PCA and UMAP for Tahoe Rainblow Trout

Tom Parchman

Code below sets chunk width so code wraps and doesn't run off the page

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

PCA_UMAP_RBT.R Following Faske, Jahner, Diaz-Papkovich 2019 Plos Genetics 07/04/2021 Input data is genotypes from Galland's rainbow trout sampled from around lake tahoe

loading required libraries

```
library(data.table)
library(ggplot2)
library(ggsci)
library(umap)
library(LEA)
library(readr)
library(ggpubr)
```

Function for running PCA, written by Trevor Faske

- PCA for 012 coded vcf files
- Following method in Patterson et al 2006

Input files:

`df_gen`: genotypic data with individuals as rows and snps as columns. Can include missing data. Either genotype probabilities or 012 format

Output:

`df_out`:
`$pca_df`: dataframe with rows as individuals and columns as PC1-X, Pop, ID
`$pve`: list of proportion of variance explained for each PC

Function:

```
PCA_gen <- function(df_gen, num = 10, tw = FALSE, tw_pvalue = 0.01) {

    df_gen <- apply(df_gen, 2, function(df) gsub(-1, NA, df,
```

```
        fixed = TRUE))
    df_gen <- apply(df_gen, 2, function(df) as.numeric(df))

    colmean <- apply(df_gen, 2, mean, na.rm = TRUE)

    normalize <- matrix(nrow = nrow(df_gen), ncol = ncol(df_gen))
    af <- colmean/2

    for (m in 1:length(af)) {
        nr <- df_gen[, m] - colmean[m]
        dn <- sqrt(af[m] * (1 - af[m]))
        normalize[, m] <- nr/dn
    }

    normalize[is.na(normalize)] <- 0

    method1 <- prcomp(normalize, scale. = FALSE, center = FALSE)
    pve <- summary(method1)$importance[2, ]
    print(pve[1:5])

    ### adjust number of PC axes ###

    if (nrow(df_gen) < num) {
        num <- nrow(df_gen)
    }

    #### Tracy Widom, PC axes ####
    if (tw == TRUE) {
        cat("\nRunning Tracy Widom test....\n\n")
        write.lfmm(normalize, "temp.lfmm")
        pca_tw <- pca("temp.lfmm", center = FALSE)
        tw <- tracy.widom(pca_tw)
        tw_sign <- tw$pvalues[tw$pvalues <= tw_pvalue]
        cat("\nNumber of TW sig. PC axes: ", length(tw_sign),
            "\n\n")
        num = length(tw_sign)
        unlink("temp.lfmm")
    }

    pca_X <- method1$x[, 1:num]

    pca_X <- as.data.frame(pca_X)

    pca_out <- list(pca_df = pca_X, pve = pve)

    return(pca_out)
}
```

Running PCA on all trout, plotting first by pop ID.

```
#### setwd ####
setwd("~/Desktop/files/rainbow_trout/analyses_plots/UMAP")
```

```
#### read in files #### gen_pop has 3 columns of id,
#### followed by genotypes ####
gen_pop <- fread("RBT_gprob2_pop_region.txt", sep = " ", data.table = F)
g <- gen_pop[, -c(1:3)]


#### Pop_ID_Sum has 2nd and 3rd columns from gen_pop. 1st
#### is sample site, 2nd is lake region (NO, WE, SO) ####
Pop_ID_Sum <- gen_pop[, 2:3]

##### Run PCA ####
pca_out <- PCA_gen(g, tw = TRUE)
```

```
##     PC1     PC2     PC3     PC4     PC5
## 0.02746 0.01611 0.01330 0.01201 0.01174
##
## Running Tracy Widom test....
##
## [1] "******************************"
## [1] " Principal Component Analysis "
## [1] "******************************"
## summary of the options:
##
##         -n (number of individuals)          150
##         -L (number of loci)                 12807
##         -K (number of principal components) 150
##         -x (genotype file)                  /Users/thomasparchman/Desktop/files/rainbow_trout/analys
##         -a (eigenvalue file)                /Users/thomasparchman/Desktop/files/rainbow_trout/analys
##         -e (eigenvector file)               /Users/thomasparchman/Desktop/files/rainbow_trout/analys
##         -d (standard deviation file)        /Users/thomasparchman/Desktop/files/rainbow_trout/analys
##         -p (projection file)                /Users/thomasparchman/Desktop/files/rainbow_trout/analys
##
## [1] "******************"
## [1] " Tracy-Widom tests "
## [1] "******************"
## summary of the options:
##
##         -n (number of eigenvalues)          150
##         -i (input file)                     /Users/thomasparchman/Desktop/files/rainbow_trout/analys
##         -o (output file)                    /Users/thomasparchman/Desktop/files/rainbow_trout/analys
##
## Number of TW sig. PC axes:  11
```

```
pve <- pca_out$pve[1:5]
pve
```

```
##     PC1     PC2     PC3     PC4     PC5
## 0.02746 0.01611 0.01330 0.01201 0.01174
```

```
# PC1 PC2 PC3 PC4 PC5 0.02746 0.01611 0.01330 0.01201
# 0.01174
```

```
ncol(pca_out$pca_df)  # 11, number of tw PC axes
```

```
## [1] 11
```

