Data for:

Nicholas S. Marzolf

Updated March 13, 2023

Introduction

RMarkdown file to accompany [].

To recreate this HTML, load the data files into a folder named ‘Data’ at the same path as the .rmd file, open the .rmd file, and click ‘Knit’.

Load packages

```
# data manipulation
library(tidyverse)
library(readxl)
library(dplyr)
library(forcats)
library(purrr)

# plotting
library(ggplot2)
library(ggeffects)
library(grid)
library(ggpubr)
library(ggrepel)
library(lemon)
source('C:/Users/Nick Marzolf/Desktop/Research/R code/theme_nick.R')
theme_set(theme_nick())

# statistics
library(car)
library(drc)
library(nlme)
library(nlstools)
library(qpcR)

# spatial
library(sf)
library(raster)
library(spData)
library(tmap)
library(leaflet)
```

```r
library(spDataLarge)

# community
library(vegan)
library(pander)
library(lattice)
library(permute)
```

Load data

```r
chem <- readxl::read_excel('Data/LT_CWD_datasheet.xlsx',
                           sheet = 'Chemistry')

cwd <- readxl::read_excel('Data/LT_CWD_datasheet.xlsx',
                          sheet = 'Sheet1')
```

Stream chemistry for 5 sites

```r
chem <- chem %>%
  dplyr::rename(site = 'Site',
                srp = `SRP (ug/L)`,
                no3_n = `NO3-N (ug/L)`,
                nh4_n = `NH4-N (ug/L)`)

# calculate mean of all measurements
chem_sum <- chem %>%
  dplyr::group_by(site) %>%
  dplyr::summarise(dplyr::across(.cols = 2:7,
                                 .fns = mean, na.rm = TRUE))

# calculate SD from stream chemistry data
chem_sd <- chem %>%
  dplyr::group_by(site) %>%
  dplyr::summarise(dplyr::across(.cols = 2:7,
                                 .fns = sd, na.rm = TRUE))

# create object that sorts sites by decreasing mean conductivity
sites <- chem_sum %>%
  dplyr::group_by(site) %>%
  dplyr::summarise(mean_cond = mean(Cond, na.rm = TRUE)) %>%
  dplyr::arrange(desc(mean_cond))

# create a character vector of sites in order
sites <- as.character(sites$site)

# create a vector for longer names used in plotting
sites_long <- c(`Arb` = 'Arboleda 30',
                `Sur30` = 'Sura 30',
                `Tito60` = 'Saltito 60',
                `Piper` = 'Piper',
                `Tac` = 'Taconazo 30')


# fill in NA from Arb and Tac with long-term means
# the monthly Arb and Tac data from those sites isn't available yet
chem_sum[1,2] = 201.8
chem_sum[1,3] = 200.0
chem_sum[1,4] = 31.3

chem_sum[4,2] = 3.92
chem_sum[4,3] = 175.4
chem_sum[4,4] = 37.9

# calculate DIN and N:P ratio
chem_sum <- chem_sum %>%
  mutate(din = no3_n + nh4_n,
         n_p = (din/14.0067)/(srp/30.973762))


# re-level the site factor
chem_sum$site <- forcats::fct_relevel(chem_sum$site,
                                      sites)

table1 <- dplyr::arrange(chem_sum, desc(Cond))
table1
```

```
## # A tibble: 5 x 9
##   site      srp no3_n nh4_n    pH  Cond  Temp   din    n_p
##   <fct>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 Arb    202.    200   31.3  6.22 260.   24.8  231.   2.53
## 2 Sur30   80.8    259   75.5  6.53 146.   24.8  334.   9.16
## 3 Tito60  32.1    211.  46.2  6.25  94.3  24.2  257.  17.7
## 4 Piper    7.37   292   113.  5.75  23.6  25.3  405. 122.
## 5 Tac      3.92   175.  37.9  5.73  18.4  24.8  213. 120.
```

Figure 1: Map

```
streams_gps <- readr::read_csv('C:/Users/Nick Marzolf/Desktop/NCSU/STREAMS/La Selva GIS data/LTREB Data
```

```
## Rows: 17 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Site
## dbl (3): Lat, Long, Elevation
##
## 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.
```

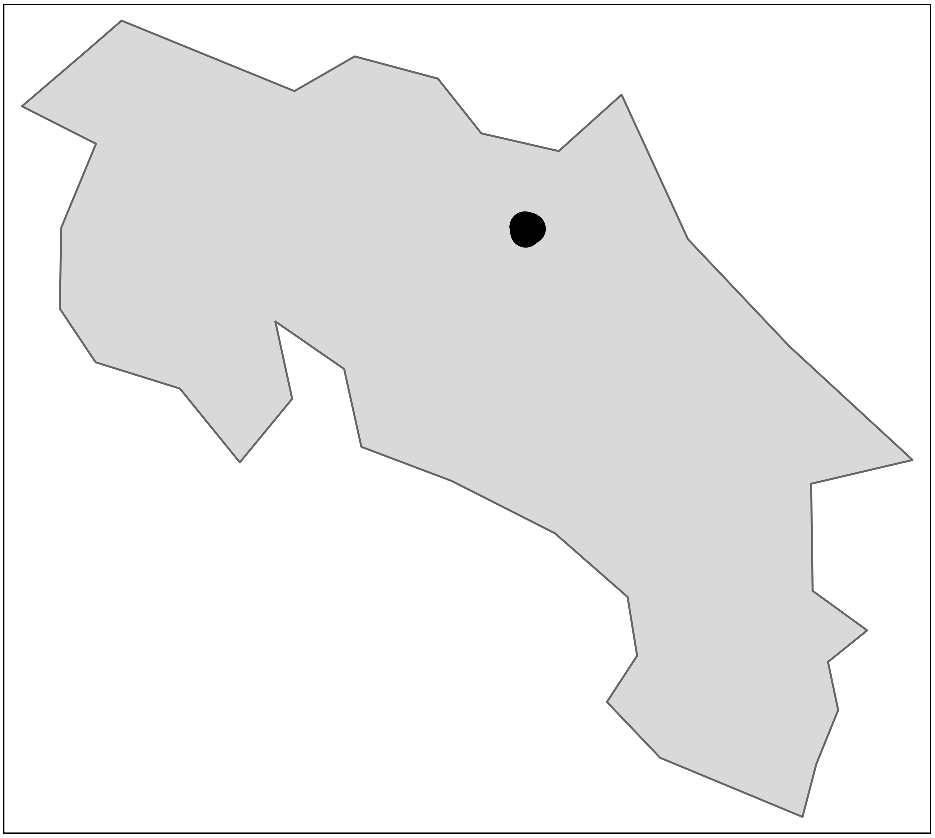
```r
cwd_coords <- streams_gps %>%
  dplyr::filter(Site %in% sites_long) %>%
  sf::st_as_sf(.,
               coords = c('Long', 'Lat'),
               crs = st_crs(4326))

cwd_coords$Site <- forcats::fct_relevel(cwd_coords$Site,
                                        c('Arboleda 30', 'Sura 30', 'Saltito 60',
                                          'Piper', 'Taconazo 30'))

# La Selva boundary
lsbs <- sf::st_read(dsn = 'Data/Spatial/laselvaboundary.shp')
```

```
## Reading layer ‘laselvaboundary’ from data source
##   ‘C:\Users\Nick Marzolf\Desktop\NCSU\STREAMS\Projects\Long-term Wood decomp\LTCWD\Data\Spatial\lasel
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 4 features and 8 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 823381.8 ymin: 1151045 xmax: 828917.3 ymax: 1156986
## Projected CRS: WGS 84 / UTM zone 16N
```

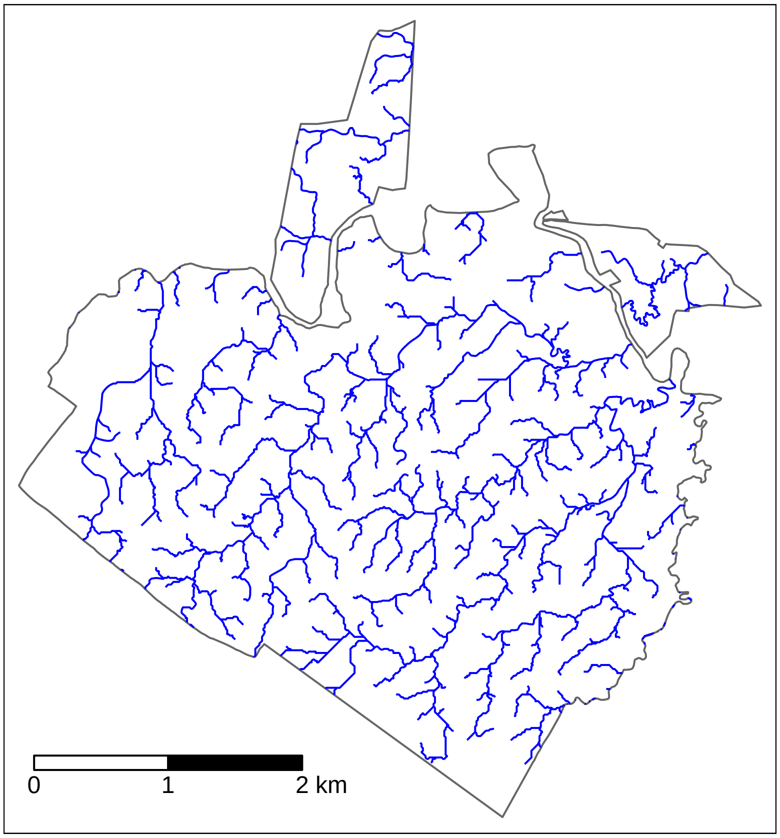
```r
# Streams at La Selva shapefile
streams <- sf::st_read(dsn = 'Data/Spatial/streamsclip.shp')
```

```
## Reading layer ‘streamsclip’ from data source
##   ‘C:\Users\Nick Marzolf\Desktop\NCSU\STREAMS\Projects\Long-term Wood decomp\LTCWD\Data\Spatial\strea
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 521 features and 10 fields
## Geometry type: MULTILINESTRING
## Dimension:     XY
## Bounding box:  xmin: 823758 ymin: 1151440 xmax: 828668.2 ymax: 1156895
## Projected CRS: WGS 84 / UTM zone 16N
```

