```
## [1] 50
```

```r
lglk2 <- function(logit_Prmat)
  sum(log(inv.logit(logit_Prmat)[samediff] * (Pr_if_mat - Pr_if_pat) + Pr_if_pat))
environment(lglk2) <- list2env(haps) # so it finds Pr_if_mat/pat

lglk1 <- function(Prmat1) {lp <- gtools::logit(Prmat1); lglk2(c(S = lp, D = lp)) }

# If no "river effect":
fit1 <- optimize(mvbutils::NEG(lglk1), lower = 0, upper = 1)
Prmatsame <- fit1$minimum

# Allowing "river effect":
lp <- logit( Prmatsame)
fit2 <- optim(c(S = lp, D = lp), mvbutils::NEG(lglk2), method = 'BFGS')

Delta <- -2*(fit2$value - fit1$objective)


# Null distribution via permutation test
nsame <- sum(haps$samediff == 'S')
npair <- nrow(haps)

Dnull <- function(iter) {
  which.same <- gbasics::rsample(nsame, 1:npair, replace = FALSE)
  diffo <- rep('D', npair)
  diffo[which.same] <- 'S'
  assign('samediff', diffo, environment( lglk2))
  sfit1 <- optimize(mvbutils::NEG( lglk1), lower = 0, upper = 1)
  sfit2 <- optim(c(S = lp, D = lp), mvbutils::NEG(lglk2), method = 'BFGS')
  Dsim <- -2*(sfit2$value - sfit1$objective)
  return(Dsim)
}
null_samps <- mvbutils::do.on(1:10000, Dnull(.))

cat(sprintf('Observed lglk ratio Delta: %5.3f', Delta), '\n')
```

```
## Observed lglk ratio Delta: 14.195
```

```r
cat('MLE(ppn mat) under H1:\n')
```

```
## MLE(ppn mat) under H1:
```

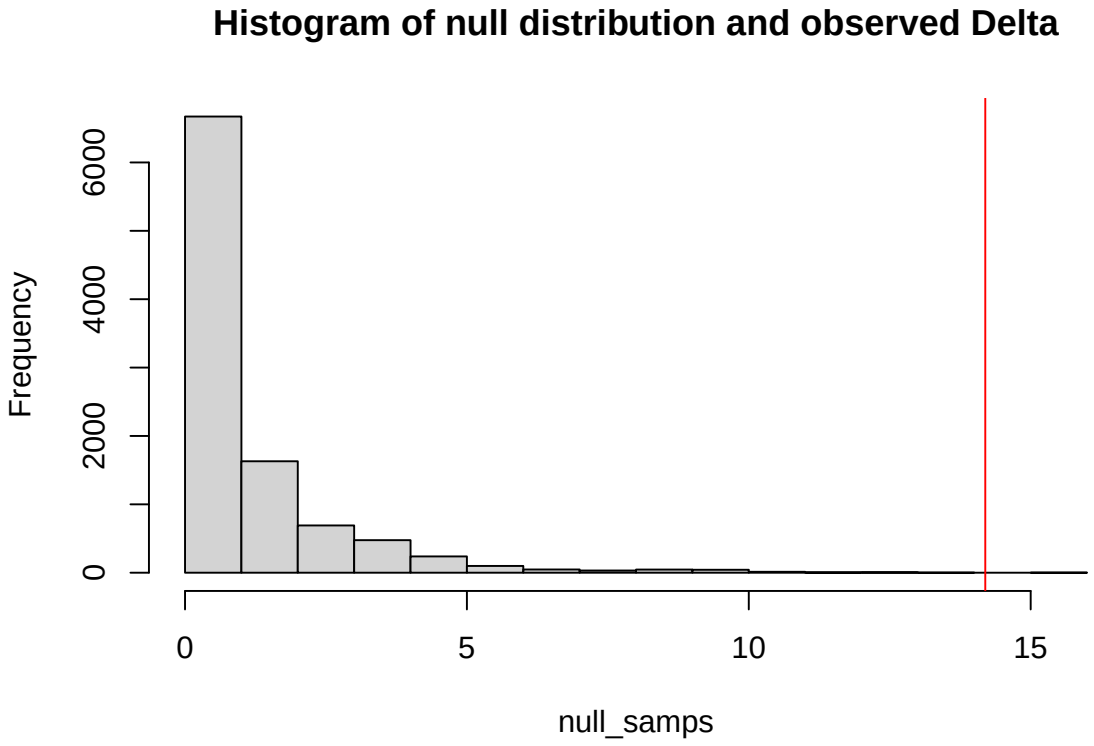
```
print(fit2$par)
```

```
##         S         D
##  1.379609 -1.829370
```

```
cat(sprintf('Propportion of null-distro Deltas above observed value: %5.3f (n=%i simulations)',
            mean(null_samps >= Delta), length(null_samps)), '\n')
```

```
## Propportion of null-distro Deltas above observed value:
0.000 (n=10000 simulations)
```

```
{
  hist(null_samps, breaks = 20, main = "Histogram of null distribution and observed Delta")
  abline(v = Delta, col = "red")
}
```




```
sum(pwdata_prob$relation == "HSP" &
      !pwdata_prob$Cohort_overlap_relaxed &
      pwdata_prob$RIVER_i == pwdata_prob$RIVER_j &
      pwdata_prob$HT_i != pwdata_prob$HT_j
      , na.rm = TRUE) ## 6 cross-cohort, same-river paternal HSPs
```

10.3.3.2 male male philopatry

```
## [1] 6
```

```
  sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        !pwdata_prob$Cohort_overlap_relaxed &
        pwdata_prob$RIVER_i == pwdata_prob$RIVER_j &
        pwdata_prob$HT_i == pwdata_prob$HT_j
        , na.rm = TRUE) ## 27 cross-cohort, same-river maternal HSPs with higher probability to be p
```

```
## [1] 27
```

```
sum(pwdata_prob$relation == "HSP" &
      !pwdata_prob$Cohort_overlap_relaxed &
      pwdata_prob$RIVER_i != pwdata_prob$RIVER_j &
      pwdata_prob$HT_i != pwdata_prob$HT_j
      , na.rm = TRUE) ## 10 cross-cohort, cross-river paternal HSPs
```

```
## [1] 10
```

```
  sum(pwdata_prob$relation == "HSP" & pwdata_prob$Pr_if_pat > 0.25 &
        !pwdata_prob$Cohort_overlap_relaxed &
        pwdata_prob$RIVER_i != pwdata_prob$RIVER_j &
        pwdata_prob$HT_i == pwdata_prob$HT_j
        , na.rm = TRUE) ## 3 cross-cohort, cross-river maternal HSPs with higher probability to be p
```

```
## [1] 3
```

```
hapcount <- as.data.frame(table(meta$HT[meta$pop == "Van Diemen Gulf"]),
                          stringsAsFactors = FALSE)

colnames(hapcount) <- c("H", "n")
hapcount <- hapcount[!hapcount$H == "",]
hapfreq <- hapcount$n / sum( hapcount$n)
names( hapfreq) <- hapcount$H


haps <- pwdata[pwdata$relation == "HSP" & !pwdata$Cohort_overlap_relaxed &
                 !is.na(pwdata$Cohort_overlap_relaxed) &
                 !is.na(pwdata$HT_i) & !is.na(pwdata$HT_j),c(12,13,15,16,17,18)]

haps$samediff <- ifelse( haps$RIVER_i == haps$RIVER_j, "S", "D")
haps <- haps[order(haps$samediff),]
haps <- data.frame(samediff = haps$samediff, h1 = haps$HT_i, h2 = haps$HT_j,
                   h1_freq = haps$HTfreq_i, h2_freq = haps$HTfreq_j,
                   stringsAsFactors = FALSE)


haps <- within( haps, {
  Pr_if_mat <- ifelse( h1 == h2, pmin(haps$h1_freq, haps$h2_freq), 0)
  Pr_if_pat <- haps$h1_freq * haps$h2_freq
})


knitr::kable(haps, format = "latex") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")
```

```
sum(haps$Pr_if_pat > haps$Pr_if_mat) ## 16
```

```
## [1] 16
```

```
sum(haps$Pr_if_mat > haps$Pr_if_pat) ## 56
```

```
## [1] 56
```

```
sum(haps$Pr_if_pat > haps$Pr_if_mat & haps$samediff == "D") ## 10
```

```
## [1] 10
```

```
sum(haps$Pr_if_mat > haps$Pr_if_pat & haps$samediff == "D") ## 6
```

```
## [1] 6
```

samediff	h1	h2	h1_freq	h2_freq	Pr_if_pat	Pr_if_mat
D	HT13	HT06	0.27869	0.37624	0.1048543	0.00000
D	HT13	HT06	0.27869	0.37624	0.1048543	0.00000
D	HT13	HT06	0.27869	0.37624	0.1048543	0.00000
D	HT07	HT07	0.06557	0.06667	0.0043716	0.06557
D	HT13	HT14	0.92683	0.02970	0.0275269	0.00000
D	HT25	HT21	0.16000	0.02222	0.0035552	0.00000
D	HT25	HT21	0.40984	0.03333	0.0136600	0.00000
D	HT21	HT21	0.03333	0.02222	0.0007406	0.02222
D	HT13	HT25	0.92683	0.40984	0.3798520	0.00000
D	HT13	HT25	0.92683	0.40984	0.3798520	0.00000
D	HT13	HT13	0.51485	0.92683	0.4771784	0.51485
D	HT13	HT13	0.51485	0.92683	0.4771784	0.51485
D	HT13	HT13	0.51485	0.92683	0.4771784	0.51485
D	HT06	HT13	0.52000	0.92683	0.4819516	0.00000
D	HT25	HT24	0.06931	0.04000	0.0027724	0.00000
D	HT06	HT06	0.14754	0.52000	0.0767208	0.14754
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT06	HT13	0.37624	0.51485	0.1937072	0.00000
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT06	HT06	0.37624	0.37624	0.1415565	0.37624
S	HT14	HT13	0.02970	0.51485	0.0152910	0.00000
S	HT06	HT13	0.37624	0.51485	0.1937072	0.00000
S	HT13	HT13	0.51485	0.51485	0.2650705	0.51485
S	HT25	HT25	0.06931	0.06931	0.0048039	0.06931
S	HT25	HT25	0.06931	0.06931	0.0048039	0.06931
S	HT25	HT25	0.06931	0.06931	0.0048039	0.06931
S	HT22	HT22	0.20000	0.20000	0.0400000	0.20000
S	HT22	HT22	0.20000	0.20000	0.0400000	0.20000
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT22	HT13	0.00990	0.51485	0.0050970	0.00000
S	HT06	HT06	0.14754	0.14754	0.0217681	0.14754
S	HT06	HT06	0.14754	0.14754	0.0217681	0.14754
S	HT13	HT13	0.92683	0.92683	0.8590138	0.92683
S	HT25	HT25	0.40984	0.40984	0.1679688	0.40984
S	HT25	HT25	0.40984	0.40984	0.1679688	0.40984
S	HT13	HT13	0.92683	0.92683	0.8590138	0.92683
S	HT13	HT13	0.92683	0.92683	0.8590138	0.92683
S	HT25	HT25	0.06931	0.06931	0.0048039	0.06931
S	HT22	HT22	0.20000	0.20000	0.0400000	0.20000
S	HT22	HT22	0.20000	0.20000	0.0400000	0.20000
S	HT22	HT22	0.20000	0.20000	0.0400000	0.20000
S	HT13	HT13	0.92683	0.92683	0.8590138	0.92683
S	HT06	HT06	0.07317	0.07317	0.0053538	0.07317
S	HT22	HT25	0.20000	0.04444	0.0088880	0.00000
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889
S	HT13	HT13	0.68889	0.68889	0.4745694	0.68889

```r
sum(haps$Pr_if_pat > haps$Pr_if_mat & haps$samediff == "S") ## 6
```

```
## [1] 6
```

```r
sum(haps$Pr_if_mat > haps$Pr_if_pat & haps$samediff == "S") ## 50
```

```
## [1] 50
```

```r
lglk2 <- function(logit_Prpat)
  sum(log(inv.logit(logit_Prpat)[samediff] * (Pr_if_pat)))
environment(lglk2) <- list2env(haps) # so it finds Pr_if_mat/pat

lglk1 <- function(Prpat1) {lp <- gtools::logit(Prpat1); lglk2(c(S = lp, D = lp)) }

# If no "river effect":
fit1 <- optimize(mvbutils::NEG(lglk1), lower = 0, upper = 1)
Prpatsame <- fit1$minimum

# Allowing "river effect":
lp <- logit( Prpatsame)
fit2 <- optim(c(S = lp, D = lp), mvbutils::NEG(lglk2), method = 'BFGS')

Delta <- -2*(fit2$value - fit1$objective)


# Null distribution via permutation test
nsame <- sum(haps$samediff == 'S')
npair <- nrow(haps)

Dnull <- function(iter) {
  which.same <- gbasics::rsample(nsame, 1:npair, replace = FALSE)
  diffo <- rep('D', npair)
  diffo[which.same] <- 'S'
  assign('samediff', diffo, environment( lglk2))
  sfit1 <- optimize(mvbutils::NEG( lglk1), lower = 0, upper = 1)
  sfit2 <- optim(c(S = lp, D = lp), mvbutils::NEG(lglk2), method = 'BFGS')
  Dsim <- -2*(sfit2$value - sfit1$objective)
  return(Dsim)
}
null_samps <- mvbutils::do.on(1:10000, Dnull(.))

cat(sprintf('Observed lglk ratio Delta: %5.3f', Delta), '\n')
```

```
## Observed lglk ratio Delta: 0.008
```

```r
cat('MLE(ppn mat) under H1:\n')
```

```
## MLE(ppn mat) under H1:
```

```r
print(fit2$par)
```

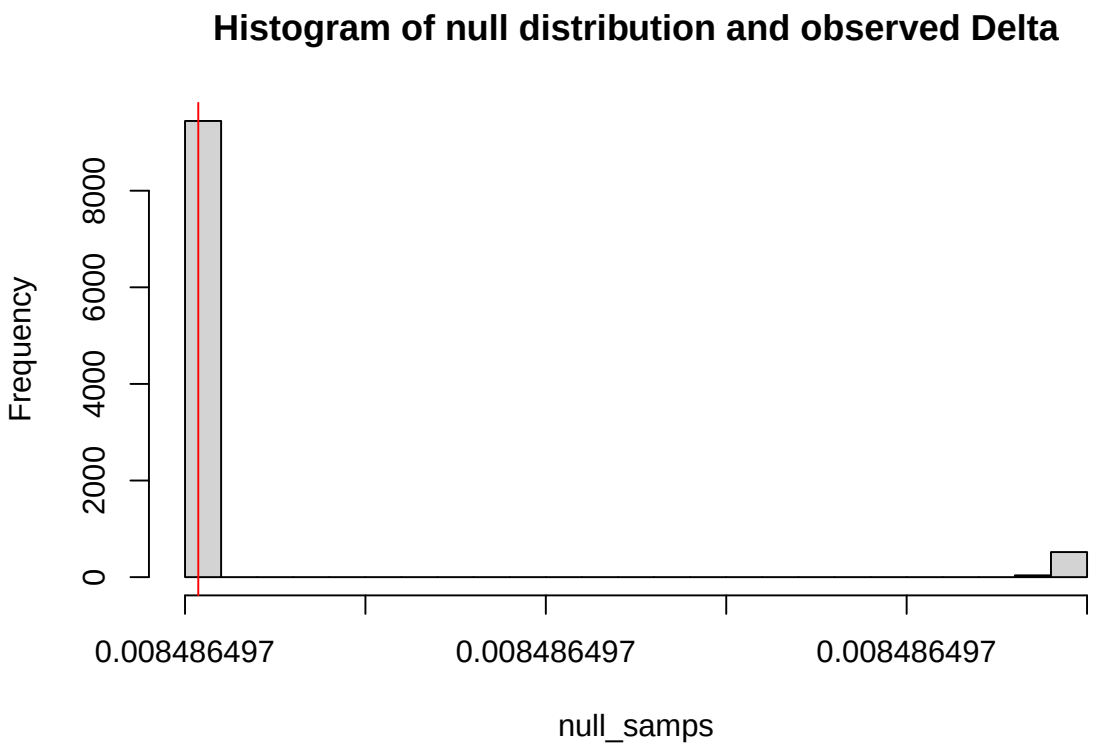
```
##        S        D
## 13.04384 10.60803
```

```r
cat(sprintf('Propportion of null-distro Deltas above observed value: %5.3f (n=%i simulations)',
             mean(null_samps >= Delta), length(null_samps)), '\n')
```

```
## Propportion of null-distro Deltas above observed value:
1.000 (n=10000 simulations)
```

```r
{
  hist(null_samps, breaks = 20, main = "Histogram of null distribution and observed Delta")
```

```
  abline(v = Delta, col = "red")
}
```



10.4 Unbiased HSP ratio

As the sample size per sampling location can bias how many HSPs are found, we corrected the number HSP by the number of pairwise comparisons performed (HSPcorr). The ratio of HSPcorr within over the sum of HSPcorr between sampling locations in Van Diemen Gulf gives an estimate of philopatric and dispersive behaviours (HSPcorrwithin/sum(HSPcorrbetween)). Specifically, a ratio larger than 1 indicates stronger philopatric behaviour and smaller than 1 shows a higher connectivity.