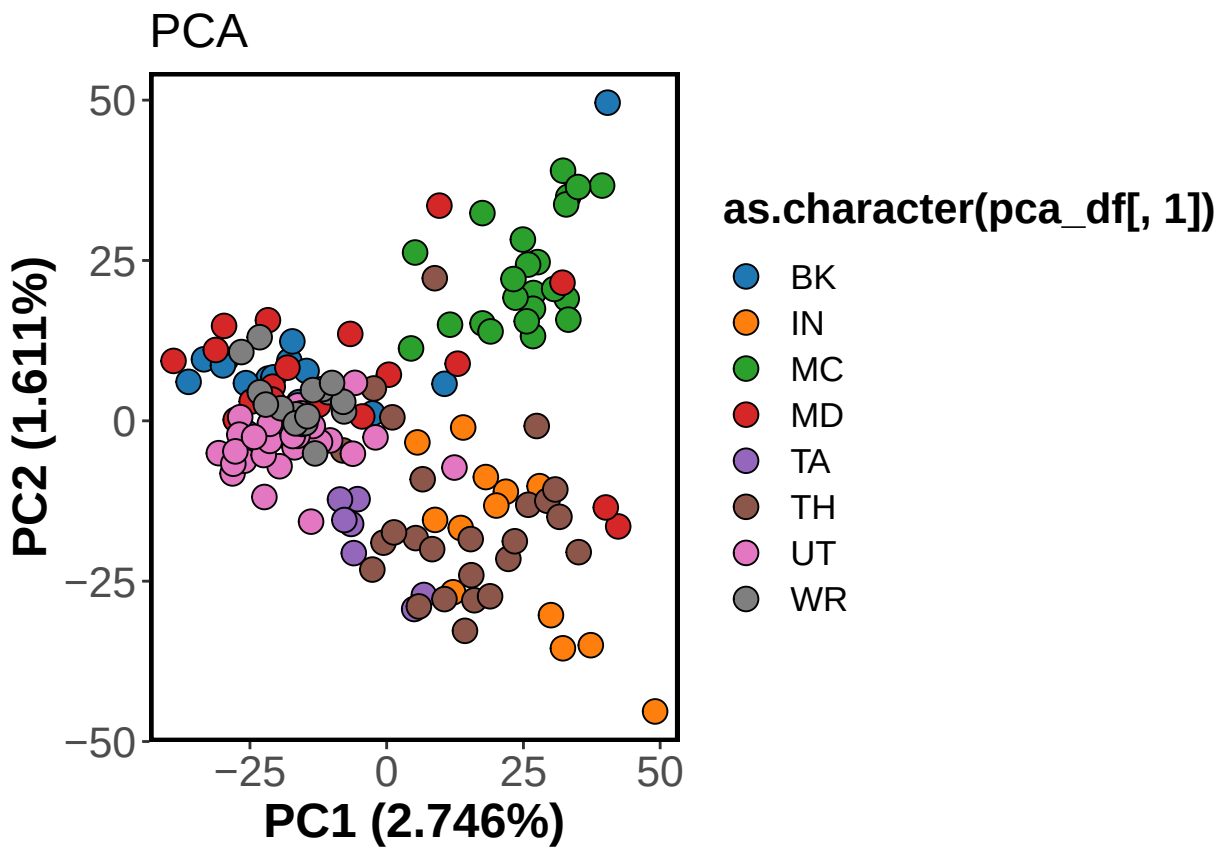
```
pca_df <- pca_out$pca_df
pca_df <- cbind(Pop_ID_Sum, pca_df)
```

Running UMAP below on three different data matrices: 1. genotypes, 2. Number of PC axes based on TW test, and 3. Just the first 10 PCs. Here we are saving umap output into three data frames, plotting code comes below.

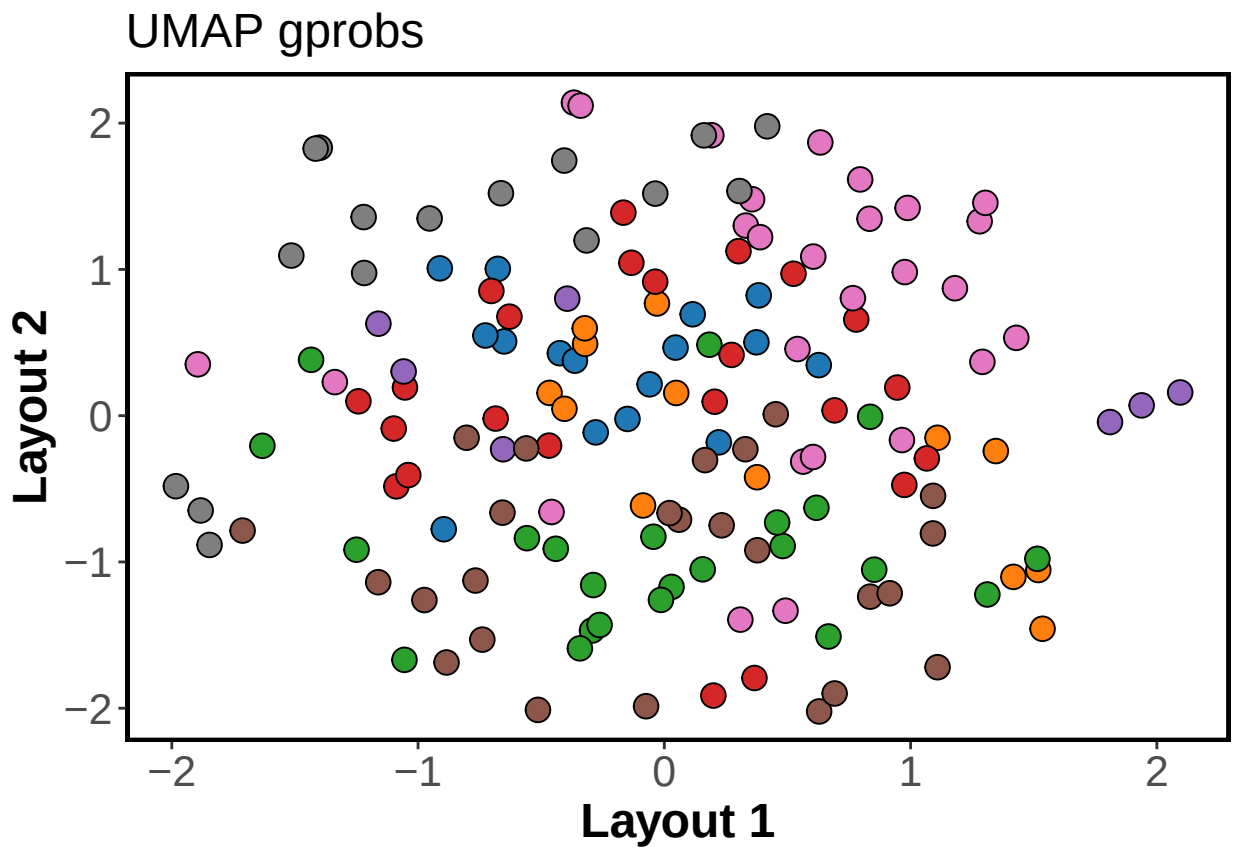
```
umap_g <- as.data.frame(umap(g)$layout)
names(umap_g) <- c("layout1", "layout2")
umap_g <- cbind(Pop_ID_Sum, umap_g)

umap_tw_pcs <- as.data.frame(umap(pca_out$pca_df)$layout)  #number of tw PC axes
names(umap_tw_pcs) <- c("layout1", "layout2")
umap_tw_pcs <- cbind(Pop_ID_Sum, umap_tw_pcs)

umap_ten_pcs <- as.data.frame(umap(pca_out$pca_df[, 1:10])$layout)
names(umap_ten_pcs) <- c("layout1", "layout2")
umap_ten_pcs <- cbind(Pop_ID_Sum, umap_ten_pcs)
```