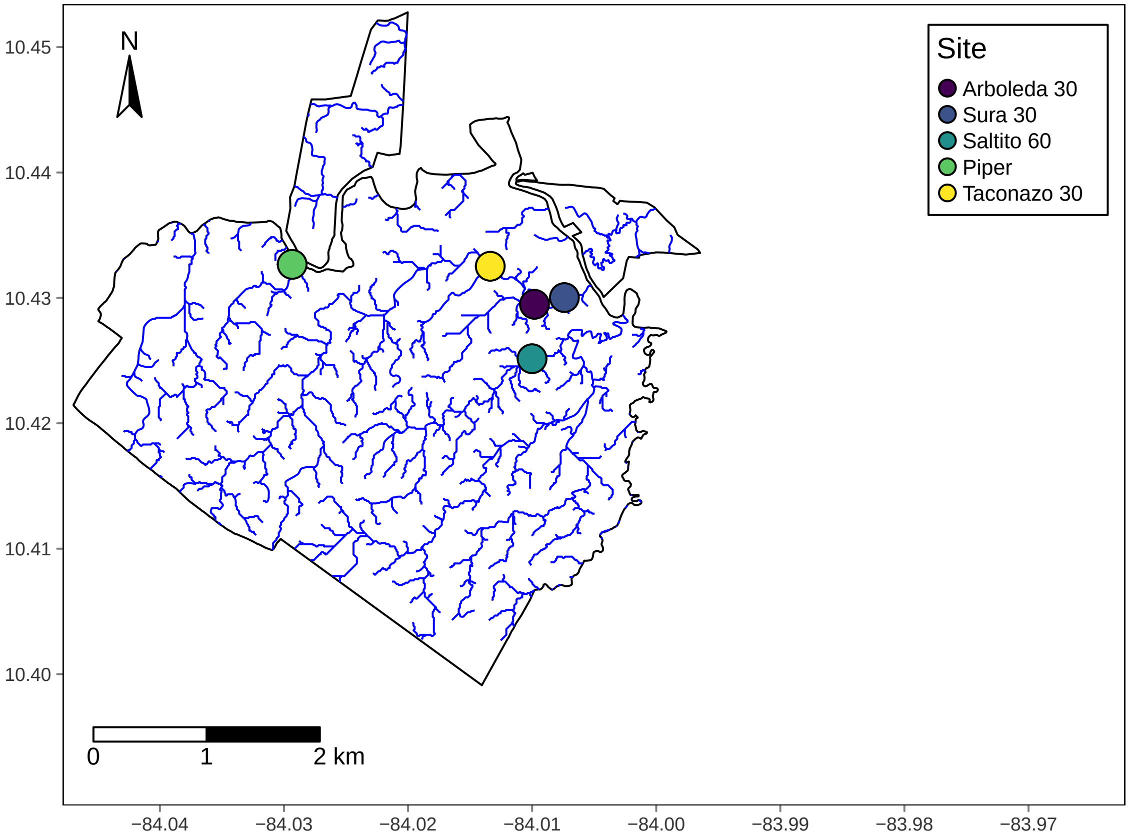
```r
cr <- world %>%
  dplyr::filter(name_long == 'Costa Rica')

# make a basic map of Costa Rica
cr_map <- tmap::tm_shape(cr)+   # create shape based on Costa Rica object
  tmap::tm_polygons()+          # add cr as a polygon
  tmap::tm_shape(lsbs)+         # create shape for the boundary of La Selva
  tmap::tm_dots(size = 1)       # add lsbs as a dot
cr_map
```



```
# La Selva boundary and stream layer ----

# Create a map of La Selva boundary and the stream network
map_lsbs <- tmap::tm_shape(lsbs)+            # new shape: La Selva boundary
  tmap::tm_borders()+                        # add as a border/line layer
  tmap::tm_shape(streams)+                   # new shape: stream network
  tmap::tm_lines(col = 'blue')+              # add as a line, colored blue
  tmap::tm_scale_bar(breaks = c(0, 1, 2),    # add a scale bar, with demarkations for 0, 1, and 2 km
                     text.size = 0.75,       # change text size
                     position = c('left',    # put the scale bar in the bottom left
                                  'bottom'))
map_lsbs
```



```r
fig1 <- tmap::tm_shape(lsbs)+                   # create La Selva boundary layer
  tmap::tm_borders(col = 'black')+
  tmap::tm_shape(streams)+                              # create stream network layer
  tmap::tm_lines(col = 'blue')+
  tmap::tm_shape(cwd_coords)+                             # map the locations of pH sites
  tmap::tm_symbols(size = 1,                            # change the size
                   col = 'Site',                        # colored by site name
                   border.col = 'black',                # with black boundary color
                   palette = "viridis", n = 5)+         # change the color palette
  tmap::tm_scale_bar(breaks = c(0, 1, 2),               # add scale bar
                     text.size = 0.75,
                     position = c('left', 'bottom'))+
  tmap::tm_layout(inner.margins = c(.15,.01, .01, .4),  # change the margins to fit the legend and inse
                  legend.position = c('right', 'top'),
                  legend.frame = TRUE)+
  tmap::tm_compass(position = c('left', 'top'))+        # add compass north star
  tmap::tm_grid(projection = 4326,
                labels.inside.frame = FALSE,
                lines = FALSE)
fig1
```



Calculate wood decomposition rates

```
# begin cleaning data
cwd <- cwd %>%
  dplyr::filter(Flag == 0) %>%
  dplyr::select(site = Site,
                month = `Collection Month`,
                rep = Rep,
                init_mass = `initial CWD mass (g)`,
                dry_mass = `CWD Pack Dry Mass (g)`,
                init_den = `init wood density (g/cm3)`,
                fin_den = `final wood density (g/cm3)`) %>%
  dplyr::filter(month < 24)

# re-level site factor based on conductivity
cwd$site <- forcats::fct_relevel(cwd$site,
                                 sites)

# calculate % dry mass remaining
cwd_calc <- cwd %>%
  dplyr::filter(site != 'Sac') %>%
  dplyr::mutate(percent_mass = (dry_mass/init_mass)*100)
```

```
# dry mass of each pack to be used in the bug analysis
final_dry_mass <- cwd_calc %>%
  dplyr::select(site, month, rep, dry_mass) %>%
  dplyr::filter(month != 0)

# calculate decay rates using ANCOVA
k_cwd_int <- lm(data = cwd_calc,
                log(percent_mass) ~ month * site)

summary(k_cwd_int)
```

```
##
## Call:
## lm(formula = log(percent_mass) ~ month * site, data = cwd_calc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41381 -0.13079  0.01234  0.12742  0.39291
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.45289    0.07116  62.577  < 2e-16 ***
## month             -0.06407    0.01154  -5.551 7.41e-07 ***
## siteSur30          0.07091    0.10063   0.705  0.48383
## siteTito60        -0.02906    0.10098  -0.288  0.77457
## sitePiper         -0.17348    0.10098  -1.718  0.09112 .
## siteTac            0.12792    0.10937   1.170  0.24695
## month:siteSur30    0.04731    0.01632   2.898  0.00529 **
## month:siteTito60   0.04650    0.01637   2.841  0.00619 **
## month:sitePiper    0.04914    0.01637   3.002  0.00395 **
## month:siteTac      0.04704    0.01683   2.795  0.00703 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.193 on 58 degrees of freedom
## Multiple R-squared:  0.539,  Adjusted R-squared:  0.4675
## F-statistic: 7.536 on 9 and 58 DF,  p-value: 3.288e-07
```

```
anova(k_cwd_int)
```

```
## Analysis of Variance Table
##
## Response: log(percent_mass)
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## month       1 0.91680 0.91680 24.6079 6.478e-06 ***
## site        4 1.10744 0.27686  7.4312 6.658e-05 ***
## month:site  4 0.50252 0.12563  3.3721   0.01505 *
## Residuals  58 2.16087 0.03726
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```r
car::Anova(k_cwd_int, type = 'III')
```

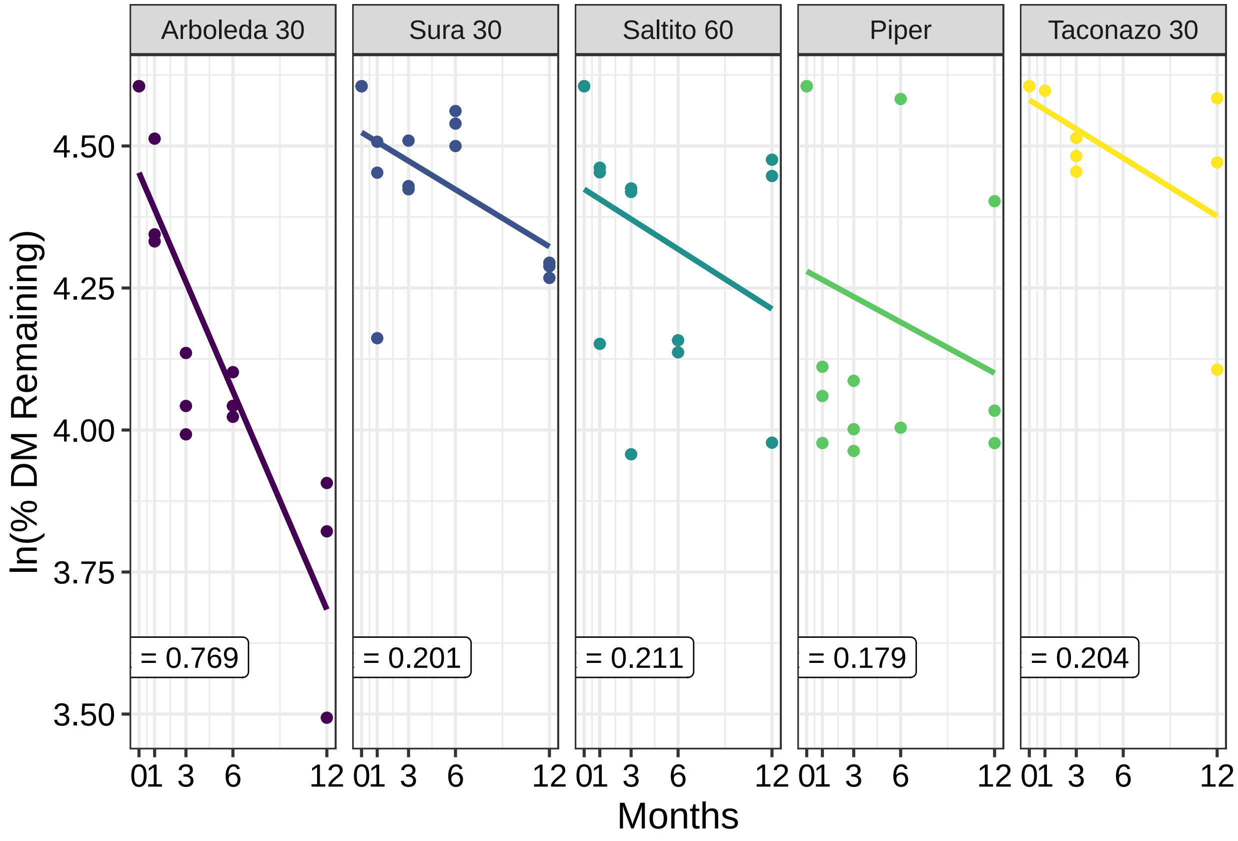
```
## Anova Table (Type III tests)
##
## Response: log(percent_mass)
##              Sum Sq Df   F value    Pr(>F)
## (Intercept) 145.894  1 3915.9413 < 2.2e-16 ***
## month         1.148  1   30.8088 7.408e-07 ***
## site          0.348  4    2.3354   0.06609 .
## month:site    0.503  4    3.3721   0.01505 *
## Residuals     2.161 58
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```r
table2 <- cwd_calc %>%
  dplyr::filter(site != 'Sac') %>%
  dplyr::group_by(site) %>%
  dplyr::summarise(
    int = round((coef(lm(log(percent_mass) ~ month))[1]), 3),
    k_yr = (round((coef(lm(log(percent_mass) ~ month))[2])*12, 3))*-1,
    error = summary(lm(log(percent_mass) ~ month))$coefficient[3],
    #df = summary(lm(log(percent_mass) ~ month))$fstatistic,
    r2 = summary(lm(log(percent_mass) ~ month))$r.squared,
    p = anova(lm(log(percent_mass) ~ month))$'Pr(>F'[1])
table2
```

```
## # A tibble: 5 x 6
##   site     int  k_yr  error     r2         p
##   <fct>  <dbl> <dbl>  <dbl>  <dbl>     <dbl>
## 1 Arb     4.45 0.769 0.0591 0.775  0.0000151
## 2 Sur30   4.52 0.201 0.0447 0.291  0.0379
## 3 Tito60  4.42 0.211 0.0818 0.128  0.209
## 4 Piper   4.28 0.179 0.103  0.0632 0.386
## 5 Tac     4.58 0.204 0.0561 0.346  0.0735
```

```r
fig2 <- ggplot(data = cwd_calc,
               aes(y = log(percent_mass),
                   x = month))+
  geom_point(aes(color = site))+
  geom_smooth(aes(color = site),
              method = 'lm',se = FALSE)+
  geom_label(data = table2 %>%
               dplyr::select(site, k_yr),
             aes(x = 2.5, y = 3.6,
                 label = paste0('k = ',k_yr)))+
  facet_grid(. ~ site,
             labeller = as_labeller(sites_long))+
  labs(x = 'Months',
       y = 'ln(% DM Remaining)')+
  scale_color_viridis_d()+
  scale_x_continuous(breaks = c(0, 1, 3, 6, 12))+
  theme(legend.position = 'none')
fig2
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



Merge chemistry with decay rates

```
merged <- table2 %>%
  dplyr::select(site, k_yr, error) %>%
  dplyr::right_join(chem_sum, 'site') %>%
  dplyr::select(-Temp) %>%
  tidyr::pivot_longer(srp:n_p)

names_long <- c(`Cond` = 'Conductivity (µS/cm)',
                `din` = 'DIN (µg/L)',
                `n_p` = 'DIN:SRP',
                `nh4_n` = 'NH4-N (µg/L)',
                `no3_n` = 'NO3-N (µg/L)',
                `pH` = 'pH',
                `srp` = 'SRP (µg/L)')
```

Run linear and non-linear models of kCWD ∼ chemistry

