

Comparisons

Ezra Valido

2025-10-30

```
library(readxl)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(vegan)
```

```
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.6-6.1
```

```
library(compositions)
```

```
## Welcome to compositions, a package for compositional data analysis.
## Find an intro with "? compositions"
##
##
## Attaching package: 'compositions'
##
## The following objects are masked from 'package:stats':
##
##   anova, cor, cov, dist, var
##
## The following object is masked from 'package:graphics':
##
##   segments
##
## The following objects are masked from 'package:base':
##
##   %*%, norm, scale, scale.default
```

```
PCA <- read_excel("C:/Users/ezrav/Documents/R_Projects/UROVAX/PCA_full.xlsx", sheet = "Tabelle1")
str(PCA)
```

```
## tibble [63 x 54] (S3: tbl_df/tbl/data.frame)
## $ Sample      : chr [1:63] "PI" "PI" "PI" "PI" ...
## $ Donor       : chr [1:63] "Donor 3" "Donor 4" "Donor 5" "Donor 6" ...
## $ ID          : num [1:63] 1 2 3 4 5 6 7 8 9 10 ...
## $ Aerococcus  : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Anaerococcus : num [1:63] 0.0203 0 0 0.1115 0 ...
## $ Aquabacterium : num [1:63] 0 0 0 0 0 ...
## $ Bacillus    : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Bacteroides  : num [1:63] 0.1046 0 0.0935 0.0792 0 ...
## $ Blautia     : num [1:63] 0 0 0 0 0 ...
## $ Brevibacterium : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Burkholderia : num [1:63] 0 0 0 0 0 ...
## $ Campylobacter : num [1:63] 0 0 0 0 0 ...
## $ Capnocytophaga : num [1:63] 0 0 0 0 0 ...
## $ Citrobacter  : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Clostridium  : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Corynebacterium : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Cutibacterium : num [1:63] 0 0.0266 0 0 0 ...
## $ Dialister    : num [1:63] 0 0 0 0 0.185 ...
## $ Eisenbergiella : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Enterobacter : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Enterococcus : num [1:63] 0 0 0 0 0 ...
## $ Escherichia  : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Ezakiella    : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Fenollaria   : num [1:63] 0 0 0 0 0.0877 ...
## $ Finegoldia   : num [1:63] 0.0289 0.0816 0 0.0937 0 ...
## $ Fusobacterium : num [1:63] 0.31 0 0 0 0.108 ...
## $ Gemella      : num [1:63] 0 0 0 0.114 0 ...
## $ Hafnia       : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Klebsiella   : num [1:63] 0 0 0 0 0 ...
## $ Lachnoclostridium : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Mogibacterium : num [1:63] 0 0 0 0 0 ...
## $ Morganella   : num [1:63] 0 0 0.0177 0 0 ...
## $ Parvimonas   : num [1:63] 0 0 0.02 0 0.0771 ...
## $ Peptoniphilus : num [1:63] 0.0124 0 0 0.0644 0.034 ...
## $ Peptostreptococcus : num [1:63] 0 0 0 0.13 0.126 ...
## $ Porphyromonas : num [1:63] 0 0.0796 0 0 0 ...
## $ Prevotella   : num [1:63] 0.0147 0.1057 0 0.3096 0.2783 ...
## $ Proteus      : num [1:63] 0.5 0 0.425 0 0 ...
## $ Pseudomonas  : num [1:63] 0 0 0.432 0 0 ...
## $ Ruminococcus : num [1:63] 0 0 0 0 0 ...
## $ Salmonella   : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Serratia     : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Solobacterium : num [1:63] 0 0 0 0.0462 0 ...
## $ Sphingobacterium : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Staphylococcus : num [1:63] 0 0.288 0 0 0 ...
## $ Streptococcus : num [1:63] 0 0.2121 0 0.0333 0 ...
## $ Subdoligranulum : num [1:63] 0 0 0 0 0 ...
## $ Veillonella  : num [1:63] 0 0.158 0 0 0 ...
## $ Raoultella   : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ Alcaligenes      : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Eikenella        : num [1:63] 0 0 0 0 0 0 0 0 0 0 ...
## $ Haemophilus      : num [1:63] 0 0 0 0 0 ...
## $ Others           : num [1:63] 0.00876 0.04796 0.01184 0.0176 0.10355 ...
## $ Observed         : num [1:63] 9 9 7 11 9 8 9 6 11 10 ...
```

```
PCA_data <- PCA %>% dplyr::select(!c(Sample, Donor, Observed))
PCA_meta <- PCA %>% dplyr::select(c(Sample, Donor, ID, Observed))
```

```
bs_base_g <- PCA_data
bs_base_g <- bs_base_g %>%
  column_to_rownames(var="ID")
bs_base_g<-bs_base_g+1 #We are adding 1 so that there are no zero values during the log transformation
bs_base_clr_g<-clr(bs_base_g, ifclose = FALSE, ifwarn = TRUE)
bs_base_pr_g<- prcomp(bs_base_clr_g)
```

```
ggbiplot <- function(pcoobj, choices = 1:2, scale = 1, pc.biplot = TRUE,
  obs.scale = 1 - scale, var.scale = scale,
  groups = NULL, ellipse = FALSE, ellipse.prob = 0.68,
  labels = NULL, labels.size = 3, alpha = 1,
  var.axes = TRUE,
  circle = FALSE, circle.prob = 0.69,
  varname.size = 3, varname.adjust = 1.5,
  varname.abbrev = FALSE, ...)
{
  library(ggplot2)
  library(plyr)
  library(scales)
  library(grid)

  stopifnot(length(choices) == 2)

  # Recover the SVD
  if(inherits(pcoobj, 'prcomp')){
    nobs.factor <- sqrt(nrow(pcoobj$x) - 1)
    d <- pcoobj$sdev
    u <- sweep(pcoobj$x, 2, 1 / (d * nobs.factor), FUN = '*')
    v <- pcoobj$rotation
  } else if(inherits(pcoobj, 'princomp')) {
    nobs.factor <- sqrt(pcoobj$n.obs)
    d <- pcoobj$sdev
    u <- sweep(pcoobj$scores, 2, 1 / (d * nobs.factor), FUN = '*')
    v <- pcoobj$loadings
  } else if(inherits(pcoobj, 'PCA')) {
    nobs.factor <- sqrt(nrow(pcoobj$call$X))
    d <- unlist(sqrt(pcoobj$eig)[1])
    u <- sweep(pcoobj$ind$coord, 2, 1 / (d * nobs.factor), FUN = '*')
    v <- sweep(pcoobj$var$coord, 2, sqrt(pcoobj$eig[1:ncol(pcoobj$var$coord),1]), FUN="/")
  } else if(inherits(pcoobj, "lda")) {
    nobs.factor <- sqrt(pcoobj$N)
    d <- pcoobj$svd
    u <- predict(pcoobj)$x/nobs.factor
    v <- pcoobj$scaling
    d.total <- sum(d^2)
```

```

} else {
  stop('Expected a object of class prcomp, princomp, PCA, or lda')
}

# Scores
choices <- pmin(choices, ncol(u))
df.u <- as.data.frame(sweep(u[,choices], 2, d[choices]^obs.scale, FUN='*'))

# Directions
v <- sweep(v, 2, d^var.scale, FUN='*')
df.v <- as.data.frame(v[, choices])

names(df.u) <- c('xvar', 'yvar')
names(df.v) <- names(df.u)

if(pc.biplot) {
  df.u <- df.u * nobs.factor
}

# Scale the radius of the correlation circle so that it corresponds to
# a data ellipse for the standardized PC scores
r <- sqrt(qchisq(circle.prob, df = 2)) * prod(colMeans(df.u^2))^(1/4)

# Scale directions
v.scale <- rowSums(v^2)
df.v <- r * df.v / sqrt(max(v.scale))

# Change the labels for the axes
if(obs.scale == 0) {
  u.axis.labs <- paste('standardized PC', choices, sep='')
} else {
  u.axis.labs <- paste('PC', choices, sep='')
}

# Append the proportion of explained variance to the axis labels
u.axis.labs <- paste(u.axis.labs,
  sprintf('(%0.1f%% explained var.)',
    100 * pcobj$sdev[choices]^2/sum(pcobj$sdev^2)))

# Score Labels
if(!is.null(labels)) {
  df.u$labels <- labels
}

# Grouping variable
if(!is.null(groups)) {
  df.u$groups <- groups
}

# Variable Names
if(varname.abbrev) {
  df.v$varname <- abbreviate(rownames(v))
} else {

