

R code for figures

Raphael Eisenhofer

30/11/2020

R code used to generate figures for the paper: Signatures of landscape and captivity in the gut microbiota of Southern Hairy-nosed Wombats (*Lasiiorhinus latifrons*)

Load libraries

```
library(phyloseq)
library(qiime2R)
library(dplyr)
library(svglite)
library(cowplot)
library(gplots)
library(ggplot2)
library(scales)
library(ggpubr)
library(tidyr)
library(knitr)
library(microbiome)
library(microbiomeutilities)
library(VennDiagram)
library(ggVennDiagram)
```

Import data

```
ps <- qza_to_phyloseq(
  features = "QIIME2_outputs/SHNW-gut-table-final.qza",
  tree = "QIIME2_outputs/sepp-tree.qza",
  taxonomy = "QIIME2_outputs/SHNW-gut-SILVA-132.qza",
  metadata = "QIIME2_outputs/SHNW_2019_Gut_Metadata.txt"
)

#PCoA matrices
pcoa_uwUniFrac_all <- read_qza(
  "QIIME2_outputs/SHNW-gut-Core-metrics-final-filtered-table-36346/unweighted_unifrac_pcoa_results.qza")
pcoa_wUniFrac_all <- read_qza(
  "QIIME2_outputs/SHNW-gut-Core-metrics-final-filtered-table-36346/weighted_unifrac_pcoa_results.qza")
pcoa_uwUniFrac_wild <- read_qza(
  "QIIME2_outputs/SHNW-gut-Core-metrics-final-filtered-table-36346-WILDonly/unweighted_unifrac_pcoa_results.qza")
pcoa_wUniFrac_wild <- read_qza(
  "QIIME2_outputs/SHNW-gut-Core-metrics-final-filtered-table-36346-WILDonly/weighted_unifrac_pcoa_results.qza")

#Alpha diversity data
faithsPD <- read.csv("QIIME2_outputs/faithsPD.tsv", sep = '\t')
ObsFeat <- read.csv("QIIME2_outputs/observed_otus.tsv", sep = '\t')

#Venn diagram data (skip second column, which is frequency of ASVs)
captive.asvs <- read.csv("QIIME2_outputs/feature-frequency-detail-captive-200.csv", header = FALSE, row.names = 1)
wild.asvs <- read.csv("QIIME2_outputs/feature-frequency-detail-wild-750.csv", header = FALSE, row.names = 1)

brookfield.asvs <- read.csv(
```

```
"QIIME2_outputs/feature-frequency-detail-Brookfield.csv", header = FALSE, row.names = 1)
```

```
kooloola.asvs <- read.csv(
```

```
"QIIME2_outputs/feature-frequency-detail-Kooloola.csv", header = FALSE, row.names = 1)
```

```
wonga.asvs <- read.csv(
```

```
"QIIME2_outputs/feature-frequency-detail-Wonga.csv", header = FALSE, row.names = 1)
```

Alpha diversity plots (figure 2):

```
#This sets the order in which you wish to display the samples
```

```
level_order_adiv <- c('Kooloola', 'Brookfield', 'Wonga', 'Captive')
```

```
#Plot the data!
```

```
sp_adiv_fig_faithsPD <- ggplot(faithsPD, aes(x = factor(Population, level=level_order_adiv), y=faith_pd))
```

```
sp_adiv_fig_ObsFeat <- ggplot(ObsFeat, aes(x = factor(Population, level=level_order_adiv), y=observed_otus))
```

```
#Colour vision deficiency-friendly palette
```

```
cbp2_adiv <- c("orange", "black", "red", "blue")
```

```
#Faith's PD
```

```
#add jitter, size, and colour
```

```
sp_adiv_fig_faithsPD +
```

```
#Boxplot
```

```
geom_boxplot(size=2, outlier.shape=4, outlier.size=3, outlier.stroke=3, aes(colour=Population)) +
```

```
#Jitter, size, colour
```

```
geom_jitter(position=position_dodge2(0.3), size=5, aes(colour=Population)) +
```

```
#Custom manual colours
```

```
scale_colour_manual(values=cbp2_adiv) +
```

```
#Tick labels
```

```
theme(axis.text.x = element_text(face="bold", size=20),  
axis.text.y = element_text(face="bold", size=20),  
axis.title.x = element_text(size=24, face="bold"),  
axis.title.y = element_text(size=24, face="bold"),  
axis.line = element_line(colour = "black"),
```

```
#Background panel
```

```
panel.background = element_rect(fill = "White"),  
panel.grid.major = element_line(colour = "white"),  
panel.grid.minor = element_line(colour = "white"),
```

```
#Legend
```

```
legend.position = "none") +
```

```
#legend.title = element_text(size=0),
```

```
#legend.text = element_text(size=0),
```

```
#legend.key = element_rect(fill = "white", color = NA),
```

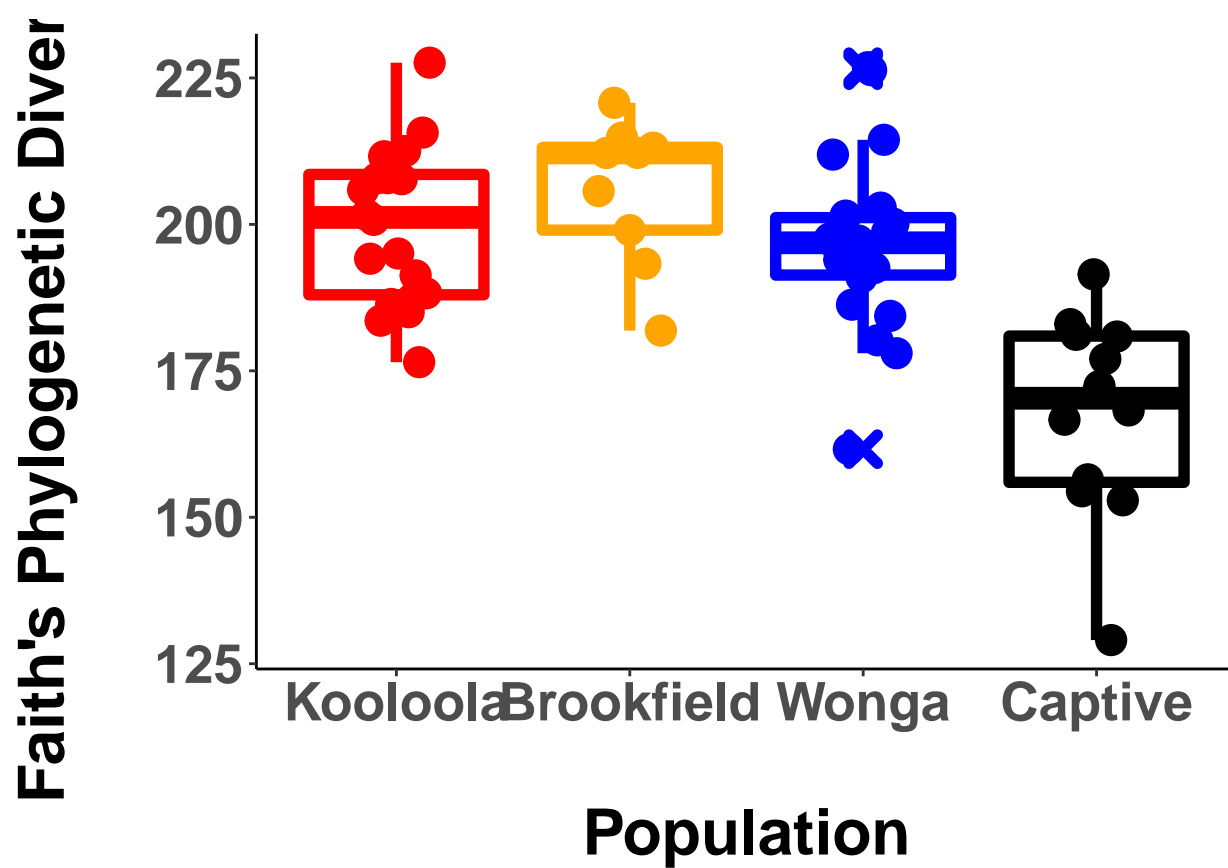
```
#legend.key.size = unit(0, "line")) +
```

```
#Axis labels
```

```
scale_y_continuous(breaks=seq(0,400,25)) +
```