```
vars <- unique(merged$name)

# linear: Y ~ a + bX

# create empty dataframe to populate in the for loop
linear <- data.frame(mod = character(),
                     var = character(),
                     a = numeric(),
                     a_2.5 = numeric(),
                     a_97.5 = numeric(),
                     b = numeric(),
                     b_2.5 = numeric(),
                     b_97.5 = numeric(),
                     AIC = numeric(),
                     p = numeric(),
                     rse = numeric(),
                     df = numeric())


# for loop across each parameter
for(i in 1:length(vars)){
  # which var is being predicted
  var <- vars[i]

  # run the lm of k ~ chemistry
  lm <- lm(data = merged %>%
             filter(name == var),
           k_yr ~ value)

  # populate the empty data frame
  linear <- linear %>%
    add_row(
      mod = 'Linear',                    # what kind of model
      var = var,                         # for which variable
      a = coef(lm)[1],                   # intercept
      a_2.5 = confint(lm)[1,1],          # intercept low CI
      a_97.5 = confint(lm)[1,2],         # intercept high CI
      b = coef(lm)[2],                   # slope
      b_2.5 = confint(lm)[2,1],          # slope low CI
      b_97.5 = confint(lm)[2,2],         # slope high CI
      AIC = AIC(lm),                     # model AIC
      p = summary(lm)$coefficients[2,4], # p value
      rse = summary(lm)$sigma,           # RSE
      df = summary(lm)$df[2]             # df
    )
}
```

```
# empty data frame
micmen <- data.frame(mod = character(),
                     fit = character(),
                     var = character(),
                     a = numeric(),
                     a_2.5 = numeric(),
                     a_97.5 = numeric(),
                     b = numeric(),
                     b_2.5 = numeric(),
                     b_97.5 = numeric(),
                     AIC = numeric(),
                     p = numeric())

# for loop for each variable
for(i in 1:length(vars)) {
  var <- vars[i]
  mm_mod <- try(nls(data = merged %>%
                      filter(name == var),
                    k_yr ~ SSmicmen(value, a, b))
  )

  # fault tolerance, won't work for every parameter
  if(inherits(mm_mod, 'try-error')){
    micmen <- micmen %>%
      add_row(fit = 'error',
              var = var,
              mod = 'Michaelis-Menten')
    next
  }

  # populate the data frame
  micmen <- micmen %>%
    add_row(
      mod = 'Michaelis-Menten',
      fit = 'success',
      var = var,
      a = coef(mm_mod)[1],
      a_2.5 = confint2(mm_mod, level = 0.95)['a',1],
      a_97.5 = confint2(mm_mod, level = 0.95)['a',2],
      b = coef(mm_mod)[2],
      b_2.5 = confint2(mm_mod, level = 0.95)['b',1],
      b_97.5 = confint2(mm_mod, level = 0.95)['b',2],
      AIC = AIC(mm_mod),
      p = summary(mm_mod)$coefficients[8]
    )
}
```

```
## Error in nls(y ~ x/(K + x), data = xy, start = list(K = abs(pars[2L]/pars[1L])),  :
##   step factor 0.000488281 reduced below ’minFactor’ of 0.000976562
## Error in nls(y ~ x/(K + x), data = xy, start = list(K = abs(pars[2L]/pars[1L])),  :
##   step factor 0.000488281 reduced below ’minFactor’ of 0.000976562
## Error in nls(y ~ x/(K + x), data = xy, start = list(K = abs(pars[2L]/pars[1L])),  :
##   step factor 0.000488281 reduced below ’minFactor’ of 0.000976562
```

```
## Logarithmic: Y ~ a + b*log(X)
logarithmic <- data.frame(mod = character(),
                          var = character(),
                          a = numeric(),
                          a_2.5 = numeric(),
                          a_97.5 = numeric(),
                          b = numeric(),
                          b_2.5 = numeric(),
                          b_97.5 = numeric(),
                          AIC = numeric(),
                          R2 = numeric(),
                          p = numeric(),
                          rse = numeric(),
                          df = numeric()
)

for(i in 1:length(vars)) {
  var <- vars[i]

  loga <- lm(data = merged %>%
               filter(name == var),
             k_yr ~ log10(value))

  logarithmic <- logarithmic %>%
    add_row(
      mod = 'Logarithmic',
      var = var,
      a = coef(loga)[1],
      a_2.5 = confint(loga)[1],
      a_97.5 = confint(loga)[3],
      b = coef(loga)[2],
      b_2.5 = confint(loga)[2],
      b_97.5 = confint(loga)[4],
      AIC = AIC(loga),
      R2 = summary(loga)$r.squared,
      p = summary(loga)$coefficients[8],
      rse = summary(loga)$sigma,
      df = summary(loga)$df[2]
    )
}
```

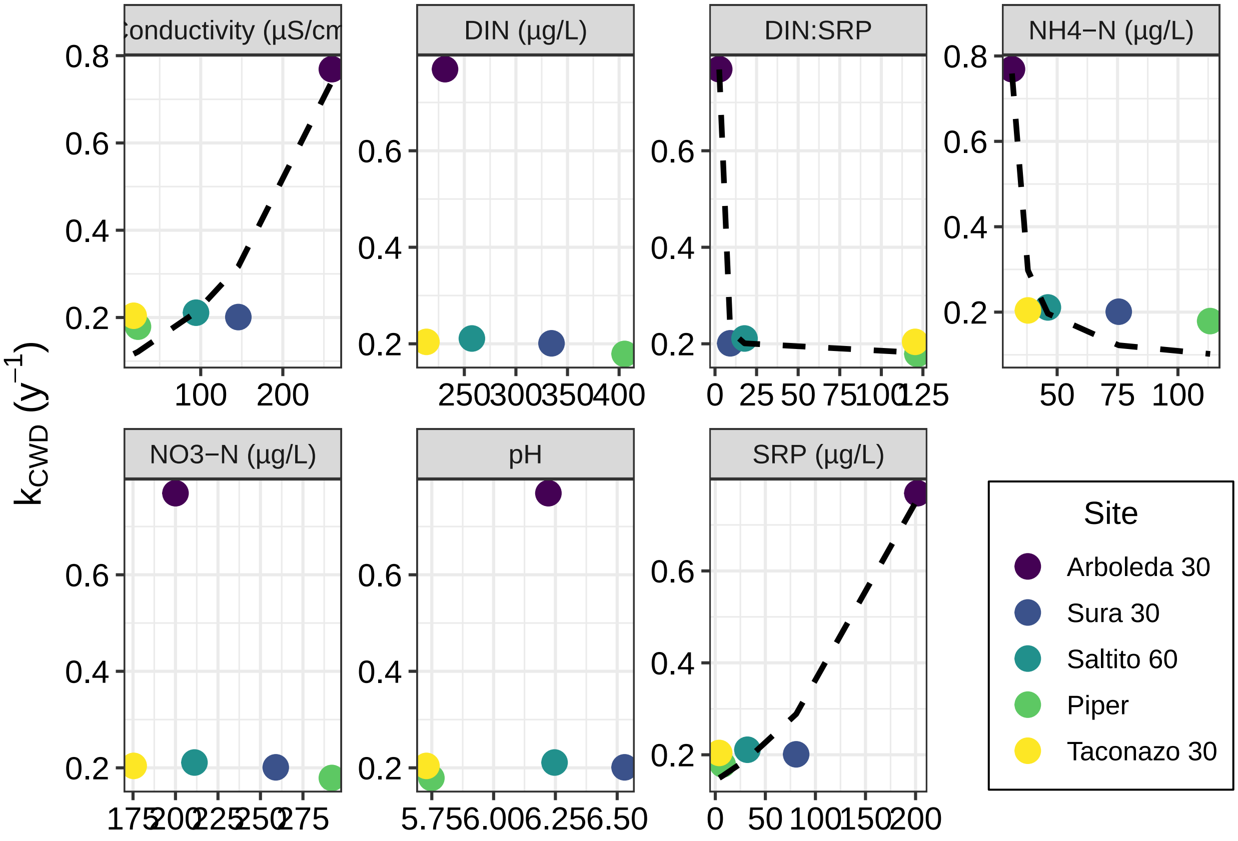
```
## Logistic: Y ~ 1/1+exp(X)
logistic <- data.frame(mod = character(),
                       var = character(),
                       b = numeric(),
                       b_2.5 = numeric(),
                       b_97.5 = numeric(),
                       d = numeric(),
                       d_2.5 = numeric(),
                       d_97.5 = numeric(),
                       e = numeric(),
                       e_2.5 = numeric(),
                       e_97.5 = numeric(),
                       AIC = numeric(),
                       rse = numeric(),
                       df = numeric())

for(i in 1:length(vars)){
  var <- vars[i]

  logi <- drm(k_yr ~ value,
              data = merged %>%
                filter(name == var),
              fct = L.3())

  logistic <- logistic %>%
    add_row(
      mod = 'Logistic',
      var = var,
      b = coef(logi)[1],
      b_2.5 = confint2(logi)['b:(Intercept)',1],
      b_97.5 = confint2(logi)['b:(Intercept)',2],
      d = coef(logi)[2],
      d_2.5 = confint2(logi)['d:(Intercept)',1],
      d_97.5 = confint2(logi)['d:(Intercept)',2],
      e = coef(logi)[3],
      e_2.5 = confint2(logi)['e:(Intercept)',1],
      e_97.5 = confint2(logi)['b:(Intercept)',2],
      AIC = AIC(logi),
      rse = summary(logi)$rseMat[1],
      df = summary(logi)$rseMat[2])
}
```

Combine and evaluate model fits

```
# combine model outputs

aic_all <- rbind(logistic %>%
                   dplyr::select(mod, var, AIC) %>%
                   mutate(p = NA),
                 logarithmic %>%
                   dplyr::select(mod, var, AIC, p),
                 linear %>%
                   dplyr::select(mod, var, AIC, p),
                 micmen %>%
                   dplyr::select(mod, var, AIC, p))

best_fits <- aic_all %>%
  arrange(var,
          AIC) %>%
  group_by(var) %>%
  dplyr::slice_head(n=1)


aic_wts <- data.frame()
for(i in 1:length(vars)) {
  use <- vars[i]

  df <- aic_all %>%
    filter(!is.na(AIC),
           var %in% use) %>%
    mutate(delAIC = akaike.weights(AIC)$deltaAIC,
           weights = akaike.weights(AIC)$weights)

  aic_wts <- rbind(aic_wts, df)
}


table3 <- aic_wts %>%
  dplyr::select(var, mod, weights) %>%
  pivot_wider(names_from = var,
              values_from = weights)
table3
```

```
## # A tibble: 4 x 8
##   mod                   srp no3_n   nh4_n     pH    Cond   din        n_p
##   <chr>               <dbl> <dbl>   <dbl>  <dbl>   <dbl> <dbl>      <dbl>
## 1 Logistic          0.835   0.129 0.00816  0.210  0.797  0.116 0.00000697
## 2 Logarithmic       0.00519 0.300 0.00610  0.397  0.0263 0.308 0.0000129
## 3 Linear            0.160   0.313 0.00416  0.392  0.177  0.309 0.00000345
## 4 Michaelis-Menten NA       0.258 0.982   NA     NA      0.267 1.00
```

```
# plot decay rates as a function of chemistry
# add lines for SRP (linear), NH4 (M-M), N:P(M-M), and cond (linear)

merged_preds <- data.frame()
for(i in 1:length(best_fits$var)) {
  fit <- best_fits[i,]

  type <- pull(fit[1])
  var <- pull(fit[2])

  if(type == 'Logistic'){
    logis <- logistic %>%
      filter(var == !!var)

    df <- merged %>%
      filter(name == var) %>%
      mutate(pred_k = logis$d/(1+exp(logis$b*(value - logis$e))),
             best_fit = type)

