

Coevolution with a seed bank

03 February, 2023

Analyze composition of mutations from pooled population sequencing

Setup Work Environment

```
# Load dependencies  
library(here)
```

```
## here() starts at C:/Users/danschw/GitHub/coevolution/coevo-seedbank-seq
```

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
```

```
## v ggplot2 3.4.0      v purrr   0.3.5  
## v tibble  3.1.8      v dplyr   1.0.10  
## v tidyr   1.2.1      v stringr 1.4.1  
## v readr   2.1.3      v forcats 0.5.2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()     masks stats::lag()
```

```
library(vegan)
```

```
## Loading required package: permute  
## Loading required package: lattice  
## This is vegan 2.6-2
```

```
library(BiodiversityR)
```

```
## Loading required package: tcltk  
## BiodiversityR 2.14-4: Use command BiodiversityRGUI() to launch the Graphical User Interface;  
## to see changes use BiodiversityRGUI(changeLog=TRUE, backward.compatibility.messages=TRUE)
```

Load data

Matrix of multiplicity data organized as population X gene.

```
mutdat <- read_csv(here("data/mult_phage.csv")) %>%
  rename(trt = 1)

## New names:
## Rows: 6 Columns: 21
## -- Column specification
## ----- Delimiter: "," chr
## (1): ...1 dbl (20): SP01_6, SP01_8, SP01_15, SP01_30, SP01_39, SP01_53,
## SP01_74, SP01_...
## 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.
## * `` -> `...1`
```

PCoA procedures

```
# Define treatments and data
seed <- str_detect(mutdat$trt, "long")

# multiplicity data only
mut <- mutdat %>% select(-trt)

# Calculate pairwise distances
mut.dist <- vegdist(mut, method = "bray", binary = "FALSE")

# Principal Coordinates Analysis (PCoA)
pc <- cmdscale(mut.dist, eig = TRUE, k = nrow(mut)-1)
```

```
## Warning in cmdscale(mut.dist, eig = TRUE, k = nrow(mut) - 1): only 4 of the
## first 5 eigenvalues are > 0
```

```
explainvar1 <- round(pc$eig[1] / sum(pc$eig), 3) * 100
explainvar2 <- round(pc$eig[2] / sum(pc$eig), 3) * 100
explainvar3 <- round(pc$eig[3] / sum(pc$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

p <-
  as_tibble(pc$points, .name_repair = "universal" ) %>%
  rename_with(~gsub("...", "PCoA", .x, fixed = TRUE)) %>%
  mutate(seed = seed) %>%
  relocate(seed, .before = 1) %>%
  mutate(seed = if_else(seed==1, "+seed bank", "-seed bank" ) %>%
  ggplot(aes(x=PCoA1,y=PCoA2)) +
```

```

geom_point(aes(color = seed, fill = seed), size=3, stroke=1,alpha=0.8)+
# geom_polygon(data = dl, linetype = 3 ,fill="transparent",
#             aes(x=x, y =y, group = interaction(seed, phage),color = seed))+
theme_bw(base_size=32) +
labs(x=paste0("PCo 1 (",round(explainvar1,1),"%)" ),
      y=paste0("PCo 2 (",round(explainvar2,1),"%)" )) +
geom_hline(yintercept = 0, linetype = 3)+
geom_vline(xintercept = 0, linetype = 3)+
scale_shape_manual(values = c(21,24))+
scale_fill_grey(end = 0.8, name = "seed bank")+
scale_color_grey(end = 0.6, name = "seed bank")+
scale_x_continuous(sec.axis = dup_axis(name = NULL, labels = NULL),
                   limits = c(-0.6,0.6)) +
scale_y_continuous(sec.axis = dup_axis(name = NULL, labels = NULL),
                   limits = c(-0.6,0.6))+
theme_classic(base_size = 16)

```