```

```

df.v$varname <- rownames(v)
}

# Variables for text label placement
df.v$angle <- with(df.v, (180/pi) * atan(yvar / xvar))
df.v$hjust = with(df.v, (1 - varname.adjust * sign(xvar)) / 2)

# Base plot
g <- ggplot(data = df.u, aes(x = xvar, y = yvar)) +
  xlab(u.axis.labs[1]) + ylab(u.axis.labs[2]) + coord_equal()

if(var.axes) {
  # Draw circle
  if(circle)
  {
    theta <- c(seq(-pi, pi, length = 50), seq(pi, -pi, length = 50))
    circle <- data.frame(xvar = r * cos(theta), yvar = r * sin(theta))
    g <- g + geom_path(data = circle, color = muted('white'),
                      size = 1/2, alpha = 1/3)
  }

  # Draw directions
  g <- g +
    geom_segment(data = df.v,
                 aes(x = 0, y = 0, xend = xvar, yend = yvar),
                 arrow = arrow(length = unit(1/2, 'picas')),
                 color = muted('red'))
}

# Draw either labels or points
if(!is.null(df.u$labels)) {
  if(!is.null(df.u$groups)) {
    g <- g + geom_text(aes(label = labels, color = groups),
                      size = labels.size)
  } else {
    g <- g + geom_text(aes(label = labels), size = labels.size)
  }
} else {
  if(!is.null(df.u$groups)) {
    g <- g + geom_point(aes(color = groups), alpha = alpha)
  } else {
    g <- g + geom_point(alpha = alpha)
  }
}

# Overlay a concentration ellipse if there are groups
if(!is.null(df.u$groups) && ellipse) {
  theta <- c(seq(-pi, pi, length = 50), seq(pi, -pi, length = 50))
  circle <- cbind(cos(theta), sin(theta))

  ell <- ddply(df.u, 'groups', function(x) {
    if(nrow(x) <= 2) {
      return(NULL)
    }
  })
}