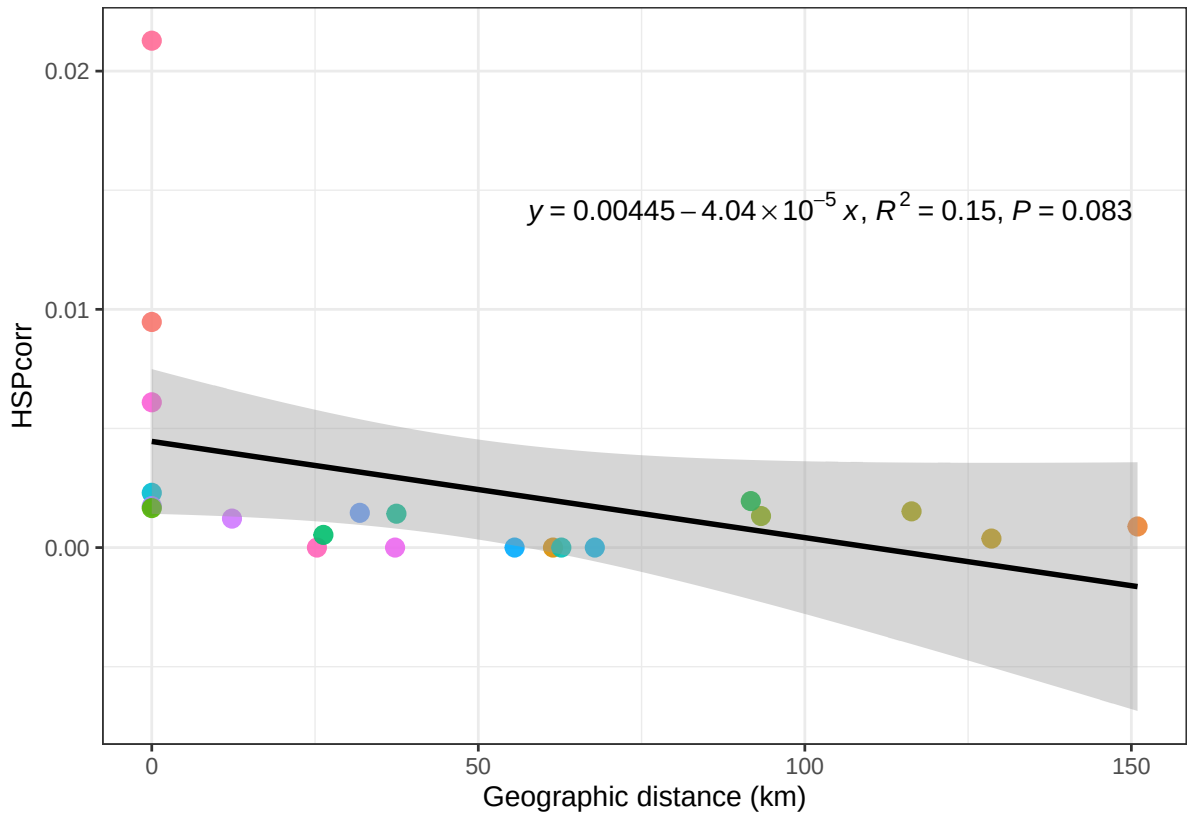
Below diagonal: cross-cohort, cross-river; *above diagonal*: same-cohort, cross-river; **diagonal (bold)**: a mix of cross-cohort and same-cohort.

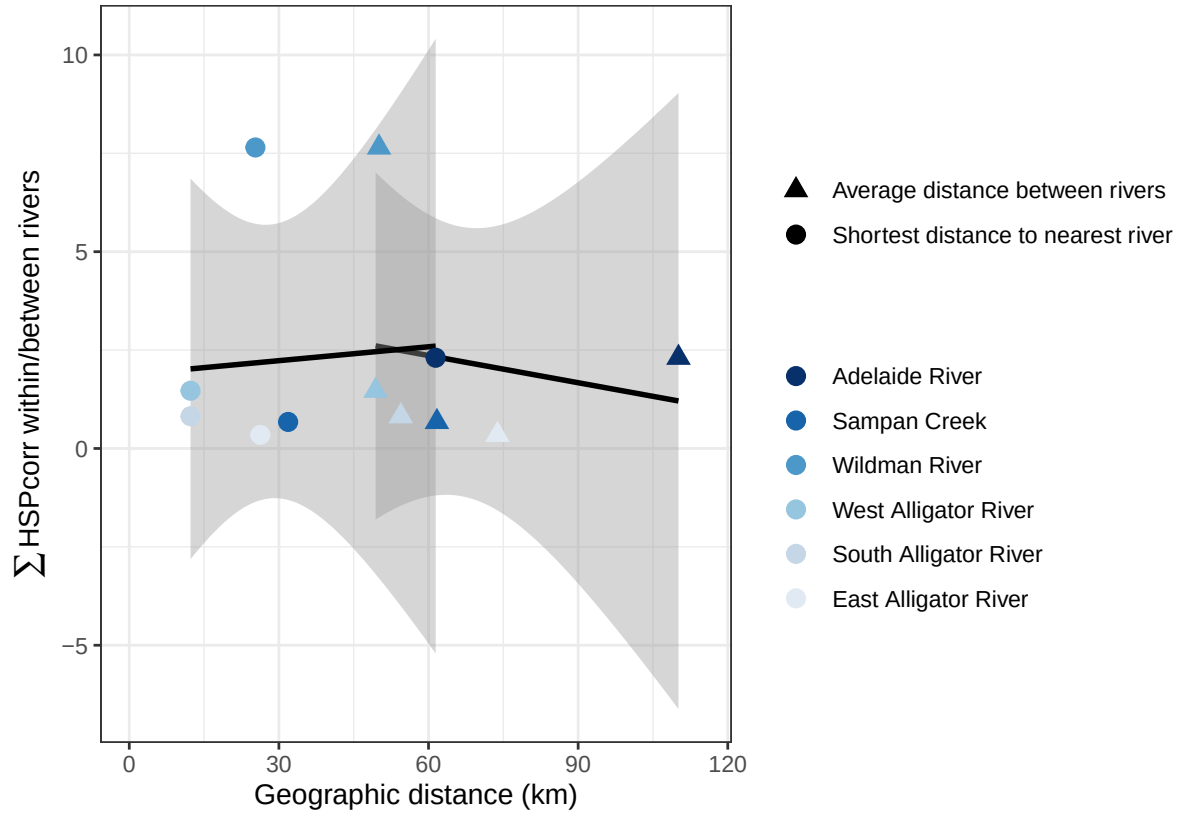
Number of HSPs		A	S	W	WA	SA	EA
		33	**30**	**47**	**41**	**163**	**70**
A	**33**	**8**	0	0	0	0	0
S	**30**	0	**1**	0	0	0	3
W	**47**	1	1	**29**	0	0	0
WA	**41**	1	0	0	**8**	1	1
SA	**163**	1	0	0	4	**60**	3
EA	**70**	1	2	0	2	3	**7**

Number of comparisons		A	S	W	WA	SA	EA
		33	**30**	**47**	**41**	**163**	**70**
A	**33**	**528**	479	756	658	2641	1129
S	**30**	479	**435**	686	597	2397	1025
W	**47**	756	686	**1081**	942	3778	1616
WA	**41**	658	597	942	**820**	3291	1407
SA	**163**	2641	2397	3778	3291	**13203**	5647
EA	**70**	1129	1025	1616	1407	5647	**2415**

Ratio of HSP/comparisons		A	S	W	WA	SA	EA
		33	**30**	**47**	**41**	**163**	**70**
A	**33**	**0.01515**	0.00000	0.00000	0.00000	0.00000	0.00000
S	**30**	0.00000	**0.00230**	0.00000	0.00000	0.00000	0.00293
W	**47**	0.00132	0.00146	**0.02683**	0.00000	0.00000	0.00000
WA	**41**	0.00152	0.00000	0.00000	**0.00976**	0.00030	0.00071
SA	**163**	0.00038	0.00000	0.00000	0.00122	**0.00454**	0.00053
EA	**70**	0.00089	0.00195	0.00000	0.00142	0.00053	**0.00290**

	A	S	W	WA	SA	EA
$\sum HSPcorr_{between}$	0.00411	0.00634	0.00278	0.00517	0.00296	0.00896
$\frac{HSPcorr_{within}}{\sum HSPcorr_{between}}$	3.68862	0.36281	9.64347	1.88666	1.53486	0.32357


$y = 0.00445 - 4.04 \times 10^{-5}\, x$, $R^2 = 0.15$, $P = 0.083$
HSPcorr
0.02
0.01
0.00
0
50
100
150
Geographic distance (km)


∑ HSPcorr within/between rivers
10
5
0
−5
0
30
60
90
120
Geographic distance (km)
Average distance between rivers
Shortest distance to nearest river
Adelaide River
Sampan Creek
Wildman River
West Alligator River
South Alligator River
East Alligator River

10.5 Breeding biology from kinship

Based on same-cohort FSP and HSP, we can get an insight on the breeding biology of this threatened shark, without destructive sampling.

- Litter size: If multiple FSP/HSP are born in the same year and from the same mother (matching haplotypes), then the number of siblings in a family group could indicate litter size.
- Multiple paternity: HSPs that are born in the same year and have the same mother may indicate multiple paternity rates.
- Male breeding events per season: HSPs born in the same year that have the same father (different haplotypes), would indicate that the father mated with multiple female in a breeding season. Alternatively, sperm storage of females may explain this as well.

ASSUMPTIONS

- Juvenile can be assigned to the same cohort. We are relatively confident we can assign juveniles to the same cohort, since all, except one, FSP were assigned to the same cohort.
- Siblings with the same haplotype share the same mother. Given that we only have 26 haplotype and three of those occur at high frequency, this assumption is likely not true. This leads to over-estimating the litter size and multiple paternity, and under-estimating multiple male breeding events.
- All sibling in litter are sampled. This assumption is certainly violated. This will lead to an under-estimation of litter size, multiple paternity, and male breeding events.

```
data <- pwdata
kinNWdata <- data %>%
  dplyr::filter(relation %in% c("HSP", "FSP")) %>%
  dplyr::mutate(Cohort_gap = pmax(Cohort_i,Cohort_j) - pmin(Cohort_i,Cohort_j)) %>%
  dplyr::filter(Cohort_overlap_relaxed) %>%
  dplyr::select(iname, jname, relation, Cohort_gap)
network <- igraph::graph_from_data_frame(d = kinNWdata, directed = TRUE)


df <- data.frame(id = igraph::V(network)$name)
vertices <- dplyr::left_join(df, meta, by = "id") %>%
  dplyr::select(id,`sample location`, sex, HSP, yearcollect, Age,
                Cohort2, HT, Htfreq_perRiver)
vertices$HT[vertices$HT == ""] <- NA
vertices <- as_data_frame(vertices, what = "vertices")
```

```
## Warning: `as_data_frame()` was deprecated in tibble 2.0.0.
## i Please use `as_tibble()` (with slightly different semantics) to convert to a
##   tibble, or `as.data.frame()` to convert to a data frame.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

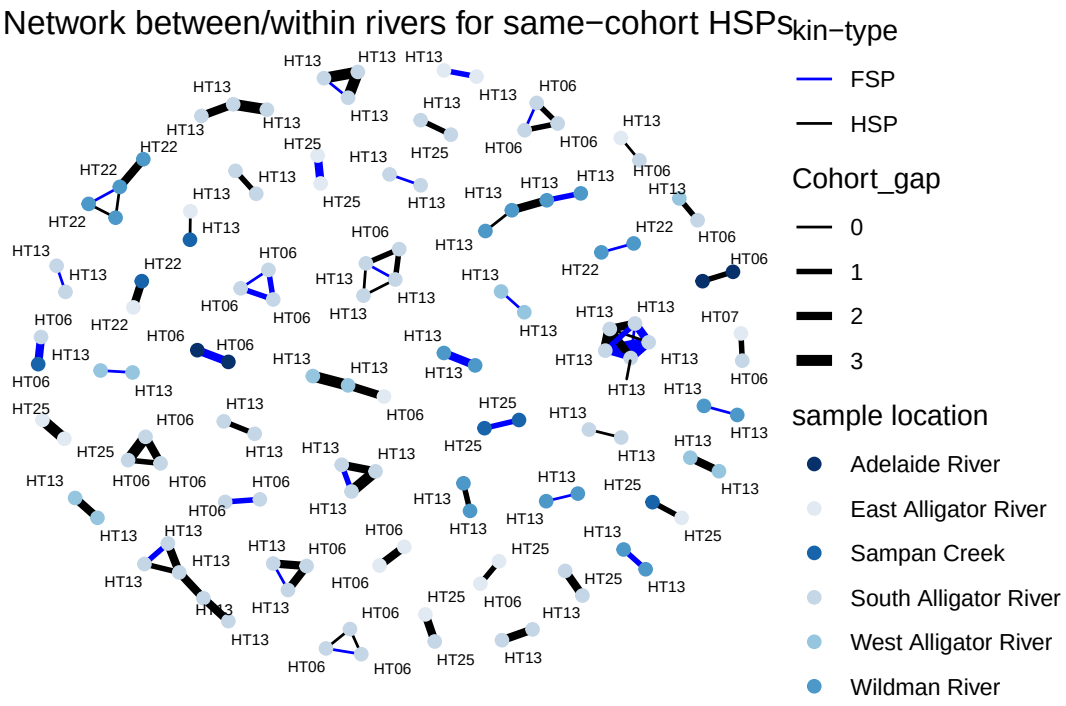
```
network <- igraph::graph_from_data_frame(d = kinNWdata, directed = TRUE,
                                         vertices = vertices )

layout <- ggraph::create_layout(network, layout = 'igraph',
                                circular = FALSE, algorithm = 'fr')


kin_network3 <- ggraph::ggraph(network, layout = layout) +
  ggraph::geom_edge_link(
    aes(width = Cohort_gap,
        edge_colour = factor(relation)),
    edge_alpha = 1) +
```

```
  ggraph::scale_edge_width(range = c(0.5, 2), name = "Cohort_gap") +
  ggraph::scale_edge_colour_manual(values = c("blue","black"),
                                   name = "kin-type",
                                   aesthetics = "edge_colour",
                                   na.value = "grey50") +
  ggraph::geom_node_point(aes(color = `sample location`),
                          size = 2) +
  ggraph::geom_node_text( aes(label = HT), repel = TRUE,
                          size = 2, color = "black") +
  ggplot2::ggtitle("Network between/within rivers for same-cohort HSPs") +
  ggplot2::scale_color_manual(values = colours.6, na.value = "grey50") +
  ggplot2::theme_void() +
  ggplot2::theme(
    legend.position = "right",
    plot.margin = unit(rep(1,4), "cm")
  )
```

```
## Warning: Ignoring `graph` as layout is already calculated
## i Pass the calculated layout to the `graph` argument to silence this warning
```

```
print(kin_network3)
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_text_repel()`).
```



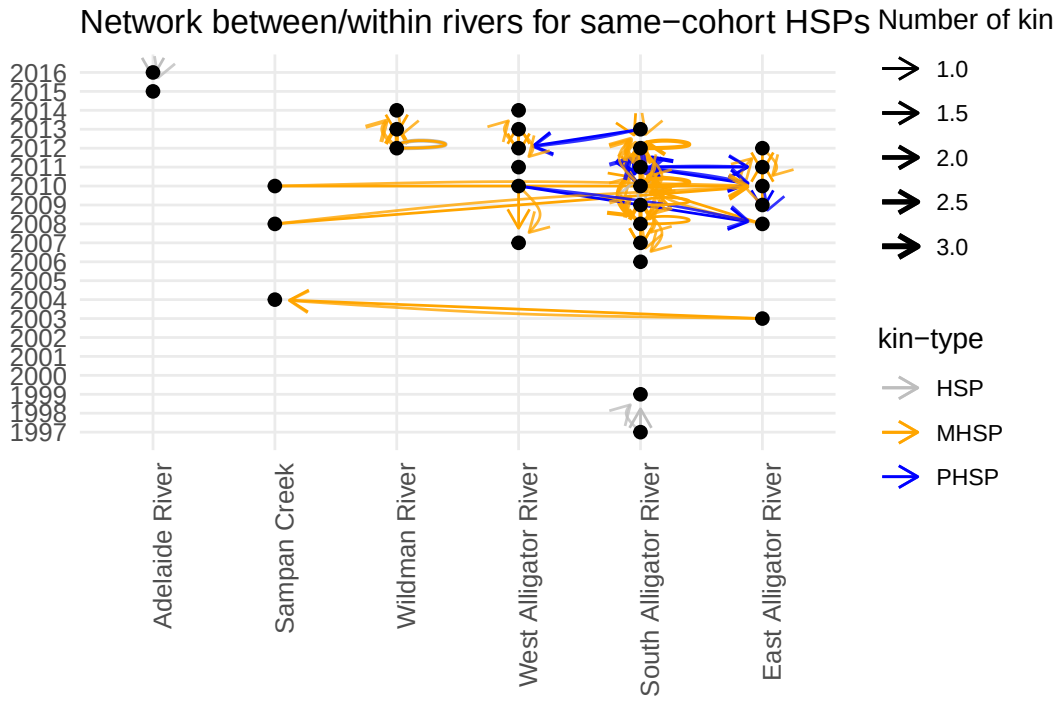

```
# ggplot2::ggsave(filename = "0.Kinship_network_within_cohort2.png",bg = "white",
#                 plot = kin_network3, device = "png", width = 40,
#                 height = 30, units = "cm")
```