```

## New names:
## * `` -> `...1`
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`

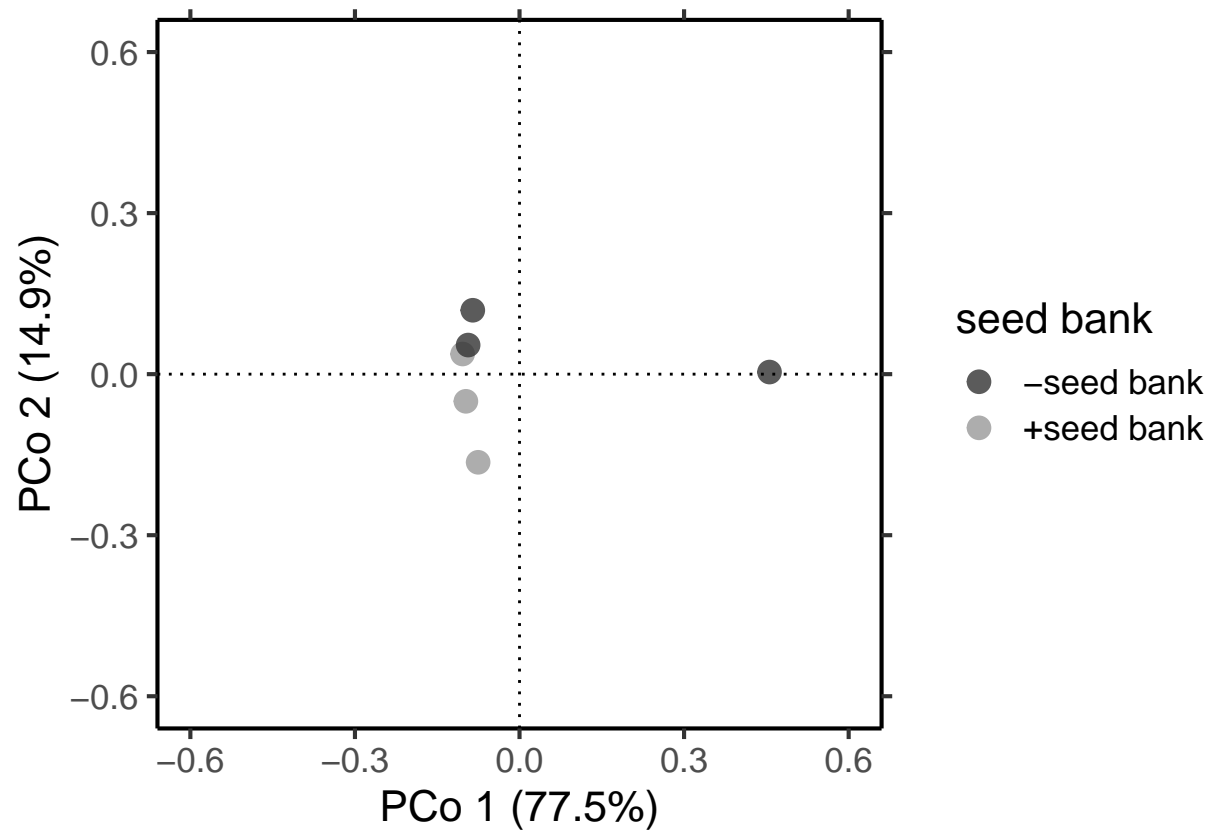
```

```

ggsave(here("analysis","PCoA_mult_phage.png"),p, width = 5, height = 3)

```

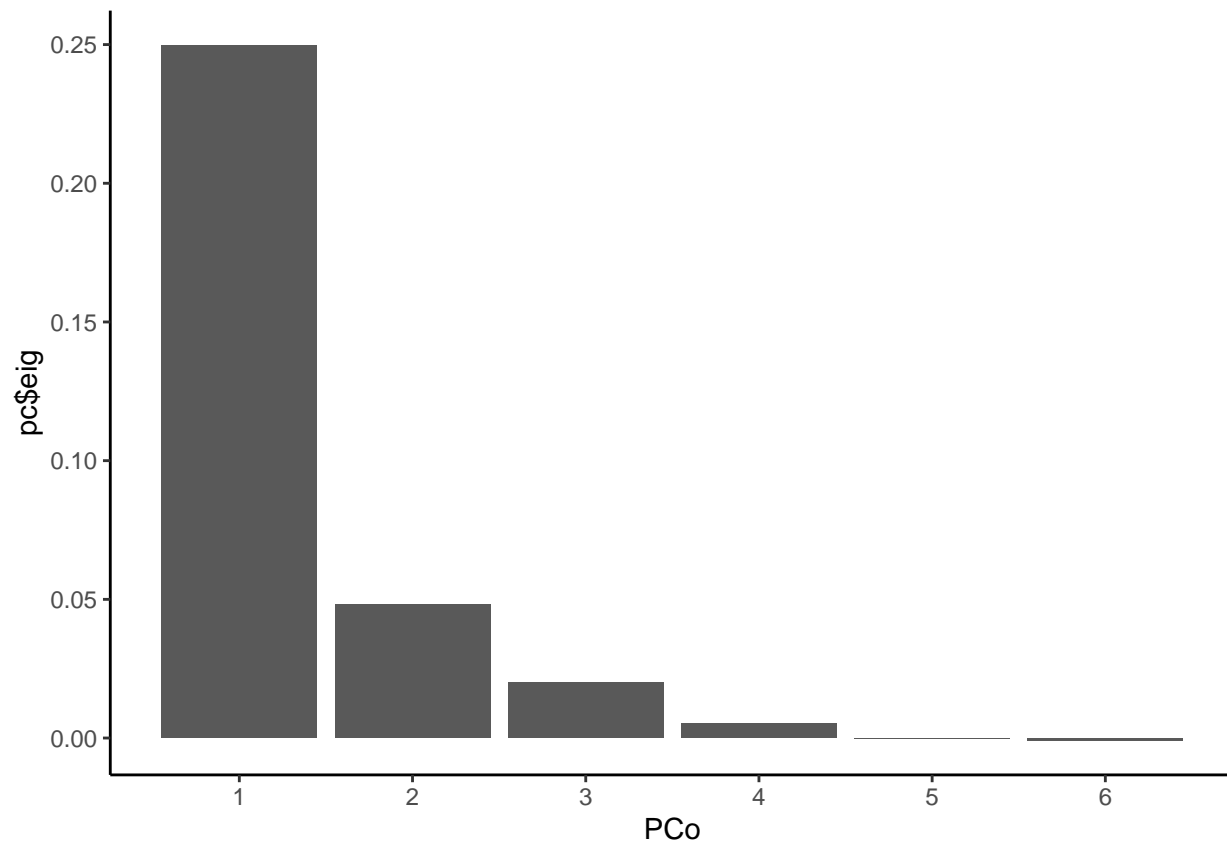
p



There are negative eigenvalues