  } # end logistic if statement

  if(type == 'Linear') {
    line <- linear %>%
      filter(var == !!var)

    if(line$p <= 0.05){
    df <- merged %>%
      filter(name == var) %>%
      mutate(pred_k = line$a + (value*line$b),
             best_fit = type)
    } else {
      df <- merged %>%
        filter(name == var) %>%
        mutate(pred_k = NA,
               best_fit = NA)
    }
  } # end linear if statement

  if(type == 'Michaelis-Menten'){
    mm <- micmen %>%
      filter(var == !!var)

    if(mm$p <= 0.05){
    df <- merged %>%
      filter(name == var) %>%
      mutate(pred_k = (mm$a*value)/(value + mm$b),
             best_fit = type)
    } else {
      df <- merged %>%
        filter(name == var) %>%
        mutate(pred_k = NA,
               best_fit = NA)
    }
  } # end M-M for loop

  if(type == 'Logarithmic'){
    logar <- logarithmic %>%
      filter(var == !!var)

    if(logar$p <= 0.05) {
      df <- merged %>%
        filter(name == var) %>%
        mutate(pred_k = logar$a + (logar$b*value),
               best_fit = type)
    } else {
      df <- merged %>%
        filter(name == var) %>%
        mutate(pred_k = NA,
               best_fit = NA)
    }
  } # end logarithmic if statement

  merged_preds <- rbind(merged_preds, df)
} # end for loop


fig3 <- ggplot(merged_preds)+
  geom_point(aes(x = value,
                 y = k_yr,
                 color = site),
             size = 4)+
  geom_line(aes(x = value,
                y = pred_k),
            size = 1,
            linetype = 'dashed')+
  ylab(expression(paste(k[CWD], ' (', y^-1,')')))+
  facet_wrap(. ~ name,
             nrow = 2,
             scales = 'free',
             labeller = as_labeller(names_long),
             drop = FALSE)+
  scale_color_viridis_d(name = 'Site',
                        labels = as_labeller(sites_long))+
  theme(axis.title.x = element_blank(),
        legend.background = element_blank(),
        legend.box.background = element_rect(colour = "black"))
shift_legend2 <- function(p) {
  # ...
  # to grob
  gp <- ggplotGrob(p)
  facet.panels <- grep("^panel", gp[["layout"]][["name"]])
  empty.facet.panels <- sapply(facet.panels, function(i) "zeroGrob" %in% class(gp[["grobs"]][[i]]))
  empty.facet.panels <- facet.panels[empty.facet.panels]

  # establish name of empty panels
  empty.facet.panels <- gp[["layout"]][empty.facet.panels, ]
  names <- empty.facet.panels$name
  # example of names:
  #[1] "panel-3-2" "panel-3-3"

# now we just need a simple call to reposition the legend
  lemon::reposition_legend(p, 'center', panel=names)
}

shift_legend2(fig3)
```



Macroinvertebrate analysis

```
data <- readxl::read_excel('Data/Samples_Nick.xlsx')
taxonomy <- readxl::read_excel('Data/Samples_Nick.xlsx',
                                sheet = 'taxonomy')

data$Stream <- forcats::fct_relevel(data$Stream,
                           c('Arboleda 30', 'Sura 30', 'Saltito 60',
                             'Piper', 'Taconazo 30'))

# build taxonomy for each ID
tax_data <- list()
for(i in 1:length(unique(data$Taxa))){
  rank <- unique(data$Taxa)[i]

  tax_info <- taxonomy %>%
    dplyr::filter(Taxa %in% rank)

  tax_data[[i]] <- data %>%
    data.frame() %>%
    dplyr::filter(Taxa == rank) %>%
    dplyr::mutate(rank = tax_info$Rank,
          family = tax_info$Family,
            order = tax_info$Order,
            class = tax_info$Class,
            phylum = tax_info$Phylum,
            month = as.numeric(gsub(".*?([0-9]+).*", "\\1", Sample)),
            rep = gsub("\\d+", "", Sample))
}

# create df
tax_data <- purrr::reduce(tax_data, rbind)

# clean the data
tax_data_clean <- tax_data %>%
  dplyr::filter(QAQC == 0,
         month != 24)
```

```
final_dry_mass <- readr::read_csv('Data/final_dry_mass.csv')
```

```
## Rows: 53 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): site, rep
## dbl (2): month, dry_mass
##
## 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.
```

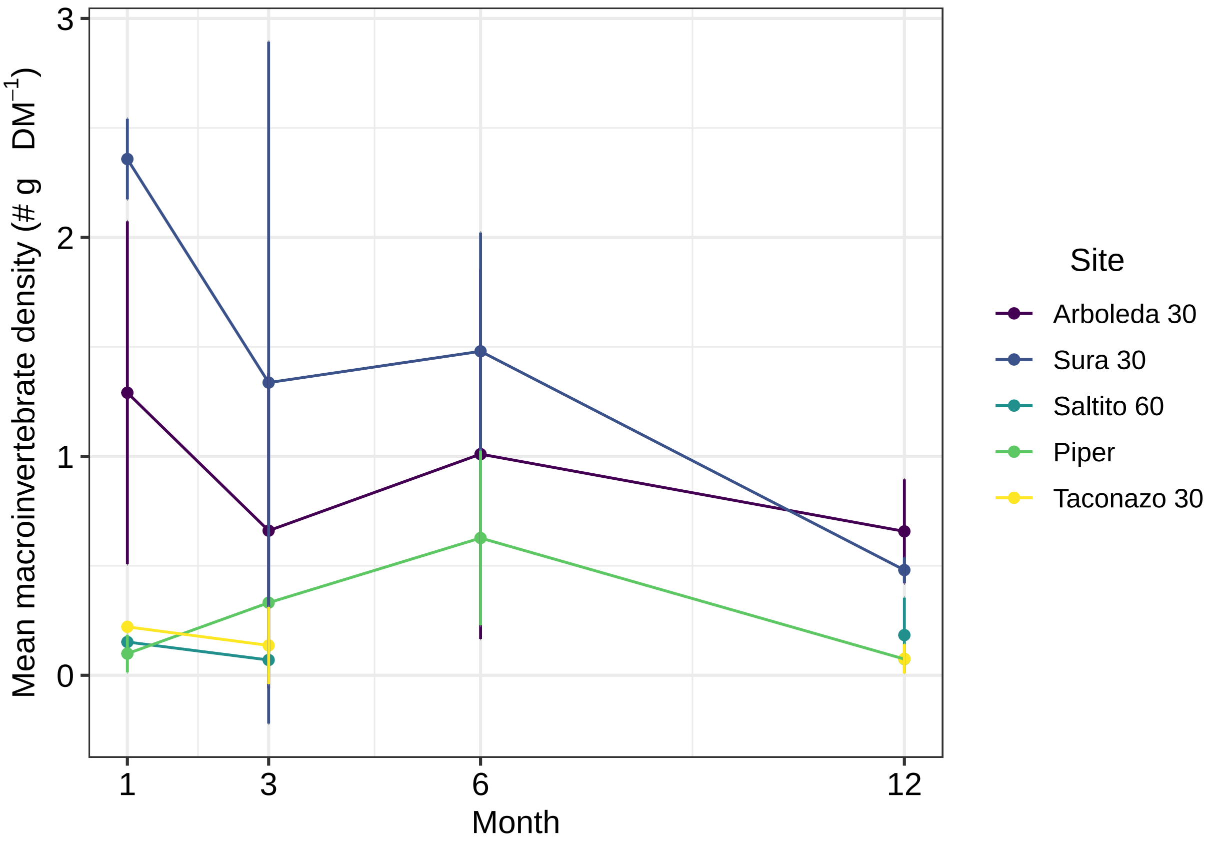
```
final_dry_mass$site <- recode_factor(final_dry_mass$site,
                                     Arb = 'Arboleda 30',
                                     Sur30 = 'Sura 30',
                                     Tito60 = 'Saltito 60',
                                     Piper = 'Piper',
                                     Tac = 'Taconazo 30')



macro_density <- tax_data_clean %>%
  dplyr::filter(!is.na(order)) %>%                              # remove NAs from Order
  dplyr::select(site = Stream, order, month, rep, Total) %>%    # get the necessary columns
  dplyr::group_by(site, month, rep) %>%                         # and do the grouping
  dplyr::summarise(total_order = sum(Total)) %>%
  dplyr::left_join(final_dry_mass,
            by = c('site', 'month', 'rep')) %>%
  dplyr::mutate(abund_per_dm = total_order/dry_mass)
```

```
## `summarise()` has grouped output by 'site', 'month'. You can override using the
## `.groups` argument.
```

```
macro_density_sum <- macro_density %>%
  group_by(site, month) %>%
  summarise(mean_den = mean(abund_per_dm, na.rm = TRUE),
            se_den = sd(abund_per_dm, na.rm = TRUE)/length(mean_den))
```

```
## `summarise()` has grouped output by 'site'. You can override using the
## `.groups` argument.
```

```
figure4 <- ggplot(macro_density_sum,
       aes(x = month,
           y = mean_den,
           color = site,
           group = site))+
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin = mean_den - se_den,
                    ymax = mean_den + se_den),
                width = 0)+
  scale_x_continuous(breaks = c(0, 1, 3, 6, 12))+
  labs(y = expression(paste('Mean macroinvertebrate density (# g   ',DM^-1,')')),
       x = 'Month')+
  scale_color_viridis_d(name = 'Site')+
  theme(axis.title = element_text(size = 12))
figure4
```



```
macro_lm <- lm(data = macro_density,
               log10(abund_per_dm) ~ factor(month) * site)

summary(macro_lm)
```

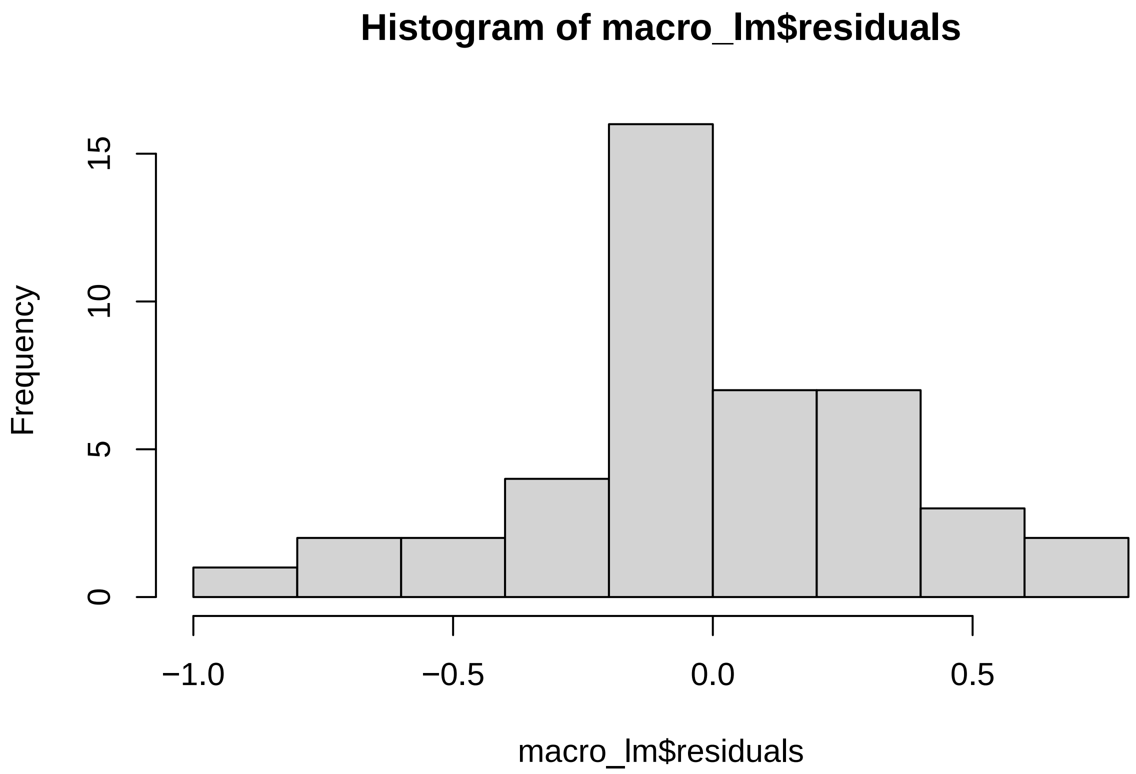
```
##
## Call:
## lm(formula = log10(abund_per_dm) ~ factor(month) * site, data = macro_density)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.89963 -0.13207 -0.00966  0.21450  0.66694
##
## Coefficients: (2 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.06241    0.25786   0.242  0.81066
## factor(month)3                 -0.45734    0.36467  -1.254  0.22095
## factor(month)6                 -0.34557    0.36467  -0.948  0.35204
## factor(month)12                -0.26362    0.36467  -0.723  0.47619
## siteSura 30                     0.30926    0.36467   0.848  0.40414
## siteSaltito 60                 -0.88172    0.40771  -2.163  0.03995 *
## sitePiper                      -1.16984    0.40771  -2.869  0.00806 **
## siteTaconazo 30                -0.71737    0.51572  -1.391  0.17601
## factor(month)3:siteSura 30     -0.02133    0.51572  -0.041  0.96733
## factor(month)6:siteSura 30      0.12579    0.51572   0.244  0.80922
## factor(month)12:siteSura 30    -0.42844    0.51572  -0.831  0.41367
## factor(month)3:siteSaltito 60   0.01225    0.57659   0.021  0.98322
## factor(month)6:siteSaltito 60        NA         NA      NA       NA
## factor(month)12:siteSaltito 60  0.22308    0.57659   0.387  0.70198
## factor(month)3:sitePiper        1.08525    0.65741   1.651  0.11081
## factor(month)6:sitePiper        1.20111    0.57659   2.083  0.04721 *
## factor(month)12:sitePiper       0.23221    0.54700   0.425  0.67468
## factor(month)3:siteTaconazo 30 -0.13879    0.65741  -0.211  0.83445
## factor(month)6:siteTaconazo 30       NA         NA      NA       NA
## factor(month)12:siteTaconazo 30 -0.43232    0.63162  -0.684  0.49974
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4466 on 26 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.7075, Adjusted R-squared:  0.5163
## F-statistic:   3.7 on 17 and 26 DF,  p-value: 0.001375
```