```
data <- pwdata
NWdata <- data %>%
  dplyr::filter(relation == "HSP") %>%
  dplyr::mutate(Cohort_gap = pmax(Cohort_i,Cohort_j) - pmin(Cohort_i,Cohort_j)) %>%
  dplyr::filter(Cohort_overlap_relaxed) %>%
  dplyr::mutate(relation2 =
                  dplyr::case_when(relation == "HSP" & HT_i == HT_j ~ 'MHSP',
                                   relation == "HSP" & HT_i != HT_j ~ 'PHSP',
                                   relation == "HSP" &
                                     (is.na(HT_i) |is.na(HT_j)) ~ 'HSP')) %>%
  dplyr::mutate(River_year_i = paste0(RIVER_i, "_", Cohort_i )) %>%
  dplyr::mutate(River_year_j = paste0(RIVER_j, "_", Cohort_j )) %>%
  dplyr::select(iname, jname, River_year_i, River_year_j,
                RIVER_i, RIVER_j, relation, relation2,
                Cohort_i, Cohort_j ) %>%
  dplyr::filter(!is.na(Cohort_i)) %>%
  dplyr::filter(!is.na(Cohort_j))

group_data <- NWdata %>%
  dplyr::group_by( River_year_i, River_year_j, relation2) %>%
  dplyr::count()
nodes <- unique(c(group_data$River_year_i, group_data$River_year_j))

pwPOP <- data %>%
  dplyr::filter(relation == "HSP") %>%
  dplyr::mutate(Cohort_gap = pmax(Cohort_i,Cohort_j) - pmin(Cohort_i,Cohort_j)) %>%
  dplyr::filter(Cohort_overlap_relaxed)

vertices <- meta %>%
  dplyr::filter(id %in% c(pwPOP$iname,pwPOP$jname)) %>%
  dplyr::select(River, Cohort2) %>%
  dplyr::filter(!is.na(Cohort2)) %>%
  dplyr::group_by(River, Cohort2) %>%
  dplyr::count()
vertices$River <- as.character(vertices$River)
vertices$River[vertices$River == "Adelaide River"] <- "AR"
vertices$River[vertices$River == "Sampan Creek"] <- "SC"
vertices$River[vertices$River == "Wildman River"] <- "WR"
vertices$River[vertices$River == "West Alligator River"] <- "WAR"
vertices$River[vertices$River == "South Alligator River"] <- "SAR"
vertices$River[vertices$River == "East Alligator River"] <- "EAR"
vertices$label <- paste0(vertices$River, "_", vertices$Cohort2)
vertices <- vertices[, c(4,1,2,3)]

vertices$River <- factor(vertices$River,
                         levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))
vertices <- as_data_frame(vertices, what = "vertices")
network <- igraph::graph_from_data_frame(d = group_data, directed = TRUE,
                                         vertices = vertices )

layout <- ggraph::create_layout(network, layout = 'igraph',
                                circular = FALSE, algorithm = 'fr')
layout$y <- layout$Cohort2
layout$x <- factor(layout$River,
                   levels = c("AR", "SC", "WR", "WAR","SAR","EAR"))
```

```
kin_network5 <- ggraph::ggraph(network,  layout = layout) +
  ggraph::geom_edge_fan(
    aes(width = n,
        edge_colour = factor(relation2)),
    arrow = arrow(length = unit(3, 'mm')),
    end_cap = ggraph::circle(2, 'mm'),
    edge_alpha = 1) +
  ggraph::geom_edge_arc(aes(width = n,
                            edge_colour = factor(relation2)),
                        arrow = arrow(length = unit(3, 'mm')),
                  end_cap = ggraph::circle(2, 'mm'),
                  alpha = 0.8, strength = 0.1) +
  ggraph::geom_edge_loop(aes(width = n, strength = 0.9,
                             edge_colour = factor(relation2))) +
  ggraph::scale_edge_width(range = c(0.5, 1), name = "Number of kin") +
  ggraph::scale_edge_colour_manual(values =
                                     c("grey", "orange","blue"),
                                   name = "kin-type",
                                   aesthetics = "edge_colour",
                                   na.value = "grey50") +
  ggraph::geom_node_point(
    colour = "black",size = 2) +
  ggplot2::scale_colour_manual(values = adegenet::funky(6), na.value = "grey50") +
  ggplot2::scale_y_continuous(limits = c(1997, 2016), n.breaks = 19, minor_breaks = NULL) +
  ggplot2::scale_x_discrete(
    labels = c("AR" = "Adelaide River",
               "SC" = "Sampan Creek",
               "WR" = "Wildman River",
               "WAR" = "West Alligator River",
               "SAR" = "South Alligator River",
               "EAR" = "East Alligator River")) +
  ggplot2::ggtitle("Network between/within rivers for same-cohort HSPs") +
  ggplot2::theme_void() +
  ggplot2::theme_minimal() +
  ggplot2::theme(
    legend.position = "right",
    plot.margin = unit(rep(1,4), "cm"),
    axis.text.y = element_text(size = 10, angle = 0, hjust = 0),
    axis.text.x = element_text(size = 10, angle = 90, hjust = 1),
    axis.title.x = element_blank(),
    axis.title.y = element_blank()
  )
```

```
## Warning: Ignoring `graph` as layout is already calculated
## i Pass the calculated layout to the `graph` argument to silence this warning
```

```
print(kin_network5)
```




```
ggplot2::ggsave(filename = "0.Kinship_network_HSP_within_cohort.png",
                bg = "white", plot = kin_network5, device = "png",
                width = 40, height = 30, units = "cm")
```

10.5.1 Multiple paternity & litter size in females

Here we are looking for half siblings that were born in the same year and share the same haplotype (i.e. likely maternally related)

- 39 same-cohort HSPs with the same haplotype & 33 FSPs (46 family groups)
  - Of the 39 same-cohort HSPs with a matching haplotype, 25 have a haplotype with a frequency $< 50\%$ and could be paternally related.
  - 29 groups consist of pairs only (16 FSPs & 13 HSPs)
  - 12 groups consist of a combination of FSPs and HSPs, consisting of:
    - 2 HSP & 1 FSP (n=6)
    - 3 FSP (n=1)
    - 3 HSP (n=1)
    - 2 HSP & 1 unknown relationship (n=1) *Unknown relationship could be cousins or HSP that was not identified*
    - 2 HSP, 1 FSP & 3 unknown relationships (n=1)
    - 5 HSP, 1 FSP & 4 unknown relationships (n=1)
    - 4 HSP & 6 FSP (n=1)

Assuming that these are all born in the same cohort (however, we have uncertain in growth function) and share a mother (but common haplotypes could be inherited from two fathers by chance), then we identified multiple paternity in 24 out of the 41 litters (58%). Further, litter size estimates depend strongly on the sampling effort per year, which is not optimal for this study. However, we found that the majority consisted of 2 individuals (29 out of 41), with a maximum observed litter size of 5.

```
### SAME HAPLOTYPE
sum(pwdata$Cohort_overlap_relaxed & pwdata$HT_i == pwdata$HT_j, na.rm = TRUE) ## 72 same-cohort HSP
```

```
## [1] 72
```

```
  sum(pwdata$Cohort_overlap_relaxed &
      pwdata$relation == "HSP" &
        pwdata$HT_i == pwdata$HT_j,
      na.rm = TRUE) ## 39 same-cohort HSP
```

```
## [1] 39
```

```
  sum(pwdata$Cohort_overlap_relaxed & pwdata$relation == "FSP" &
        pwdata$HT_i == pwdata$HT_j, na.rm = TRUE) ## 33 same-cohort HSP
```

```
## [1] 33
```

```
sum(pwdata$Cohort_overlap_relaxed & pwdata$RIVER_i == pwdata$RIVER_j &
    pwdata$HT_i == pwdata$HT_j, na.rm = TRUE) ## 67 same-cohort, same-river HSP & FSP
```

```
## [1] 67
```

```
  sum(pwdata$Cohort_overlap_relaxed & pwdata$RIVER_i == pwdata$RIVER_j &
        pwdata$relation == "HSP" &
        pwdata$HT_i == pwdata$HT_j, na.rm = TRUE) ## 35 same-cohort, same-river HSP
```

```
## [1] 35
```

```
    sum(pwdata$Cohort_overlap_relaxed & pwdata$RIVER_i == pwdata$RIVER_j &
        pwdata$relation == "HSP" & pwdata$HTfreq_i > 0.47 &
        pwdata$HT_i == pwdata$HT_j, na.rm = TRUE) ## 25 same-cohort, same-river HSP have a high HT f
```

```
## [1] 25
```

```
  sum(pwdata$Cohort_overlap_relaxed & pwdata$RIVER_i == pwdata$RIVER_j &
        pwdata$relation == "FSP" &
        pwdata$HT_i == pwdata$HT_j, na.rm = TRUE) ## 32 same-cohort, same-river HSP
```

```
## [1] 32
```

```
knitr::kable(pwdata[pwdata$Cohort_overlap_relaxed &
                        !is.na(pwdata$Cohort_overlap_relaxed) &
                        pwdata$relation == "HSP" &
                        pwdata$HT_i == pwdata$HT_j &
                        !(is.na(pwdata$HT_i) | is.na(pwdata$HT_j))
            , -c(1:5,10,11,14,25)],
             format = "latex", row.names = FALSE) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
695	560	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2013	2013	2013
560	648	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2013	2013	2013	2011	2012	2012
910	837	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2010	2011
910	925	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2009	2010	2010
695	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2012	2012	2012	2013	2013
825	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2011	2012	2013	2013
840	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2012	2013	2013
840	1005	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2008	2009	2009
705	695	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2012	2012	2012	2011	2012	2012
765	705	M	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2012	2012	2012
905	945	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2010	2009	2009	2010
1030	1160	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2008	2009	2009	2005	2006	2007
1140	1160	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2006	2007	2008	2005	2006	2007
1030	1140	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2008	2009	2009	2006	2007	2008
842	1030	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2008	2008	2009
795	1030	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2008	2008	2009
740	950	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2012	2012	2009	2009	2010
745	950	M	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2012	2009	2009	2010
860	663	M	M	WAR	WAR	HT13	HT13	0.92683	0.92683	2011	2012	2012	2013	2014	2014
1080	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2010	2011
1080	865	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2011	2011
1080	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2010	2011
1070	1080	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2007	2008	2009
960	950	M	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2009	2009	2010	2009	2009	2010
910	670	F	M	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2012	2013	2013
750	910	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2012	2012	2010	2010	2011
855	675	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2012	2013	2013	2012	2012	2013
1064	935	M	M	SC	EAR	HT13	HT13	0.40000	0.27869	2010	2010	2011	2009	2010	2010
685	915	M	M	EAR	EAR	HT06	HT06	0.14754	0.14754	2012	2012	2013	2010	2010	2011
1070	935	M	F	SC	EAR	HT22	HT22	0.10000	0.01639	2007	2008	2009	2009	2010	2010
1000	775	F	M	EAR	EAR	HT25	HT25	0.40984	0.40984	2008	2009	2010	2011	2012	2012
1025	1170	M	F	WAR	WAR	HT13	HT13	0.92683	0.92683	2009	2010	2010	2007	2007	2008
795	585	M	M	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2012	2013	2013	2013
915	660	M	M	WR	WR	HT22	HT22	0.20000	0.20000	2012	2012	2013	2014	2014	2015
615	925	F	M	WAR	WAR	HT13	HT13	0.92683	0.92683	2013	2013	2013	2011	2011	2012
930	740	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2011	2012	2012	2013	2014	2014
930	932	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2011	2012	2012	2011	2012	2012
1070	895	F	F	EAR	SAR	HT25	HT25	0.40984	0.06931	2007	2008	2009	2009	2010	2010
1315	1358	F	M	EAR	SC	HT25	HT25	0.40984	0.23333	2001	2003	2004	2002	2004	2005

```
pwdata2 <- pwdata[(pwdata$Cohort_overlap_relaxed &
         !is.na(pwdata$Cohort_overlap_relaxed)) &
                  (pwdata$HT_i == pwdata$HT_j &
                     !is.na(pwdata$HT_i) & !is.na(pwdata$HT_j)),]


PWchain <- chain_pairwise(pwdata2)

PLOD <- list()
REL <- list()
for (i in 1:length(PWchain)) {
  PLOD[[i]] <- chain_with_plods(achain = PWchain[[i]],
                                all_plods = pwdata,
                                digits = 1L,
                                character. = TRUE)
  name1 <- c()
  for (c in 1:ncol(PWchain[[i]])) {
    row1 <- as.integer(colnames(PWchain[[i]])[c])
    if (!sum(pwdata$i %in% row1) == 0) {
      name1 <- c(name1, as.character(unique(pwdata$iname[pwdata$i %in%
                                                           row1])))
    } else {
      name1 <- c(name1, as.character(unique(pwdata$jname[pwdata$j %in%
                                                           row1])))
    }
  }
  rownames(PLOD[[i]]) <- name1
  colnames(PLOD[[i]]) <- name1
  REL[[i]] <-  chain_with_rel(achain = PWchain[[i]],
                              all_rel = pwdata,
                              character. = TRUE)
  rownames(REL[[i]]) <- name1
  colnames(REL[[i]]) <- name1
}
print(REL)
```

```
## [[1]]
##        GGA212 GGA182
## GGA212  .     FSP
## GGA182 FSP     .
##
## [[2]]
##        GGA340 GGA330
## GGA340  .     HSP
## GGA330 HSP     .
##
## [[3]]
##        GGA341 GGA239
## GGA341  .     FSP
## GGA239 FSP     .
##
## [[4]]
##        GGA343 GGA327
## GGA343  .     HSP
## GGA327 HSP     .
##
## [[5]]
##        GGA371 GGA114
```

```
## GGA371  .     FSP
## GGA114 FSP     .
##
## [[6]]
##        GGL033 GGL002
## GGL033  .     FSP
## GGL002 FSP     .
##
## [[7]]
##        GGS025 GGA019
## GGS025  .     FSP
## GGA019 FSP     .
##
## [[8]]
##        GGA022 GGA155
## GGA022  .     FSP
## GGA155 FSP     .
##
## [[9]]
##        GGA032 GGA086
## GGA032  .     HSP
## GGA086 HSP     .
##
## [[10]]
##        GGA112 GGA117
## GGA112  .     FSP
## GGA117 FSP     .
##
## [[11]]
##        GGA115 GGA119
## GGA115  .     FSP
## GGA119 FSP     .
##
## [[12]]
##        GGA116 GGA363
## GGA116  .     HSP
## GGA363 HSP     .
##
## [[13]]
##        GGA133 GGS018
## GGA133  .     HSP
## GGS018 HSP     .
##
## [[14]]
##        GGA147 GGA139
## GGA147  .     HSP
## GGA139 HSP     .
##
## [[15]]
##        GGA164 GGS006
## GGA164  .     HSP
## GGS006 HSP     .
##
## [[16]]
##        GGA166 GGA168
## GGA166  .     HSP
## GGA168 HSP     .
##
```

```
## [[17]]
##        GGA249 GGA250
## GGA249  .     FSP
## GGA250 FSP     .
##
## [[18]]
##        GGA356 GGA362
## GGA356  .     FSP
## GGA362 FSP     .
##
## [[19]]
##        GGS012 GGS004
## GGS012  .     FSP
## GGS004 FSP     .
##
## [[20]]
##        GGS017 GGA186
## GGS017  .     HSP
## GGA186 HSP     .
##
## [[21]]
##        GGA131 GGA267
## GGA131  .     FSP
## GGA267 FSP     .
##
## [[22]]
##        GGA214 GGA078
## GGA214  .     HSP
## GGA078 HSP     .
##
## [[23]]
##        GGA087 GGA256
## GGA087  .     FSP
## GGA256 FSP     .
##
## [[24]]
##        GGA080 GGA083
## GGA080  .     HSP
## GGA083 HSP     .
##
## [[25]]
##        GGA085 GGA038
## GGA085  .     FSP
## GGA038 FSP     .
##
## [[26]]
##        GGA365 GGA366
## GGA365  .     FSP
## GGA366 FSP     .
##
## [[27]]
##         GGSA014 GGA179
## GGSA014  .      HSP
## GGA179  HSP      .
##
## [[28]]
##        GGA175 GGA190
## GGA175  .     FSP
```

```
## GGA190 FSP     .
##
## [[29]]
##        GGA334 GGA234
## GGA334  .     HSP
## GGA234 HSP     .
##
## [[30]]
##        GGA059 GGA023 GGA095
## GGA059  .     HSP    HSP
## GGA023 HSP     .     FSP
## GGA095 HSP    FSP     .
##
## [[31]]
##        GGA061 GGA158 GGA048
## GGA061  .     HSP    HSP
## GGA158 HSP     .     FSP
## GGA048 HSP    FSP     .
##
## [[32]]
##        GGA066 GGA024 GGA027
## GGA066  .     FSP    FSP
## GGA024 FSP     .     FSP
## GGA027 FSP    FSP     .
##
## [[33]]
##        GGA018 GGA093 GGA036
## GGA018  .     HSP    FSP
## GGA093 HSP     .     HSP
## GGA036 FSP    HSP     .
##
## [[34]]
##        GGA020 GGA098 GGA221
## GGA020  .     FSP    HSP
## GGA098 FSP     .     HSP
## GGA221 HSP    HSP     .
##
## [[35]]
##        GGA030 GGA092 GGA258
## GGA030  .     HSP    FSP
## GGA092 HSP     .     HSP
## GGA258 FSP    HSP     .
##
## [[36]]
##        GGA033 GGA259 GGA226
## GGA033  .     HSP    HSP
## GGA259 HSP     .     HSP
## GGA226 HSP    HSP     .
##
## [[37]]
##        GGA354 GGA357 GGA361
## GGA354  .     HSP    HSP
## GGA357 HSP     .     FSP
## GGA361 HSP    FSP     .
##
## [[38]]
##        GGA228 GGA160 GGA219
## GGA228  .      .     HSP
```

```
## GGA160  .       .     HSP
## GGA219 HSP    HSP     .
##
## [[39]]
##        GGA370 GGA364 GGA355 GGA360
## GGA370  .       .       .     HSP
## GGA364  .       .     FSP     .
## GGA355  .     FSP     .     HSP
## GGA360 HSP     .     HSP     .
##
## [[40]]
##        GGA077 GGA026 GGA156 GGA045 GGA021
## GGA077  .       .       .     HSP    HSP
## GGA026  .       .     FSP     .     HSP
## GGA156  .     FSP     .       .     HSP
## GGA045 HSP     .       .       .     HSP
## GGA021 HSP    HSP    HSP    HSP     .
##
## [[41]]
##        GGA120 GGA076 GGA100 GGA257 GGA157
## GGA120  .     FSP    FSP    HSP    FSP
## GGA076 FSP     .     FSP    HSP    FSP
## GGA100 FSP    FSP     .     HSP    FSP
## GGA257 HSP    HSP    HSP     .     HSP
## GGA157 FSP    FSP    FSP    HSP     .
```

```
knitr::kable(pwdata2[,-c(1:5,10,11,14,25)],
         format = "latex") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

	LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
5	670	1085	M	F	SAR	SC	HT06	HT06	0.37624	0.13333	2012	2012	2012	2009	2010	2011
6	825	870	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2010	2010	2011	2010	2010	2011
7	795	870	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2010	2011	2011	2010	2010	2011
8	795	825	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2010	2011	2011	2010	2010	2011
9	1005	1243	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2008	2009	2009	2006	2007	2008
10	1140	1140	M	M	WAR	WAR	HT13	HT13	0.92683	0.92683	2006	2007	2008	2006	2007	2008
11	925	837	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2009	2010	2010	2010	2010	2011
12	880	865	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2011	2011
13	880	865	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2011	2011
14	880	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2010	2011
15	1070	865	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2011	2011
16	1070	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2010	2011
17	1070	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2010	2011
18	895	827	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2010	2011
19	750	690	F	M	SAR	SAR	HT13	HT13	0.51485	0.51485	2012	2012	2012	2012	2012	2012
20	740	570	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2012	2012	2013	2013	2013	2014
21	825	695	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2011	2011	2012	2012
22	765	695	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2011	2012	2012
23	842	795	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2010	2011	2011
24	805	915	M	F	EAR	EAR	HT13	HT13	0.27869	0.27869	2011	2011	2012	2010	2010	2011
25	780	740	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2013	2013	2014	2013	2014	2014
26	1060	1010	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2010	2010	2011	2010	2011	2011
27	695	648	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2011	2012	2012
28	745	740	M	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2012	2011	2012	2012
29	905	885	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2009	2010	2010	2010	2010	2011
30	1025	1155	M	F	EAR	EAR	HT25	HT25	0.40984	0.40984	2008	2009	2009	2006	2007	2007
31	1065	1065	F	F	WR	WR	HT22	HT22	0.20000	0.20000	2007	2008	2009	2007	2008	2009
32	915	920	M	F	WR	WR	HT22	HT22	0.20000	0.20000	2012	2012	2013	2011	2012	2012
33	840	870	F	M	WR	WR	HT13	HT13	0.68889	0.68889	2010	2011	2011	2010	2011	2011
34	900	950	M	M	WR	WR	HT13	HT13	0.68889	0.68889	2012	2012	2013	2011	2012	2012
36	1000	1225	M	F	AR	AR	HT06	HT06	0.52000	0.52000	2009	2009	2010	2006	2007	2008
37	1260	1304	M	M	SC	SC	HT25	HT25	0.23333	0.23333	2003	2004	2005	2004	2005	2007
38	770	855	F	F	WAR	WAR	HT13	HT13	0.92683	0.92683	2011	2012	2012	2011	2012	2012
39	695	560	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2013	2013	2013
41	560	648	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2013	2013	2013	2011	2012	2012
46	910	837	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2010	2010	2011
48	910	925	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2011	2009	2010	2010
52	695	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2012	2012	2012	2013	2013
53	825	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2011	2012	2013	2013
54	840	574	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2012	2013	2013
56	840	1005	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2008	2009	2009
65	705	695	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2012	2012	2012	2011	2012	2012
66	765	705	M	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2011	2012	2012	2012	2012	2012
69	905	945	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2010	2010	2009	2009	2010
75	1030	1160	F	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2008	2009	2009	2005	2006	2007
76	1140	1160	M	F	SAR	SAR	HT06	HT06	0.37624	0.37624	2006	2007	2008	2005	2006	2007
77	1030	1140	F	M	SAR	SAR	HT06	HT06	0.37624	0.37624	2008	2009	2009	2006	2007	2008
80	842	1030	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2008	2008	2009
81	795	1030	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2010	2011	2011	2008	2008	2009
82	740	950	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2012	2012	2009	2009	2010
83	745	950	M	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2011	2011	2012	2009	2009	2010
85	860	663	M	M	WAR	WAR	HT13	HT13	0.92683	0.92683	2011	2012	2012	2013	2014	2014
87	1080	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2010	2011
88	1080	865	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2011	2011
89	1080	880	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2010	2010	2011
90	1070	1080	F	F	SAR	SAR	HT13	HT13	0.51485	0.51485	2007	2008	2009	2007	2008	2009

10.5.2 Multiple mating events per breeding season in males

Here we are looking for half siblings that were born in the same year and a different haplotype (i.e. paternally related)

- 10 same-cohort HSPs with different haplotypes (8 family groups, including FSPs)
  - 6 groups consist of pairs only (6 HSPs)
  - 2 groups consist of a combination of FSPs and HSPs, consisting of:
    - 2 HSP & 1 FSP (n=2)

Assuming that these are all born in the same cohort (however, we have uncertainty in growth function) and share a father, then we see that 10 father may have successfully reproduced with a different females in a season. However, sperm storage by females has been observed in other shark species; this is en equally valid explanation for these results.

Notice that the father moved between rivers in the Alligator river system for 4 of the 10 HSPs.