Plotting PCA and UMAP first by sampling locality

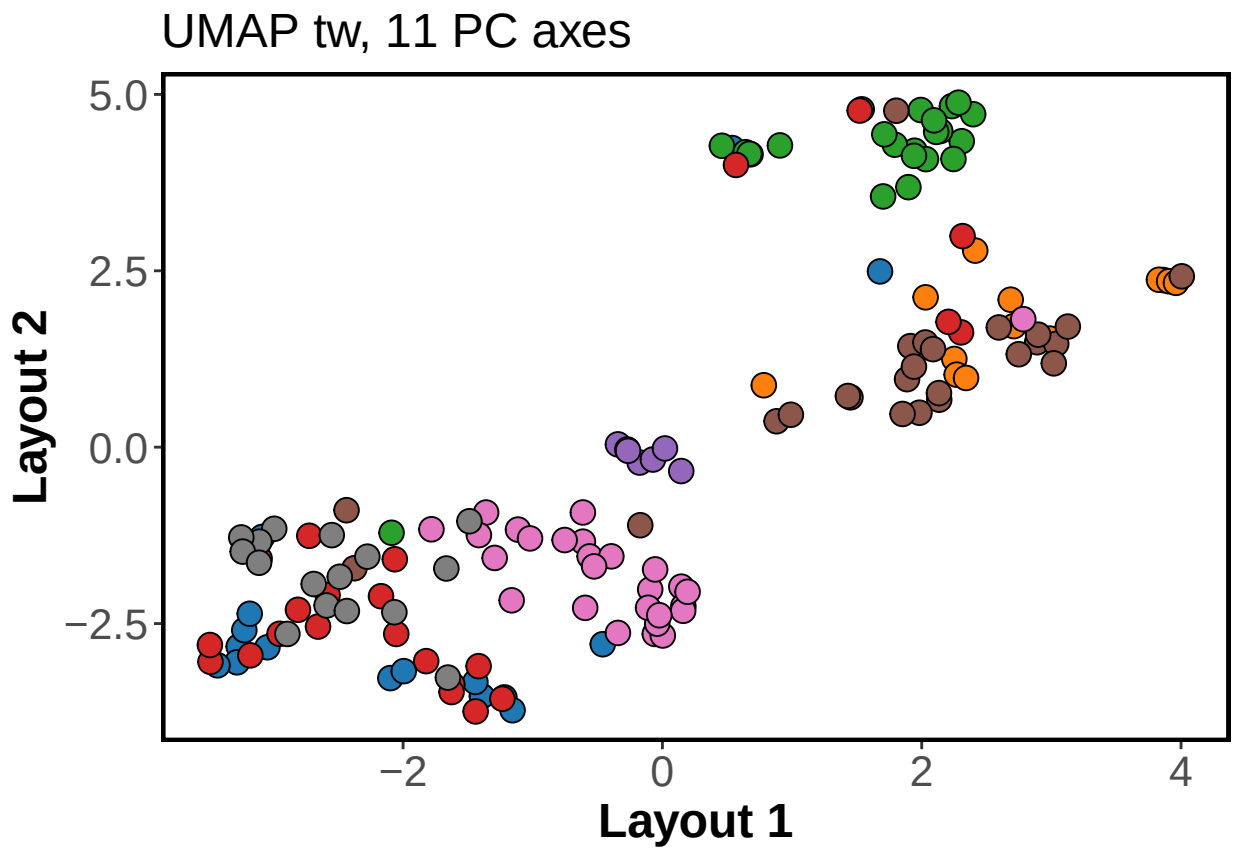
```
pve <- c(0.02746,0.01611,0.01330,0.01201,0.01174)

#col17 <- pal_d3(palette='category20')(20)[c(1:5,7,9:17,19,20)]
PCA_plot <- ggplot(data = pca_df, aes(x=PC1,y=PC2,fill=as.character(pca_df[,1]))) +
  geom_point(colour='black',size = 4,pch=21) + ggtitle("PCA") +
  xlab(paste("PC",1," (",pve[1]*100,"%)",sep="")) + ylab(paste("PC",2," (",pve[2]*100,"%)",sep=""))  +
# scale_fill_manual(values = col17) +
  scale_fill_d3(palette = 'category20') +
  theme_bw() +
theme(#legend.position = 'none', #removes legend
        plot.title = element_text(size = 18, colour="black"),
        axis.text = element_text(size=16),
        axis.title = element_text(size = 18, colour="black",face = "bold"),
        panel.border = element_rect(size = 1.5, colour = "black"),
        legend.title = element_text(size = 16, colour="black",face = "bold",vjust = 1),
        legend.text = element_text(size=13),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
PCA_plot
```