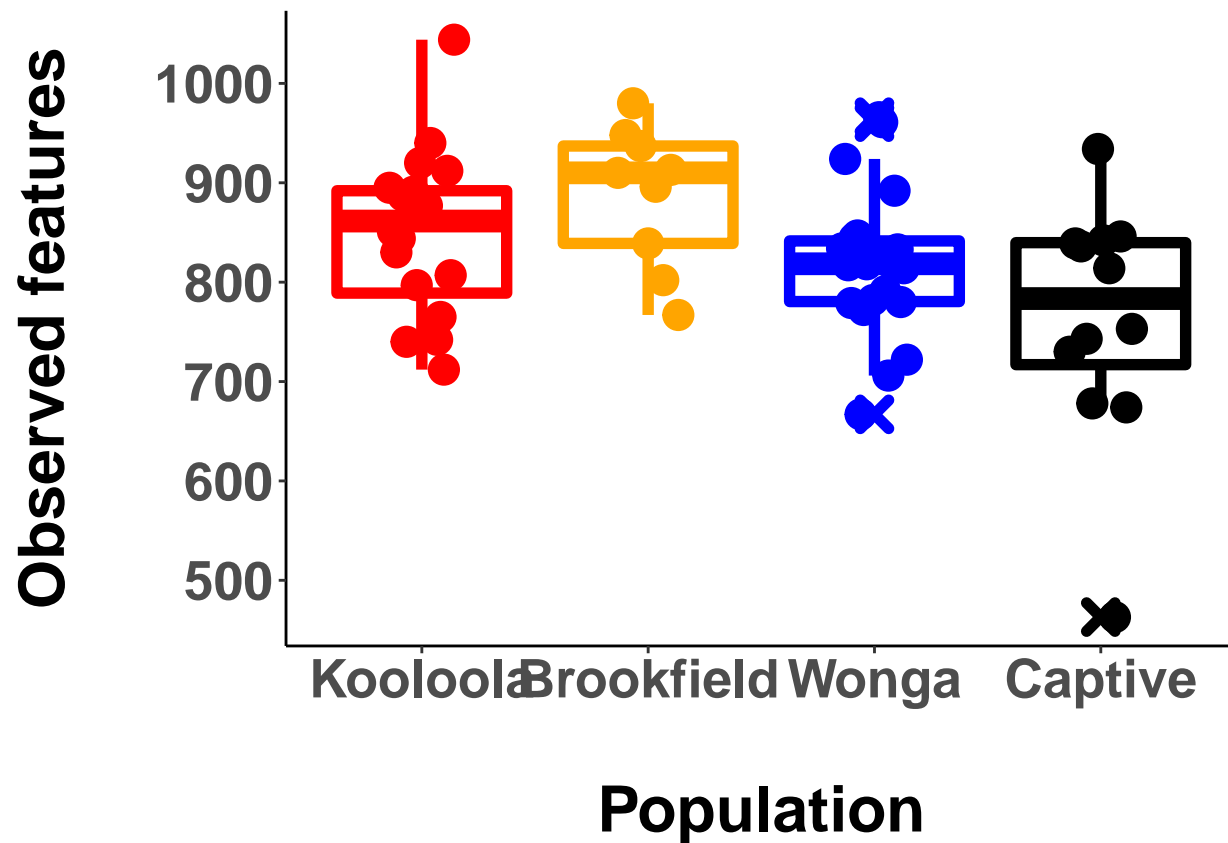
```
labs(x = "\nPopulation") +
```

```
labs(y = "Faith's Phylogenetic Diversity\n")
```



```
#Save image as .svg
ggsave(filename = "Fig_AlphaDiv-FaithsPD.svg", width = 20, height = 11, dpi = 300)

#Observed OTUs
#add jitter, size, and colour
sp_adiv_fig_ObsFeat +
  #Boxplot
  geom_boxplot(size=2, outlier.shape=4, outlier.size=3, outlier.stroke=3, aes(colour=Population)) +
  #Jitter, size, colour
  geom_jitter(position=position_dodge2(0.3), size=5, aes(colour=Population)) +
  #Custom manual colours
  scale_colour_manual(values=cbp2_adiv) +
  #Tick labels
  theme(axis.text.x = element_text(face="bold", size=20),
        axis.text.y = element_text(face="bold", size=20),
        axis.title.x = element_text(size=24, face="bold"),
        axis.title.y = element_text(size=24, face="bold"),
        axis.line = element_line(colour = "black"),
        #Background panel
        panel.background = element_rect(fill = "White"),
        panel.grid.major = element_line(colour = "white"),
        panel.grid.minor = element_line(colour = "white"),
        #Legend
        legend.position = "none") +
  #legend.title = element_text(size=0),
  #legend.text = element_text(size=0),
  #legend.key = element_rect(fill = "white", color = NA),
  #legend.key.size = unit(0, "line")) +
  #Axis labels
  scale_y_continuous(breaks=seq(0,1000,100)) +
  labs(x = "\nPopulation") +
  labs(y = "Observed features\n")
```



```
#Save image as .svg
ggsave(filename = "Fig_AlphaDiv-ObservedFeatures.svg", width = 20, height = 11, dpi = 300)
```

Beta diversity PCoA plots (figures 2 and 4):

```
#Load metadata
metadata_pcoa <- read.csv("QIIME2_outputs/SHNW_2019_Gut_Metadata.txt", sep = '\t')

#Colour palette
cbp2_beta <- c("orange", "black", "red", "blue")

#Plot PC1 vs. PC2 (uwUniFrac_all)
pcoa_uwUniFrac_all$data$Vectors %>%
  rename("X.SampleID"=SampleID) %>%
  left_join(metadata_pcoa) %>%

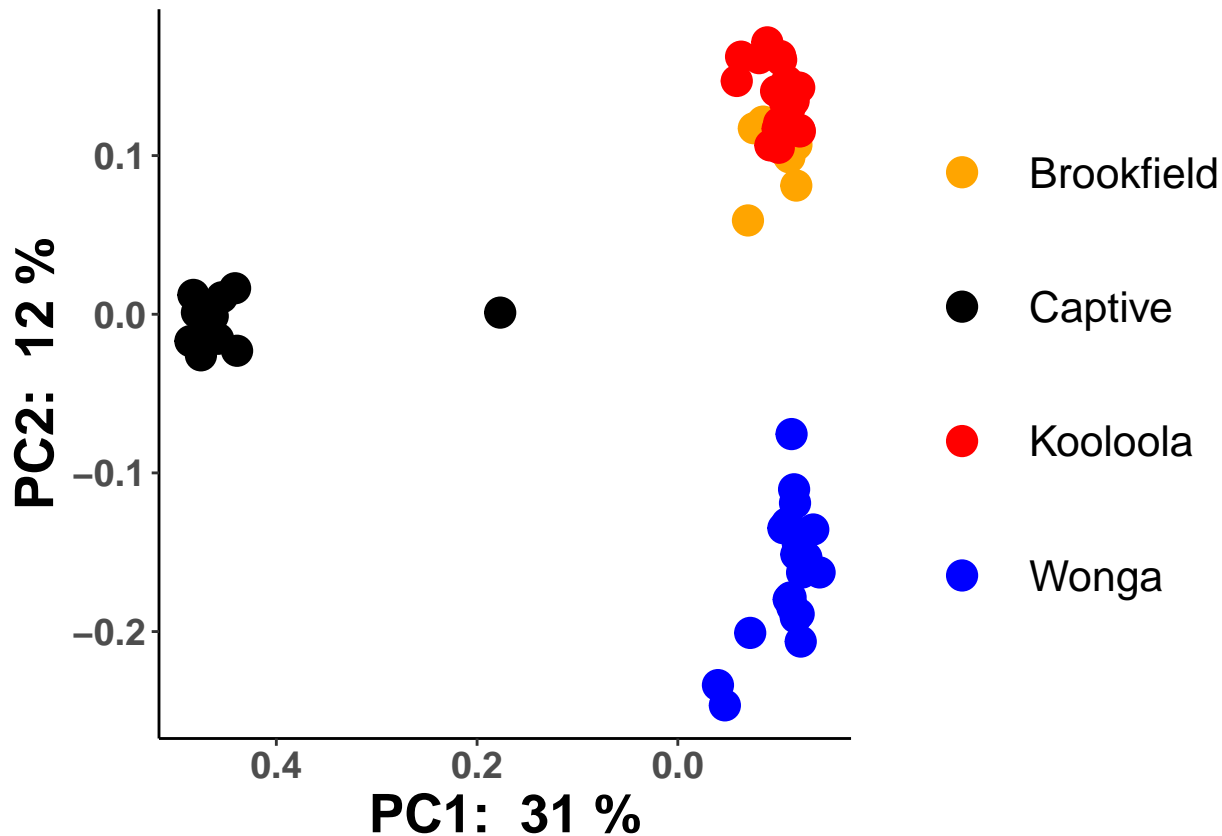
  ggplot(aes(x=PC1, y=PC2, color=Population)) +
  scale_x_reverse() +
  geom_point(size=5) +
  xlab(paste("PC1: ", round(100*pcoa_uwUniFrac_all$data$ProportionExplained[1]), "%")) +
  ylab(paste("PC2: ", round(100*pcoa_uwUniFrac_all$data$ProportionExplained[2]), "%")) +
  #Custom manual colours
  scale_colour_manual(values=cbp2_beta) +

  theme(axis.text.x = element_text(face="bold", size=14),
        axis.text.y = element_text(face="bold", size=14),
        axis.title.x = element_text(size=20, face="bold"),
        axis.title.y = element_text(size=20, face="bold"),
        axis.line = element_line(colour = "black"),
        #Background panel
        panel.background = element_rect(fill = "white"),
        panel.grid.major = element_line(colour = "white"),
        panel.grid.minor = element_line(colour = "white"),
        #Legend
```

```

legend.title = element_text(size=0),
legend.text = element_text(size=16),
legend.key = element_rect(fill = "white", color = NA),
legend.key.size = unit(3.5, "line")