```
qplot(1:6,pc$eig, geom = "col")+
  theme_classic()+
  scale_x_continuous(breaks = 1:6)+
  xlab("PCo")
```

Warning: `qplot()` was deprecated in ggplot2 3.4.0.



```
library(ape)

# Principal Coordinates Analysis (PCoA)
pc.ape <- pcoa(mut.dist, correction = "none")
pc.ape.corct <- pcoa(mut.dist, correction = "lingoes")

tibble(PCo = 1:6,
  cmdscale = pc$eig,
  ape = pc.ape$values$Eigenvalues,
  ape_corrected = pc.ape.corct$values$Eigenvalues)
```

```
## # A tibble: 6 x 4
##   PCo cmdscale      ape ape_corrected
##   <int>   <dbl>   <dbl>      <dbl>
## 1     1 2.50e- 1 0.250        0.250
## 2     2 4.80e- 2 0.0480       0.0480
## 3     3 2.00e- 2 0.0200       0.0200
## 4     4 5.22e- 3 0.00522      0.00522
## 5     5 -4.55e-18 0          0
## 6     6 -7.67e- 4 -0.000767   -0.000767
```

```
# ape pcoa has only one single negative, and small value
```

```

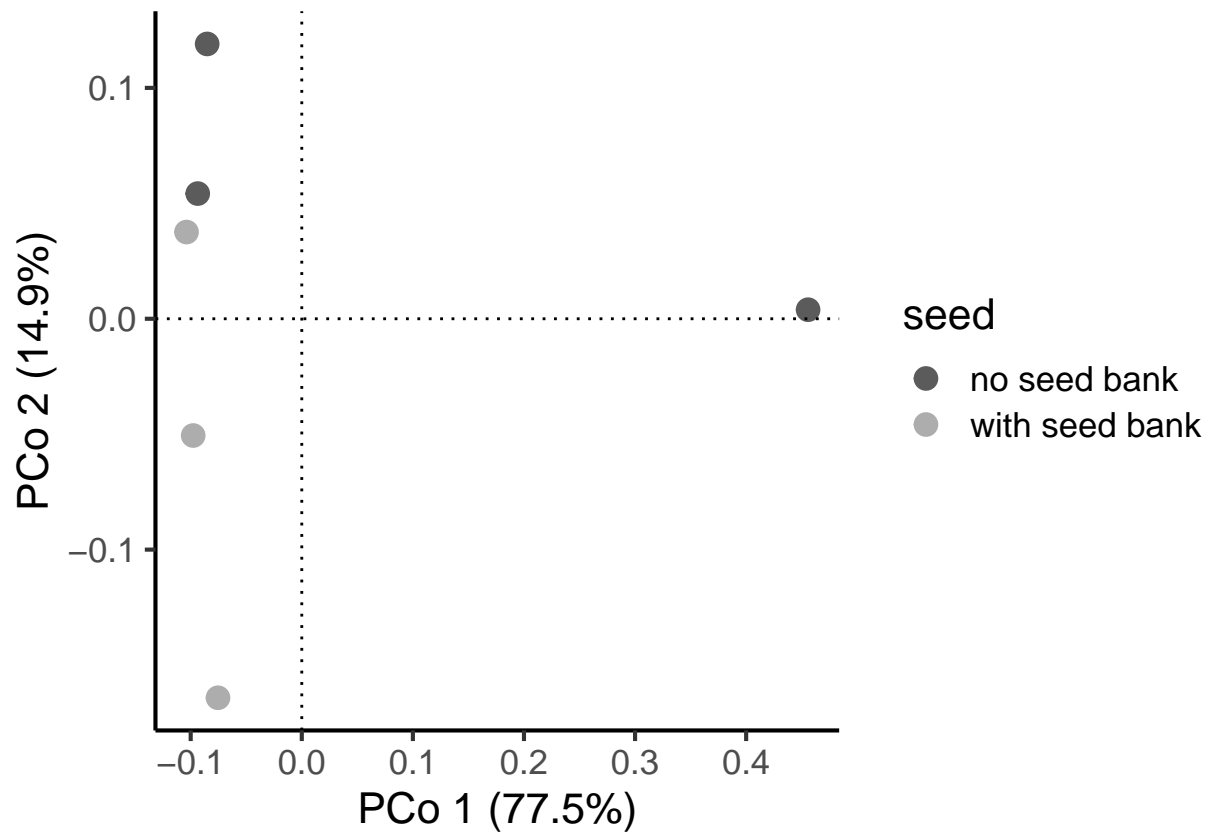
explainvar1.ape <-
  round(pc.ape.corct$values$Eigenvalues[1] / sum(pc.ape.corct$values$Eigenvalues), 3) * 100
explainvar2.ape <-
  round(pc.ape.corct$values$Eigenvalues[2] / sum(pc.ape.corct$values$Eigenvalues), 3) * 100
explainvar3.ape <-
  round(pc.ape.corct$values$Eigenvalues[3] / sum(pc.ape.corct$values$Eigenvalues), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

p <-
  as_tibble(pc.ape.corct$vectors, .name_repair = "universal" ) %>%
  rename_with(~gsub("Axis.", "PCoA", .x, fixed = TRUE)) %>%
  mutate(seed = seed) %>%
  relocate(seed, .before = 1) %>%
  mutate(seed = if_else(seed==1, "with seed bank", "no seed bank") ) %>%
  ggplot(aes(x=PCoA1,y=PCoA2)) +
  geom_point(aes(color = seed, fill = seed), size=3, stroke=1,alpha=0.8)+
  # geom_polygon(data = dl, linetype = 3 ,fill="transparent",
  #               aes(x=x, y =y, group = interaction(seed, phage),color = seed))+
  theme_bw(base_size=32) +
  labs(x=paste0("PCo 1 (",round(explainvar1,1),"%)" ),
       y=paste0("PCo 2 (",round(explainvar2,1),"%)" )) +
  geom_hline(yintercept = 0, linetype = 3)+
  geom_vline(xintercept = 0, linetype = 3)+
  scale_shape_manual(values = c(21,24))+
  scale_fill_grey(end = 0.8)+
  scale_color_grey(end = 0.6)+
  # scale_x_continuous(sec.axis = dup_axis(name = NULL, labels = NULL),
  #                    limits = c(-0.4,0.4)) +
  # scale_y_continuous(sec.axis = dup_axis(name = NULL, labels = NULL),
  #                    limits = c(-0.4,0.4))+
  theme_classic(base_size = 16)