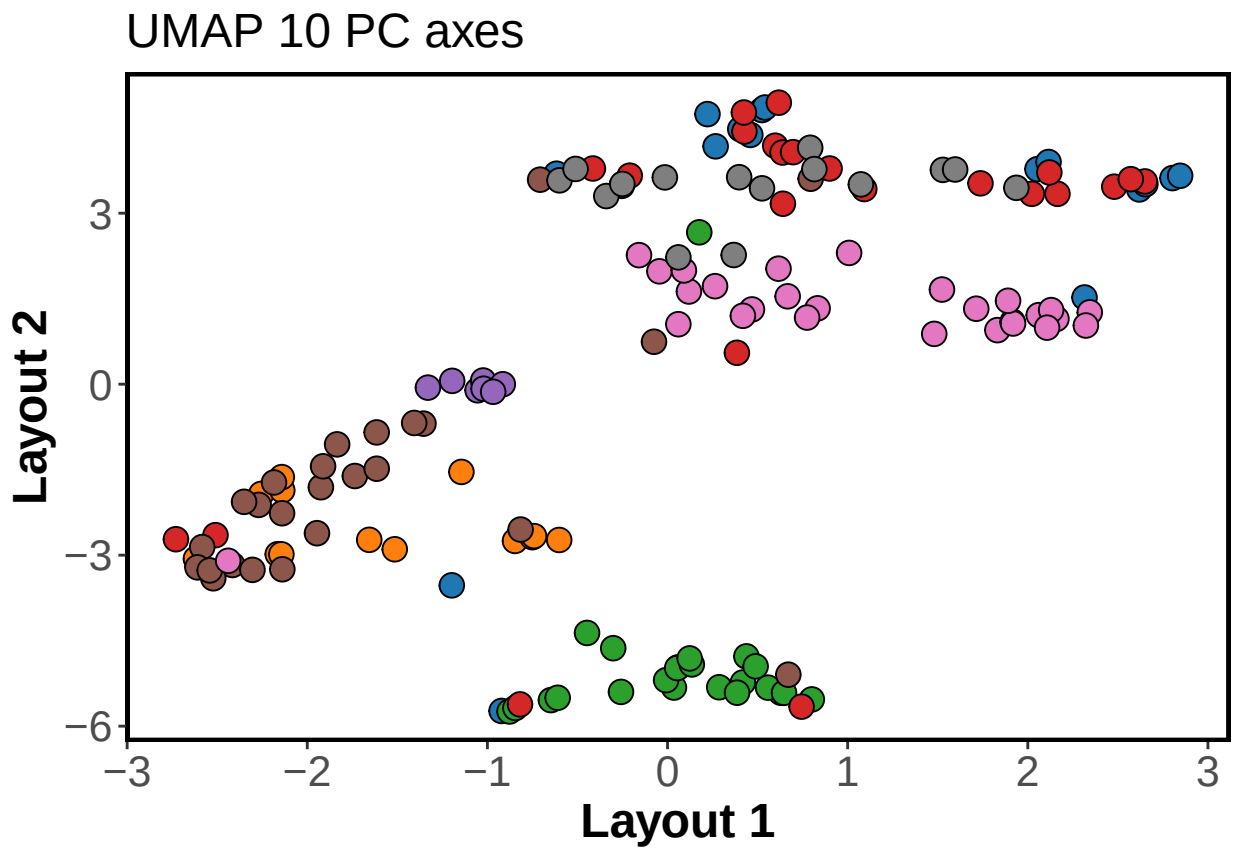

```
#### UMAP gprob ####

#col17 <- pal_d3(palette='category20')(20)[c(1:5,7,9:17,19,20)]
umap_g_plot <- ggplot(data = umap_g, aes(x=layout1,y=layout2,fill=as.character(umap_g[,1]))) +
  geom_point(colour='black',size = 4,pch=21) + ggtitle("UMAP gprobs") +
  xlab('Layout 1') + ylab('Layout 2') +
#scale_fill_manual(values = col17) +
  scale_fill_d3(palette = 'category20') +
  theme_bw() +
  theme(legend.position = 'none', #removes legend
        plot.title = element_text(size = 18, colour="black"),
        axis.text = element_text(size=16),
        axis.title = element_text(size = 18, colour="black",face = "bold"),
        panel.border = element_rect(size = 1.5, colour = "black"),
        legend.title = element_text(size = 16, colour="black",face = "bold",vjust = 1),
        legend.text = element_text(size=13),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
umap_g_plot
```



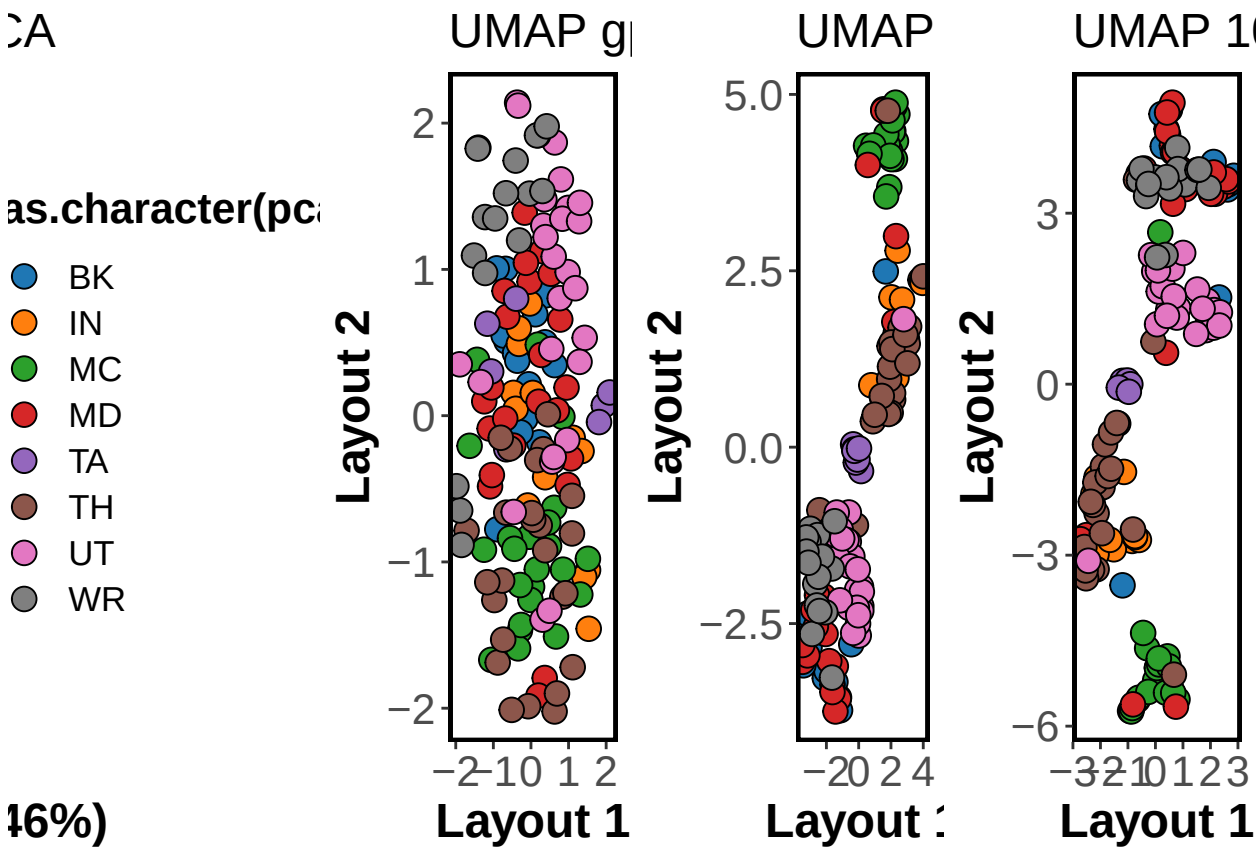

```
#### UMAP tracy widom ####

#col17 <- pal_d3(palette='category20')(20)[c(1:5,7,9:17,19,20)]
umap_tw_plot <- ggplot(data = umap_tw_pcs, aes(x=layout1,y=layout2,fill=as.character(umap_g[,1]))) +
  geom_point(colour='black',size = 4,pch=21) + ggtitle(paste0("UMAP tw, ",ncol(pca_out$pca_df)," PC axes
  xlab('Layout 1') + ylab('Layout 2') +
# scale_fill_manual(values = col17) +
  scale_fill_d3(palette = 'category20') +
  theme_bw() +
  theme(legend.position = 'none', #removes legend
        plot.title = element_text(size = 18, colour="black"),
        axis.text = element_text(size=16),
        axis.title = element_text(size = 18, colour="black",face = "bold"),
        panel.border = element_rect(size = 1.5, colour = "black"),
        legend.title = element_text(size = 16, colour="black",face = "bold",vjust = 1),
        legend.text = element_text(size=13),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
umap_tw_plot
```




```
#### UMAP 10 pcs ####

#col17 <- pal_d3(palette='category20')(20)[c(1:5,7,9:17,19,20)]
umap_ten_plot <- ggplot(data = umap_ten_pcs, aes(x=layout1,y=layout2,fill=as.character(umap_g[,1]))) +
  geom_point(colour='black',size = 4,pch=21) + ggtitle("UMAP 10 PC axes") +
  xlab('Layout 1') + ylab('Layout 2') +
# scale_fill_manual(values = col17) +
  scale_fill_d3(palette = 'category20') +
  theme_bw() +
  theme(legend.position = 'none', #removes legend
        plot.title = element_text(size = 18, colour="black"),
        axis.text = element_text(size=16),
        axis.title = element_text(size = 18, colour="black",face = "bold"),
        panel.border = element_rect(size = 1.5, colour = "black"),
        legend.title = element_text(size = 16, colour="black",face = "bold",vjust = 1),
        legend.text = element_text(size=13),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
umap_ten_plot
```