```
### DIFFERENT HAPLOTYPE
sum(pwdata$Cohort_overlap_relaxed & pwdata$HT_i != pwdata$HT_j, na.rm = TRUE) ## 10 same-cohort HSP
```

```
## [1] 10
```

```
  sum(pwdata$Cohort_overlap_relaxed & pwdata$RIVER_i == pwdata$RIVER_j &
        pwdata$relation == "HSP" &
        pwdata$HT_i != pwdata$HT_j, na.rm = TRUE) ## 6 same-cohort, same-river HSP
```

```
## [1] 6
```

```
  sum(pwdata$Cohort_overlap_relaxed & pwdata$RIVER_i != pwdata$RIVER_j &
          pwdata$relation == "HSP" &
          pwdata$HT_i != pwdata$HT_j, na.rm = TRUE) ## 4 same-cohort, cross-river HSP
```

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

```
knitr::kable(pwdata[pwdata$Cohort_overlap_relaxed &
                        !is.na(pwdata$Cohort_overlap_relaxed) &
                        pwdata$relation == "HSP" &
                        pwdata$HT_i != pwdata$HT_j &
                        !(is.na(pwdata$HT_i) |is.na(pwdata$HT_j))
            , -c(1:5,10,11,14,25)],
             format = "latex", row.names = FALSE) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
880	837	F	F	SAR	SAR	HT06	HT13	0.37624	0.51485	2010	2011	2011	2010	2010	2011
880	925	F	F	SAR	SAR	HT06	HT13	0.37624	0.51485	2010	2011	2011	2009	2010	2010
827	700	F	M	SAR	SAR	HT13	HT06	0.51485	0.37624	2010	2010	2011	2012	2012	2012
895	700	F	M	SAR	SAR	HT13	HT06	0.51485	0.37624	2010	2010	2011	2012	2012	2012
830	965	M	F	SAR	EAR	HT06	HT07	0.37624	0.06557	2010	2011	2011	2009	2010	2010
825	935	M	F	SAR	SAR	HT13	HT25	0.51485	0.06931	2010	2011	2011	2009	2010	2010
1030	1050	M	M	EAR	EAR	HT25	HT06	0.40984	0.14754	2008	2009	2009	2008	2008	2009
1025	1100	M	M	WAR	EAR	HT13	HT06	0.92683	0.14754	2009	2010	2010	2007	2008	2008
705	840	F	M	SAR	WAR	HT06	HT13	0.37624	0.92683	2012	2013	2013	2012	2012	2012
815	795	M	F	SAR	EAR	HT06	HT13	0.37624	0.27869	2010	2011	2011	2011	2011	2012

```
pwdata2 <- pwdata[(pwdata$Cohort_overlap_relaxed &
         !is.na(pwdata$Cohort_overlap_relaxed)) &
                  (pwdata$HT_i !=  pwdata$HT_j&
                     !is.na(pwdata$HT_i) & !is.na(pwdata$HT_j)),]


PWchain <- chain_pairwise(pwdata2)

PLOD <- list()
REL <- list()
for (i in 1:length(PWchain)) {
  PLOD[[i]] <- chain_with_plods(achain = PWchain[[i]],
                                all_plods = pwdata,
                                digits = 1L,
                                character. = TRUE)
  name1 <- c()
  for (c in 1:ncol(PWchain[[i]])) {
    row1 <- as.integer(colnames(PWchain[[i]])[c])
    if (!sum(pwdata$i %in% row1) == 0) {
      name1 <- c(name1, as.character(unique(pwdata$iname[pwdata$i %in%
                                                           row1])))
    } else {
      name1 <- c(name1, as.character(unique(pwdata$jname[pwdata$j %in%
                                                           row1])))
    }
  }
  rownames(PLOD[[i]]) <- name1
  colnames(PLOD[[i]]) <- name1
  REL[[i]] <-  chain_with_rel(achain = PWchain[[i]],
                              all_rel = pwdata,
                              character. = TRUE)
  rownames(REL[[i]]) <- name1
  colnames(REL[[i]]) <- name1
}
print(REL)
```

```
## [[1]]
##        GGA204 GGA082
## GGA204  .     HSP
## GGA082 HSP     .
##
## [[2]]
##        GGA336 GGA266
## GGA336  .     HSP
## GGA266 HSP     .
##
## [[3]]
##        GGA148 GGA183
## GGA148  .     HSP
## GGA183 HSP     .
##
## [[4]]
##        GGA150 GGA096
## GGA150  .     HSP
## GGA096 HSP     .
##
## [[5]]
##        GGA101 GGA079
```

```
## GGA101  .     HSP
## GGA079 HSP     .
##
## [[6]]
##        GGA171 GGA330
## GGA171  .     HSP
## GGA330 HSP     .
##
## [[7]]
##        GGA022 GGA155 GGA063
## GGA022  .     FSP    HSP
## GGA155 FSP     .     HSP
## GGA063 HSP    HSP     .
##
## [[8]]
##        GGA020 GGA098 GGA216
## GGA020  .     FSP    HSP
## GGA098 FSP     .     HSP
## GGA216 HSP    HSP     .
```

```
knitr::kable(pwdata2[,-c(1:5,10,11,14,25)],
         format = "latex") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down") %>%
  kableExtra::landscape()
```

	LEN_i	LEN_j	SEX_i	SEX_j	RIVER_i	RIVER_j	HT_i	HT_j	HTfreq_i	HTfreq_j	Cohort_i_lower	Cohort_i	Cohort_i_upper	Cohort_j_lower	Cohort_j	Cohort_j_upper
43	880	837	F	F	SAR	SAR	HT06	HT13	0.37624	0.51485	2010	2011	2011	2010	2010	2011
44	880	925	F	F	SAR	SAR	HT06	HT13	0.37624	0.51485	2010	2011	2011	2009	2010	2010
57	827	700	F	M	SAR	SAR	HT13	HT06	0.51485	0.37624	2010	2010	2011	2012	2012	2012
58	895	700	F	M	SAR	SAR	HT13	HT06	0.51485	0.37624	2010	2010	2011	2012	2012	2012
68	830	965	M	F	SAR	EAR	HT06	HT07	0.37624	0.06557	2010	2011	2011	2009	2010	2010
97	825	935	M	F	SAR	SAR	HT13	HT25	0.51485	0.06931	2010	2011	2011	2009	2010	2010
116	1030	1050	M	M	EAR	EAR	HT25	HT06	0.40984	0.14754	2008	2009	2009	2008	2008	2009
121	1025	1100	M	M	WAR	EAR	HT13	HT06	0.92683	0.14754	2009	2010	2010	2007	2008	2008
125	705	840	F	M	SAR	WAR	HT06	HT13	0.37624	0.92683	2012	2013	2013	2012	2012	2012
130	815	795	M	F	SAR	EAR	HT06	HT13	0.37624	0.27869	2010	2011	2011	2011	2011	2012

11 Comparing mtDNA, nuDNA and kinship within VDG

Results When comparing fixation indices (Table 3; supplementary material section 11), both mitochondrial and nuclear markers had high and significant fixation values. Nonetheless, the PHIst values of the mtDNA were usually one order of magnitude higher than the Fst value of the nuclear SNPs, even between close sampling locations (e.g. East Alligator vs. South Alligator Rivers). In general, high fixation indices corresponded to few kin shared between locations, as was the case for the Adelaide and Wildman rivers. Vice versa, both PHIst and Fst values for Sampan Creek vs. East Alligator River were non-significant, suggesting high connectivity. This was corroborated by the six kin pairs between Sampan Creek and East Alligator River. Generally, we found that Sampan Creek and East Alligator River share many kin with other VDG rivers, relative to the number of kin pairs found within these rivers (Table 4; Fig.3; supplementary material section 10.4). Interestingly, we also observed some discordance between marker types. A non-significant PHIst, but significant Fst, was observed between King Sound vs. East Alligator River, and Sampan Creek vs. Wildman River. Nineteen of the 177 kin pairs were not caught independently (i.e. within two weeks of each other and likely born in the same cohort) and could potentially generate a false signal of genetic structure due to biased sampling toward family members (see Devloo-Delva et al. 2019; Feutry et al. 2017). Most of these individuals were from the Wildman and South Alligator Rivers, where the sample sizes were large. Removing one individual per kin pair did not affect the fixation indices (supplementary material section 11.2). A disconnect was also observed between fixation indices and the number of shared kin pairs. For example, we found three, six, and nine kin pairs shared between the West Alligator, South Alligator and East Alligator Rivers (Fig. 3), but both PHIst and Fst values were high and significant (Table 3).

Discussion Our study applied three different methods to assess fine-scale population structure at different timescales. We show that each geographic region forms its own genetically and demographically distinct population. Within VDG, we consider the Adelaide River and the Wildman River to host demographically independent entities, demonstrated by the higher fixation indices and the many kin pairs retained within these sampling locations. For the Adelaide River this can be explained by its geographic distance from suitable habitats, although it is not certain why sharks from the Wildman River are less likely to disperse. Demographic independence in an island model is often considered to lay around a migration rate <0.1 (Hastings 1993) or less than one migrant per generation (Spieth 1974), and Bentzen (1998) argued that significant fixation indices provide strong evidence of demographic independence. In this study, we also found reproductive connectivity between spatially close rivers in VDG (i.e. low fixation indices and cross-river HSPs). For example, the increased connectivity of Sampan Creek and the East Alligator River is not explained solely by geographic proximity. Low mating opportunities, food availability, or environmental fluctuations may be responsible for the relatively few same-river kin pairs in these sampling locations (Comins et al. 1980; Greenwood 1980; Hamilton and May 1977). Overall, when comparing the 26 cross-river HSPs to recent effective and adult population size estimates from VDG: Ne = 168 [16–1 411] (Feutry et al. 2020), Nadults = [582–1 116] (Bravington et al. 2019), we would expect these locations to be demographically dependent. Nonetheless, we see low, but significant fixation indices between most of these rivers. The small population sizes and powerful genomic markers (Waples and Gaggiotti 2006), or the spatiotemporal stochasticity of demographic parameters (such as population growth rate or reproductive success; Lowe and Allendorf 2010), may explain why significant differentiation is identified while these sampling locations are still demographically dependent. For example, Petit et al. (2001) showed that population structure can still be found in a bat, the Common Noctule (Nyctalus noctula), even if the number of migrants per generation is high (>50). Regarding our study, we consider each sampling region (King Sound, Cambridge Gulf, Daly River, VDG, and PNG) as separate populations in both the genetic and demographic sense, but we believe VDG forms a metapopulation where the sampling locations are variably connected.