# ggsave(here("analysis", "PCoA_mult_phage.png"),p, width = 5, height = 3)

p

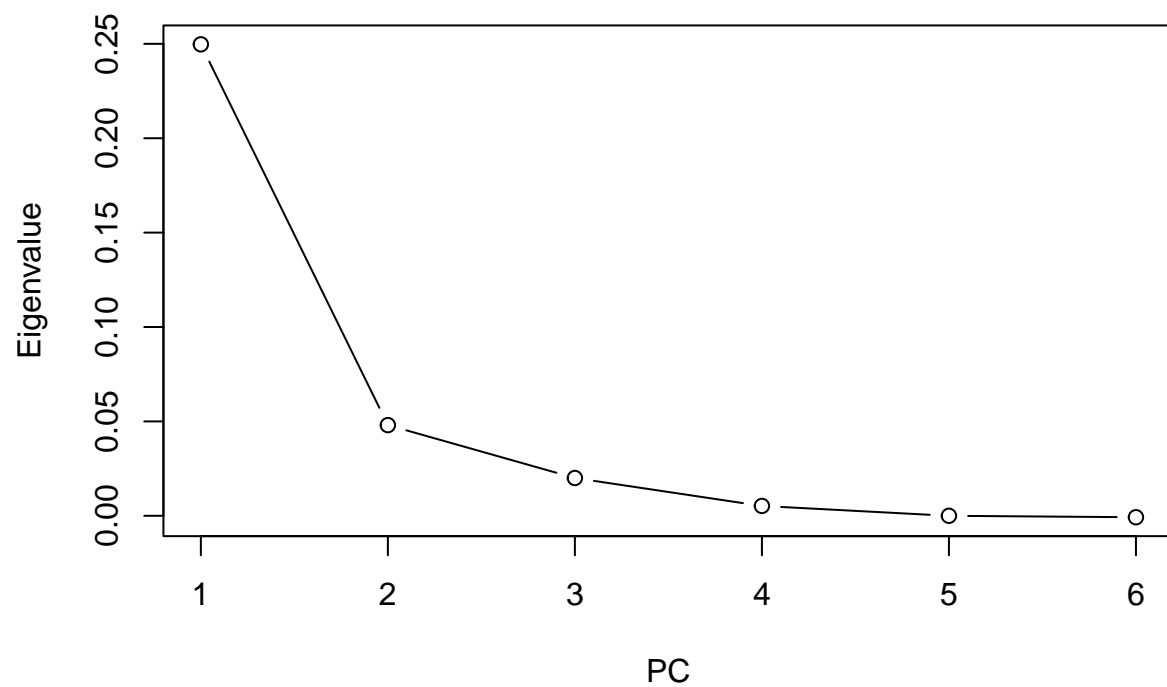
```



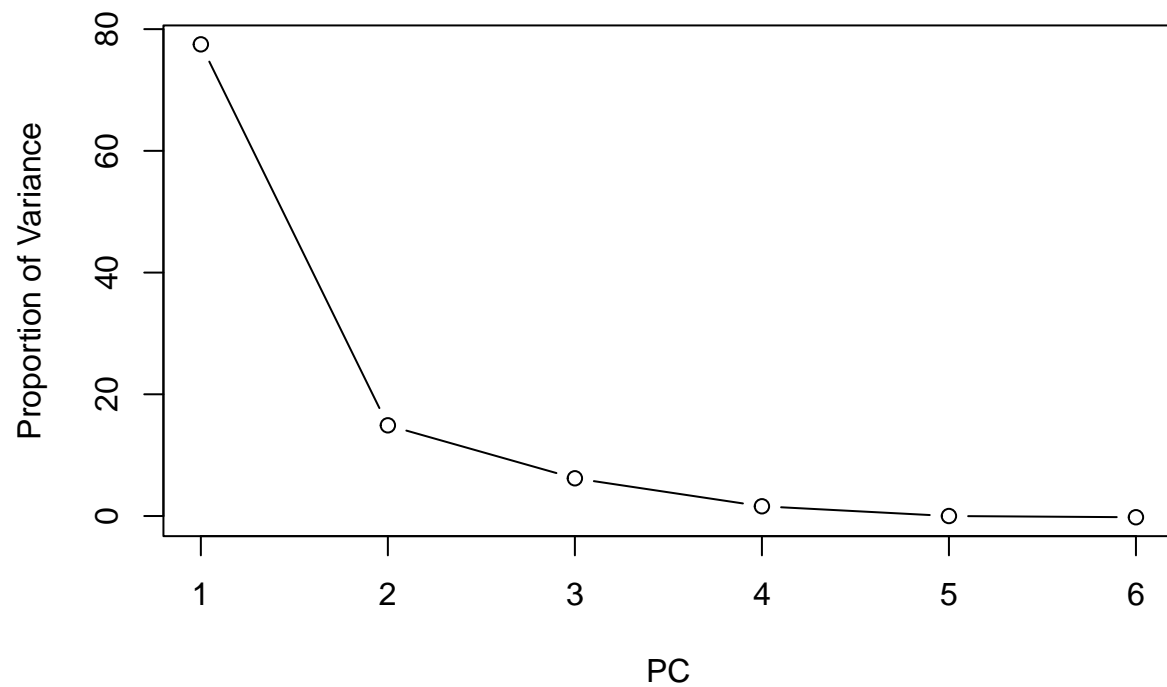
Using APE pcoa with correction does not change