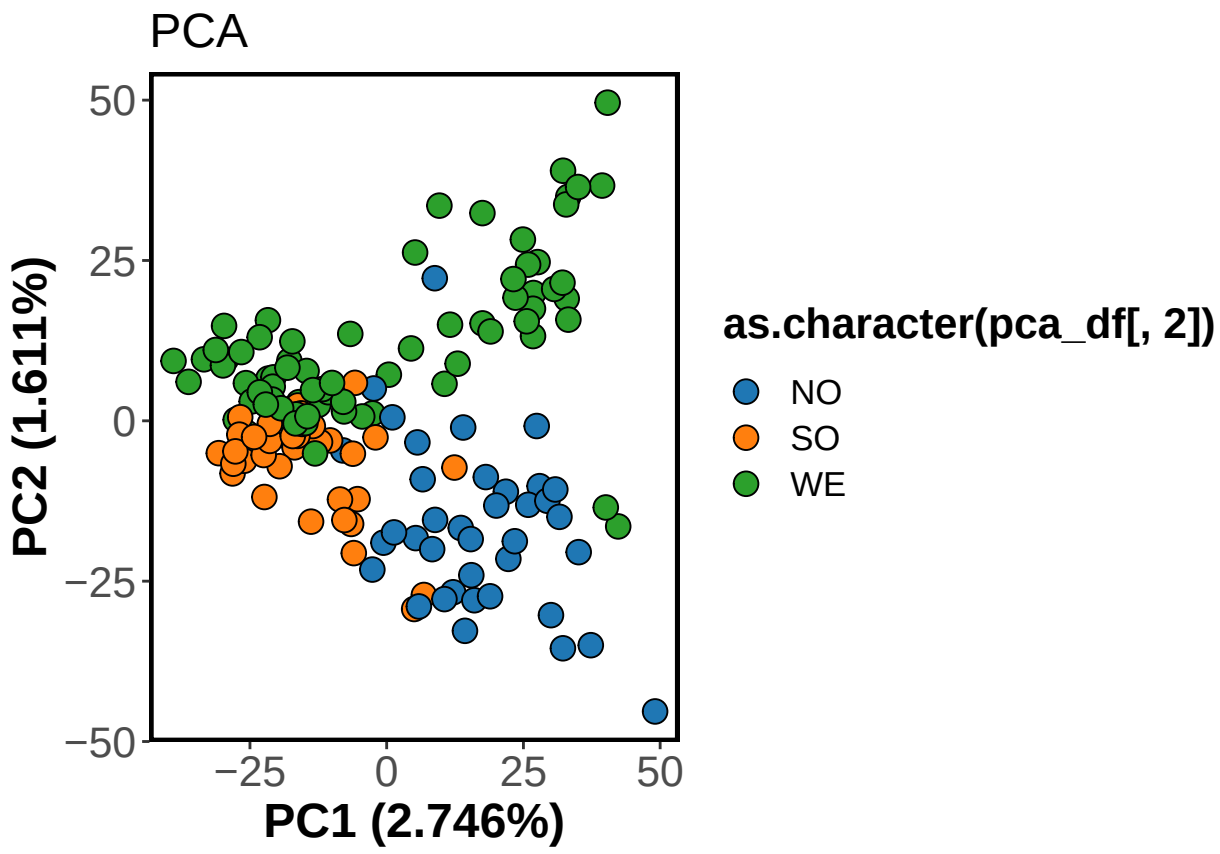

```
### Combine plots for PCA and umap colored by sampling locality

all_plots <- ggarrange(PCA_plot,umap_g_plot, umap_tw_plot,umap_ten_plot,ncol=4)
all_plots
```




```
ggsave('RBT_PCA_UMAP_sample_locale.pdf',all_plots,height=5,width = 20,units = 'in')
```