```



```

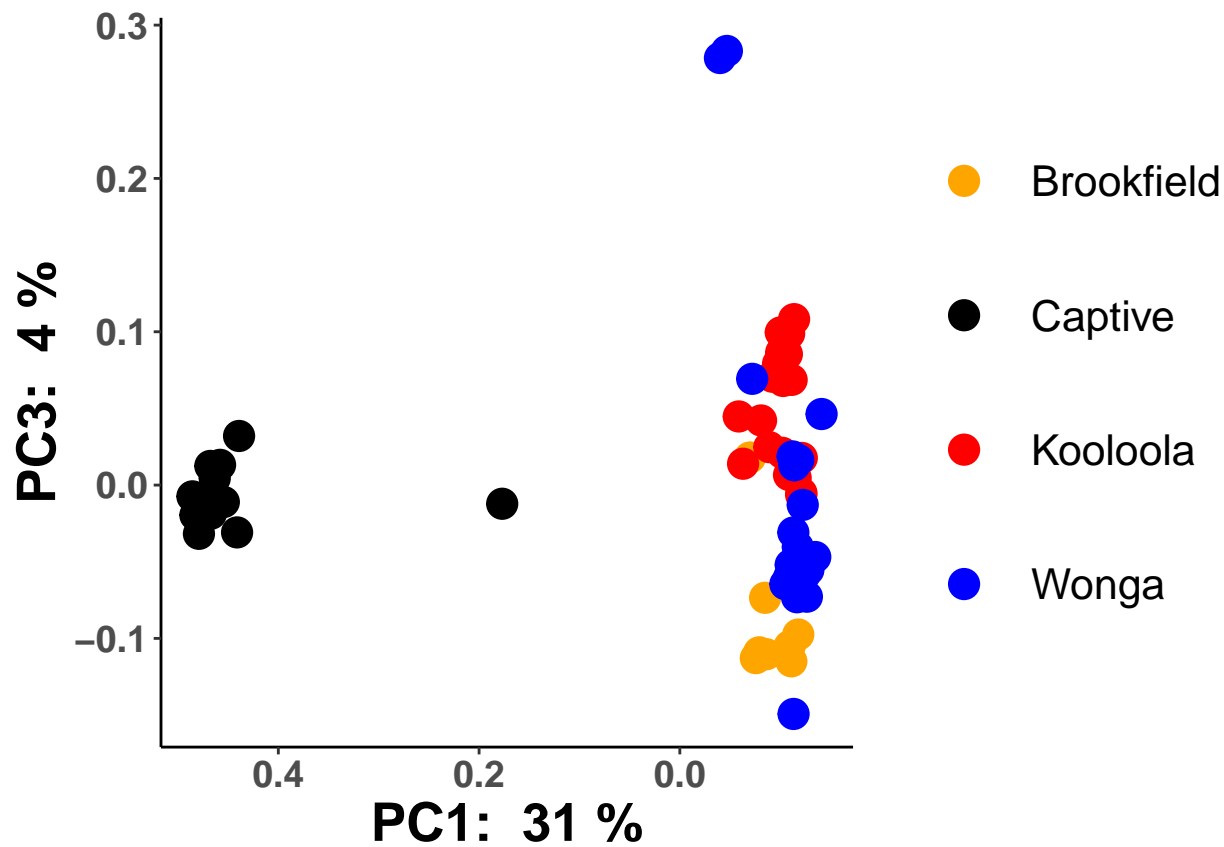
#Save image
ggsave(filename = "PC1-2_uwUniFrac_all.svg", width = 20, height = 11, dpi = 300)

#Plot PC1 vs. PC3 (uwUniFrac_all)
pcoa_uwUniFrac_all$data$Vectors %>%
  rename("X.SampleID"=SampleID) %>%
  left_join(metadata_pcoa) %>%

ggplot(aes(x=PC1, y=PC3, color=Population)) +
  scale_x_reverse() +
  geom_point(size=5) +
  xlab(paste("PC1: ", round(100*pcoa_uwUniFrac_all$data$ProportionExplained[1]), "%")) +
  ylab(paste("PC3: ", round(100*pcoa_uwUniFrac_all$data$ProportionExplained[3]), "%")) +
  #Custom manual colours
  scale_colour_manual(values=cbp2_beta) +

theme(axis.text.x = element_text(face="bold", size=14),
      axis.text.y = element_text(face="bold", size=14),
      axis.title.x = element_text(size=20, face="bold"),
      axis.title.y = element_text(size=20, face="bold"),
      axis.line = element_line(colour = "black"),
      #Background panel
      panel.background = element_rect(fill = "White"),
      panel.grid.major = element_line(colour = "white"),
      panel.grid.minor = element_line(colour = "white"),
      #Legend
      legend.title = element_text(size=0),
      legend.text = element_text(size=16),
      legend.key = element_rect(fill = "white", color = NA),
      legend.key.size = unit(3.5, "line"))