anything. Defaulting back to the same procedure used for host.

How many PCs to include in stats?

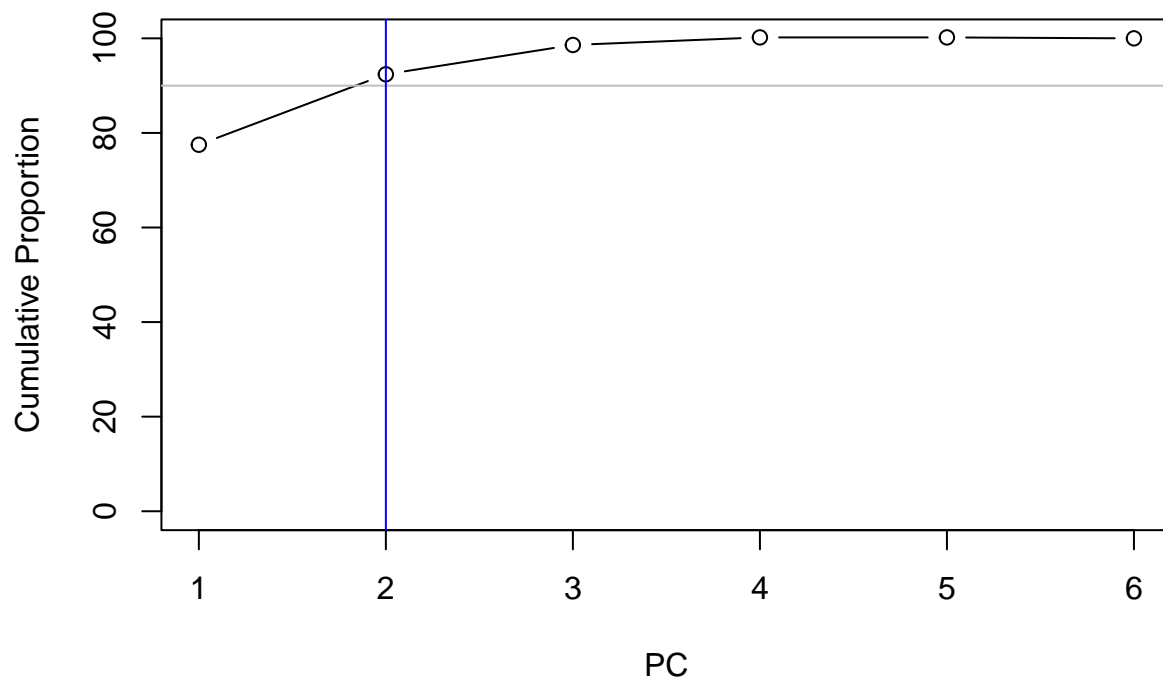
```
plot(1:length(pc$eig),pc$eig,type = "b",
     xlab = "PC", ylab = "Eigenvalue")
```