```
anova(macro_lm)
```

```
## Analysis of Variance Table
##
## Response: log10(abund_per_dm)
##                    Df Sum Sq Mean Sq F value    Pr(>F)
## factor(month)       3 2.8965 0.96551  4.8403  0.008302 **
## site                4 7.8264 1.95660  9.8089 5.646e-05 ***
## factor(month):site 10 1.8238 0.18238  0.9143  0.535020
## Residuals          26 5.1863 0.19947
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
hist(macro_lm$residuals)
```



```
(agricolae::HSD.test(macro_lm, 'site'))
```

```
## Registered S3 methods overwritten by 'klaR':
##   method      from 
##   predict.rda vegan
##   print.rda   vegan
##   plot.rda    vegan
```

```
## $statistics
##    MSerror Df      Mean        CV
##   0.199472 26 -0.497135 -89.83936
## 
## $parameters
##    test name.t ntr StudentizedRange alpha
##   Tukey   site   5         4.141455  0.05
## 
## $means
##             log10(abund_per_dm)       std  r         Min         Max        Q25
## Arboleda 30         -0.20422589 0.4626163 12 -1.1827949  0.34095794 -0.3674263
## Piper               -0.82684017 0.4681730  8 -1.4183013 -0.04139269 -1.1965817
## Saltito 60          -0.98118917 0.3701599  6 -1.5843312 -0.51648452 -1.1385314
## Sura 30              0.02403965 0.3714742 12 -0.6345892  0.49391505 -0.2824285
## Taconazo 30         -1.20164157 0.6479210  6 -2.0813473 -0.58415719 -1.7196216
##                     Q50        Q75
## Arboleda 30 -0.07686845  0.1212471
## Piper       -0.90836418 -0.4752450
## Saltito 60  -0.89888760 -0.8023205
## Sura 30      0.06738877  0.3516841
## Taconazo 30 -0.98567543 -0.7029616
##
## $comparison
## NULL
##
## $groups
##             log10(abund_per_dm) groups
## Sura 30               0.02403965      a
## Arboleda 30         -0.20422589      a
## Piper               -0.82684017      b
## Saltito 60          -0.98118917      b
## Taconazo 30         -1.20164157      b
##
## attr(,"class")
## [1] "group"
```

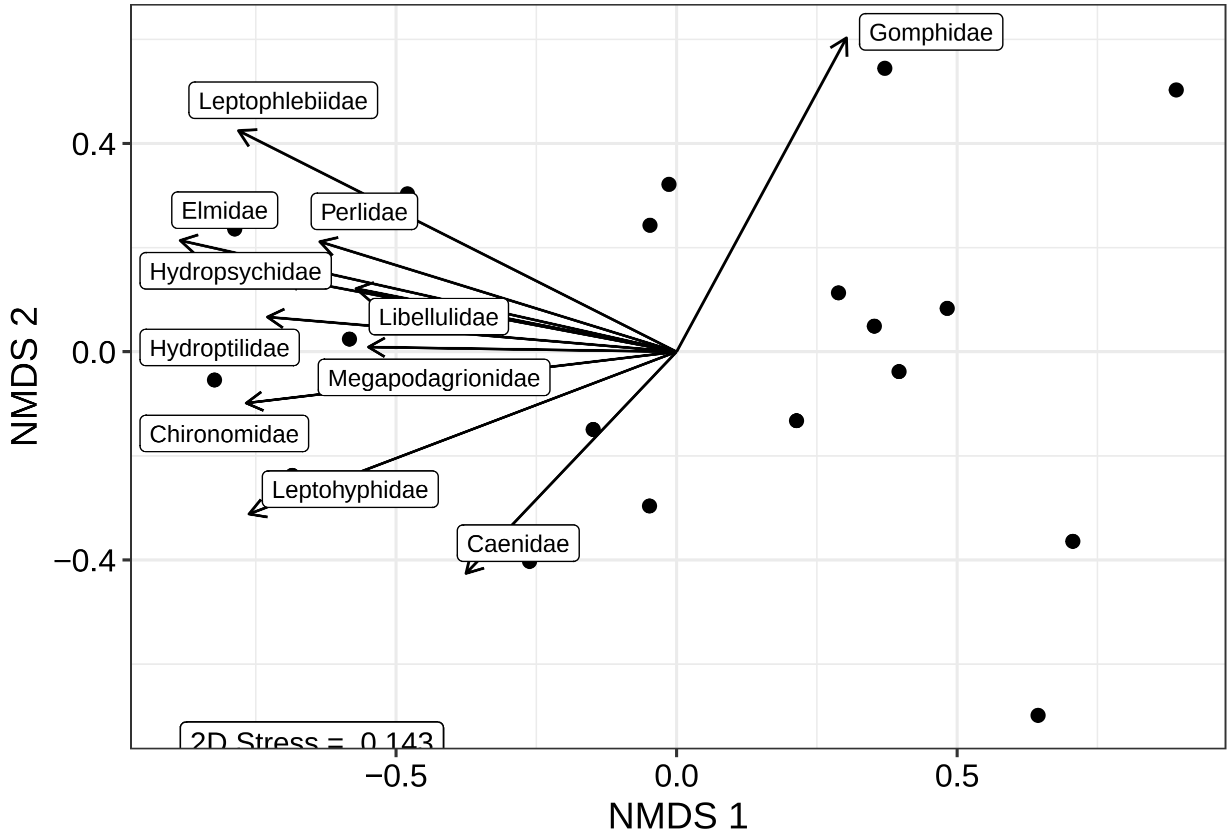
```
tax_data_clean_family_log <- tax_data_clean %>%
  dplyr::filter(!is.na(family)) %>%                               # remove NAs from Order
  dplyr::select(Stream, family, month, rep, Total) %>%    # get the necessary columns
  dplyr::group_by(Stream, month, family) %>%         # and do the grouping
  dplyr::summarise(mean_family = mean(Total, na.rm = TRUE)) %>%   # sum by order in each possible group
  dplyr::mutate(log_total = log10(1 + mean_family)) %>%          # log10 + 1 transform data
  dplyr::select(-mean_family) %>%
  pivot_wider(names_from = family,                         # pivot data
              values_from = log_total,
              values_fill = 0)
```

```
## `summarise()` has grouped output by 'Stream', 'month'. You can override using
## the `.groups` argument.
```

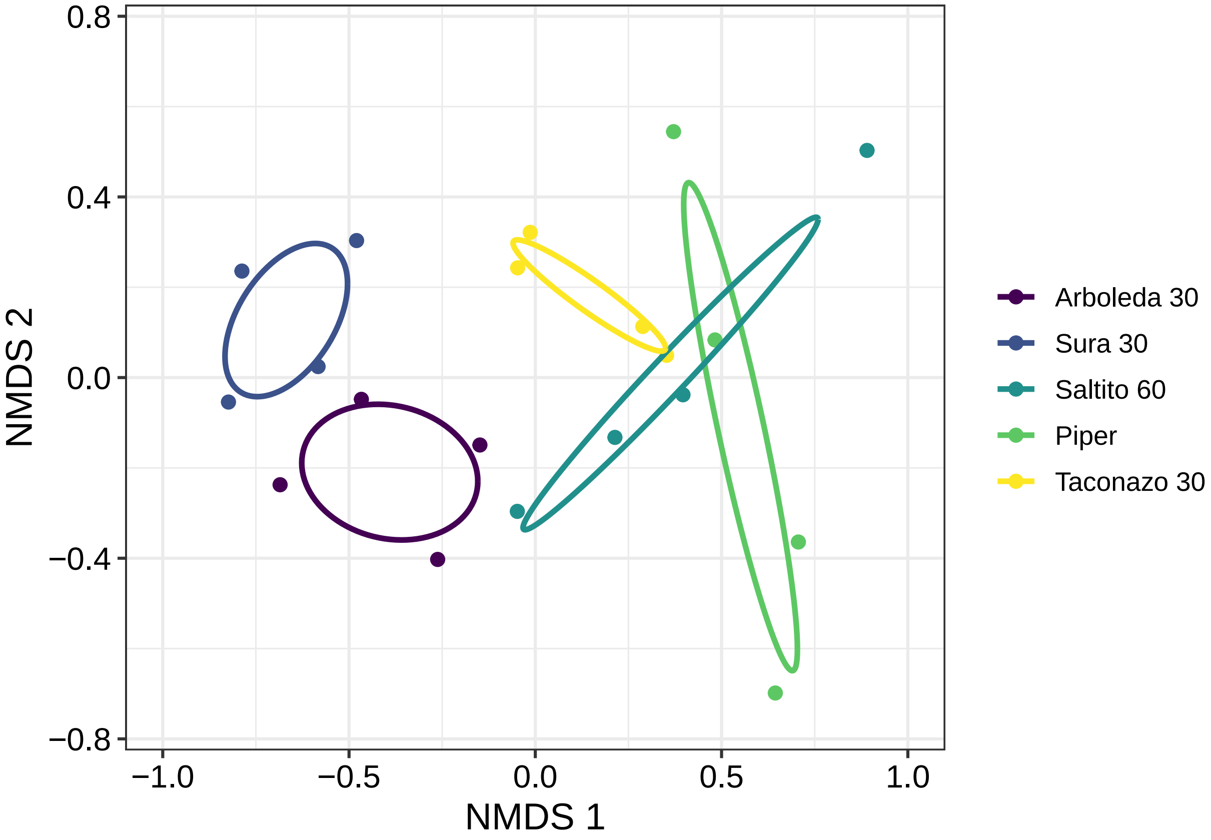
```
tax_data_log_matrix_fam <- tax_data_clean_family_log[,-c(1:2)]
tax_data_log_meta_fam <- tax_data_clean_family_log[,c(1:2)]

nmds_family_log_2d <- vegan::metaMDS(tax_data_log_matrix_fam,
                                     distance = 'bray',
                                     k = 2,
                                     autotransform = FALSE)
```