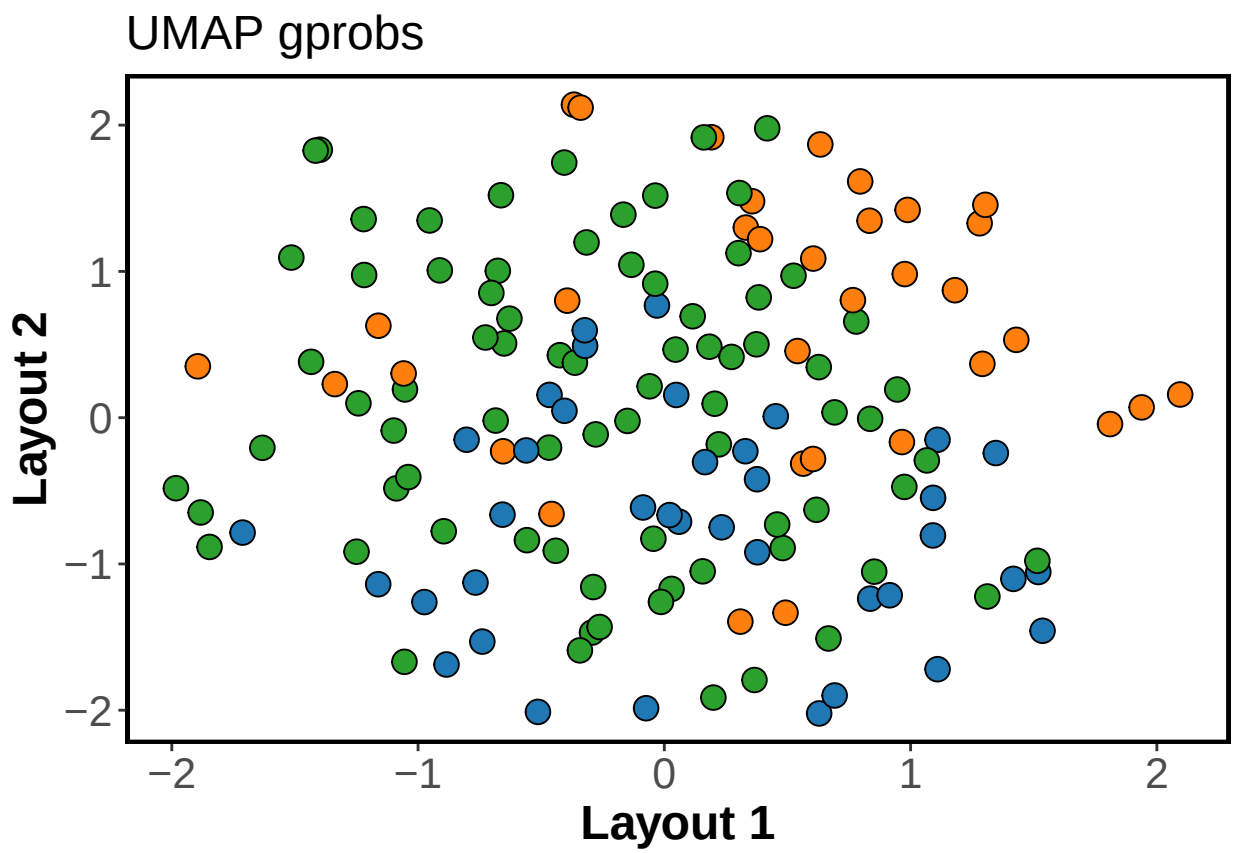
Plotting PCA and UMAP first by sampling locality

```
pve <- c(0.02746,0.01611,0.01330,0.01201,0.01174)

#col17 <- pal_d3(palette='category20')(20)[c(1:5,7,9:17,19,20)]
PCA_plot <- ggplot(data = pca_df, aes(x=PC1,y=PC2,fill=as.character(pca_df[,2]))) +
  geom_point(colour='black',size = 4,pch=21) + ggtitle("PCA") +
  xlab(paste("PC",1," (",pve[1]*100,"%)",sep="")) + ylab(paste("PC",2," (",pve[2]*100,"%)",sep=""))  +
# scale_fill_manual(values = col17) +
  scale_fill_d3(palette = 'category20') +
  theme_bw() +
theme(#legend.position = 'none', #removes legend
        plot.title = element_text(size = 18, colour="black"),
        axis.text = element_text(size=16),
        axis.title = element_text(size = 18, colour="black",face = "bold"),
        panel.border = element_rect(size = 1.5, colour = "black"),
        legend.title = element_text(size = 16, colour="black",face = "bold",vjust = 1),
        legend.text = element_text(size=13),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
PCA_plot
```