```
prop_var <- round(pc$eig / sum(pc$eig), 3) * 100
plot(1:length(pc$eig),prop_var, type = "b",
     xlab = "PC", ylab = "Proportion of Variance")
```

```
# PCs explaining 90% variation
pc_var90 <- min(which(cumsum(prop_var)>90))
plot(1:length(pc$eig),cumsum(prop_var), type = "b",
     xlab = "PC", ylab = "Cumulative Proportion", ylim = c(0,100))
abline(v=pc_var90,h=90, col = c("grey", "blue"))
```



First 2 PCs explain >90% of the variation

PERMANOVA

```
## on multiplicity data
# perm <-
#   adonis2(mut.dist ~ seed * phage,
#           binary = FALSE, permutations = 9999)

# on PCs explaining >90% var
perm <-
  adonis2(pc$points[,1:pc_var90] ~ seed ,method = "euclidean",
          binary = FALSE, permutations = 9999)
```

```
## 'nperm' >= set of all permutations: complete enumeration.
```

```
## Set of permutations < 'minperm'. Generating entire set.
```

```
perm
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
```

```
## Permutation: free
## Number of permutations: 719
##
## adonis2(formula = pc$points[, 1:pc_var90] ~ seed, permutations = 9999, method = "euclidean", binary =
##           Df SumOfSqs      R2      F Pr(>F)
## seed      1 0.072048 0.24196 1.2768    0.2
## Residual   4 0.225718 0.75804
## Total      5 0.297766 1.00000
```

Gene correlations