```
## Run 0 stress 0.1462039
## Run 1 stress 0.1462039
## ... New best solution
## ... Procrustes: rmse 5.981685e-05  max resid 0.0001207771
## ... Similar to previous best
## Run 2 stress 0.1525832
## Run 3 stress 0.15402
## Run 4 stress 0.1533036
## Run 5 stress 0.1462039
## ... Procrustes: rmse 0.0001468947  max resid 0.0003201662
## ... Similar to previous best
## Run 6 stress 0.1484103
## Run 7 stress 0.1540663
## Run 8 stress 0.1428578
## ... New best solution
## ... Procrustes: rmse 0.0640511  max resid 0.2064098
## Run 9 stress 0.1540666
## Run 10 stress 0.1540664
## Run 11 stress 0.1532954
## Run 12 stress 0.1506147
## Run 13 stress 0.1425758
## ... New best solution
## ... Procrustes: rmse 0.007922573  max resid 0.02783095
## Run 14 stress 0.1530004
## Run 15 stress 0.143685
## Run 16 stress 0.1551354
## Run 17 stress 0.1509932
## Run 18 stress 0.1462039
## Run 19 stress 0.153263
## Run 20 stress 0.1439368
## *** Best solution was not repeated -- monoMDS stopping criteria:
##     18: stress ratio > sratmax
##      2: scale factor of the gradient < sfgrmin
```

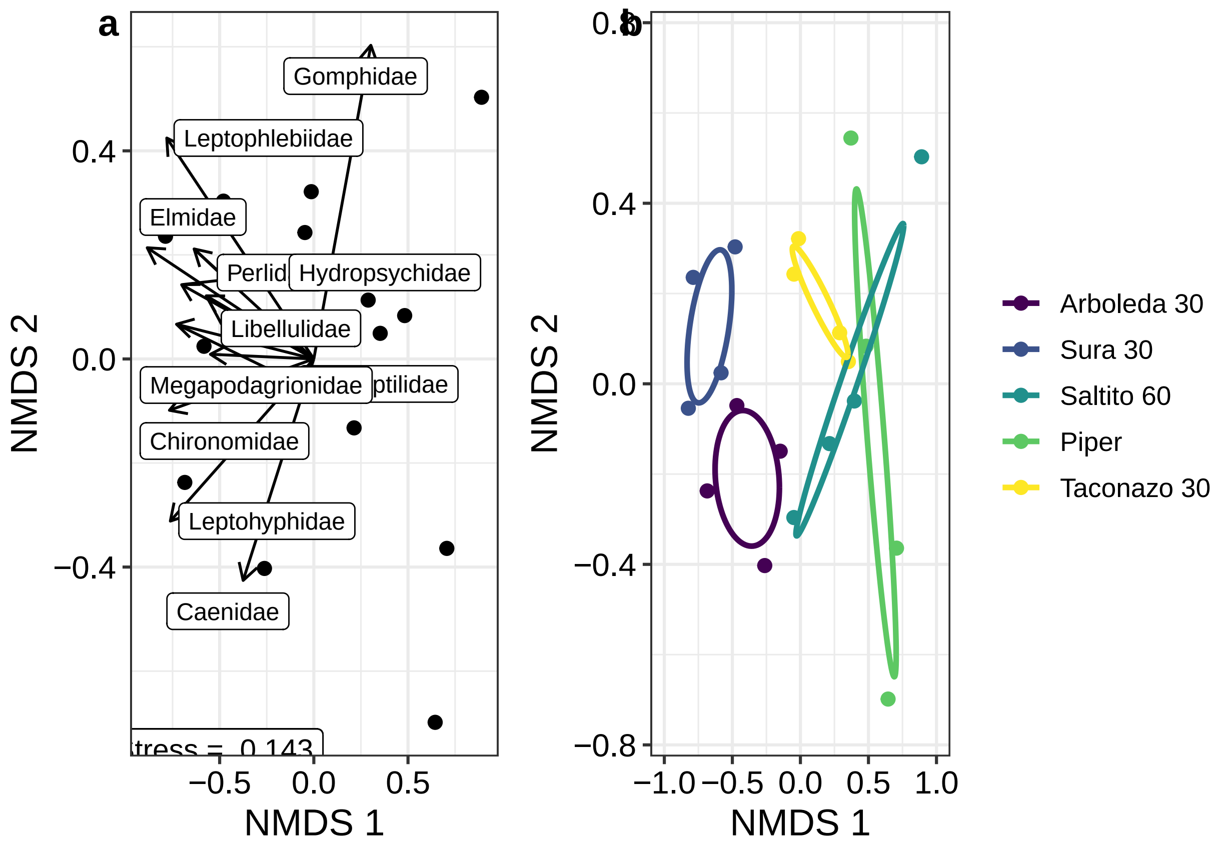
```r
stress_fam_log_2d <- round(nmds_family_log_2d$stress, 3)

# pick the log transformed 2d NMDS (14.4% stress)

nmds_family_out <- data.frame(x = nmds_family_log_2d$points[,1],
                              y = nmds_family_log_2d$points[,2])

nmds_family_out <- cbind(tax_data_log_meta_fam,
                         nmds_family_out)

fit_fam <- (vegan::envfit(nmds_family_log_2d,
                          tax_data_log_matrix_fam,
                          perm = 9999))

scrs_fam <- data.frame(vegan::scores(fit_fam, 'vectors'))

scrs_fam$pvals <- fit_fam$vectors$pvals

scrs_fam_sig <- subset(scrs_fam, pvals <= 0.05)

scrs_fam_sig$env.variables <- row.names(scrs_fam_sig)


plot_nmds_family_scrs <- ggplot(nmds_family_out,
                                aes(x = x, y = y))+
  geom_point(size = 2)+
  geom_segment(data = scrs_fam_sig,
               aes(x = 0, xend = NMDS1,
                   y = 0, yend = NMDS2),
               arrow = arrow(length = unit(0.25, 'cm')),
               color = 'black')+
  ggrepel::geom_label_repel(data = scrs_fam_sig,
                            aes(NMDS1, NMDS2,
                                label = env.variables),
                            size = 3)+
  labs(x = 'NMDS 1',
       y = 'NMDS 2')+
  # lims(x = c(-1,1),
  #      y = c(-0.75, 0.75))+
  geom_label(label = paste('2D Stress = ', stress_fam_log_2d),
             x = -0.65,
             y = -0.75)
plot_nmds_family_scrs
```



```
source('Code/veganCovEllipse.R')
streams <- data.frame()
for(i in unique(nmds_family_out$Stream)){
  streams <- rbind(streams,
                   cbind(
                     as.data.frame(
                       with(nmds_family_out[nmds_family_out$Stream == i,],
                            veganCovEllipse(cov.wt(cbind(x, y),
                                                   wt = rep(1/length(x),
                                                            length(x)))$cov,
                                            center = c(mean(x),
                                                       mean(y)
                                            )
                            )
                       )
                     ),
                     Stream = i)
  )
} # end for loop

plot_nmds_streams <- ggplot(data = nmds_family_out,
                            aes(x = x, y = y))+
  geom_point(aes(color = Stream),
             size = 2)+
  geom_path(data = streams,
            linewidth = 1,
            aes(x = x, y = y, color = Stream))+
  labs(x = "NMDS 1",
       y = "NMDS 2")+
  lims(x = c(-1,1),
       y = c(-0.75,0.75))+
  theme(legend.background = element_blank())+
  scale_color_viridis_d(name = element_blank())
plot_nmds_streams
```



```r
fig5 <- ggpubr::ggarrange(plot_nmds_family_scrs,
                  plot_nmds_streams,
                  align = 'h',
                  widths = c(1,1.4),
                  labels = 'auto',
                  label.x = c(0.18,0.13))
fig5
```



```r
# Permanova
# The goal of this test is to tell you if there are significant differences in your response variables
permanova <- vegan::adonis2(tax_data_log_matrix_fam ~ Stream * month,
                            data = tax_data_log_meta_fam,
                            permutations = 9999,
                            method = 'bray',
                            by = 'terms')
permanova
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## vegan::adonis2(formula = tax_data_log_matrix_fam ~ Stream * month, data = tax_data_log_meta_fam, per
##               Df SumOfSqs      R2      F Pr(>F)
## Stream         4  1.52075 0.52397 4.1548 0.0003 ***
## month         1  0.14162 0.04879 1.5476 0.1729
## Stream:month  4  0.32495 0.11196 0.8878 0.6093
## Residual     10  0.91505 0.31528
## Total        19  2.90237 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(permanova)
```

```
##        Df          SumOfSqs            R2                 F
##  Min.   : 1.0   Min.   :0.1416   Min.   :0.04879   Min.   :0.8878
##  1st Qu.: 4.0   1st Qu.:0.3250   1st Qu.:0.11196   1st Qu.:1.2177
##  Median : 4.0   Median :0.9151   Median :0.31528   Median :1.5476
##  Mean   : 7.6   Mean   :1.1609   Mean   :0.40000   Mean   :2.1967
##  3rd Qu.:10.0   3rd Qu.:1.5208   3rd Qu.:0.52397   3rd Qu.:2.8512
##  Max.   :19.0   Max.   :2.9024   Max.   :1.00000   Max.   :4.1548
##                                                    NA's   :2
##      Pr(>F)
##  Min.   :0.0003
##  1st Qu.:0.0866
##  Median :0.1729
##  Mean   :0.2608
##  3rd Qu.:0.3911
##  Max.   :0.6093
##  NA's   :2
```

```
# SIMPER
sim <- with(tax_data_log_meta_fam,
            simper(tax_data_log_matrix_fam, Stream))