11.1 Unbiased PHIst - Fst

Kin pairs from the same cohort that were sampled dependently (within 14 days) could influence the population signal (Feutry 2017, Waples and Anderson 2017 and Devloo-Delva 2019). Therefore we removed the 19 kin pairs that were not independent.

```
Nindep <- (!independent_catch & pwdata$relation == "FSP") |
  (!independent_catch & pwdata$relation == "HSP" & pwdata$Cohort_overlap_relaxed)
Nnames <- pwdata$iname[Nindep]
```

```
sum(unique(Nnames) %in% ggar.gt@data$id) ## 19
```

```
## [1] 19
```

```
ggar.gt.unbiased <- ggar.gt[ggar.gt@data$id[
  (!ggar.gt@data$id %in% Nnames) &
    ggar.gt@schemes$pop == "Van Diemen Gulf"],,drop = TRUE]
ggar.gt.unbiased@schemes <- ggar.gt.unbiased@schemes[
  (!ggar.gt.unbiased@schemes$id %in% Nnames) &
    ggar.gt.unbiased@schemes$pop == "Van Diemen Gulf",]


ggar.snp.gt.unbiased <- ggar.snp.gt[ggar.snp.gt@data$id[
  (!ggar.snp.gt@data$id %in% Nnames) &
    ggar.snp.gt@schemes$pop == "Van Diemen Gulf"],,drop = TRUE]
ggar.snp.gt.unbiased@schemes <- ggar.snp.gt.unbiased@schemes[
  (!ggar.snp.gt.unbiased@schemes$id %in% Nnames) &
    ggar.snp.gt.unbiased@schemes$pop == "Van Diemen Gulf",]
```

```
strata <- as.character(ggar.gt.unbiased@schemes$river)
names(strata) <- as.character(ggar.gt.unbiased@schemes$id)
strataG::setStrata(ggar.gt.unbiased) <- strata

popstr.mt.unbiased.river <-
  strataG::popStructTest(
    ggar.gt.unbiased,
    nrep = 10000,
    stats = "all",
    type = "both",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores()-1,
    write.output = TRUE
  )

strata <- as.character(ggar.snp.gt.unbiased@schemes$river)
names(strata) <- as.character(ggar.snp.gt.unbiased@schemes$id)
strataG::setStrata(ggar.snp.gt.unbiased) <- strata

popstr.snp.unbiased.river <-
  strataG::popStructTest(
    ggar.snp.gt.unbiased,
    nrep = 10000,
    stats = "all",
    type = "both",
    keep.null = FALSE,
    quietly = TRUE,
    max.cores = parallel::detectCores()-1,
    write.output = TRUE
  )
save(popstr.snp.unbiased.river,popstr.mt.unbiased.river,
     file = "6.Ggar_Fst_PHIst_objects_unbiased.Rdata")
```

11.2 Compare biased and unbiased fixation indices

```
load("6.Ggar_Fst_objects.Rdata")
load("6.Ggar_Fst_PHIst_objects_unbiased.Rdata")

mtdata <- popstr.mt.river$pairwise$result[,c(9,10)]
nudata <- popstr.snp.river$pairwise$result[,c(11,12)]
rm <- c("King Sound", "Ord River", "West Cambridge Gulf", "Daly River",
        "Papua New Guinea")
sub <- grepl(paste(rm,collapse = "|"), rownames(mtdata))
mtdata <- mtdata[!sub,]
nudata <- nudata[!sub,]

mtdata_unb <- popstr.mt.unbiased.river$pairwise$result[,c(9,10)]
nudata_unb <- popstr.snp.unbiased.river$pairwise$result[,c(11,12)]

mtdata <- cbind(mtdata, mtdata_unb)
colnames(mtdata) <- c("PHIst_bias", "PHIst.pval_bias","PHIst_unbias",
                      "PHIst.pval_unbias")
nudata <- cbind(nudata, nudata_unb)
colnames(nudata) <- c("Fst_bias", "Fst.pval_bias","Fst_unbias", "Fst.pval_unbias")


res1 <- pmin(as.character(pwdata$RIVER_i), as.character(pwdata$RIVER_j))
res2 <- pmax(as.character(pwdata$RIVER_i), as.character(pwdata$RIVER_j))
res.levels <- t(combn(unique(c(res1,res2) ), 2 ))
paste0(res.levels[,1], "vs",res.levels[,2])
```

```
## [1] "EARvsSAR" "EARvsWR" "EARvsWAR" "EARvsAR" "EARvsSC"
"SARvsWR"
## [7] "SARvsWAR" "SARvsAR" "SARvsSC" "WRvsWAR" "WRvsAR"
"WRvsSC"
## [13] "WARvsAR" "WARvsSC" "ARvsSC"
```

```
res.levels <- c("ARvsEAR",
                "ARvsSC",
                "ARvsSAR",
                "ARvsWAR",
                "ARvsWR",
                "EARvsSC",
                "EARvsSAR",
                "EARvsWAR",
                "EARvsWR",
                "SARvsSC",
                "WARvsSC",
                "SCvsWR",
                "SARvsWAR",
                "SARvsWR",
                "WRvsWAR")

res3 <- factor(paste0(res1, "vs",res2), levels = res.levels)
cross.river <- table(res3)

table(pwdata$RIVER_i[pwdata$RIVER_i == pwdata$RIVER_j])
```

```
##
##  AR EAR SAR  SC WAR  WR
##   9   9  79   2  10  37
```

	Kin.river_i	Kin.river_j	Xriver.kin	PHIst_bias	PHIst.pval_bias	PHIst_unbias	PHIst.pval_unbias	Fst_bias	Fst.pval_bias	Fst_unbias	Fst.pval_unbias
Adelaide River (25) v. East Alligator River (61)	13	28	1	0.2215	0.0001	0.2191	0.0002	0.0138	0.0001	0.0133	0.0001
Adelaide River (25) v. Sampan Creek (30)	13	10	0	0.2325	0.0002	0.2325	0.0002	0.0126	0.0001	0.0126	0.0001
Adelaide River (25) v. South Alligator River (101)	13	97	1	0.2494	0.0003	0.2357	0.0003	0.0146	0.0001	0.0146	0.0001
Adelaide River (25) v. West Alligator River (41)	13	20	1	0.7147	0.0001	0.7019	0.0001	0.0151	0.0001	0.0148	0.0001
Adelaide River (25) v. Wildman River (45)	13	40	1	0.4909	0.0001	0.4765	0.0001	0.0154	0.0001	0.0135	0.0001
East Alligator River (61) v. Sampan Creek (30)	28	10	6	0.0047	0.2881	0.0022	0.3270	0.0012	0.0497	0.0010	0.0899
East Alligator River (61) v. South Alligator River (101)	28	97	9	0.0823	0.0005	0.0754	0.0035	0.0023	0.0001	0.0021	0.0001
East Alligator River (61) v. West Alligator River (41)	28	20	3	0.3653	0.0001	0.3471	0.0001	0.0041	0.0001	0.0036	0.0001
East Alligator River (61) v. Wildman River (45)	28	40	0	0.1801	0.0001	0.1579	0.0003	0.0061	0.0001	0.0038	0.0001
Sampan Creek (30) v. South Alligator River (101)	10	97	1	0.0357	0.0754	0.0347	0.0723	0.0024	0.0030	0.0024	0.0026
Sampan Creek (30) v. West Alligator River (41)	10	20	0	0.2939	0.0001	0.2787	0.0001	0.0037	0.0001	0.0034	0.0005
Sampan Creek (30) v. Wildman River (45)	10	40	1	0.0807	0.0130	0.0658	0.0261	0.0059	0.0002	0.0042	0.0002
South Alligator River (101) v. West Alligator River (41)	97	20	6	0.2213	0.0001	0.2238	0.0003	0.0050	0.0001	0.0047	0.0001
South Alligator River (101) v. Wildman River (45)	97	40	1	0.1404	0.0004	0.1355	0.0007	0.0076	0.0001	0.0056	0.0001
West Alligator River (41) v. Wildman River (45)	20	40	0	0.1269	0.0016	0.1235	0.0026	0.0074	0.0001	0.0053	0.0001

```
table(pwdata$RIVER_i[pwdata$RIVER_i != pwdata$RIVER_j])
```

```
##
##  AR EAR SAR  SC WAR  WR
##   2  11   9   4   4   1
```

```
table(pwdata$RIVER_j[pwdata$RIVER_i != pwdata$RIVER_j])
```

```
##
##  AR EAR SAR  SC WAR  WR
##   2   8   9   4   6   2
```

```
table(c(pwdata$RIVER_i, pwdata$RIVER_j))
```

```
##
##  AR EAR SAR  SC WAR  WR
##  22  37 176  12  30  77
```

```
Total_kin_i <- c(13,13,13,13,13,28,28,28,28,10,10,10,97,97,20)
Total_kin_j <- c(28,10,97,20,40,10,97,20,40,97,20,40,20,40,40)
res4 <- data.frame(`Kin river_i` = Total_kin_i,
                   `Kin river_j` = Total_kin_j,
                   `Xriver kin` = as.integer(cross.river),
                   mtdata, nudata)
knitr::kable(res4,digits = 4,
             format = "latex") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "scale_down")
```

11.3 Compare kin distribution against fixation indices

```
coord <- readr::read_csv("VDG_coord.csv")
```

```
## Rows: 6 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): River
## dbl (2): lat, lon
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
geo.dist <- geodist::geodist(coord[,2:3])/1000
```

```
## Maximum distance is > 100km. The 'cheap' measure is inaccurate over such
## large distances, you'd likely be better using a different 'measure',
## one of 'haversine', 'vincenty', or 'geodesic'.
```

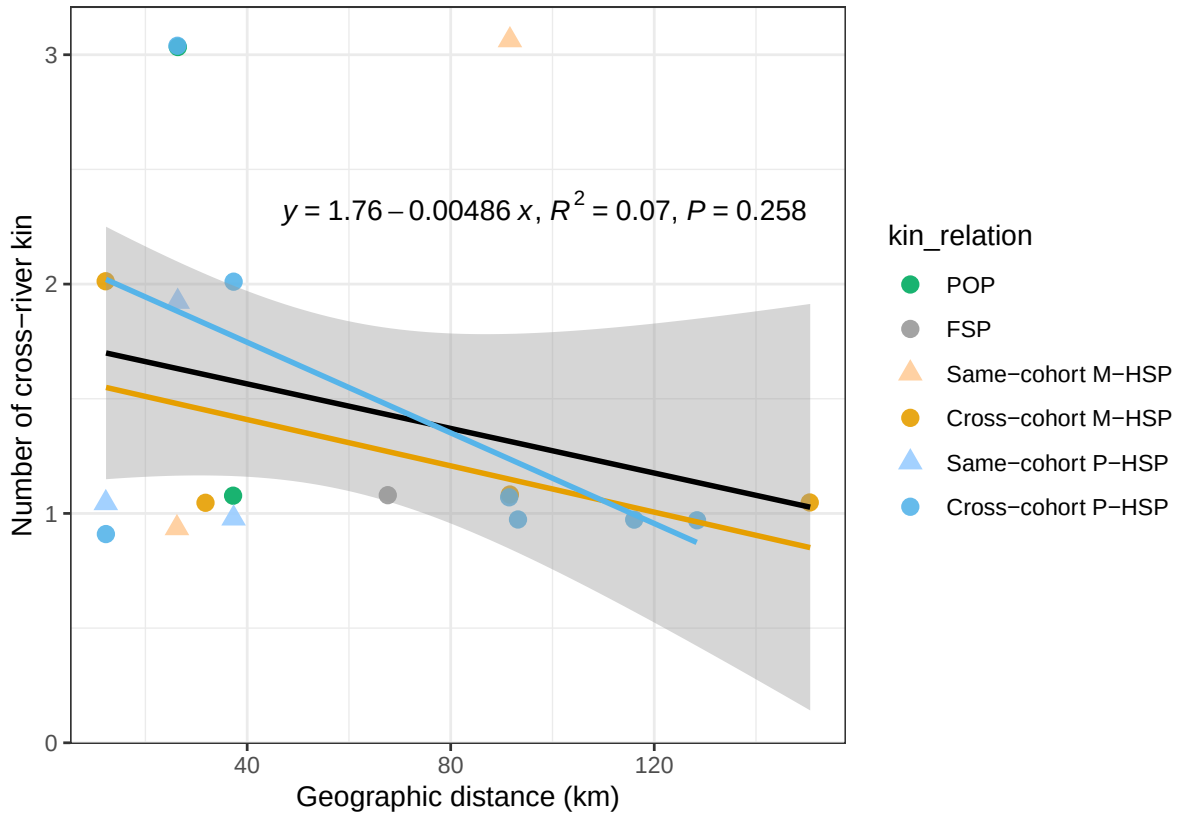
```
geo.dist <- as.matrix(geo.dist)
dimnames(geo.dist) <- list(coord$River, coord$River)
xy <- t(combn(colnames(geo.dist), 2))
distance <- data.frame(xy, dist = geo.dist[xy])

ord <- c(2,5,4,3,1,12,11,10,6,15,14,9,13,8,7)
distance <- distance[order(ord),]
rownames(res4)
```

```
## [1] "Adelaide River (25) v. East Alligator River (61)"
## [2] "Adelaide River (25) v. Sampan Creek (30)"
## [3] "Adelaide River (25) v. South Alligator River (101)"
## [4] "Adelaide River (25) v. West Alligator River (41)"
## [5] "Adelaide River (25) v. Wildman River (45)"
## [6] "East Alligator River (61) v. Sampan Creek (30)"
## [7] "East Alligator River (61) v. South Alligator River
(101)"
## [8] "East Alligator River (61) v. West Alligator River
(41)"
## [9] "East Alligator River (61) v. Wildman River (45)"
## [10] "Sampan Creek (30) v. South Alligator River (101)"
## [11] "Sampan Creek (30) v. West Alligator River (41)"
## [12] "Sampan Creek (30) v. Wildman River (45)"
## [13] "South Alligator River (101) v. West Alligator
River (41)"
## [14] "South Alligator River (101) v. Wildman River (45)"
## [15] "West Alligator River (41) v. Wildman River (45)"
```

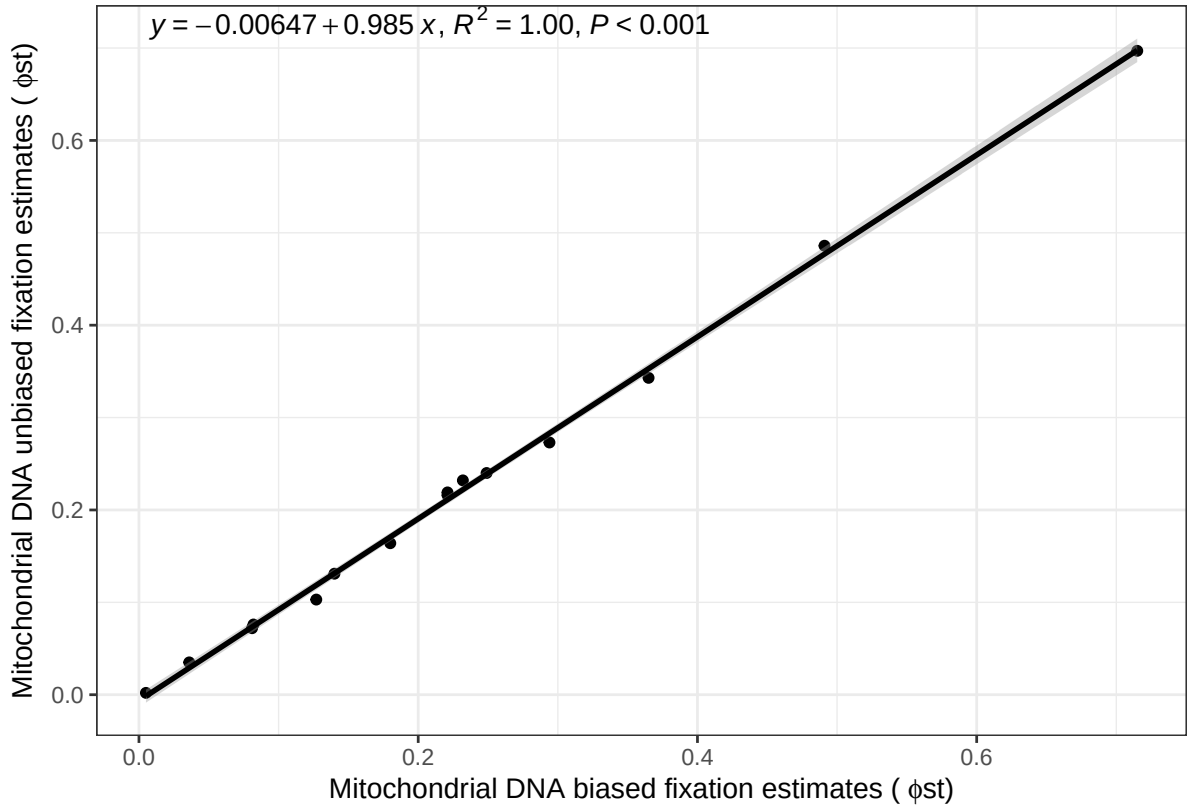
```
res5 <- data.frame(res4, distance$dist)

kin.fst <- readr::read_csv("Kin-Fst_data2.csv")
```

```
## Rows: 15 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): comparison
## dbl (24): Geographic.distance, n1, n2, Number of pairwise comparisons, Total...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

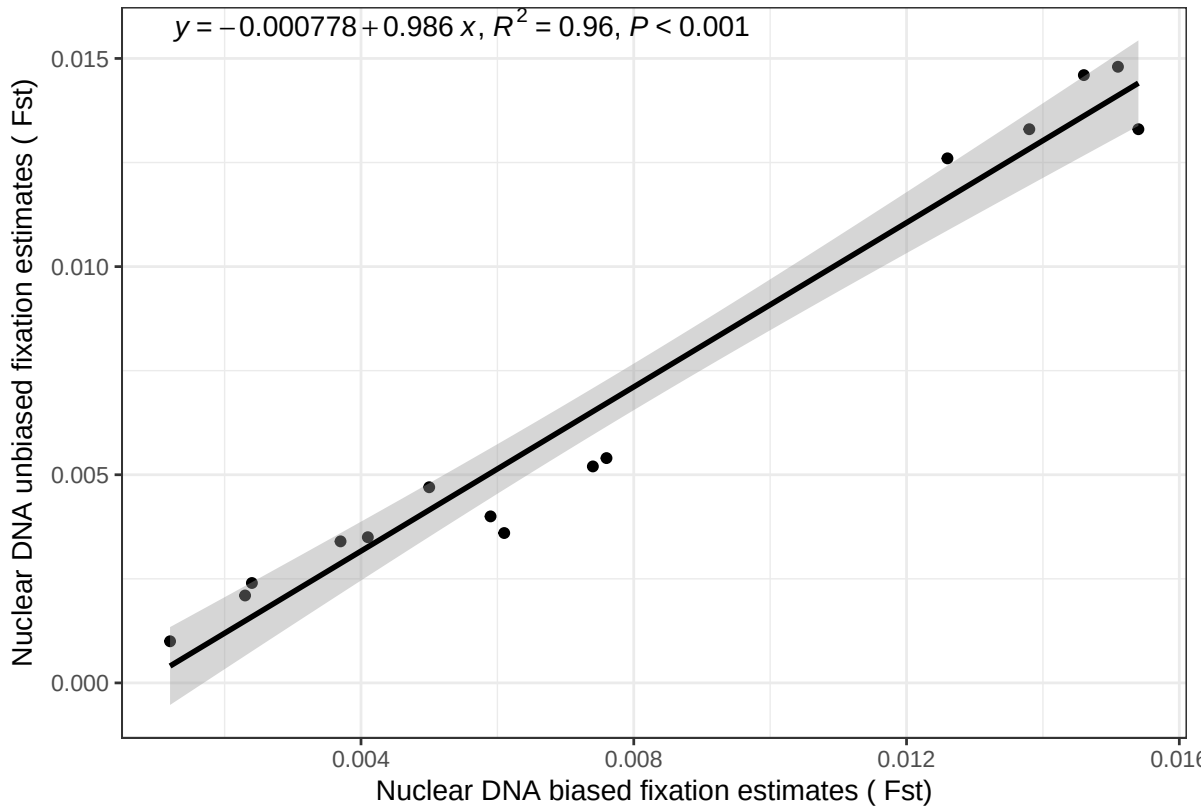
```
kin.fst$PHIst.p.adj <- p.adjust(kin.fst$PHIst.pval_unbias, "bonferroni")
kin.fst$Fst.p.adj <- p.adjust(kin.fst$Fst.pval_unbias, "bonferroni")