```

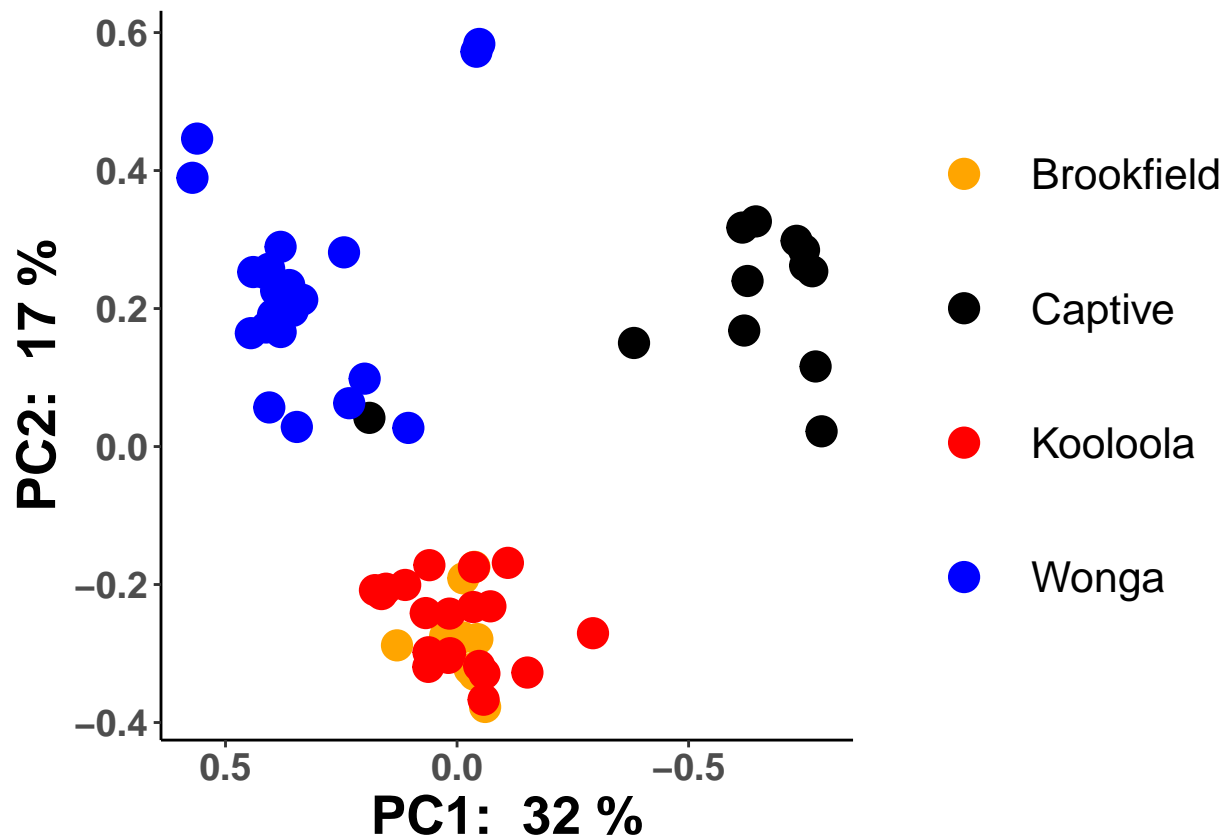


```
#Save image
ggsave(filename = "PC1-3_uwUniFrac_all.svg", width = 20, height = 11, dpi = 300)

#Plot PC1 vs. PC2 (wUniFrac_all)
pcoa_wUniFrac_all$data$Vectors %>%
  rename("X.SampleID"=SampleID) %>%
  left_join(metadata_pcoa) %>%

ggplot(aes(x=PC1, y=PC2, color=Population)) +
  scale_x_reverse() +
  geom_point(size=5) +
  xlab(paste("PC1: ", round(100*pcoa_wUniFrac_all$data$ProportionExplained[1]), "%")) +
  ylab(paste("PC2: ", round(100*pcoa_wUniFrac_all$data$ProportionExplained[2]), "%")) +
  #Custom manual colours
  scale_colour_manual(values=cbp2_beta) +

theme(axis.text.x = element_text(face="bold", size=14),
      axis.text.y = element_text(face="bold", size=14),
      axis.title.x = element_text(size=20, face="bold"),
      axis.title.y = element_text(size=20, face="bold"),
      axis.line = element_line(colour = "black"),
      #Background panel
      panel.background = element_rect(fill = "white"),
      panel.grid.major = element_line(colour = "white"),
      panel.grid.minor = element_line(colour = "white"),
      #Legend
      legend.title = element_text(size=0),
      legend.text = element_text(size=16),
      legend.key = element_rect(fill = "white", color = NA),
      legend.key.size = unit(3.5, "line"))
```