summary(sim)
```

```
##
## Contrast: Arboleda 30_Sura 30
##
##                     average      sd   ratio     ava     avb cumsum     p
## Hydroptilidae      0.04481 0.03065 1.46200 0.19450 0.61820  0.105 0.122
## Perlidae           0.03594 0.02521 1.42600 0.07530 0.45040  0.189 0.096 .
## Hydropsychidae     0.03450 0.01878 1.83740 0.34510 0.74720  0.270 0.931
## Leptohyphidae      0.02993 0.01977 1.51410 0.50550 0.72550  0.341 0.992
## Chironomidae       0.02972 0.01540 1.92960 0.77320 0.87940  0.410 0.989
## Leptophlebiidae    0.02359 0.01571 1.50230 0.47580 0.67660  0.466 0.997
## Ceratopogonidae    0.02238 0.01655 1.35230 0.22580 0.28650  0.518 0.901
## Caenidae           0.02190 0.01592 1.37540 0.28650 0.07530  0.570 0.522
## Polycentropodidae 0.02170 0.02304 0.94170 0.00000 0.23860  0.620 0.070 .
## Elmidae            0.02061 0.01098 1.87720 0.83920 1.03930  0.669 1.000
## Libellulidae       0.02048 0.02191 0.93470 0.07530 0.25000  0.717 0.156
## Simuliidae         0.01804 0.01912 0.94320 0.00000 0.20600  0.759 0.082 .
## Coenagrionidae     0.01715 0.01414 1.21250 0.22580 0.07530  0.799 0.730
## Gyrinidae          0.01450 0.01517 0.95540 0.15050 0.00000  0.833 0.147
## Megapodagrionidae 0.01390 0.01472 0.94420 0.07530 0.15050  0.866 0.181
## Empididae          0.01350 0.01413 0.95540 0.00000 0.15050  0.898 0.429
## Baetidae           0.01147 0.02069 0.55410 0.11930 0.00000  0.924 0.406
## Scirtidae         0.00993 0.01352 0.73430 0.07530 0.07530  0.948 0.157
## Leptoceridae      0.00779 0.01403 0.55500 0.00000 0.07530  0.966 0.646
## Calamoceratidae   0.00723 0.01306 0.55410 0.07530 0.00000  0.983 0.406
## Planariidae       0.00722 0.01300 0.55550 0.00000 0.07530  1.000 0.398
## Gomphidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.953
## Forcypominae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.390
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Contrast: Arboleda 30_Saltito 60
##
##                   average      sd   ratio     ava     avb cumsum     p
## Leptophlebiidae   0.07680 0.03481 2.20630 0.47580 0.00000  0.138 0.034 *
## Elmidae           0.07590 0.03697 2.05320 0.83920 0.38910  0.274 0.236
## Chironomidae      0.05637 0.02984 1.88920 0.77320 0.51800  0.376 0.284
## Caenidae          0.04185 0.02899 1.44360 0.28650 0.11930  0.451 0.034 *
## Leptohyphidae     0.04184 0.03988 1.04920 0.50550 0.22580  0.526 0.833
## Hydropsychidae    0.03945 0.03462 1.13960 0.34510 0.22580  0.597 0.841
## Coenagrionidae    0.03678 0.02375 1.54870 0.22580 0.00000  0.663 0.056 .
## Ceratopogonidae   0.03389 0.02920 1.16090 0.22580 0.15050  0.724 0.546
## Gyrinidae         0.02763 0.02888 0.95700 0.15050 0.00000  0.774 0.006 **
## Hydroptilidae     0.02737 0.02858 0.95790 0.19450 0.00000  0.823 0.602
## Gomphidae         0.02646 0.02827 0.93590 0.00000 0.15050  0.870 0.375
## Baetidae          0.02182 0.03933 0.55470 0.11930 0.00000  0.909 0.022 *
## Calamoceratidae   0.01377 0.02482 0.55470 0.07530 0.00000  0.934 0.022 *
## Libellulidae      0.00915 0.01642 0.55720 0.07530 0.00000  0.951 0.505
## Megapodagrionidae 0.00915 0.01642 0.55720 0.07530 0.00000  0.967 0.477
## Perlidae          0.00915 0.01642 0.55720 0.07530 0.00000  0.984 0.918
## Scirtidae         0.00915 0.01642 0.55720 0.07530 0.00000  1.000 0.240
## Polycentropodidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.661
## Simuliidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.661
## Empididae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.831
## Planariidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.418
## Leptoceridae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.649
## Forcypominae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.433
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Contrast: Arboleda 30_Piper
##
##                   average      sd   ratio     ava     avb cumsum     p
## Elmidae           0.11913 0.04934 2.41420 0.83920 0.15050  0.181 0.002 **
## Leptophlebiidae   0.06800 0.04375 1.55440 0.47580 0.07530  0.284 0.096 .
## Leptohyphidae     0.06552 0.03424 1.91330 0.50550 0.07530  0.384 0.070 .
## Hydropsychidae    0.05504 0.04064 1.35430 0.34510 0.07530  0.468 0.243
## Chironomidae      0.04927 0.03647 1.35120 0.77320 0.46110  0.543 0.502
## Caenidae          0.04527 0.02751 1.64600 0.28650 0.00000  0.612 0.021 *
## Ceratopogonidae   0.04377 0.02778 1.57570 0.22580 0.36930  0.678 0.270
## Coenagrionidae    0.03131 0.02630 1.19070 0.22580 0.07530  0.726 0.146
## Gyrinidae         0.02848 0.02966 0.96010 0.15050 0.00000  0.769 0.004 **
## Hydroptilidae     0.02804 0.02921 0.96000 0.19450 0.00000  0.812 0.563
## Leptoceridae      0.02534 0.04590 0.55200 0.00000 0.17470  0.850 0.390
## Baetidae          0.02248 0.04044 0.55590 0.11930 0.00000  0.884 0.006 **
## Calamoceratidae   0.01418 0.02552 0.55590 0.07530 0.00000  0.906 0.006 **
## Empididae         0.01360 0.02478 0.54910 0.00000 0.07530  0.927 0.428
## Gomphidae         0.01091 0.01977 0.55200 0.00000 0.07530  0.943 0.935
## Libellulidae      0.00934 0.01675 0.55750 0.07530 0.00000  0.957 0.489
## Megapodagrionidae 0.00934 0.01675 0.55750 0.07530 0.00000  0.972 0.447
## Perlidae          0.00934 0.01675 0.55750 0.07530 0.00000  0.986 0.892
## Scirtidae         0.00934 0.01675 0.55750 0.07530 0.00000  1.000 0.183
## Polycentropodidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.632
## Simuliidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.632
## Planariidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.392
## Forcypominae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.412
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Contrast: Arboleda 30_Taconazo 30
##
##                   average      sd   ratio     ava     avb cumsum     p
## Leptohyphidae     0.06981 0.02312 3.01980 0.50550 0.00000  0.137 0.055 .
## Elmidae           0.05320 0.02811 1.89270 0.83920 0.47970  0.242 0.757
## Leptophlebiidae   0.05219 0.03501 1.49090 0.47580 0.28260  0.344 0.502
## Ceratopogonidae   0.04216 0.03214 1.31200 0.22580 0.40050  0.427 0.286
## Caenidae          0.03992 0.02426 1.64560 0.28650 0.00000  0.505 0.049 *
## Chironomidae      0.03808 0.02417 1.57520 0.77320 0.63870  0.580 0.876
## Hydropsychidae    0.03377 0.02631 1.28370 0.34510 0.25700  0.646 0.931
## Coenagrionidae    0.02783 0.02324 1.19750 0.22580 0.07530  0.701 0.249
## Hydroptilidae     0.02499 0.02610 0.95760 0.19450 0.00000  0.750 0.629
## Gyrinidae         0.02464 0.02566 0.96020 0.15050 0.00000  0.798 0.020 *
## Gomphidae         0.02153 0.02287 0.94140 0.00000 0.15050  0.841 0.575
## Baetidae          0.01946 0.03500 0.55590 0.11930 0.00000  0.879 0.107
## Perlidae          0.01438 0.01930 0.74500 0.07530 0.07530  0.907 0.799
## Calamoceratidae   0.01228 0.02208 0.55590 0.07530 0.00000  0.931 0.107
## Forcypominae      0.00973 0.01758 0.55320 0.00000 0.07530  0.950 0.370
## Libellulidae      0.00847 0.01520 0.55750 0.07530 0.00000  0.967 0.541
## Megapodagrionidae 0.00847 0.01520 0.55750 0.07530 0.00000  0.983 0.560
## Scirtidae         0.00847 0.01520 0.55750 0.07530 0.00000  1.000 0.364
## Polycentropodidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.662
## Simuliidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.662
## Empididae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.819
## Planariidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.393
## Leptoceridae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.676
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Contrast: Sura 30_Saltito 60
##
##                   average      sd   ratio     ava     avb cumsum     p
## Leptophlebiidae   0.07961 0.01422 5.59800 0.67660 0.00000  0.121 0.021 *
## Elmidae           0.07683 0.01676 4.58400 1.03930 0.38910  0.237 0.240
## Hydroptilidae     0.06684 0.04381 1.52600 0.61820 0.00000  0.339 0.003 **
## Hydropsychidae    0.06287 0.02451 2.56500 0.74720 0.22580  0.434 0.079 .
## Leptohyphidae     0.06061 0.03318 1.82700 0.72550 0.22580  0.526 0.167
## Perlidae          0.04998 0.03367 1.48400 0.45040 0.00000  0.602 0.003 **
## Chironomidae      0.04908 0.03311 1.48200 0.87940 0.51800  0.676 0.547
## Polycentropodidae 0.02889 0.03096 0.93300 0.23860 0.00000  0.720 0.004 **
## Libellulidae      0.02542 0.02962 0.85800 0.25000 0.00000  0.759 0.028 *
## Ceratopogonidae   0.02421 0.02136 1.13400 0.28650 0.15050  0.795 0.881
## Simuliidae        0.02373 0.02484 0.95500 0.20600 0.00000  0.831 0.004 **
## Gomphidae         0.01885 0.02000 0.94300 0.00000 0.15050  0.860 0.682
## Megapodagrionidae 0.01884 0.01996 0.94400 0.15050 0.00000  0.889 0.027 *
## Empididae         0.01779 0.01862 0.95600 0.15050 0.00000  0.915 0.376
## Caenidae          0.01743 0.02258 0.77200 0.07530 0.11930  0.942 0.687
## Leptoceridae      0.01076 0.01933 0.55600 0.07530 0.00000  0.958 0.413
## Scirtidae         0.00971 0.01743 0.55700 0.07530 0.00000  0.973 0.166
## Planariidae       0.00971 0.01743 0.55700 0.07530 0.00000  0.988 0.020 *
## Coenagrionidae    0.00808 0.01450 0.55800 0.07530 0.00000  1.000 0.947
## Gyrinidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.662
## Baetidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.410
## Calamoceratidae   0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.410
## Forcypominae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.367
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Contrast: Sura 30_Piper
##
##                   average      sd   ratio     ava     avb cumsum     p
## Elmidae           0.10786 0.02607 4.13700 1.03930 0.15050  0.149 0.003 **
## Hydropsychidae    0.08135 0.02132 3.81600 0.74720 0.07530  0.262 0.002 **
## Leptohyphidae     0.07875 0.03093 2.54600 0.72550 0.07530  0.370 0.014 *
## Leptophlebiidae   0.07303 0.02391 3.05400 0.67660 0.07530  0.471 0.058 .
## Hydroptilidae     0.06808 0.04451 1.53000 0.61820 0.00000  0.565 0.002 **
## Chironomidae      0.05236 0.02745 1.90700 0.87940 0.46110  0.638 0.408
## Perlidae          0.05094 0.03427 1.48600 0.45040 0.00000  0.708 0.003 **
## Polycentropodidae 0.02950 0.03161 0.93300 0.23860 0.00000  0.749 0.003 **
## Libellulidae      0.02586 0.03011 0.85900 0.25000 0.00000  0.785 0.015 *
## Leptoceridae      0.02524 0.03090 0.81700 0.07530 0.17470  0.820 0.383
## Simuliidae        0.02421 0.02530 0.95700 0.20600 0.00000  0.853 0.003 **
## Megapodagrionidae 0.01925 0.02038 0.94400 0.15050 0.00000  0.879 0.014 *
## Ceratopogonidae   0.01910 0.01615 1.18200 0.28650 0.36930  0.906 0.959
## Empididae         0.01828 0.01927 0.94800 0.15050 0.07530  0.931 0.320
## Coenagrionidae    0.01355 0.01844 0.73500 0.07530 0.07530  0.950 0.854
## Scirtidae         0.00992 0.01779 0.55700 0.07530 0.00000  0.964 0.130
## Planariidae       0.00992 0.01779 0.55700 0.07530 0.00000  0.977 0.009 **
## Caenidae          0.00823 0.01475 0.55800 0.07530 0.00000  0.989 0.884
## Gomphidae         0.00820 0.01484 0.55300 0.00000 0.07530  1.000 0.963
## Gyrinidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.640
## Baetidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.406
## Calamoceratidae   0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.406
## Forcypominae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.439
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Contrast: Sura 30_Taconazo 30
##
##                   average      sd   ratio     ava     avb cumsum     p
## Leptohyphidae     0.08016 0.02461 3.25800 0.72550 0.00000  0.138 0.008 **
## Hydroptilidae     0.06237 0.04101 1.52100 0.61820 0.00000  0.245 0.007 **
## Elmidae           0.06031 0.01065 5.66300 1.03930 0.47970  0.349 0.631
## Hydropsychidae    0.05406 0.02088 2.58900 0.74720 0.25700  0.442 0.252
## Leptophlebiidae   0.04662 0.03425 1.36100 0.67660 0.28260  0.522 0.702
## Perlidae          0.04339 0.02972 1.46000 0.45040 0.07530  0.597 0.024 *
## Chironomidae      0.03647 0.02351 1.55100 0.87940 0.63870  0.659 0.925
## Polycentropodidae 0.02669 0.02846 0.93800 0.23860 0.00000  0.705 0.021 *
## Libellulidae      0.02383 0.02779 0.85800 0.25000 0.00000  0.746 0.080 .
## Simuliidae        0.02201 0.02308 0.95400 0.20600 0.00000  0.784 0.021 *
## Ceratopogonidae   0.02066 0.01886 1.09500 0.28650 0.40050  0.820 0.924
## Megapodagrionidae 0.01738 0.01834 0.94800 0.15050 0.00000  0.850 0.074 .
## Empididae         0.01650 0.01723 0.95800 0.15050 0.00000  0.878 0.382
## Gomphidae         0.01621 0.01712 0.94700 0.00000 0.15050  0.906 0.790
## Coenagrionidae    0.01204 0.01624 0.74100 0.07530 0.07530  0.926 0.879
## Leptoceridae      0.00983 0.01765 0.55700 0.07530 0.00000  0.943 0.469
## Scirtidae         0.00895 0.01605 0.55700 0.07530 0.00000  0.959 0.252
## Planariidae       0.00895 0.01605 0.55700 0.07530 0.00000  0.974 0.108
## Caenidae          0.00755 0.01354 0.55800 0.07530 0.00000  0.987 0.911
## Forcypominae      0.00751 0.01357 0.55400 0.00000 0.07530  1.000 0.397
## Gyrinidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.646
## Baetidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.403
## Calamoceratidae   0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.403
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Contrast: Saltito 60_Piper
##
##                   average      sd   ratio     ava     avb cumsum     p
## Elmidae           0.07628 0.06081 1.25430 0.38910 0.15050  0.149 0.245
## Ceratopogonidae   0.06975 0.06210 1.12320 0.15050 0.36930  0.286 0.017 *
## Chironomidae      0.06370 0.05167 1.23300 0.51800 0.46110  0.410 0.110
## Hydropsychidae    0.05484 0.04653 1.17840 0.22580 0.07530  0.517 0.251
## Leptohyphidae     0.05454 0.04724 1.15460 0.22580 0.07530  0.624 0.370
## Gomphidae         0.05143 0.05729 0.89780 0.15050 0.07530  0.725 0.012 *
## Leptoceridae      0.04123 0.07475 0.55170 0.00000 0.17470  0.805 0.061 .
## Caenidae          0.02967 0.05356 0.55390 0.11930 0.00000  0.863 0.274
## Empididae         0.02629 0.04848 0.54230 0.00000 0.07530  0.914 0.094 .
## Coenagrionidae    0.02596 0.04783 0.54280 0.00000 0.07530  0.965 0.315
## Leptophlebiidae   0.01776 0.03219 0.55170 0.00000 0.07530  1.000 0.999
## Hydroptilidae     0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.952
## Libellulidae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.816
## Megapodagrionidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.816
## Perlidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.949
## Scirtidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.666
## Gyrinidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.658
## Baetidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.416
## Calamoceratidae   0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.416
## Polycentropodidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.673
## Simuliidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.673
## Planariidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.418
## Forcypominae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.389
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Contrast: Saltito 60_Taconazo 30
##
##                   average      sd   ratio     ava     avb cumsum     p
## Ceratopogonidae   0.06596 0.06082 1.08460 0.15050 0.40050  0.156 0.015 *
## Chironomidae      0.06392 0.04497 1.42140 0.51800 0.63870  0.306 0.100 .
## Leptophlebiidae   0.06218 0.06913 0.89940 0.00000 0.28260  0.453 0.167
## Leptohyphidae     0.05115 0.03168 1.61460 0.22580 0.00000  0.574 0.497
## Gomphidae         0.03758 0.04084 0.92020 0.15050 0.15050  0.662 0.072 .
## Hydropsychidae    0.03621 0.04191 0.86410 0.22580 0.25700  0.748 0.923
## Elmidae           0.03478 0.02899 1.19980 0.38910 0.47970  0.830 0.988
## Caenidae          0.02462 0.04442 0.55430 0.11930 0.00000  0.888 0.449
## Coenagrionidae    0.01783 0.03232 0.55160 0.00000 0.07530  0.930 0.706
## Perlidae          0.01482 0.02675 0.55400 0.00000 0.07530  0.965 0.781
## Forcypominae      0.01482 0.02675 0.55400 0.00000 0.07530  1.000 0.011 *
## Hydroptilidae     0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.958
## Libellulidae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.828
## Megapodagrionidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.823
## Scirtidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.661
## Gyrinidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.668
## Baetidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.416
## Calamoceratidae   0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.416
## Polycentropodidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.655
## Simuliidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.655
## Empididae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.798
## Planariidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.393
## Leptoceridae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Contrast: Piper_Taconazo 30
##
##                   average      sd   ratio     ava     avb cumsum     p
## Elmidae           0.09030 0.06020 1.49990 0.15050 0.47970  0.190 0.055 .
## Leptophlebiidae   0.06594 0.06569 1.00390 0.07530 0.28260  0.329 0.106
## Chironomidae      0.05922 0.03812 1.55340 0.46110 0.63870  0.453 0.183
## Hydropsychidae    0.05485 0.04208 1.30350 0.07530 0.25700  0.569 0.241
## Gomphidae         0.03714 0.03952 0.93970 0.07530 0.15050  0.647 0.091 .
## Leptoceridae      0.03561 0.06424 0.55440 0.17470 0.00000  0.721 0.171
## Ceratopogonidae   0.02996 0.03183 0.94110 0.36930 0.40050  0.784 0.704
## Coenagrionidae    0.02950 0.04027 0.73250 0.07530 0.07530  0.847 0.174
## Leptohyphidae     0.02120 0.03850 0.55070 0.07530 0.00000  0.891 0.998
## Empididae         0.02120 0.03850 0.55070 0.07530 0.00000  0.936 0.219
## Perlidae          0.01530 0.02754 0.55540 0.00000 0.07530  0.968 0.788
## Forcypominae      0.01530 0.02754 0.55540 0.00000 0.07530  1.000 0.007 **
## Caenidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.947
## Hydroptilidae     0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.944
## Libellulidae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.786
## Megapodagrionidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.800
## Scirtidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.661
## Gyrinidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.680
## Baetidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.420
## Calamoceratidae   0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.420
## Polycentropodidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.630
## Simuliidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.630
## Planariidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.418
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```
simper_use <- do.call(rbind, summary(sim)) %>%
  tibble::rownames_to_column() %>%
  dplyr::rowwise() %>%
  dplyr::mutate(comparison = strsplit(rowname, "[.]")[[1]][1],
                family = strsplit(rowname, "[.]")[[1]][2]) %>%
  dplyr::select(-rowname)