kin.fst2 <- tidyr::gather(kin.fst, kin_relation,number, POP:`Same-cohort P-HSP`)
kin.fst2$kin_relation <- factor(kin.fst2$kin_relation,
                                levels = c("POP","FSP", "Same-cohort M-HSP",
                                           "Cross-cohort M-HSP","Same-cohort P-HSP",
                                           "Cross-cohort P-HSP"))


plot <- ggplot2::ggplot(kin.fst2, ggplot2::aes(x = Geographic.distance, y = number)) +
  ggplot2::geom_jitter(ggplot2::aes(colour = kin_relation, shape = kin_relation),
                       width = 0.1, height = 0.1, size = 3, alpha = 0.9) +
  ggpmisc::stat_poly_line(method = "lm", colour = "black") +
  ggpmisc::stat_poly_eq(label.x = "right", label.y = 0.75,
                        ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::geom_smooth(data = subset(kin.fst2, subset = kin_relation == "Cross-cohort M-HSP"),
                       ggplot2::aes(x = Geographic.distance, y = number),se = FALSE,
                       method = 'lm', formula = y~x, colour = "#E69F00") +
  ggplot2::geom_smooth(data = subset(kin.fst2, subset = kin_relation == "Cross-cohort P-HSP"),
                       ggplot2::aes(x = Geographic.distance, y = number),se = FALSE,
                       method = 'lm', formula = y~x, colour = "#56B4E9") +
  ggplot2::xlab("Geographic distance (km)") +
  ggplot2::ylab("Number of cross-river kin") +
  ggplot2::scale_shape_manual(values = c(16, 16, 17, 16, 17, 16)) +
  ggplot2::scale_color_manual(values = c("#00ab61","#999999", "#FFCC99","#E69F00",
                                         "#99CCFF","#56B4E9")) +
  ggplot2::theme_bw()
print(plot)
```



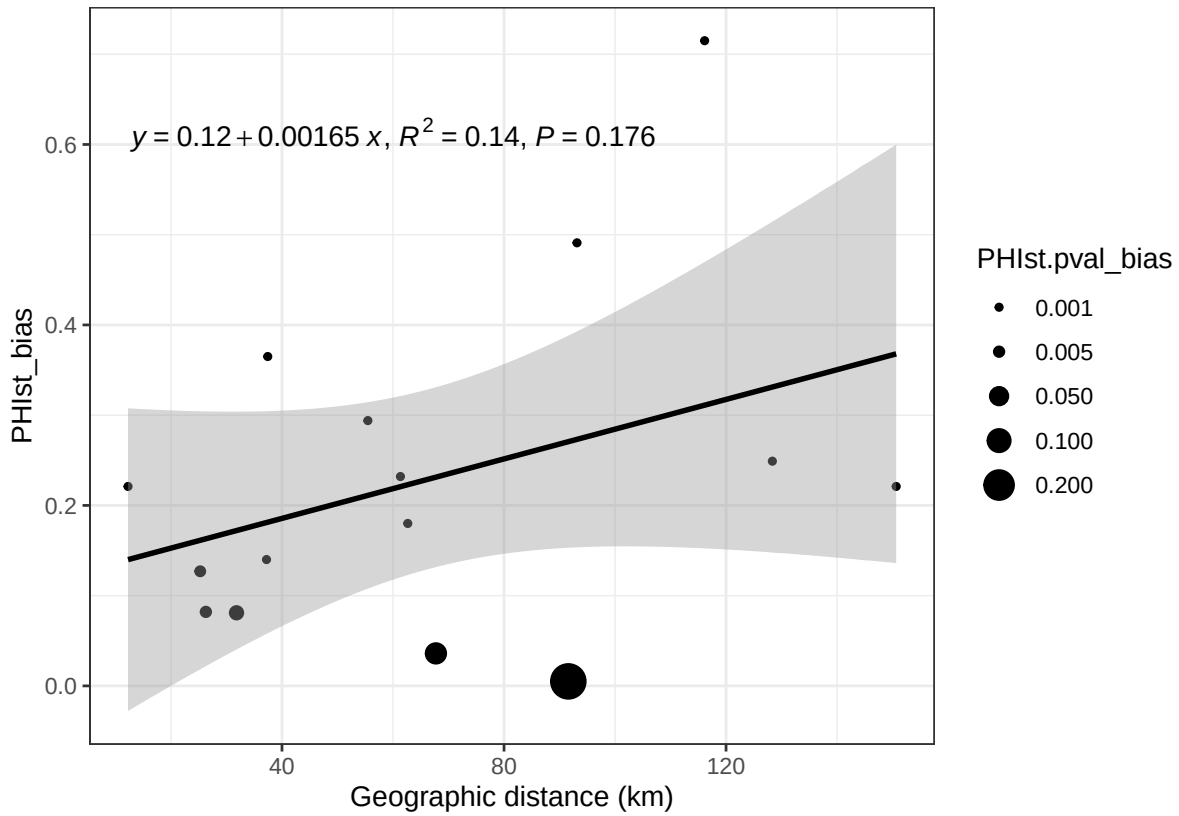
```
ggplot2::ggsave(plot = plot, filename = "Kin_distribtion_by_geographic_distance.png",
                device = "png", width = 30, height = 15, units = "cm")



# The removal of biased kin pairs does not affect the PHIst and FST
ggplot2::ggplot(kin.fst, ggplot2::aes(x = PHIst_bias, y = PHIst_unbias)) +
  ggplot2::geom_point() +
  ggpmisc::stat_poly_line(method = "lm", colour = "black") +
  ggpmisc::stat_poly_eq( label.x = "left", label.y = 1,
                         ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::xlab(expression(paste("Mitochondrial DNA biased fixation estimates ( ", phi, "st)"))) +
  ggplot2::ylab(expression(paste("Mitochondrial DNA unbiased fixation estimates ( ", phi, "st)"))) +
  ggplot2::theme_bw()
```



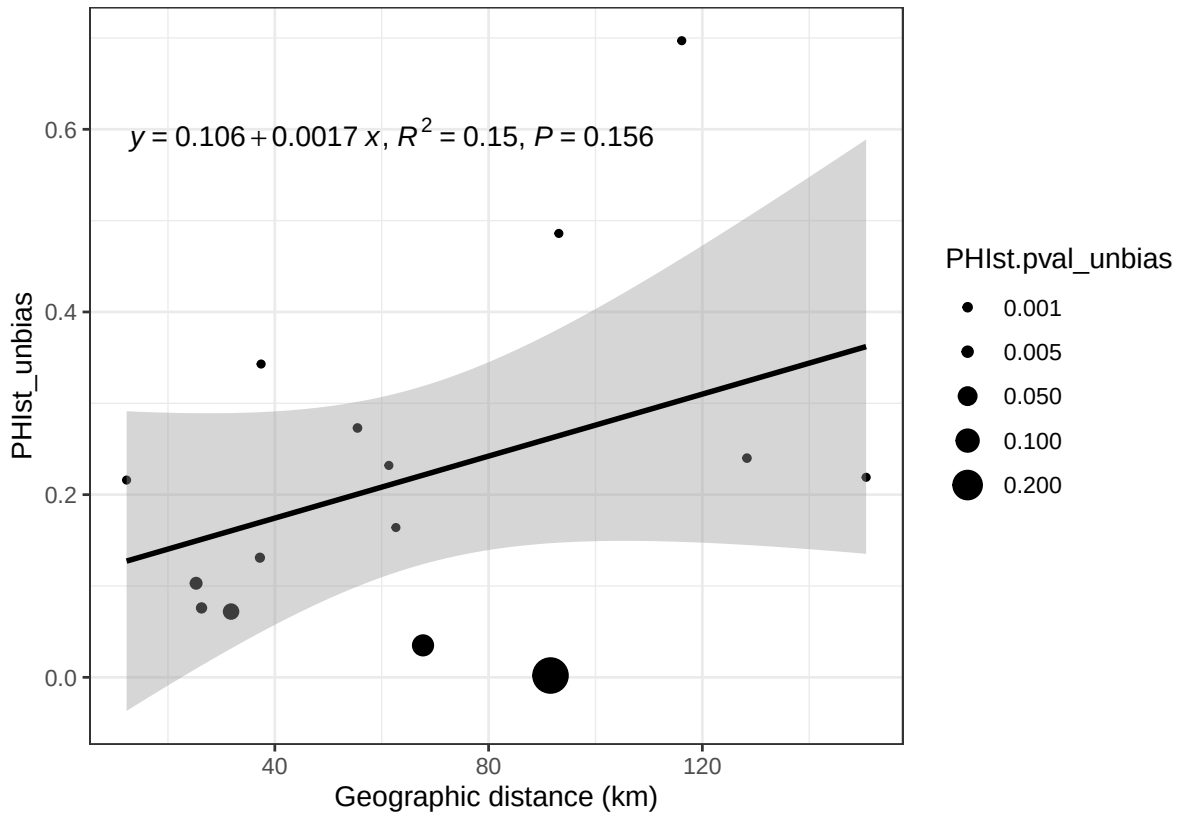

```
ggplot2::ggplot(kin.fst, ggplot2::aes(x = Fst_bias, y = Fst_unbias)) +
  ggplot2::geom_point() +
  ggpmisc::stat_poly_line(method = "lm", colour = "black") +
  ggpmisc::stat_poly_eq( label.x = "left", label.y = 1,
                         ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::xlab(expression(paste("Nuclear DNA biased fixation estimates ( ", F, "st)"))) +
  ggplot2::ylab(expression(paste("Nuclear DNA unbiased fixation estimates ( ", F, "st)"))) +
  ggplot2::theme_bw()
```



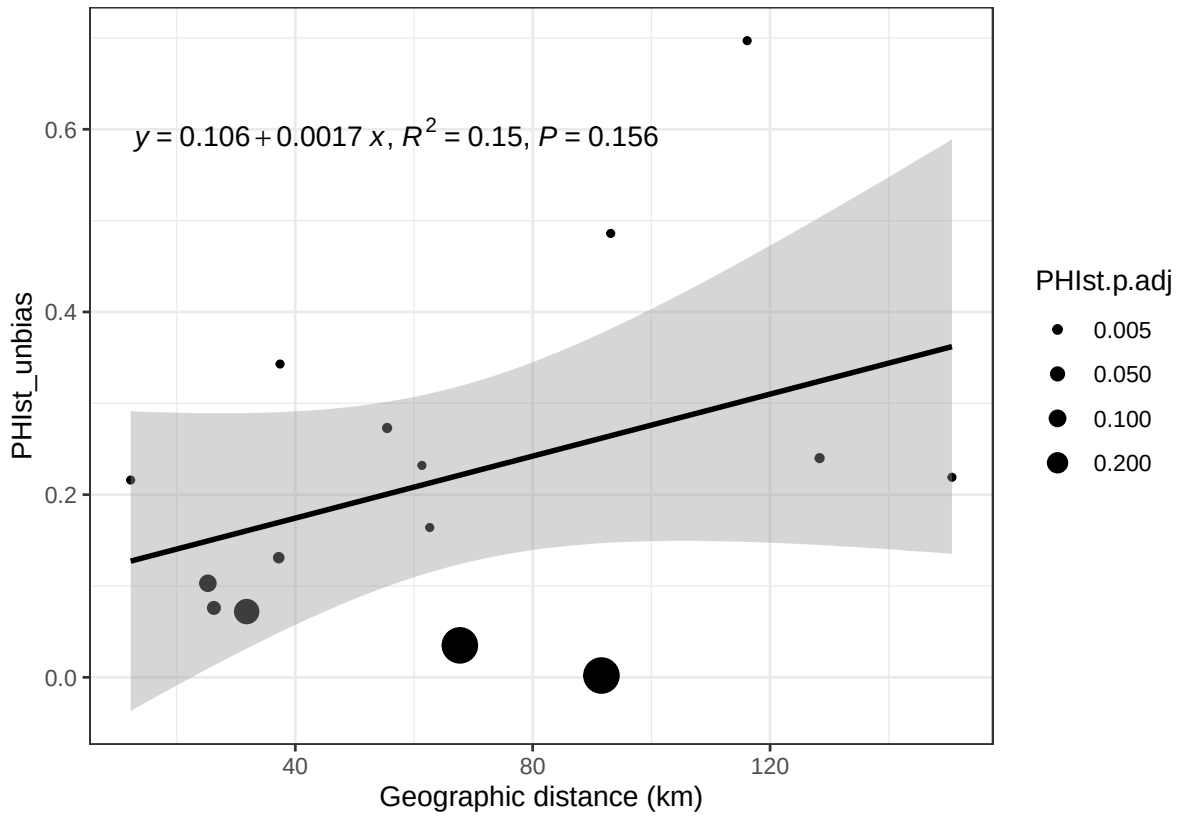

```
## TRYING TO PLOT PHIST/FST/KIN by geographic distance
kin.fst3 <- tidyr::gather(kin.fst, "Marker", "fixation_index",
                          PHIst_bias, PHIst_unbias, Fst_bias, Fst_unbias)
# kin.fst2 <- tidyr::gather(kin.fst2, "Marker.pval", "pvalue",
#                           PHIst.pval_bias, PHIst.pval_unbias,
#                           Fst.pval_bias, Fst.pval_unbias,
#                           PHIst.p.adj,Fst.p.adj)

ggplot2::ggplot(kin.fst,
                ggplot2::aes(x = Geographic.distance, y = PHIst_bias)) +
  ggplot2::geom_point(ggplot2::aes(size =  PHIst.pval_bias), colour = "black") +
  ggpmisc::stat_poly_line(method = "lm", colour = "black") +
  ggpmisc::stat_poly_eq(label.x = "left", label.y = 0.85,
                        ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::scale_size_continuous(breaks = c(0.001, 0.005, 0.050, 0.100,0.200)) +
  ggplot2::xlab("Geographic distance (km)") +
  ggplot2::theme_bw()
```



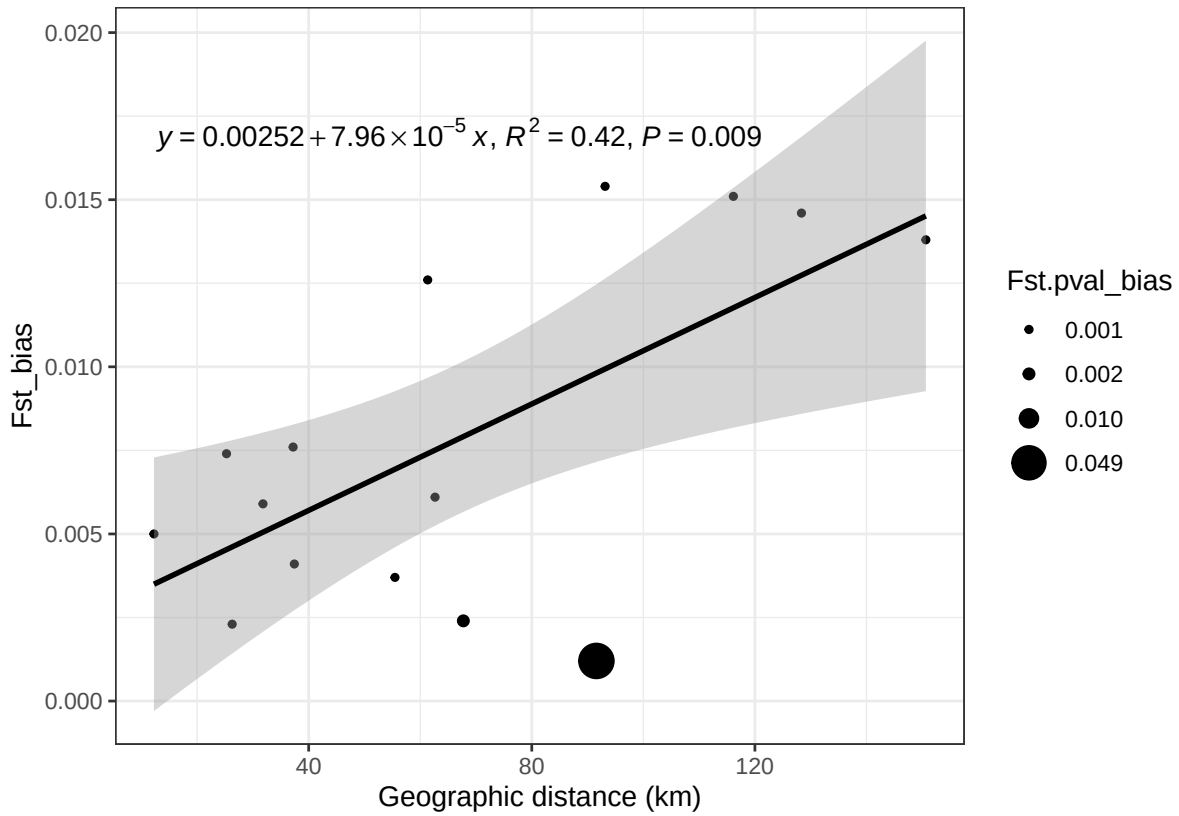