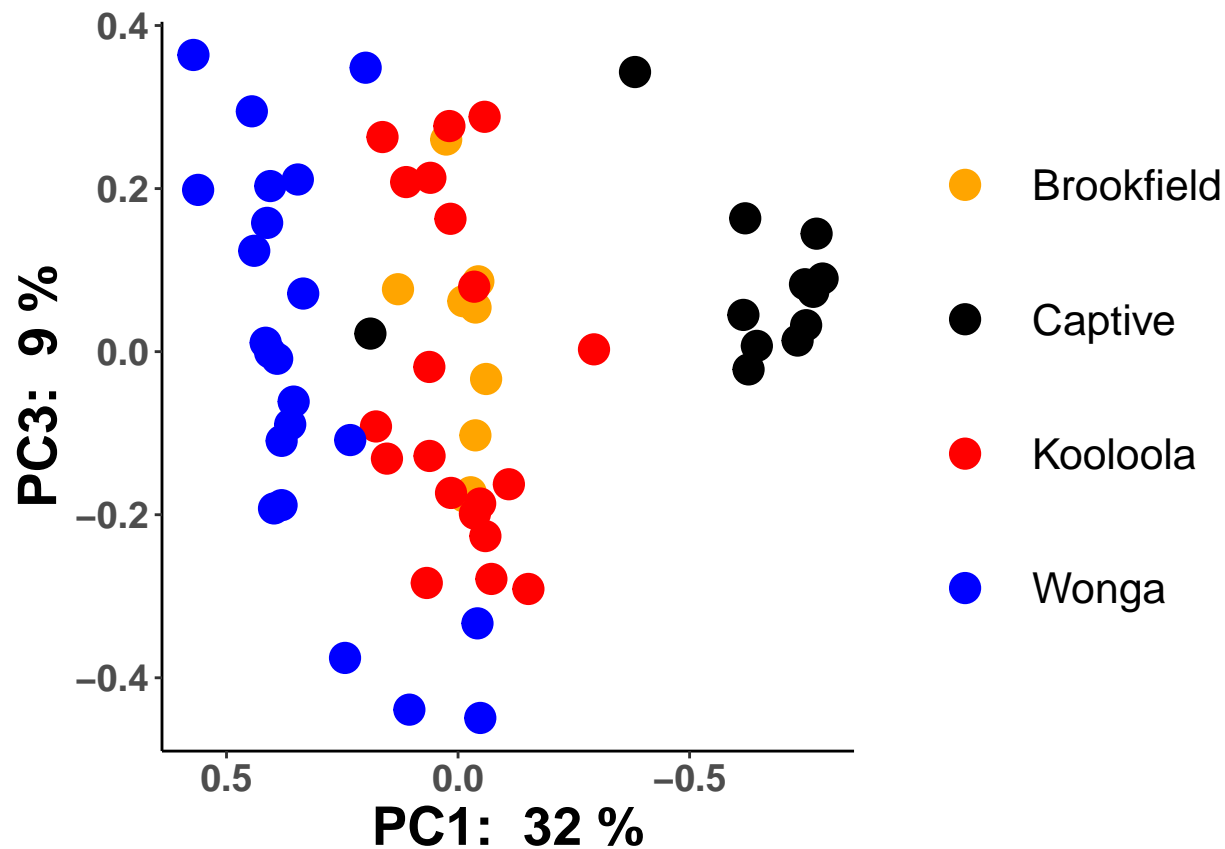


```
#Save image
ggsave(filename = "PC1-2_wUniFrac_all.svg", width = 20, height = 11, dpi = 300)

#Plot PC1 vs. PC3 (wUniFrac_all)
pcoa_wUniFrac_all$data$Vectors %>%
  rename("X.SampleID"=SampleID) %>%
  left_join(metadata_pcoa) %>%

ggplot(aes(x=PC1, y=PC3, color=Population)) +
  scale_x_reverse() +
  geom_point(size=5) +
  xlab(paste("PC1: ", round(100*pcoa_wUniFrac_all$data$ProportionExplained[1]), "%")) +
  ylab(paste("PC3: ", round(100*pcoa_wUniFrac_all$data$ProportionExplained[3]), "%")) +
  #Custom manual colours
  scale_colour_manual(values=cbp2_beta) +

  theme(axis.text.x = element_text(face="bold", size=14),
        axis.text.y = element_text(face="bold", size=14),
        axis.title.x = element_text(size=20, face="bold"),
        axis.title.y = element_text(size=20, face="bold"),
        axis.line = element_line(colour = "black"),
        #Background panel
        panel.background = element_rect(fill = "white"),
        panel.grid.major = element_line(colour = "white"),
        panel.grid.minor = element_line(colour = "white"),
        #Legend
        legend.title = element_text(size=0),
        legend.text = element_text(size=16),
        legend.key = element_rect(fill = "white", color = NA),
        legend.key.size = unit(3.5, "line"))
```



```
#Save image
ggsave(filename = "PC1-3_wUniFrac_all.svg", width = 20, height = 11, dpi = 300)

#####
### WILD ONLY ###
#####

cbp2_beta_wild <- c("orange", "red", "blue")

#Plot PC1 vs. PC2 (uwUniFrac_wild)
pcoa_uwUniFrac_wild$data$Vectors %>%
  rename("X.SampleID"=SampleID) %>%
  left_join(metadata_pcoa) %>%

ggplot(aes(x=PC1, y=PC2, color=Population)) +
  scale_x_reverse() +
  geom_point(size=5) +
  xlab(paste("PC1: ", round(100*pcoa_uwUniFrac_wild$data$ProportionExplained[1]), "%")) +
  ylab(paste("PC2: ", round(100*pcoa_uwUniFrac_wild$data$ProportionExplained[2]), "%")) +
  #Custom manual colours
  scale_colour_manual(values=cbp2_beta_wild) +

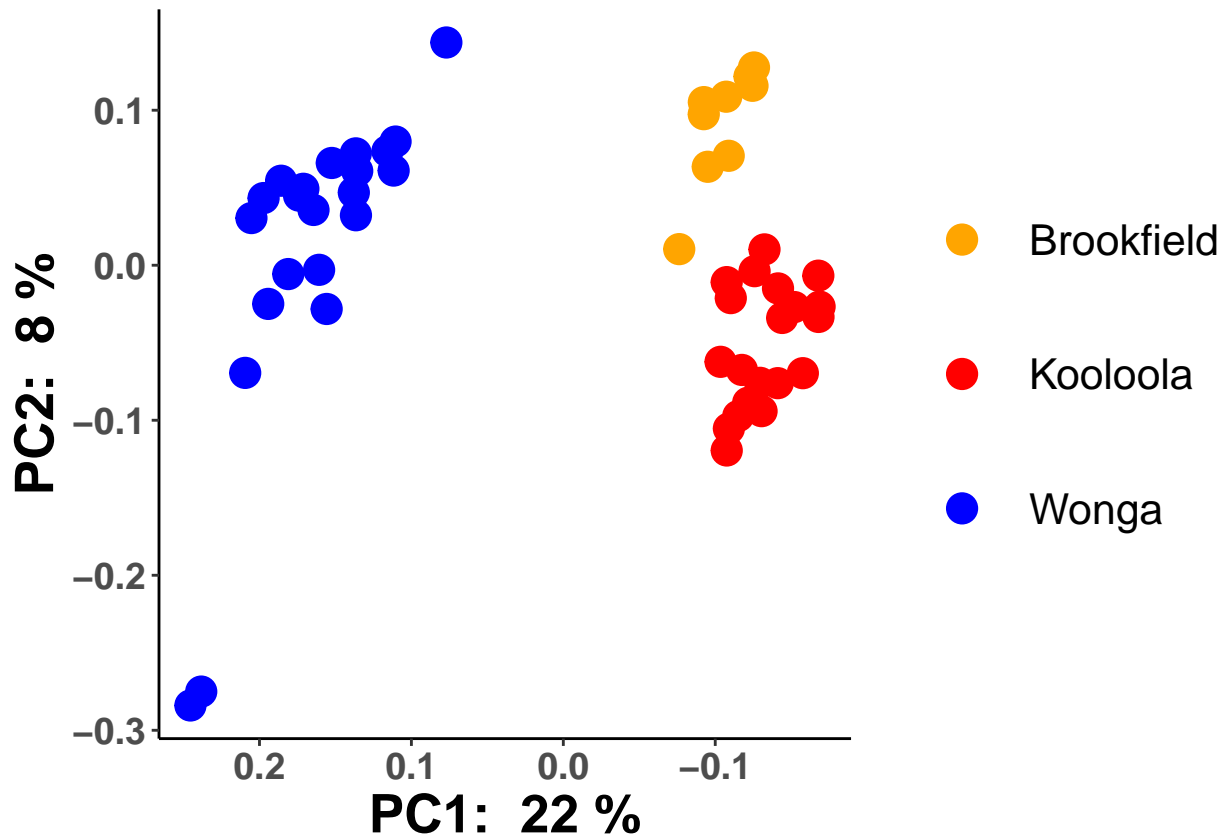
theme(axis.text.x = element_text(face="bold", size=14),
      axis.text.y = element_text(face="bold", size=14),
      axis.title.x = element_text(size=20, face="bold"),
      axis.title.y = element_text(size=20, face="bold"),
      axis.line = element_line(colour = "black"),
      #Background panel
      panel.background = element_rect(fill = "white"),
      panel.grid.major = element_line(colour = "white"),
      panel.grid.minor = element_line(colour = "white"),
      #Legend
      legend.title = element_text(size=0),
```



```

legend.text = element_text(size=16),
legend.key = element_rect(fill = "white", color = NA),
legend.key.size = unit(3.5, "line")

```



```

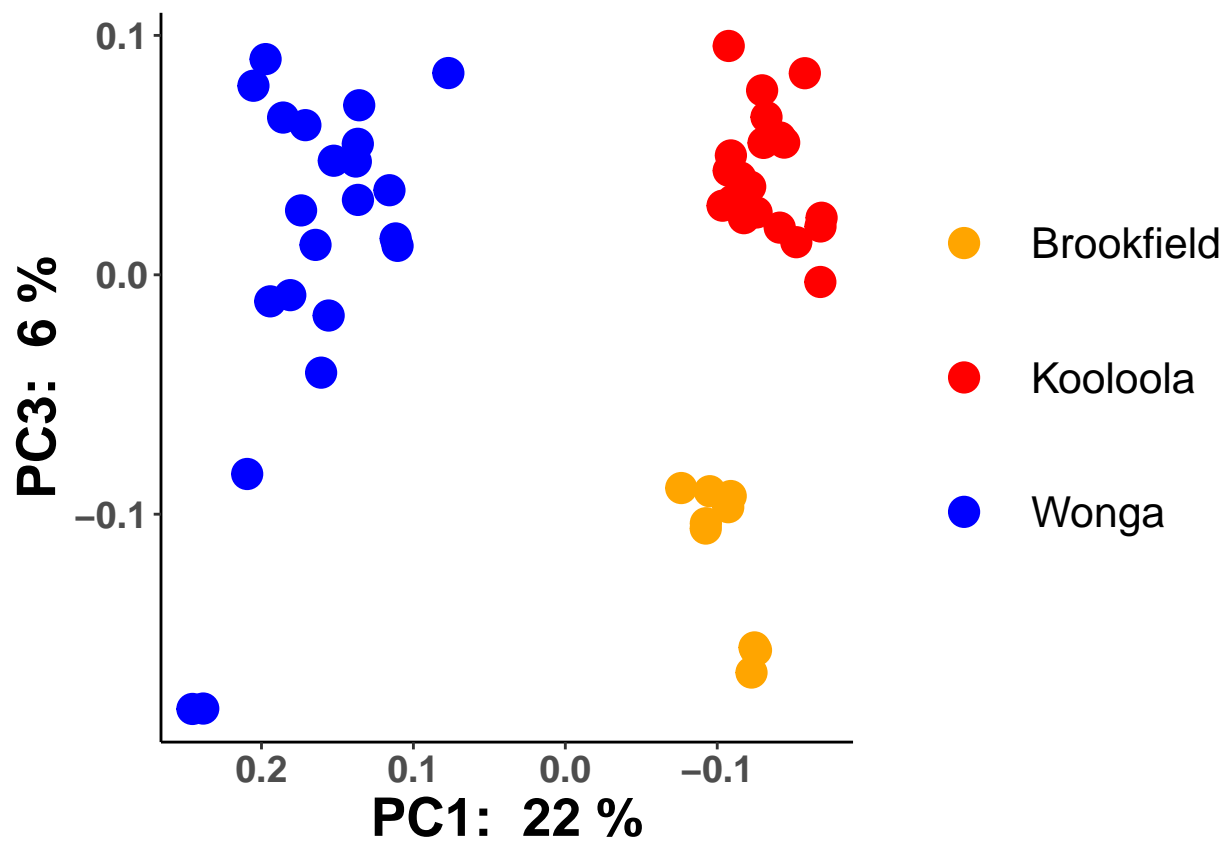
#Save image
ggsave(filename = "PC1-2_uwUniFrac_wild.svg", width = 20, height = 11, dpi = 300)

#Plot PC1 vs. PC3 (uwUniFrac_wild)
pcoa_uwUniFrac_wild$data$Vectors %>%
  rename("X.SampleID"=SampleID) %>%
  left_join(metadata_pcoa) %>%

ggplot(aes(x=PC1, y=PC3, color=Population)) +
  scale_x_reverse() +
  geom_point(size=5) +
  xlab(paste("PC1: ", round(100*pcoa_uwUniFrac_wild$data$ProportionExplained[1]), "%")) +
  ylab(paste("PC3: ", round(100*pcoa_uwUniFrac_wild$data$ProportionExplained[3]), "%")) +
  #Custom manual colours
  scale_colour_manual(values=cbp2_beta_wild) +

theme(axis.text.x = element_text(face="bold", size=14),
      axis.text.y = element_text(face="bold", size=14),
      axis.title.x = element_text(size=20, face="bold"),
      axis.title.y = element_text(size=20, face="bold"),
      axis.line = element_line(colour = "black"),
      #Background panel
      panel.background = element_rect(fill = "White"),
      panel.grid.major = element_line(colour = "white"),
      panel.grid.minor = element_line(colour = "white"),
      #Legend
      legend.title = element_text(size=0),
      legend.text = element_text(size=16),
      legend.key = element_rect(fill = "white", color = NA),
      legend.key.size = unit(3.5, "line"))