```
# Phage annotation -----
d.spo1 <-
  read_delim(here("data/SP01-ANC_no_fasta.gff3"),
    col_names = c(letters[1:9]),
    trim_ws = T,
    comment = "#") %>%
  select(seqname = a, feature = c, start = d, end = e,
    strand = g, frame = h, attribute = i)

## Rows: 425 Columns: 9
## -- Column specification -----
## Delimiter: "\t"
## chr (7): a, b, c, f, g, h, i
## dbl (2): d, e
##
## 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.
```

```
d.spo1_genes <-
  d.spo1 %>%
  filter(feature == "gene") %>%
  separate(attribute, into = c("Alias", "ID", "Name", "Pseudo"), sep = ";") %>%
  mutate(Alias = str_remove(Alias, ".*="),
    ID = str_remove(ID, ".*="),
    Name = str_remove(Name, ".*="),
    Pseudo = str_remove(Pseudo, ".*="))
```

```
## Warning: Expected 4 pieces. Missing pieces filled with `NA` in 208 rows [1, 2,
## 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
# Test correlation to PCoA axes -----
gene.corr <- add.spec.scores(pc, mut, method = "cor.scores")$cproj
gene.corr <-
  tibble(gene = rownames(gene.corr)) %>%
  bind_cols(as_tibble(gene.corr))

# Genes correlated with PCo1 -----

fit <- envfit(pc, mut, choices=1, perm = 999)
```

```
## 'nperm' >= set of all permutations: complete enumeration.
## Set of permutations < 'minperm'. Generating entire set.
## 'nperm' >= set of all permutations: complete enumeration.
## Set of permutations < 'minperm'. Generating entire set.
```

```
d.fit1 <- tibble(gene = names(fit$vectors$r),
  r = fit$vectors$r,
  pvals = fit$vectors$pvals)

# combine with Correlation for significant genes
sig_genes1 <- d.fit1 %>%
  filter(pvals<0.05) %>%
  left_join(., gene.corr %>% select(gene, cor = Dim1))
```

```
## Joining, by = "gene"
```

```
# add annotations
sig_genes1 <- d.spo1_genes %>%
  select(Alias, Name, strand) %>%
  left_join(sig_genes1, ., by = c("gene" = "Alias"))

# export positively correlated
sig_genes1 %>%
  filter(cor > 0) %>%
  arrange(desc(abs(cor))) %>%
  rename(P_value = pvals) %>% write_csv(here("data", "significant_PHAGE_genes_pc1_positive.csv"))

# export negatively correlated
sig_genes1 %>%
  filter(cor < 0) %>%
  arrange(desc(abs(cor))) %>%
  rename(P_value = pvals) %>%
  write_csv(here("data", "significant_PHAGE_genes_pc1_negative.csv"))

# Genes correlated wit PCo2 -----
fit <- envfit(pc, mut, choices=2, perm = 999)
```

```
## 'nperm' >= set of all permutations: complete enumeration.
## Set of permutations < 'minperm'. Generating entire set.
## 'nperm' >= set of all permutations: complete enumeration.
## Set of permutations < 'minperm'. Generating entire set.
```

```
d.fit2 <- tibble(gene = names(fit$vectors$r),
  r = fit$vectors$r,
  pvals = fit$vectors$pvals)

# Correlation of significant genes
sig_genes2 <- d.fit2 %>%
  filter(pvals<0.05) %>%
  left_join(., gene.corr %>% select(gene, cor = Dim2))
```

```
## Joining, by = "gene"
```

```
# none significant
```

Ellipses

The ggplot function of `stat_ellipse` does not allow CI ellipses on less than 4 data points. We have three points per treatment. However three points should be allowed “because your CI depends on the variance, which takes two degrees of freedom”.

According to `stat_ellipses` help “The method for calculating the ellipses has been modified from `car::dataEllipse` (Fox and Weisberg, 2011)”. The limit on 3 points does not exist in the original function.