```
ggplot2::ggplot(kin.fst,
                ggplot2::aes(x = Geographic.distance, y = PHIst_unbias)) +
  ggplot2::geom_point(ggplot2::aes(size =  PHIst.pval_unbias)) +
  ggpmisc::stat_poly_line(method = "lm", colour = "black") +
  ggpmisc::stat_poly_eq(label.x = "left", label.y = 0.85,
                        ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::scale_size_continuous(breaks = c(0.001, 0.005, 0.050, 0.100,0.200)) +
  ggplot2::xlab("Geographic distance (km)") +
  ggplot2::theme_bw()
```



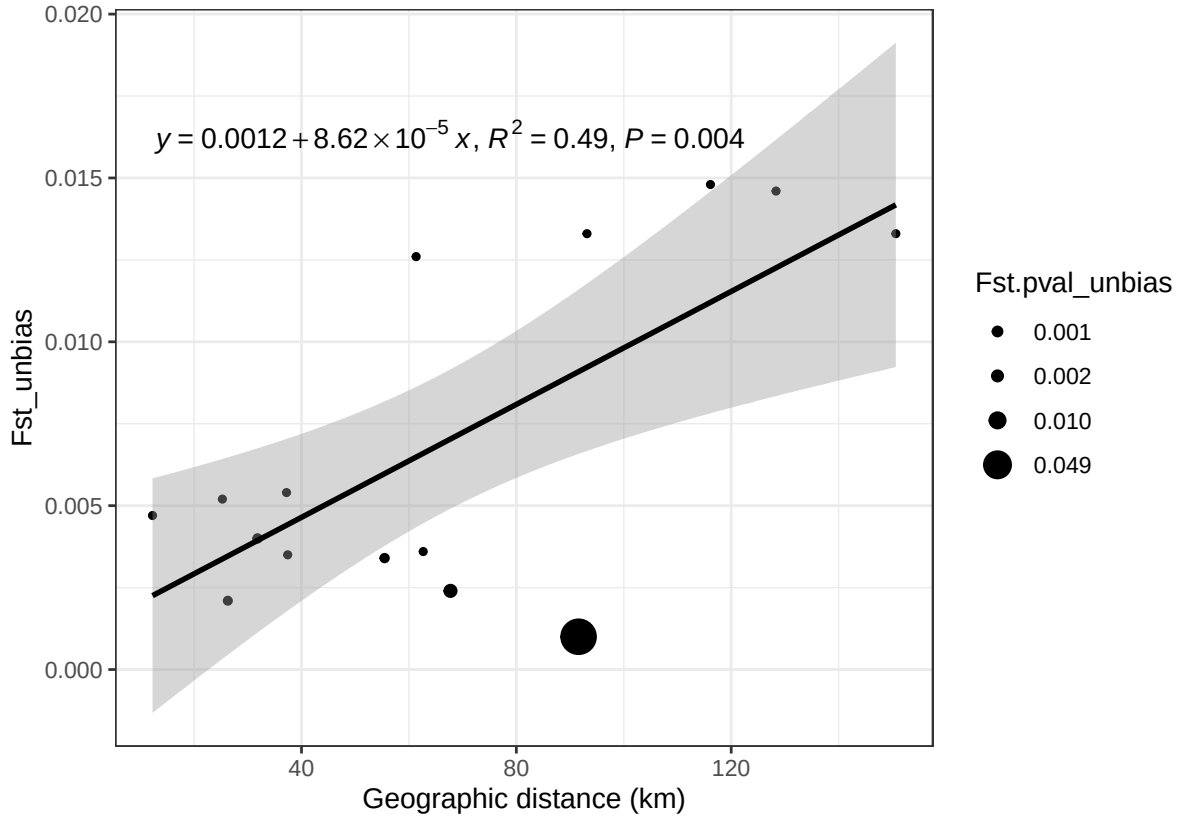

```
ggplot2::ggplot(kin.fst,
                ggplot2::aes(x = Geographic.distance, y = PHIst_unbias)) +
  ggplot2::geom_point(ggplot2::aes(size =  PHIst.p.adj)) +
  ggpmisc::stat_poly_line(method = "lm", colour = "black") +
  ggpmisc::stat_poly_eq(label.x = "left", label.y = 0.85,
                        ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::scale_size_continuous(breaks = c(0.001, 0.005, 0.050, 0.100,0.200)) +
  ggplot2::xlab("Geographic distance (km)") +
  ggplot2::theme_bw()
```



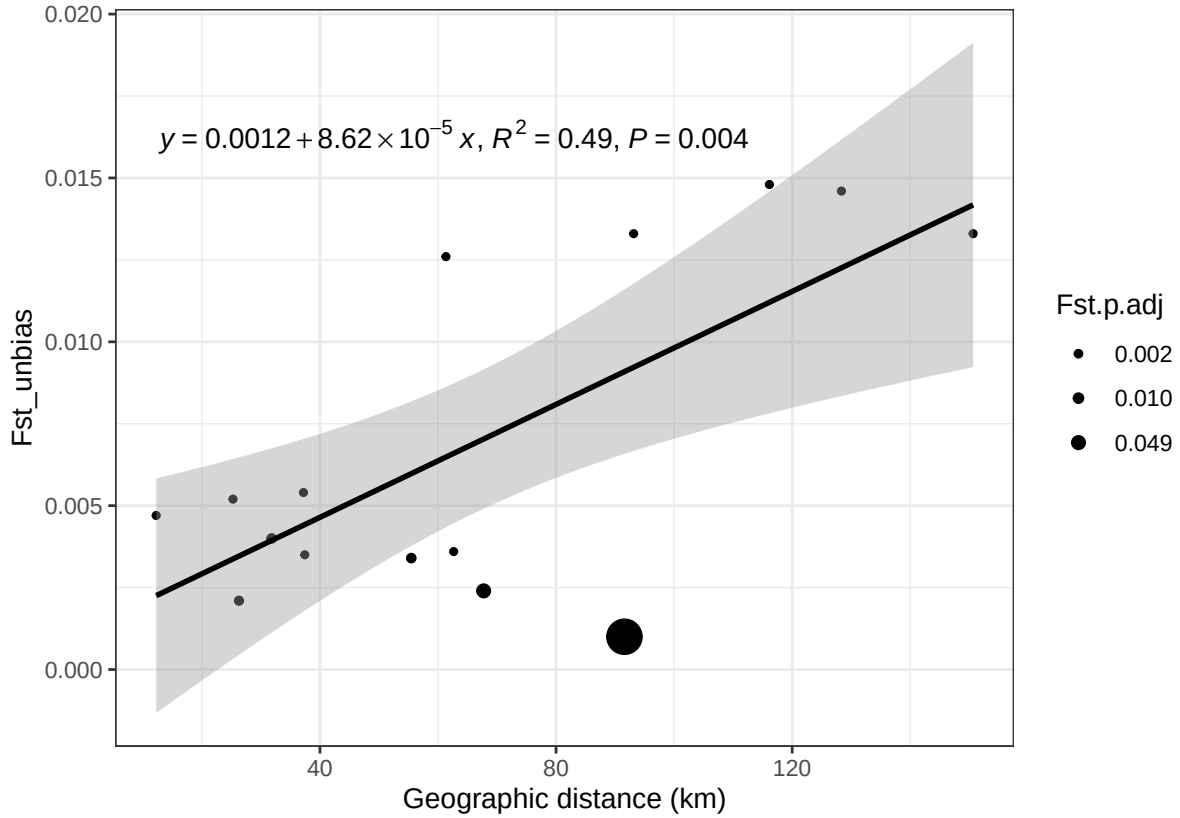

```
ggplot2::ggplot(kin.fst,
                ggplot2::aes(x = Geographic.distance, y = Fst_bias)) +
  ggplot2::geom_point(ggplot2::aes(size =  Fst.pval_bias)) +
  ggpmisc::stat_poly_line(method = "lm", colour = "black") +
  ggpmisc::stat_poly_eq(label.x = "left", label.y = 0.85,
                        ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::scale_size_continuous(breaks = c(0.001, 0.002, 0.01, 0.049)) +
  ggplot2::xlab("Geographic distance (km)") +
  ggplot2::theme_bw()
```




```
ggplot2::ggplot(kin.fst,
                ggplot2::aes(x = Geographic.distance, y = Fst_unbias)) +
  ggplot2::geom_point(ggplot2::aes(size =  Fst.pval_unbias)) +
  ggpmisc::stat_poly_line(method = "lm", colour = "black") +
  ggpmisc::stat_poly_eq(label.x = "left", label.y = 0.85,
                        ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::scale_size_continuous(breaks = c(0.001, 0.002, 0.01, 0.049)) +
  ggplot2::xlab("Geographic distance (km)") +
  ggplot2::theme_bw()
```




```
ggplot2::ggplot(kin.fst,
                ggplot2::aes(x = Geographic.distance, y = Fst_unbias)) +
  ggplot2::geom_point(ggplot2::aes(size =  Fst.p.adj)) +
  ggpmisc::stat_poly_line(method = "lm", colour = "black") +
  ggpmisc::stat_poly_eq(label.x = "left", label.y = 0.85,
                        ggpmisc::use_label(c("eq", "R2", "p"))) +
  ggplot2::scale_size_continuous(breaks = c(0.001, 0.002, 0.01, 0.049)) +
  ggplot2::xlab("Geographic distance (km)") +
  ggplot2::theme_bw()
```


0.020
0.015
0.010
0.005
0.000
Fst_unbias
$y = 0.0012 + 8.62 \times 10^{-5}\, x$, $R^2 = 0.49$, $P = 0.004$
40
80
120
Geographic distance (km)
Fst.p.adj
0.002
0.010
0.049

12 References

Bravington MV, Feutry P, Pillans RD, Hillary RM, Johnson G, Saunders T et al. (2019). Report to the National Environmental Science Program, Marine Biodiversity Hub. CSIRO Oceans & Atmosphere: Hobart, Australia.

Brée, Baptiste, Fabien L. Condamine, and Guillaume Guinot. "Combining palaeontological and neontological data shows a delayed diversification burst of carcharhiniform sharks likely mediated by environmental change." Scientific Reports 12.1 (2022): 21906.

Devloo-Delva F, Maes GE, Hernández SI, McAllister JD, Gunasekera RM, Grewe PM et al. (2019). Accounting for kin sampling reveals genetic connectivity in Tasmanian and New Zealand school sharks, Galeorhinus galeus. Ecol Evol 9: 4465-4472.

Feutry P, Berry O, Kyne PM, Pillans RD, Hillary RM, Grewe PM et al. (2017). Inferring contemporary and historical genetic connectivity from juveniles. Mol Ecol 26(2): 444-456.

Feutry P, Devloo-Delva F, Tran Lu Y A, Mona S, Rasanthi GM, Grant J et al. (2020). One panel to rule them all: DArTcap genotyping for population structure, historical demography, and kinship analyses, and its application to a threatened shark. Mol Ecol Resour 20: 1470–1485

Galván-Tirado, C., Díaz-Jaimes, P., García-de León, F. J., Galván-Magaña, F., & Uribe-Alcocer, M. (2013). Historical demography and genetic differentiation inferred from the mitochondrial DNA of the silky shark (Carcharhinus falciformis) in the Pacific Ocean. Fisheries Research, 147, 36-46.

Waples, R.S. , and Anderson, E.C. (2017) Purging putative siblings from population genetic data sets: a cautionary view. Mol Ecol, 26, 1211-1224.

13 Citations for packages

```
packages <-
  c(
  "knitr",
  "adegenet",
  "SNPRelate",
  "radiator",
  "strataG",
  "plyr",
  "ggplot2",
  "dartRverse",
  "coda",
  "ggtree"
  )
for (P in packages) {
  p <- utils::citation(P)
  print(strwrap(attr(unclass(p)[[1]], "textVersion"), width = 0.7 *
                  getOption("width"),
          indent = 0,exdent = 3, prefix = "", simplify = TRUE))
  cat("\n\n")
}
```

```
## 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. (2020). radiator: RADseq Data
Exploration,"
## [2] " Manipulation and Visualization using R. R package"
## [3] " version 1.1.9"
## [4] " https://thierrygosselin.github.io/radiator/. doi
:"
## [5] " 10.5281/zenodo.3687060"
##
##
## [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] "Hadley Wickham (2011). The Split-Apply-Combine
Strategy"
## [2] " for Data Analysis. Journal of Statistical
Software,"
## [3] " 40(1), 1-29. URL
https://www.jstatsoft.org/v40/i01/."
##
##
## [1] "H. Wickham. ggplot2: Elegant Graphics for Data"
## [2] " Analysis. Springer-Verlag New York, 2016."
##
##
## [1] "Gruber, B, Unmack, PJ, Berry, OF, Georges, A.
(2018)."
## [2] " dartr: An r package to facilitate analysis of SNP"
## [3] " data generated from reduced representation genome"
## [4] " sequencing. Molecular Ecology Resources 18:
691-699."
## [5] " https://doi.org/10.1111/1755-0998.12745"
##
##
## [1] "Martyn Plummer, Nicky Best, Kate Cowles and Karen
Vines"
## [2] " (2006). CODA: Convergence Diagnosis and Output"
## [3] " Analysis for MCMC, R News, vol 6, 7-11"
##
##
## [1] "Guangchuang Yu. (2022). Data Integration,
Manipulation"
## [2] " and Visualization of Phylogenetic Trees (1st"
## [3] " edition). Chapman and Hall/CRC."
## [4] " doi:10.1201/9781003279242"
```

14 Session info

```
devtools::session_info()
```

```
## - Session info
-------------------------------------------------------------
## setting value
## version R version 4.4.0 (2024-04-24 ucrt)
## os Windows 11 x64 (build 22631)
## system x86_64, mingw32
## ui RTerm
## language (EN)
## collate English_Australia.utf8
## ctype English_Australia.utf8
## tz Australia/Hobart
## date 2025-02-02
## pandoc 3.1.11 @ C:/Program
Files/RStudio/resources/app/bin/quarto/bin/tools/ (via
rmarkdown)
##
## - Packages
-------------------------------------------------------------------
## ! package * version date (UTC) lib source
## ade4 * 1.7-22 2023-02-06 [1] CRAN (R 4.4.0)
## adegenet * 2.1.10 2023-01-26 [1] CRAN (R 4.4.1)
## ADGofTest 0.3 2011-12-28 [1] CRAN (R 4.4.0)
## ape * 5.8 2024-04-11 [1] CRAN (R 4.4.0)
## apex * 1.0.6 2024-01-31 [1] CRAN (R 4.4.0)
## aplot 0.2.4 2024-12-17 [1] CRAN (R 4.4.0)
## arsenal 3.6.3 2021-06-04 [1] CRAN (R 4.4.1)
## assignPOP * 1.3.0 2024-03-13 [1] CRAN (R 4.4.0)
## atease 1.0.32 2024-09-27 [1] local
## babette * 2.3.4 2024-06-25 [1] CRAN (R 4.4.2)
## backports 1.5.0 2024-05-23 [1] CRAN (R 4.4.0)
## base64enc 0.1-3 2015-07-28 [1] CRAN (R 4.4.0)
## bdskytools * 0.0.1.0 2024-11-27 [1] Github
(laduplessis/bdskytools@c474052)
## beastier * 2.5.2 2024-10-06 [1] CRAN (R 4.4.2)
## beastio * 0.3.3 2024-11-27 [1] Github
(laduplessis/beastio@ff276c2)
## beautier * 2.6.12 2024-04-01 [1] CRAN (R 4.4.2)
## bit 4.5.0.1 2024-12-03 [1] CRAN (R 4.4.2)
## bit64 4.5.2 2024-09-22 [1] CRAN (R 4.4.1)
## bitops 1.0-9 2024-10-03 [1] CRAN (R 4.4.1)
## brio 1.1.5 2024-04-24 [1] CRAN (R 4.4.1)
## cachem 1.1.0 2024-05-16 [1] CRAN (R 4.4.0)
## caret 7.0-1 2024-12-10 [1] CRAN (R 4.4.2)
## caTools 1.18.3 2024-09-04 [1] CRAN (R 4.4.2)
## checkmate 2.3.2 2024-07-29 [1] CRAN (R 4.4.1)
## class 7.3-22 2023-05-03 [2] CRAN (R 4.4.0)
## cli 3.6.3 2024-06-21 [1] CRAN (R 4.4.1)
## cluster 2.1.6 2023-12-01 [2] CRAN (R 4.4.0)
## coda * 0.19-4.1 2024-01-31 [1] CRAN (R 4.4.0)
## codetools 0.2-20 2024-03-31 [2] CRAN (R 4.4.0)
## colorspace 2.1-0 2023-01-23 [1] CRAN (R 4.4.0)
## confintr 1.0.2 2023-06-04 [1] CRAN (R 4.4.2)
## copula 1.1-4 2024-08-17 [1] CRAN (R 4.4.2)
## crayon 1.5.3 2024-06-20 [1] CRAN (R 4.4.0)
```