```

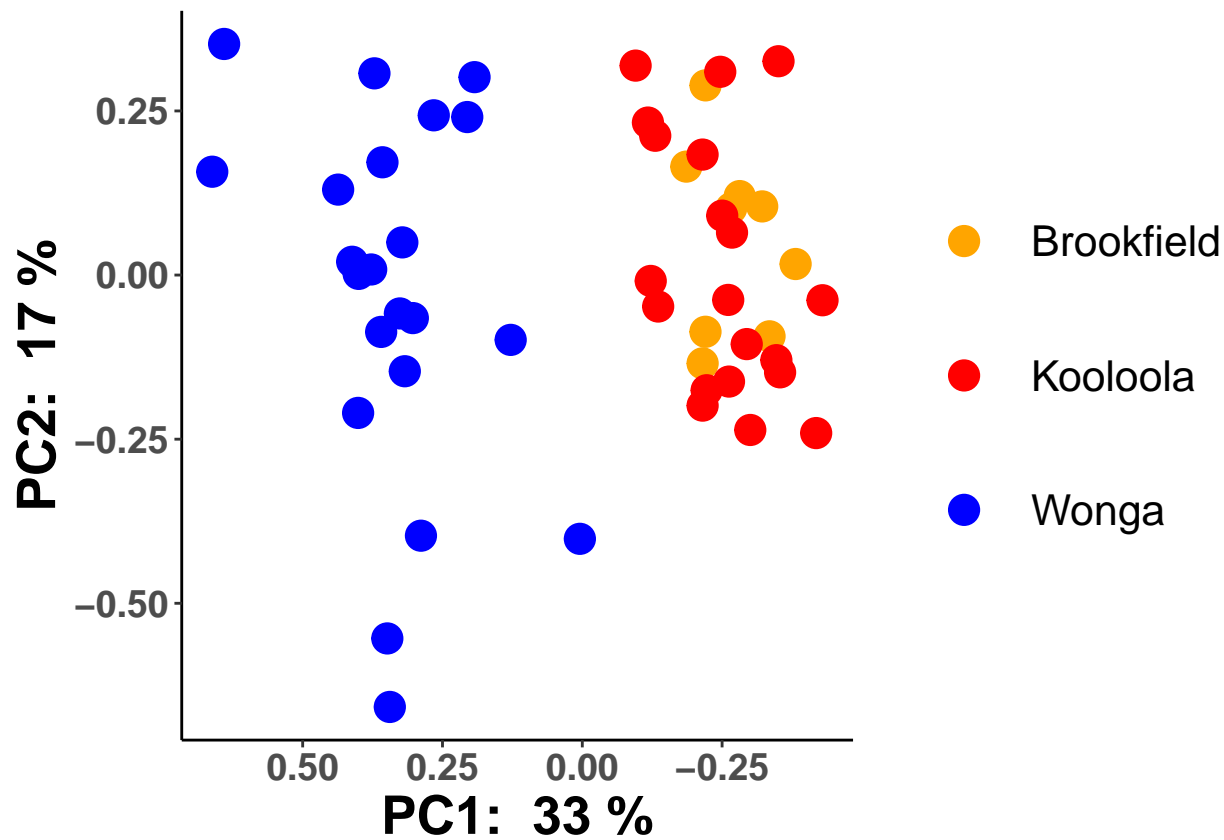


```
#Save image
ggsave(filename = "PC1-3_uwUniFrac_wild.svg", width = 20, height = 11, dpi = 300)

#Plot PC1 vs. PC2 (wUniFrac_wild)
pcoa_wUniFrac_wild$data$Vectors %>%
  rename("X.SampleID"=SampleID) %>%
  left_join(metadata_pcoa) %>%

  ggplot(aes(x=PC1, y=PC2, color=Population)) +
  scale_x_reverse() +
  geom_point(size=5) +
  xlab(paste("PC1: ", round(100*pcoa_wUniFrac_wild$data$ProportionExplained[1]), "%")) +
  ylab(paste("PC2: ", round(100*pcoa_wUniFrac_wild$data$ProportionExplained[2]), "%")) +
  #Custom manual colours
  scale_colour_manual(values=cbp2_beta_wild) +

  theme(axis.text.x = element_text(face="bold", size=14),
        axis.text.y = element_text(face="bold", size=14),
        axis.title.x = element_text(size=20, face="bold"),
        axis.title.y = element_text(size=20, face="bold"),
        axis.line = element_line(colour = "black"),
        #Background panel
        panel.background = element_rect(fill = "white"),
        panel.grid.major = element_line(colour = "white"),
        panel.grid.minor = element_line(colour = "white"),
        #Legend
        legend.title = element_text(size=0),
        legend.text = element_text(size=16),
        legend.key = element_rect(fill = "white", color = NA),
        legend.key.size = unit(3.5, "line"))
```