```

```

    }
    sigma <- var(cbind(x$xvar, x$yvar))
    mu <- c(mean(x$xvar), mean(x$yvar))
    ed <- sqrt(qchisq(ellipse.prob, df = 2))
    data.frame(sweep(circle %% chol(sigma) * ed, 2, mu, FUN = '+'),
               groups = x$groups[1])
  })
  names(ell)[1:2] <- c('xvar', 'yvar')
  g <- g + geom_path(data = ell, aes(color = groups, group = groups))
}

# Label the variable axes
if(var.axes) {
  g <- g +
    geom_text(data = df.v,
              aes(label = varname, x = xvar, y = yvar,
                  angle = angle, hjust = hjust),
              color = 'darkred', size = varname.size)
}

# Change the name of the legend for groups
# if(!is.null(groups)) {
#   g <- g + scale_color_brewer(name = deparse(substitute(groups)),
#                               palette = 'Dark2')
# }

# TODO: Add a second set of axes

return(g)
}

ggscreeplot <- function(pcobj, type = c('pev', 'cev'))
{
  type <- match.arg(type)
  d <- pcobj$sdev^2
  yvar <- switch(type,
                 pev = d / sum(d),
                 cev = cumsum(d) / sum(d))

  yvar.lab <- switch(type,
                    pev = 'proportion of explained variance',
                    cev = 'cumulative proportion of explained variance')

  df <- data.frame(PC = 1:length(d), yvar = yvar)

  ggplot(data = df, aes(x = PC, y = yvar)) +
    xlab('principal component number') + ylab(yvar.lab) +
    geom_point() + geom_path()
}

ggbiplot(bs_base_pr_g,
         ellipse=T,
         choices = c(1,2), #The choices can be changed depending on the component
         var.axes=F,

```

```

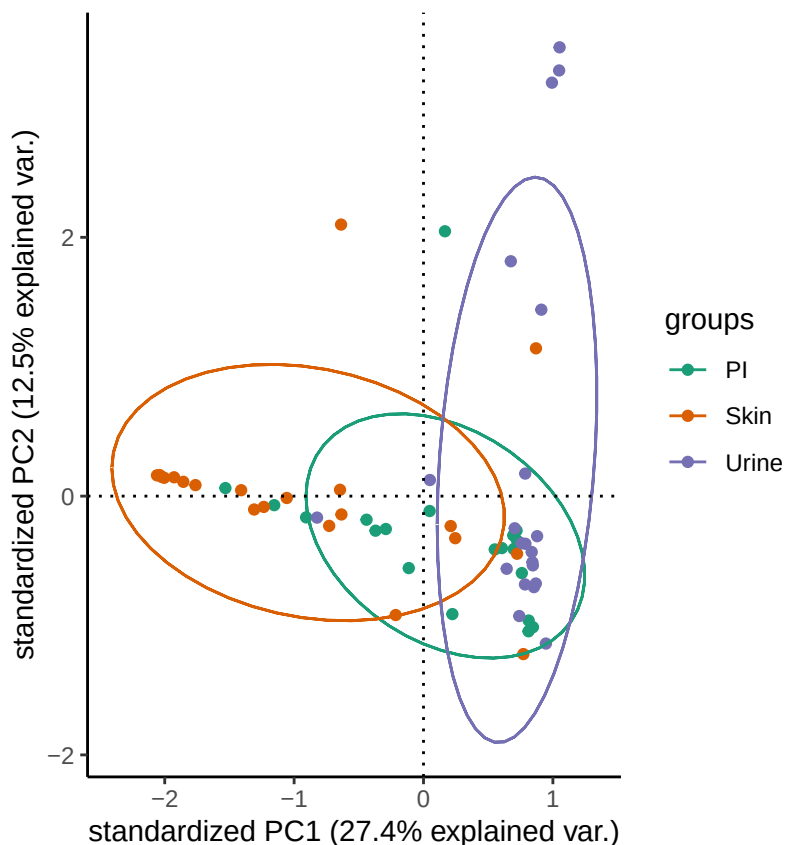
    #labels=rownames(bs_base_clr_g),
    groups=PCA_meta$Sample)+
geom_vline(xintercept = 0,linetype="dotted") +
geom_hline(yintercept = 0,linetype="dotted")+
theme_classic()+
scale_color_brewer(palette="Dark2")