```
## dartR.base * 0.98 2024-09-19 [1] CRAN (R 4.4.1)
## dartR.captive * 0.75 2023-11-27 [1] CRAN (R 4.4.1)
## dartR.data * 1.0.8 2024-05-30 [1] CRAN (R 4.4.0)
## dartR.popgen * 1.0.0 2024-06-27 [1] CRAN (R 4.4.1)
## dartR.sexlinked * 1.0.5 2024-06-24 [1] CRAN (R 4.4.1)
## dartR.sim * 0.70 2023-11-20 [1] CRAN (R 4.4.1)
## dartR.spatial * 0.78 2023-11-15 [1] CRAN (R 4.4.1)
## dartRverse * 1.0.2 2024-05-24 [1] CRAN (R 4.4.1)
## data.table * 1.16.4 2024-12-06 [1] CRAN (R 4.4.2)
## desc 1.4.3 2023-12-10 [1] CRAN (R 4.4.1)
## devtools 2.4.5 2022-10-11 [1] CRAN (R 4.4.1)
## digest 0.6.36 2024-06-23 [1] CRAN (R 4.4.1)
## doParallel 1.0.17 2022-02-07 [1] CRAN (R 4.4.0)
## dotCall64 1.2 2024-10-04 [1] CRAN (R 4.4.1)
## dplyr * 1.1.4 2023-11-17 [1] CRAN (R 4.4.0)
## e1071 1.7-16 2024-09-16 [1] CRAN (R 4.4.1)
## ellipsis 0.3.2 2021-04-29 [1] CRAN (R 4.4.0)
## evaluate 1.0.1 2024-10-10 [1] CRAN (R 4.4.1)
## farver 2.1.2 2024-05-13 [1] CRAN (R 4.4.0)
## fastmap 1.2.0 2024-05-15 [1] CRAN (R 4.4.0)
## fastmatch 1.1-6 2024-12-23 [1] CRAN (R 4.4.2)
## fields 16.3 2024-09-30 [1] CRAN (R 4.4.1)
## forcats * 1.0.0 2023-01-29 [1] CRAN (R 4.4.0)
## foreach 1.5.2 2022-02-02 [1] CRAN (R 4.4.0)
## foreign 0.8-86 2023-11-28 [2] CRAN (R 4.4.0)
## formatR * 1.14 2023-01-17 [1] CRAN (R 4.4.0)
## Formula 1.2-5 2023-02-24 [1] CRAN (R 4.4.0)
## fs 1.6.5 2024-10-30 [1] CRAN (R 4.4.2)
## furrr 0.3.1 2022-08-15 [1] CRAN (R 4.4.1)
## future 1.34.0 2024-07-29 [1] CRAN (R 4.4.1)
## future.apply 1.11.3 2024-10-27 [1] CRAN (R 4.4.2)
## gbasics 1.0.27 2024-09-27 [1] local
## gdsfmt 1.40.0 2024-05-01 [1] Bioconductor 3.19 (R 4.4.0)
## generics 0.1.3 2022-07-05 [1] CRAN (R 4.4.0)
## geodist 0.1.0 2024-05-23 [1] CRAN (R 4.4.2)
## ggdendro 0.2.0 2024-02-23 [1] CRAN (R 4.4.1)
## ggforce 0.4.2 2024-02-19 [1] CRAN (R 4.4.1)
## ggfun 0.1.8 2024-12-03 [1] CRAN (R 4.4.2)
## ggplot2 * 3.5.1 2024-04-23 [1] CRAN (R 4.4.0)
## ggplotify 0.1.2 2023-08-09 [1] CRAN (R 4.4.1)
## ggpmisc 0.6.1 2024-11-14 [1] CRAN (R 4.4.2)
## ggpp 0.5.8-1 2024-07-01 [1] CRAN (R 4.4.2)
## ggraph 2.2.1 2024-03-07 [1] CRAN (R 4.4.2)
## ggrepel 0.9.6 2024-09-07 [1] CRAN (R 4.4.1)
## ggtree * 3.12.0 2024-05-01 [1] Bioconductor 3.19 (R
4.4.0)
## globals 0.16.3 2024-03-08 [1] CRAN (R 4.4.0)
## glue 1.7.0 2024-01-09 [1] CRAN (R 4.4.0)
## gower 1.0.2 2024-12-17 [1] CRAN (R 4.4.2)
## gplots * 3.2.0 2024-10-05 [1] CRAN (R 4.4.2)
## graphlayouts 1.2.2 2025-01-23 [1] CRAN (R 4.4.2)
## gridExtra * 2.3 2017-09-09 [1] CRAN (R 4.4.0)
## gridGraphics 0.5-1 2020-12-13 [1] CRAN (R 4.4.1)
## gsl 2.1-8 2023-01-24 [1] CRAN (R 4.4.0)
## gtable 0.3.6 2024-10-25 [1] CRAN (R 4.4.0)
## gtools * 3.9.5 2023-11-20 [1] CRAN (R 4.4.0)
## haplo.stats 1.9.7 2024-09-19 [1] CRAN (R 4.4.1)
## haplotypes * 1.1.3.1 2023-07-15 [1] CRAN (R 4.4.0)
```

```
## hardhat 1.4.0 2024-06-02 [1] CRAN (R 4.4.0)
## Hmisc 5.2-1 2024-12-02 [1] CRAN (R 4.4.2)
## hms 1.1.3 2023-03-21 [1] CRAN (R 4.4.0)
## htmlTable 2.4.3 2024-07-21 [1] CRAN (R 4.4.1)
## htmltools 0.5.8.1 2024-04-04 [1] CRAN (R 4.4.0)
## htmlwidgets 1.6.4 2023-12-06 [1] CRAN (R 4.4.0)
## httpuv 1.6.15 2024-03-26 [1] CRAN (R 4.4.0)
## igraph 2.0.3 2024-03-13 [1] CRAN (R 4.4.0)
## ipred 0.9-15 2024-07-18 [1] CRAN (R 4.4.1)
## iterators 1.0.14 2022-02-05 [1] CRAN (R 4.4.0)
## jsonlite 1.8.9 2024-09-20 [1] CRAN (R 4.4.2)
## kableExtra * 1.4.0 2024-01-24 [1] CRAN (R 4.4.0)
## KernSmooth 2.23-22 2023-07-10 [2] CRAN (R 4.4.0)
## knitr * 1.49 2024-11-08 [1] CRAN (R 4.4.2)
## labeling 0.4.3 2023-08-29 [1] CRAN (R 4.4.0)
## later 1.3.2 2023-12-06 [1] CRAN (R 4.4.0)
## lattice * 0.22-6 2024-03-20 [2] CRAN (R 4.4.0)
## lava 1.8.0 2024-03-05 [1] CRAN (R 4.4.0)
## lazyeval 0.2.2 2019-03-15 [1] CRAN (R 4.4.0)
## lifecycle 1.0.4 2023-11-07 [1] CRAN (R 4.4.0)
## listenv 0.9.1 2024-01-29 [1] CRAN (R 4.4.0)
## lubridate * 1.9.4 2024-12-08 [1] CRAN (R 4.4.2)
## magrittr * 2.0.3 2022-03-30 [1] CRAN (R 4.4.0)
## maps 3.4.2.1 2024-11-10 [1] CRAN (R 4.4.2)
## MASS 7.3-60.2 2024-04-24 [2] local
## Matrix 1.7-0 2024-03-22 [2] CRAN (R 4.4.0)
## MatrixModels 0.5-3 2023-11-06 [1] CRAN (R 4.4.1)
## mauricer * 2.5.4 2024-06-11 [1] CRAN (R 4.4.2)
## memoise 2.0.1 2021-11-26 [1] CRAN (R 4.4.0)
## mgcv 1.9-1 2023-12-21 [2] CRAN (R 4.4.0)
## mime 0.12 2021-09-28 [1] CRAN (R 4.4.0)
## miniUI 0.1.1.1 2018-05-18 [1] CRAN (R 4.4.1)
## ModelMetrics 1.2.2.2 2020-03-17 [1] CRAN (R 4.4.0)
## multcomp 1.4-26 2024-07-18 [1] CRAN (R 4.4.1)
## munsell 0.5.1 2024-04-01 [1] CRAN (R 4.4.0)
## mvbutils * 2.8.501 2024-09-27 [1] local
## mvtnorm 1.3-2 2024-11-04 [1] CRAN (R 4.4.2)
## network 1.19.0 2024-12-09 [1] CRAN (R 4.4.2)
## nlme 3.1-164 2023-11-27 [2] CRAN (R 4.4.0)
## nnet 7.3-19 2023-05-03 [2] CRAN (R 4.4.0)
## numDeriv 2016.8-1.1 2019-06-06 [1] CRAN (R 4.4.0)
## OutFLANK * 0.2 2024-11-24 [1] Github
(whitlock/OutFLANK@e502e82)
## parallelly 1.41.0 2024-12-18 [1] CRAN (R 4.4.2)
## patchwork 1.3.0 2024-09-16 [1] CRAN (R 4.4.1)
## pcaPP 2.0-5 2024-08-19 [1] CRAN (R 4.4.2)
## pegas * 1.3 2023-12-13 [1] CRAN (R 4.4.0)
## permute * 0.9-7 2022-01-27 [1] CRAN (R 4.4.0)
## phangorn * 2.12.1 2024-09-17 [1] CRAN (R 4.4.2)
## phylotools 0.2.2 2017-12-10 [1] CRAN (R 4.4.2)
## pillar 1.10.0 2024-12-17 [1] CRAN (R 4.4.0)
## pkgbuild 1.4.5 2024-10-28 [1] CRAN (R 4.4.2)
## pkgconfig 2.0.3 2019-09-22 [1] CRAN (R 4.4.0)
## pkgload 1.4.0 2024-06-28 [1] CRAN (R 4.4.1)
## plotrix 3.8-4 2023-11-10 [1] CRAN (R 4.4.0)
## plyr * 1.8.9 2023-10-02 [1] CRAN (R 4.4.0)
## poisbinom 1.0.1 2017-05-19 [1] CRAN (R 4.4.1)
## polspline 1.1.25 2024-05-10 [1] CRAN (R 4.4.0)
```

```
## polyclip 1.10-7 2024-07-23 [1] CRAN (R 4.4.1)
## polynom 1.4-1 2022-04-11 [1] CRAN (R 4.4.2)
## pROC 1.18.5 2023-11-01 [1] CRAN (R 4.4.0)
## prodlim 2024.06.25 2024-06-24 [1] CRAN (R 4.4.1)
## profvis 0.4.0 2024-09-20 [1] CRAN (R 4.4.1)
## promises 1.3.2 2024-11-28 [1] CRAN (R 4.4.2)
## proxy 0.4-27 2022-06-09 [1] CRAN (R 4.4.0)
## pspline 1.0-21 2024-12-11 [1] CRAN (R 4.4.0)
## purrr * 1.0.2 2023-08-10 [1] CRAN (R 4.4.0)
## quadprog 1.5-8 2019-11-20 [1] CRAN (R 4.4.0)
## quantreg 5.99.1 2024-11-22 [1] CRAN (R 4.4.2)
## qvalue * 2.36.0 2024-05-01 [1] Bioconductor 3.19 (R
4.4.0)
## R6 2.5.1 2021-08-19 [1] CRAN (R 4.4.0)
## radiator 1.3.4 2024-07-23 [1] local
## ragg 1.3.3 2024-09-11 [1] CRAN (R 4.4.1)
## randomForest 4.7-1.2 2024-09-22 [1] CRAN (R 4.4.1)
## rappdirs 0.3.3 2021-01-31 [1] CRAN (R 4.4.0)
## raster 3.6-30 2024-10-02 [1] CRAN (R 4.4.1)
## RColorBrewer 1.1-3 2022-04-03 [1] CRAN (R 4.4.0)
## Rcpp 1.0.12 2024-01-09 [1] CRAN (R 4.4.0)
## readr * 2.1.5 2024-01-10 [1] CRAN (R 4.4.0)
## recipes 1.1.0 2024-07-04 [1] CRAN (R 4.4.1)
## remotes 2.5.0 2024-03-17 [1] CRAN (R 4.4.1)
## reshape2 1.4.4 2020-04-09 [1] CRAN (R 4.4.0)
## D rJava 1.0-11 2024-01-26 [1] CRAN (R 4.4.0)
## rlang 1.1.4 2024-06-04 [1] CRAN (R 4.4.0)
## rmarkdown 2.29 2024-11-04 [1] CRAN (R 4.4.2)
## rms 6.9-0 2024-12-12 [1] CRAN (R 4.4.2)
## rpart 4.1.23 2023-12-05 [2] CRAN (R 4.4.0)
## rprojroot 2.0.4 2023-11-05 [1] CRAN (R 4.4.1)
## rstudioapi 0.17.1 2024-10-22 [1] CRAN (R 4.4.1)
## sandwich 3.1-1 2024-09-15 [1] CRAN (R 4.4.1)
## scales 1.3.0 2023-11-28 [1] CRAN (R 4.4.0)
## seqinr 4.2-36 2023-12-08 [1] CRAN (R 4.4.0)
## sessioninfo 1.2.2 2021-12-06 [1] CRAN (R 4.4.0)
## shiny 1.10.0 2024-12-14 [1] CRAN (R 4.4.2)
## sna 2.8 2024-09-08 [1] CRAN (R 4.4.2)
## SNPassoc 2.1-2 2024-10-28 [1] CRAN (R 4.4.2)
## SNPRelate 1.38.0 2024-05-01 [1] Bioconductor 3.19 (R
4.4.0)
## sp 2.1-4 2024-04-30 [1] CRAN (R 4.4.0)
## spam 2.11-0 2024-10-03 [1] CRAN (R 4.4.1)
## SparseM 1.84-2 2024-07-17 [1] CRAN (R 4.4.1)
## stabledist 0.7-2 2024-08-17 [1] CRAN (R 4.4.2)
## StAMPP 1.6.3 2021-08-08 [1] CRAN (R 4.4.0)
## statnet.common 4.10.0 2024-10-06 [1] CRAN (R 4.4.2)
## strataG * 2.5.01 2024-11-20 [1] Github
(ericarcher/strataG@f38fd47)
## stringi 1.8.4 2024-05-06 [1] CRAN (R 4.4.0)
## stringr * 1.5.1 2023-11-14 [1] CRAN (R 4.4.0)
## survival 3.5-8 2024-02-14 [2] CRAN (R 4.4.0)
## svglite 2.1.3 2023-12-08 [1] CRAN (R 4.4.0)
## swfscMisc 1.6.5 2023-09-08 [1] CRAN (R 4.4.2)
## systemfonts 1.1.0 2024-05-15 [1] CRAN (R 4.4.0)
## terra 1.8-5 2024-12-12 [1] CRAN (R 4.4.2)
## testit 0.13 2021-04-14 [1] CRAN (R 4.4.2)
## testthat 3.2.2 2024-12-10 [1] CRAN (R 4.4.0)
```

```
## textshaping 0.4.1 2024-12-06 [1] CRAN (R 4.4.2)
## TH.data 1.1-2 2023-04-17 [1] CRAN (R 4.4.1)
## tibble * 3.2.1 2023-03-20 [1] CRAN (R 4.4.0)
## tidygraph 1.3.1 2024-01-30 [1] CRAN (R 4.4.2)
## tidyr * 1.3.1 2024-01-24 [1] CRAN (R 4.4.0)
## tidyselect 1.2.1 2024-03-11 [1] CRAN (R 4.4.0)
## tidytree 0.4.6 2023-12-12 [1] CRAN (R 4.4.2)
## tidyverse * 2.0.0 2023-02-22 [1] CRAN (R 4.4.0)
## timechange 0.3.0 2024-01-18 [1] CRAN (R 4.4.0)
## timeDate 4041.110 2024-09-22 [1] CRAN (R 4.4.1)
## tracerer * 2.2.3 2023-09-27 [1] CRAN (R 4.4.2)
## tree 1.0-44 2024-12-11 [1] CRAN (R 4.4.2)
## treeio * 1.28.0 2024-05-01 [1] Bioconductor 3.19 (R
4.4.0)
## tweenr 2.0.3 2024-02-26 [1] CRAN (R 4.4.1)
## tzdb 0.4.0 2023-05-12 [1] CRAN (R 4.4.0)
## urlchecker 1.0.1 2021-11-30 [1] CRAN (R 4.4.1)
## usethis 3.1.0 2024-11-26 [1] CRAN (R 4.4.2)
## vctrs 0.6.5 2023-12-01 [1] CRAN (R 4.4.0)
## vegan * 2.6-8 2024-08-28 [1] CRAN (R 4.4.2)
## viridis 0.6.5 2024-01-29 [1] CRAN (R 4.4.1)
## viridisLite 0.4.2 2023-05-02 [1] CRAN (R 4.4.0)
## vroom 1.6.5 2023-12-05 [1] CRAN (R 4.4.0)
## withr 3.0.2 2024-10-28 [1] CRAN (R 4.4.2)
## xfun 0.49 2024-10-31 [1] CRAN (R 4.4.2)
## xml2 1.3.6 2023-12-04 [1] CRAN (R 4.4.0)
## xtable 1.8-4 2019-04-21 [1] CRAN (R 4.4.0)
## yaml 2.3.10 2024-07-26 [1] CRAN (R 4.4.2)
## yulab.utils 0.1.8 2024-11-07 [1] CRAN (R 4.4.2)
## zoo 1.8-12 2023-04-13 [1] CRAN (R 4.4.1)
##
## [1] C:/Users/dev093/AppData/Local/R/win-library/4.4
## [2] C:/Program Files/R/R-4.4.0/library
##
## D -- DLL MD5 mismatch, broken installation.
##
##
------------------------------------------------------------------------------
```