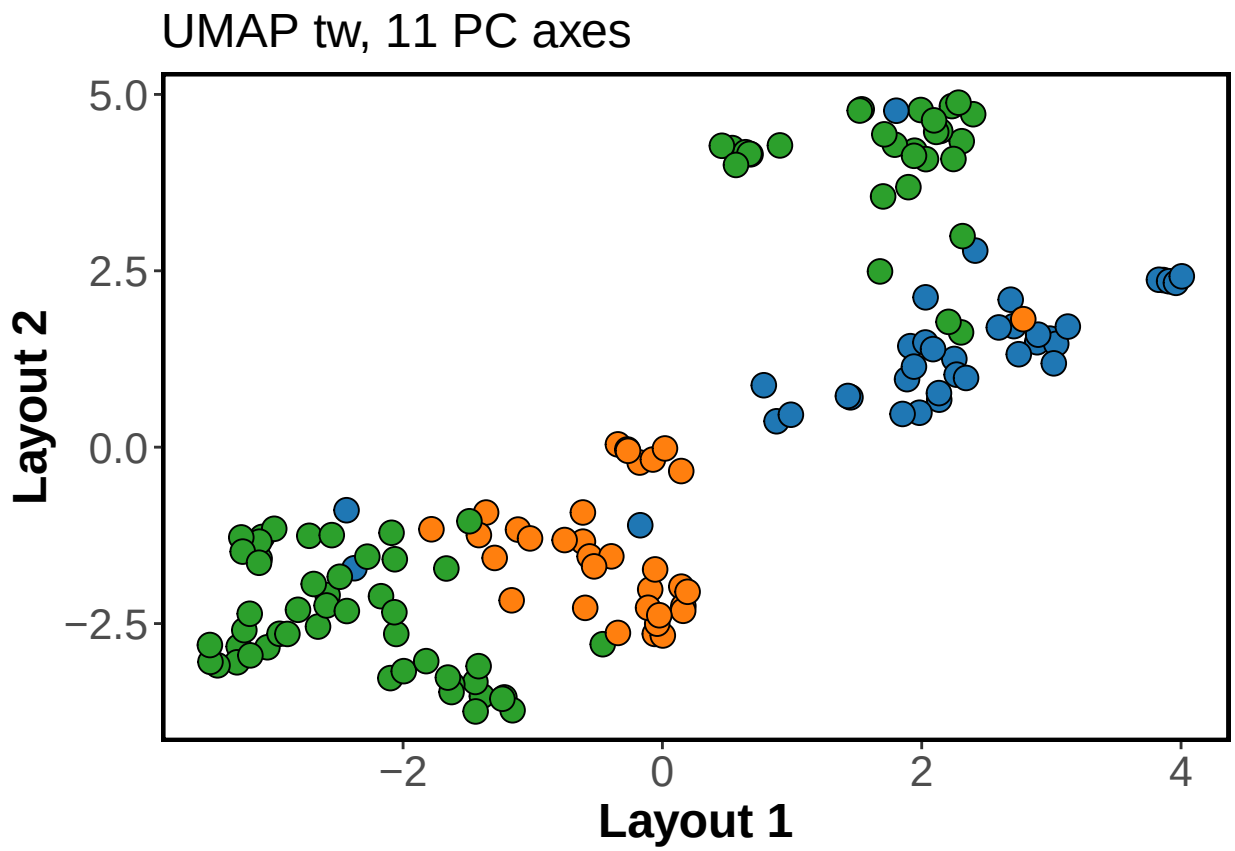


plotting UMAP by lake region

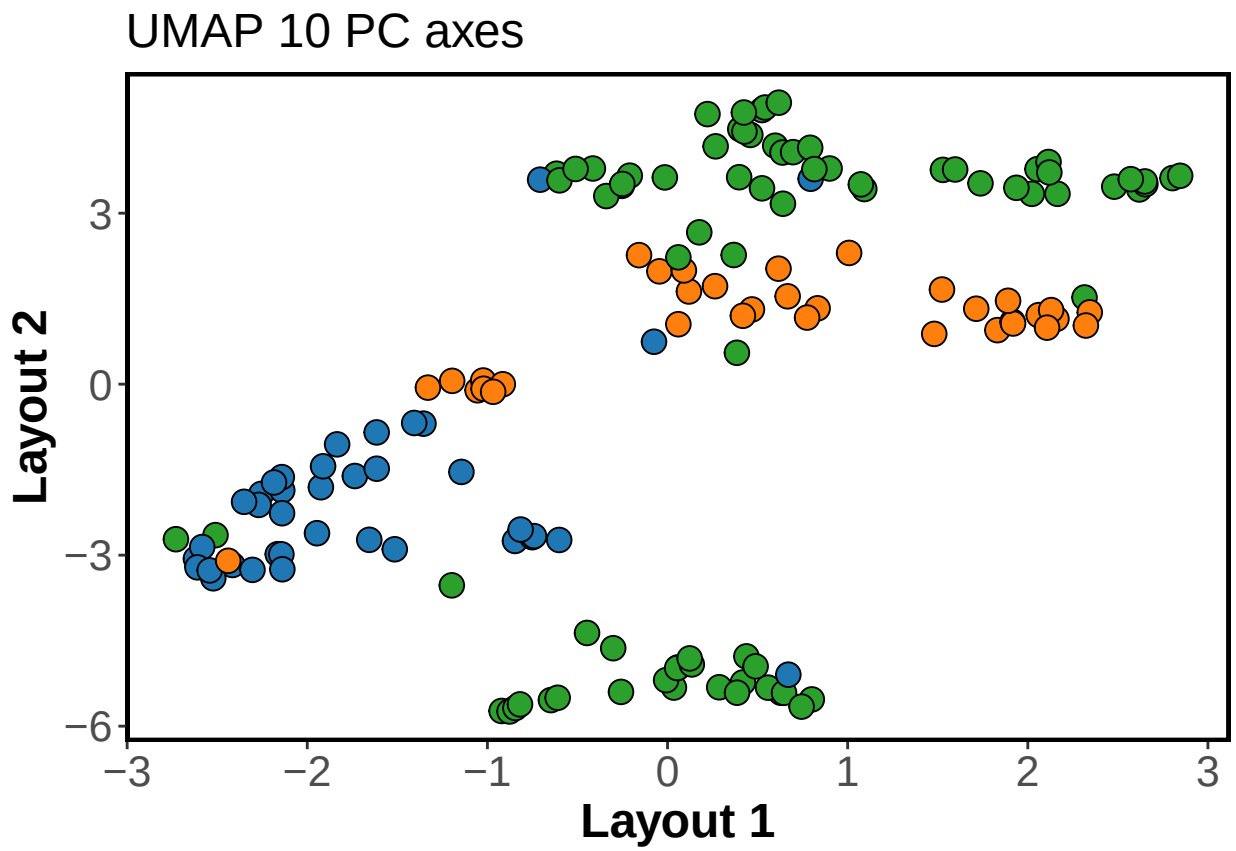
```
#### UMAP gprob ####

#col17 <- pal_d3(palette='category20')(20)[c(1:5,7,9:17,19,20)]
umap_g_plot <- ggplot(data = umap_g, aes(x=layout1,y=layout2,fill=as.character(umap_g[,2]))) +
  geom_point(colour='black',size = 4,pch=21) + ggtitle("UMAP gprobs") +
  xlab('Layout 1') + ylab('Layout 2') +
#scale_fill_manual(values = col17) +
  scale_fill_d3(palette = 'category20') +
  theme_bw() +
  theme(legend.position = 'none', #removes legend
        plot.title = element_text(size = 18, colour="black"),
        axis.text = element_text(size=16),
        axis.title = element_text(size = 18, colour="black",face = "bold"),
        panel.border = element_rect(size = 1.5, colour = "black"),
        legend.title = element_text(size = 16, colour="black",face = "bold",vjust = 1),
        legend.text = element_text(size=13),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
umap_g_plot
```