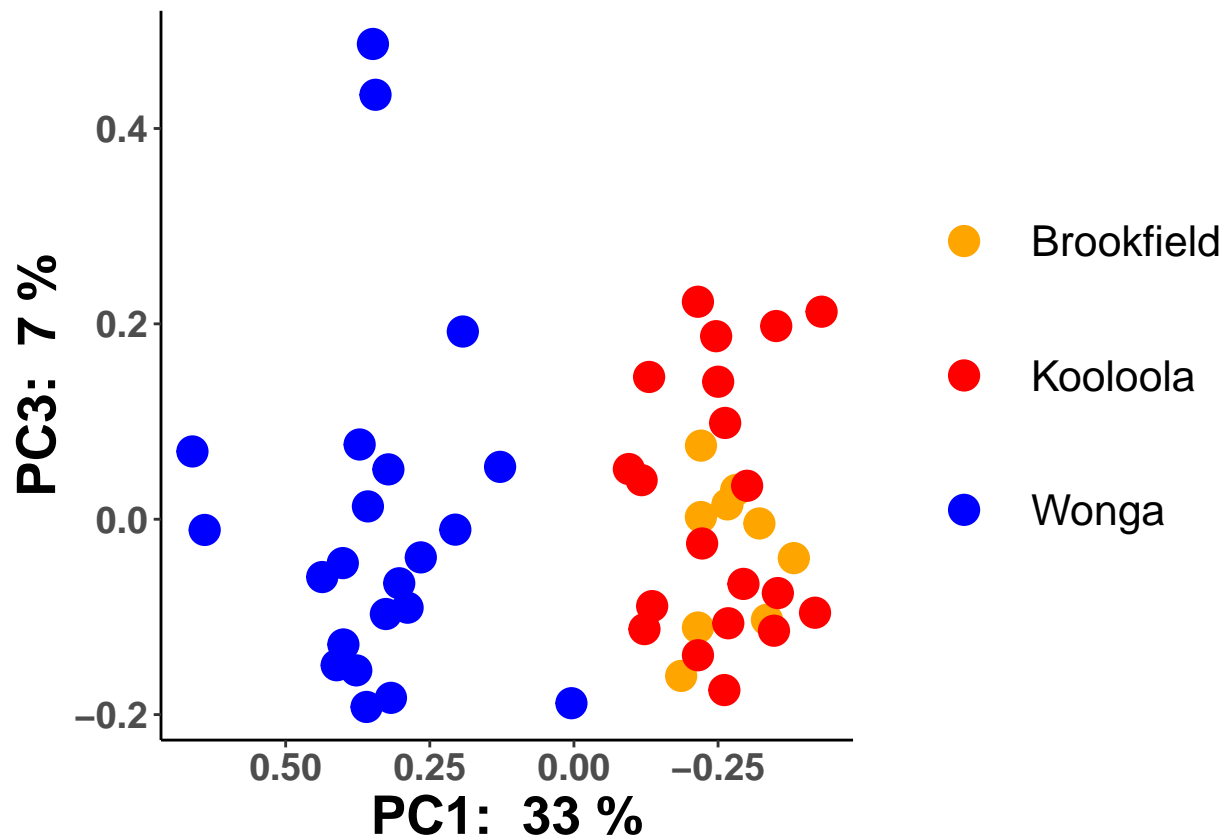


```
#Save image
ggsave(filename = "PC1-2_wUniFrac_wild.svg", width = 20, height = 11, dpi = 300)

#Plot PC1 vs. PC3 (wUniFrac_wild)
pcoa_wUniFrac_wild$data$Vectors %>%
  rename("X.SampleID"=SampleID) %>%
  left_join(metadata_pcoa) %>%

ggplot(aes(x=PC1, y=PC3, color=Population)) +
  scale_x_reverse() +
  geom_point(size=5) +
  xlab(paste("PC1: ", round(100*pcoa_wUniFrac_wild$data$ProportionExplained[1]), "%")) +
  ylab(paste("PC3: ", round(100*pcoa_wUniFrac_wild$data$ProportionExplained[3]), "%")) +
  #Custom manual colours
  scale_colour_manual(values=cbp2_beta_wild) +

  theme(axis.text.x = element_text(face="bold", size=14),
        axis.text.y = element_text(face="bold", size=14),
        axis.title.x = element_text(size=20, face="bold"),
        axis.title.y = element_text(size=20, face="bold"),
        axis.line = element_line(colour = "black"),
        #Background panel
        panel.background = element_rect(fill = "white"),
        panel.grid.major = element_line(colour = "white"),
        panel.grid.minor = element_line(colour = "white"),
        #Legend
        legend.title = element_text(size=0),
        legend.text = element_text(size=16),
        legend.key = element_rect(fill = "white", color = NA),
        legend.key.size = unit(3.5, "line"))
```



```
#Save image
ggsave(filename = "PC1-3_wUniFrac_wild.svg", width = 20, height = 11, dpi = 300)
```

Venn diagrams (figures 3 and 5)

```
##Captive vs. Wild
#Store only row names (ASV names) as character vectors
captive.asv.names <- row.names(captive.asvs)
wild.asv.names <- row.names(wild.asvs)

captive.vs.wild <- list(Captive = captive.asv.names, Wild = wild.asv.names)

venn.diagram(captive.vs.wild, print.mode = c("raw","percent"), fill = c("black", "white"), cat.fontface = 2,
             cat.pos = 180, imagetype = "png", filename = "R_figures/captive-vs-wild-venn.png")

## [1] 1

##Wild populations
#Store only row names (ASV names) as character vectors
brookfield.asv.names <- row.names(brookfield.asvs)
kooloola.asv.names <- row.names(kooloola.asvs)
wonga.asv.names <- row.names(wonga.asvs)

wild.comparison <- list(Brookfield = brookfield.asv.names, Kooloola = kooloola.asv.names, Wonga = wonga.asv.names)

venn.diagram(wild.comparison, print.mode = c("raw","percent"), fill = c("orange", "red", "blue"),
             cat.fontface = 2, cat.pos = c(320, 40, 180), imagetype = "png",
             filename = "R_figures/wild-populations-venn.png")

## [1] 1
```