```
library(car)
```

```
## Loading required package: carData
```

```
##
```

```
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

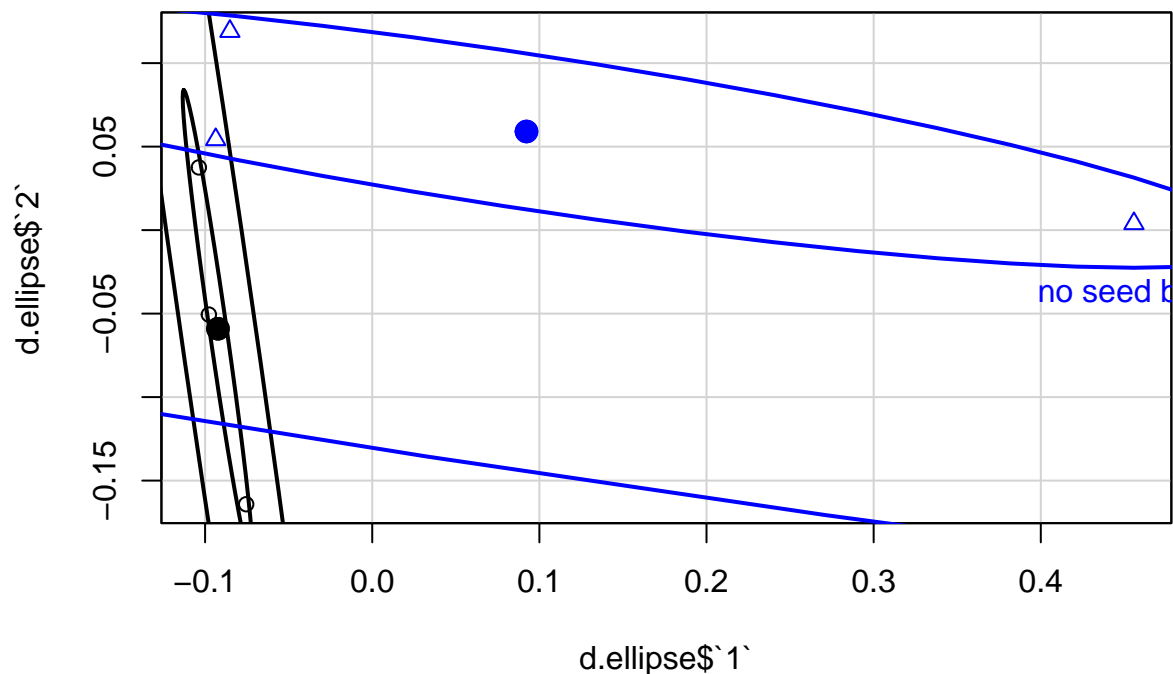
```
##      recode
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
##      some
```

```
d.ellipse <- cbind(mut[,1:2],pc$points) %>%  
  as.data.frame %>%  
  mutate(seed = if_else(seed, "with seed bank", "no seed bank") %>% as_factor())  
  
el <- dataEllipse(d.ellipse$`1`, d.ellipse$`2`, groups = d.ellipse$seed)
```



```
# unpack list
dl <- rbind(
  cbind("with seed bank", el$`with seed bank`$`0.95`),
  cbind("no seed bank", el$`no seed bank`$`0.95`)
)
```

```
dl <- dl %>%
  as_tibble() %>%
  mutate(x= as.numeric(x), y=as.numeric(y)) %>%
  rename(seed = V1)
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
## `.name_repair` is omitted as of tibble 2.0.0.
## i Using compatibility `.name_repair`.
```

PCA with ellipses

```
p <-
  as_tibble(pc$points, .name_repair = "universal" ) %>%
  rename_with(~gsub("...", "PCoA", .x, fixed = TRUE)) %>%
  mutate(seed = seed) %>%
  relocate(seed, .before = 1) %>%
```

```

mutate(seed = if_else(seed==1, "with seed bank", "no seed bank") ) %>%
ggplot(aes(x=PCoA1,y=PCoA2)) +
geom_point(aes(color = seed, fill = seed), size=3, stroke=1,alpha=0.8)+
geom_polygon(data = dl, linetype = 1 ,fill="transparent",
aes(x=x, y =y,color = seed))+
theme_bw(base_size=32) +
labs(x=paste0("PCo 1 (",round(explainvar1,1),"%)" ),
y=paste0("PCo 2 (",round(explainvar2,1),"%)" ) +
geom_hline(yintercept = 0, linetype = 3)+
geom_vline(xintercept = 0, linetype = 3)+
scale_shape_manual(values = c(21,24))+
scale_fill_grey(end = 0.8)+
scale_color_grey(end = 0.6)+
# scale_x_continuous(sec.axis = dup_axis(name = NULL, labels = NULL),
# limits = c(-0.4,0.4)) +
# scale_y_continuous(sec.axis = dup_axis(name = NULL, labels = NULL),
# limits = c(-0.4,0.4))+
theme_classic(base_size = 16)

```

```

## New names:
## * `` -> `...1`
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`

```

p