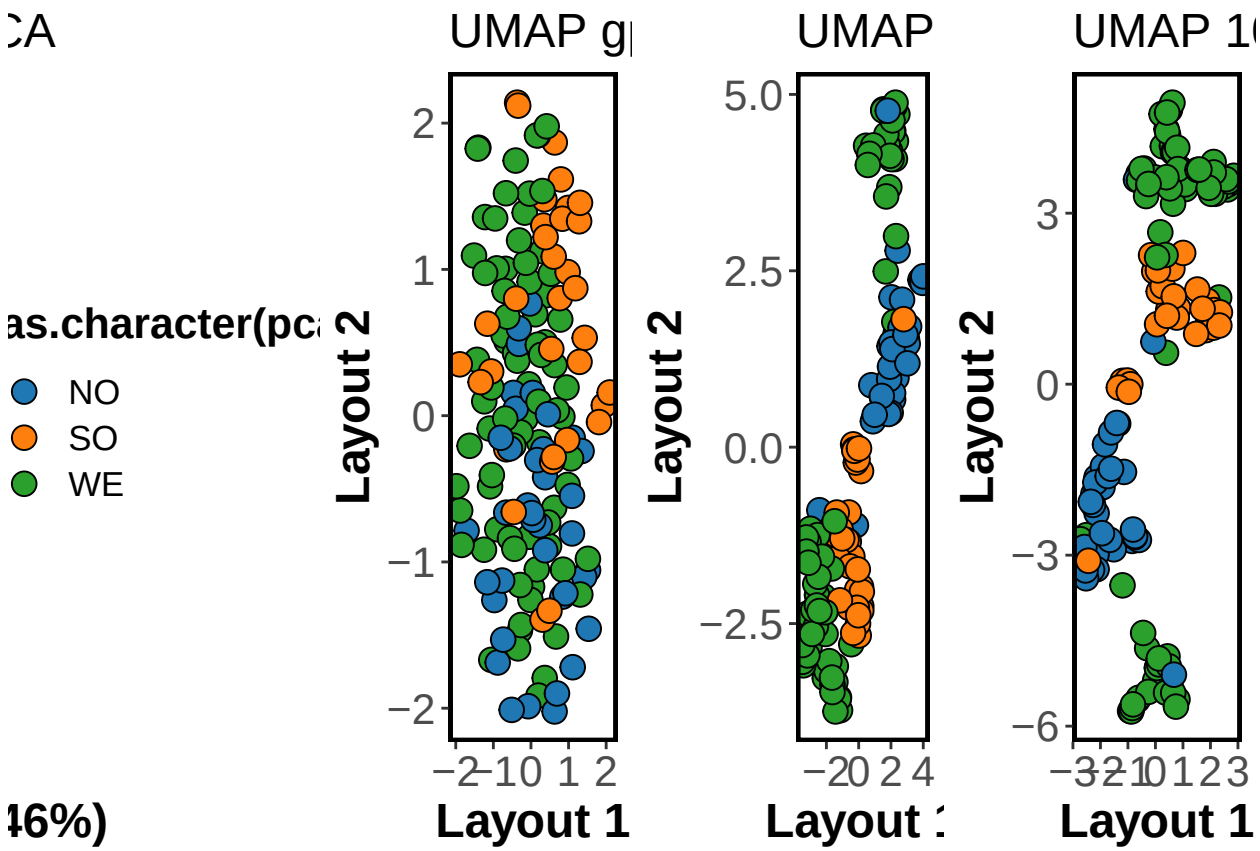

```
#### UMAP tracy widom ####

#col17 <- pal_d3(palette='category20')(20)[c(1:5,7,9:17,19,20)]
umap_tw_plot <- ggplot(data = umap_tw_pcs, aes(x=layout1,y=layout2,fill=as.character(umap_g[,2]))) +
  geom_point(colour='black',size = 4,pch=21) + ggtitle(paste0("UMAP tw, ",ncol(pca_out$pca_df)," PC axes
  xlab('Layout 1') + ylab('Layout 2') +
# scale_fill_manual(values = col17) +
  scale_fill_d3(palette = 'category20') +
  theme_bw() +
  theme(legend.position = 'none', #removes legend
        plot.title = element_text(size = 18, colour="black"),
        axis.text = element_text(size=16),
        axis.title = element_text(size = 18, colour="black",face = "bold"),
        panel.border = element_rect(size = 1.5, colour = "black"),
        legend.title = element_text(size = 16, colour="black",face = "bold",vjust = 1),
        legend.text = element_text(size=13),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
umap_tw_plot
```




```
#### UMAP 10 pcs ####

#col17 <- pal_d3(palette='category20')(20)[c(1:5,7,9:17,19,20)]
umap_ten_plot <- ggplot(data = umap_ten_pcs, aes(x=layout1,y=layout2,fill=as.character(umap_g[,2]))) +
  geom_point(colour='black',size = 4,pch=21) + ggtitle("UMAP 10 PC axes") +
  xlab('Layout 1') + ylab('Layout 2') +
# scale_fill_manual(values = col17) +
  scale_fill_d3(palette = 'category20') +
  theme_bw() +
  theme(legend.position = 'none', #removes legend
        plot.title = element_text(size = 18, colour="black"),
        axis.text = element_text(size=16),
        axis.title = element_text(size = 18, colour="black",face = "bold"),
        panel.border = element_rect(size = 1.5, colour = "black"),
        legend.title = element_text(size = 16, colour="black",face = "bold",vjust = 1),
        legend.text = element_text(size=13),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
umap_ten_plot
```




```
### Combine plots for PCA and umap colored by sampling locality

all_plots <- ggarrange(PCA_plot,umap_g_plot, umap_tw_plot,umap_ten_plot,ncol=4)
all_plots
```




```
ggsave('RBT_PCA_UMAP_lake_region.pdf',all_plots,height=5,width = 20,units = 'in')
```