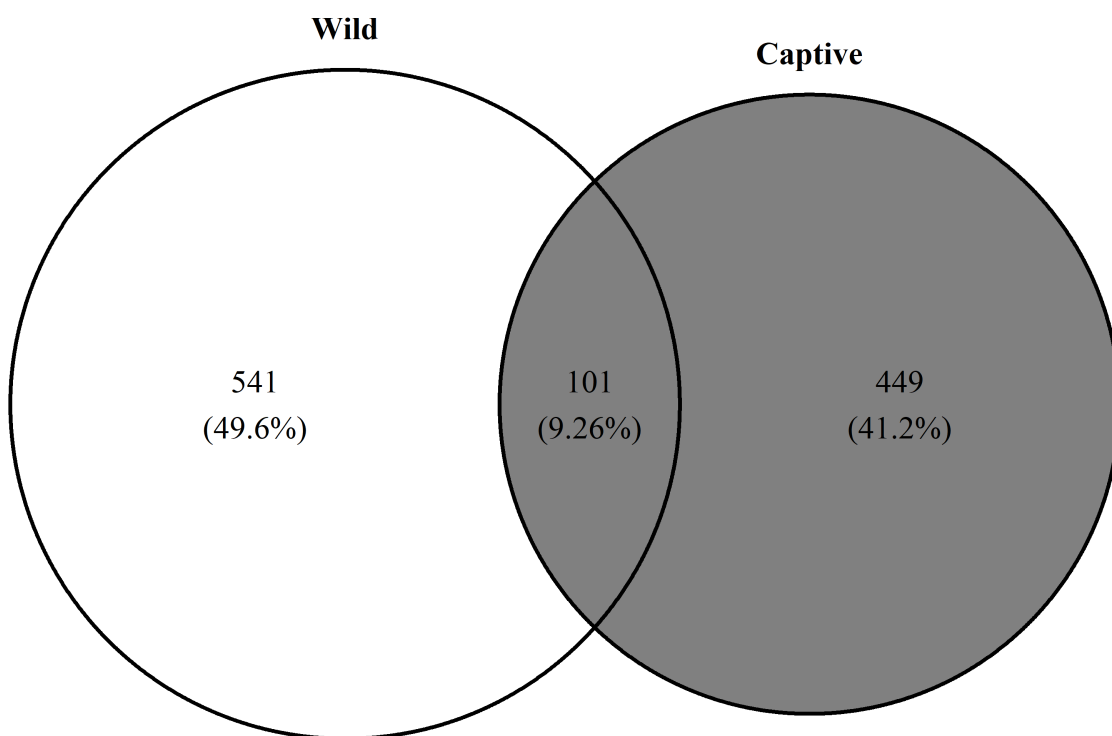


Figure 1: Captive and Wild

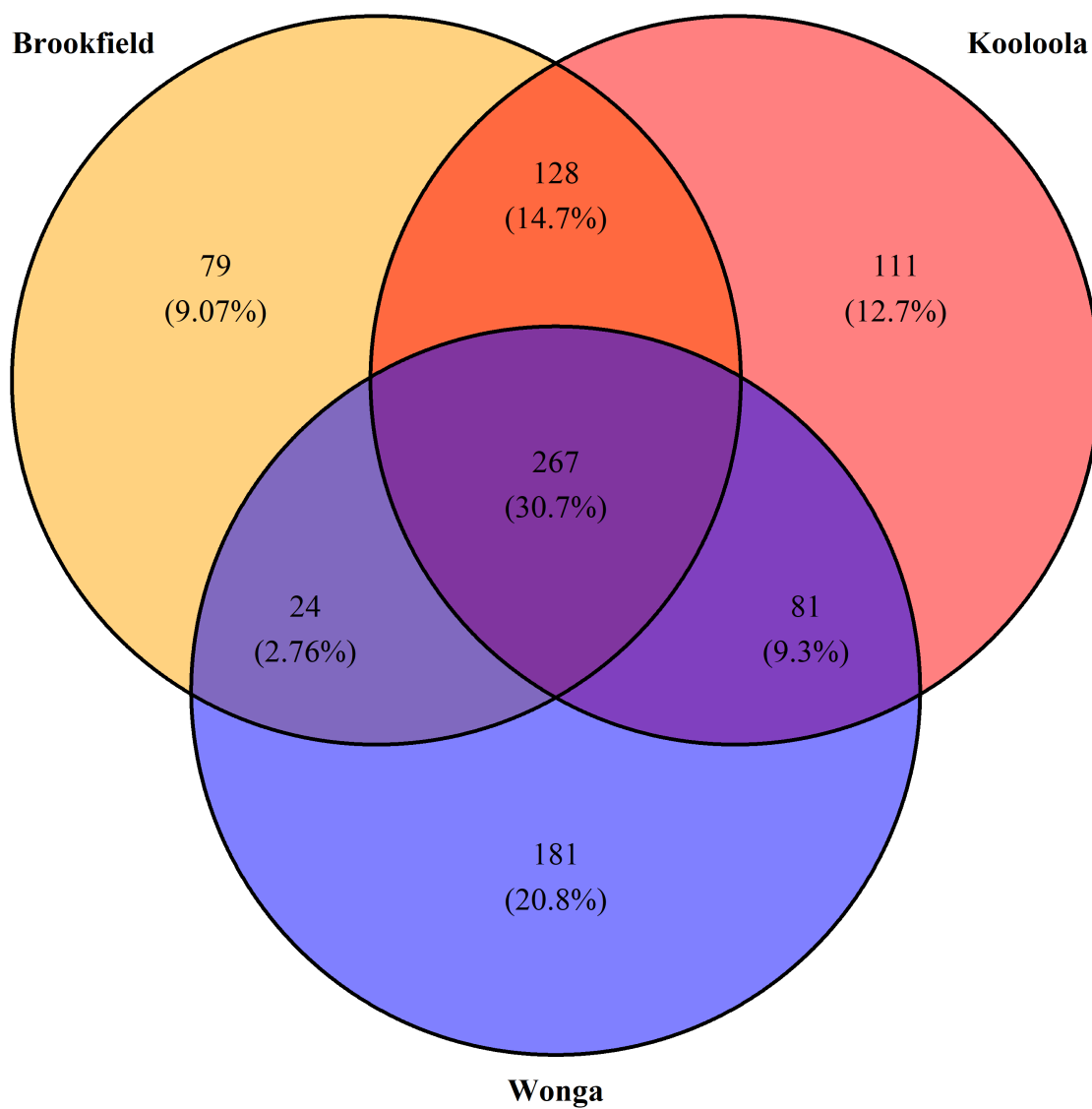


Figure 2: Captive and Wild

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19041)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Australia.1252 LC_CTYPE=English_Australia.1252
## [3] LC_MONETARY=English_Australia.1252 LC_NUMERIC=C
## [5] LC_TIME=English_Australia.1252
##
## attached base packages:
## [1] grid      stats      graphics  grDevices utils      datasets  methods
## [8] base
##
## other attached packages:
## [1] gdttools_0.2.2          ggVennDiagram_0.3.3
## [3] VennDiagram_1.6.20      futile.logger_1.4.3
## [5] microbiomeutilities_1.00.11 microbiome_1.10.0
## [7] knitr_1.29              tidyr_1.1.2
## [9] ggpubr_0.4.0            scales_1.1.1
## [11] ggplot2_3.3.2           gplots_3.1.0
## [13] cowplot_1.1.0           svglite_1.2.3.2
## [15] dplyr_1.0.2             qiime2R_0.99.34
## [17] phyloseq_1.32.0
##
## loaded via a namespace (and not attached):
## [1] Rtsne_0.15              colorspace_1.4-1        ggsignif_0.6.0
## [4] class_7.3-17            ellipsis_0.3.1          rio_0.5.16
## [7] htmlTable_2.1.0         XVector_0.28.0          base64enc_0.1-3
## [10] rstudioapi_0.11         farver_2.0.3            ggrepel_0.8.2
## [13] DT_0.15                 codetools_0.2-16        splines_4.0.2
## [16] ade4_1.7-15             Formula_1.2-3           jsonlite_1.7.1
## [19] broom_0.7.0             cluster_2.1.0           png_0.1-7
## [22] pheatmap_1.0.12         compiler_4.0.2          backports_1.1.9
## [25] Matrix_1.2-18           formatR_1.7             htmltools_0.5.0
## [28] tools_4.0.2             igraph_1.2.5            gtable_0.3.0
## [31] glue_1.4.2              reshape2_1.4.4          Rcpp_1.0.5
## [34] carData_3.0-4           Biobase_2.48.0          cellranger_1.1.0
## [37] vctr_0.3.4              Biostrings_2.56.0       zCompositions_1.3.4
## [40] multtest_2.44.0         ape_5.4-1              nlme_3.1-148
## [43] iterators_1.0.12        gghalves_0.1.1          xfun_0.17
## [46] stringr_1.4.0           openxlsx_4.2.2          lifecycle_0.2.0
## [49] gtools_3.8.2            rstatix_0.6.0           zlibbioc_1.34.0
## [52] MASS_7.3-51.6           hms_0.5.3              parallel_4.0.2
## [55] biomformat_1.16.0       rhdf5_2.32.2            lambda.r_1.2.4
## [58] RColorBrewer_1.1-2      yaml_2.2.1             curl_4.3
## [61] gridExtra_2.3           NADA_1.6-1.1           rpart_4.1-15
## [64] latticeExtra_0.6-29     stringi_1.5.3           S4Vectors_0.26.1
## [67] foreach_1.5.0          e1071_1.7-4            checkmate_2.0.0
## [70] permute_0.9-5          caTools_1.18.0         BiocGenerics_0.34.0
## [73] zip_2.1.1              truncnorm_1.0-8        cpp11_0.2.1
## [76] rlang_0.4.9            pkgconfig_2.0.3        systemfonts_0.3.1
## [79] bitops_1.0-6           evaluate_0.14          lattice_0.20-41
## [82] sf_0.9-6               purrr_0.3.4            Rhdf5lib_1.10.1
## [85] labeling_0.3           htmlwidgets_1.5.1      tidyselect_1.1.0
## [88] plyr_1.8.6             magrittr_1.5           R6_2.4.1
## [91] IRanges_2.22.2         generics_0.0.2         Hmisc_4.4-1
## [94] DBI_1.1.0              pillar_1.4.6           haven_2.3.1
## [97] foreign_0.8-80         withr_2.2.0            mgcv_1.8-31
## [100] units_0.6-7            survival_3.1-12        abind_1.4-5
## [103] nnet_7.3-14            tibble_3.0.3           crayon_1.3.4
```

```
## [106] car_3.0-9          futile.options_1.0.1 KernSmooth_2.23-17
## [109] rmarkdown_2.3      jpeg_0.1-8.1       readxl_1.3.1
## [112] data.table_1.13.0  vegan_2.5-6        forcats_0.5.0
## [115] classInt_0.4-3     digest_0.6.25      stats4_4.0.2
## [118] munsell_0.5.0
```