```

```

## -----
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## -----
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
##
## The following object is masked from 'package:purrr':
##
##   compact
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##   discard
##
## The following object is masked from 'package:readr':
##
##   col_factor

```



```
PCA2 <- read_excel("C:/Users/ezrav/Documents/R_Projects/URQVAX/PCA2.xlsx", sheet = "Sheet3")
str(PCA2)
```

```
## tibble [42 x 29] (S3: tbl_df/tbl/data.frame)
## $ Sample      : num [1:42] 1 2 3 4 5 6 7 8 9 10 ...
## $ Donor       : chr [1:42] "Donor 3" "Donor 3" "Donor 4" "Donor 4" ...
## $ Time        : chr [1:42] "Day2" "Day15" "Day2" "Day15" ...
## $ Bacteria/mL : num [1:42] 1.27e+07 1.88e+07 1.82e+08 2.81e+01 1.48e+08 ...
## $ Load1      : chr [1:42] "Above" "Above" "Above" "Below" ...
## $ Load2      : chr [1:42] "Above" "Above" "Above" "Below" ...
## $ Load3      : chr [1:42] "Above" "Above" "Above" "Below" ...
## $ Aerococcus  : num [1:42] 0 0 0 0 0 0 0 0 0 0 ...
## $ Acinetobacter : num [1:42] 0 0 0 0 0 0 0 0 0 0 ...
## $ Anaerococcus : num [1:42] 0 0 0 0 0 0 0 0 0 0 ...
## $ Blautia     : num [1:42] 0 0 0 0 0 0 0 0 0 0 ...
## $ Campylobacter : num [1:42] 0 0 0 0 0 0 0 0 0 0 ...
## $ Citrobacter  : num [1:42] 0 0 0 0 0 0 0 0 0 0 ...
## $ Corynebacterium: num [1:42] 0 0 0 0 0 ...
## $ Cutibacterium : num [1:42] 0 0 0 0 0 ...
## $ Enterobacter : num [1:42] 0 0 0 0 0 0 0 0 0 0 ...
## $ Enterococcus : num [1:42] 0 0 0 0 67295 ...
## $ Escherichia  : num [1:42] 0 0 0 0 0 ...
## $ Finegoldia   : num [1:42] 0 0 0 0 0 0 0 0 0 0 ...
## $ Hafnia       : num [1:42] 0 0 0 0 334422 ...
## $ Klebsiella   : num [1:42] 44780 27841 670964 12 0 ...
## $ Morganella   : num [1:42] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ Proteus      : num [1:42] 3824 16110 0 0 4643 ...
## $ Pseudomonas  : num [1:42] 819 111 0 0 0 ...
## $ Salmonella   : num [1:42] 0 0 0 0 0 0 0 0 0 0 ...
## $ Staphylococcus : num [1:42] 1023 5790 0 3 28723 ...
## $ Streptococcus : num [1:42] 0 0 0 0 532066 ...
## $ Others       : num [1:42] 0 0 0 0 0 133 0 12 0 7 ...
## $ Observed     : num [1:42] 4 4 1 2 5 5 1 3 1 5 ...
```

```
PCA_data <- PCA2 %>% dplyr::select(c(Sample, Aerococcus:Others))
PCA_data[is.na(PCA_data)] <-0
PCA_meta <- PCA2 %>% dplyr::select(c(Sample, Load1:Load3))

bs_base_g <-PCA_data
bs_base_g <- bs_base_g %>%
  column_to_rownames(var="Sample")
bs_base_g<-bs_base_g+1 #We are adding 1 so that there are no zero values during the log transformation
bs_base_clr_g<-clr(bs_base_g, ifclose = FALSE, ifwarn = TRUE)
bs_base_pr_g<- prcomp(bs_base_clr_g)

ggbiplot(bs_base_pr_g,
  ellipse=T,
  choices = c(1,2), #The choices can be changed depending on the component
  var.axes=F,
  #labels=rownames(bs_base_clr_g),
  groups=PCA_meta$Load3)+
  geom_vline(xintercept = 0,linetype="dotted") +
  geom_hline(yintercept = 0,linetype="dotted")+
  theme_classic()+
  scale_color_brewer(palette="Dark2")
```