simper_sig <- simper_use %>%
  dplyr::filter(p <= 0.05) %>%
  dplyr::group_by(comparison, family) %>%
  dplyr::summarise(n_sig = length(p),
                   mean_avg = mean(average, na.rm = TRUE))
```

```
## ‘summarise()‘ has grouped output by ’comparison’. You can override using the
## ‘.groups‘ argument.
```

```
table_s1 <- simper_use %>%
  dplyr::filter(p <= 0.05) %>%
  dplyr::group_by(comparison, family)

table_s1[,-c(8:9)] <- round(table_s1[,-c(8:9)],
                       digits = 2)



# ANOSIM
anosim <- anosim(tax_data_log_matrix_fam,
                 grouping = tax_data_log_meta_fam$Stream,
                 distance = 'bray',
                 permutations = 9999)
summary(anosim)
```

```
##
## Call:
## anosim(x = tax_data_log_matrix_fam, grouping = tax_data_log_meta_fam$Stream,      permutations = 999
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.5204
##       Significance: 1e-04
##
## Permutation: free
## Number of permutations: 9999
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.134 0.185 0.224 0.279
##
## Dissimilarity ranks between and within classes:
##             0%   25%   50%    75% 100%   N
## Between      1 59.50 104.5 149.25  190 160
## Arboleda 30  9 23.25  46.0  56.75  107   6
## Sura 30     10 14.25  19.5  38.25   51   6
## Saltito 60   2 17.25  33.5  96.25  140   6
## Piper       63 80.00  94.5 111.25  157   6
## Taconazo 30  6 20.00  40.0  60.75   69   6
```

```
anosim_r <- anosim$statistic
anosim_p <- anosim$signif
```

Session Info

```
pander(sessionInfo())
```

R version 4.2.1 (2022-06-23 ucrt)

Platform: x86_64-w64-mingw32/x64 (64-bit)

locale: *LC_COLLATE=English_United States.utf8*, *LC_CTYPE=English_United States.utf8*, *LC_MONETARY=English_ States.utf8*, *LC_NUMERIC=C* and *LC_TIME=English_United States.utf8*

attached base packages: *grid*, *stats*, *graphics*, *grDevices*, *utils*, *datasets*, *methods* and *base*

other attached packages: *pander(v.0.6.5)*, *vegan(v.2.6-4)*, *lattice(v.0.20-45)*, *permute(v.0.9-7)*, *spDataLarge(v.2.0.9)*, *leaflet(v.2.1.1)*, *tmap(v.3.3-3)*, *spData(v.2.2.1)*, *raster(v.3.6-14)*, *sp(v.1.6-0)*, *sf(v.1.0-9)*, *qpcR(v.1.4-1)*, *Matrix(v.1.5-3)*, *robustbase(v.0.95-0)*, *rgl(v.1.0.1)*, *minpack.lm(v.1.2-3)*, *nlstools(v.2.0-0)*, *nlme(v.3.1-157)*, *drc(v.3.0-1)*, *MASS(v.7.3-57)*, *car(v.3.1-1)*, *carData(v.3.0-5)*, *lemon(v.0.4.6)*, *ggrepel(v.0.9.2)*, *ggpubr(v.0.5.0)*, *ggeffects(v.1.2.0)*, *readxl(v.1.4.1)*, *forcats(v.0.5.2)*, *stringr(v.1.5.0)*, *dplyr(v.1.0.10)*, *purrr(v.1.0.1)*, *readr(v.2.1.4)*, *tidyr(v.1.3.0)*, *tibble(v.3.1.8)*, *ggplot2(v.3.4.0)* and *tidyverse(v.1.3.2)*

loaded via a namespace (and not attached): *backports(v.1.4.1)*, *lwgeom(v.0.2-11)*, *plyr(v.1.8.8)*, *splines(v.4.2.1)*, *AlgDesign(v.1.2.1)*, *crosstalk(v.1.2.0)*, *TH.data(v.1.1-1)*, *digest(v.0.6.31)*, *htmltools(v.0.5.4)*, *leaflet.providers(v.1.9.0)*, *fansi(v.1.0.4)*, *magrittr(v.2.0.3)*, *googlesheets4(v.1.0.1)*, *cluster(v.2.1.3)*, *tzdb(v.0.3.0)*, *modelr(v.0.1.10)*, *vroom(v.1.6.1)*, *sandwich(v.3.0-2)*, *timechange(v.0.2.0)*, *colorspace(v.2.1-0)*, *rvest(v.1.0.3)*, *haven(v.2.5.1)*, *xfun(v.0.37)*, *leafem(v.0.2.0)*, *crayon(v.1.5.2)*, *jsonlite(v.1.8.4)*, *survival(v.3.3-1)*, *zoo(v.1.8-11)*, *glue(v.1.6.2)*, *stars(v.0.6-0)*, *gtable(v.0.3.1)*, *gargle(v.1.2.1)*, *questionr(v.0.7.8)*, *DEoptimR(v.1.0-11)*, *abind(v.1.4-5)*, *scales(v.1.2.1)*, *mvtnorm(v.1.1-3)*, *DBI(v.1.1.3)*, *rstatix(v.0.7.1)*, *miniUI(v.0.1.1.1)*, *Rcpp(v.1.0.9)*, *plotrix(v.3.8-2)*, *xtable(v.1.8-4)*, *viridisLite(v.0.4.1)*, *units(v.0.8-1)*, *bit(v.4.0.5)*, *proxy(v.0.4-27)*, *htmlwidgets(v.1.6.1)*, *httr(v.1.4.4)*, *RColorBrewer(v.1.1-3)*, *wk(v.0.7.1)*, *ellipsis(v.0.3.2)*, *farver(v.2.1.1)*, *pkgconfig(v.2.0.3)*, *XML(v.3.99-0.13)*, *dbplyr(v.2.3.0)*, *utf8(v.1.2.2)*, *later(v.1.3.0)*, *labeling(v.0.4.2)*, *tidyselect(v.1.2.0)*, *rlang(v.1.0.6)*, *tmaptools(v.3.1-1)*, *munsell(v.0.5.0)*, *cellranger(v.1.1.0)*, *tools(v.4.2.1)*, *cli(v.3.4.1)*, *generics(v.0.1.3)*, *broom(v.1.0.2)*, *evaluate(v.0.20)*, *fastmap(v.1.1.0)*, *yaml(v.2.3.7)*, *leafsync(v.0.1.0)*, *knitr(v.1.42)*, *bit64(v.4.0.5)*, *fs(v.1.6.0)*, *s2(v.1.1.2)*, *mime(v.0.12)*, *xml2(v.1.3.3)*, *compiler(v.4.2.1)*, *rstudioapi(v.0.14)*, *png(v.0.1-8)*, *e1071(v.1.7-12)*, *ggsignif(v.0.6.4)*, *klaR(v.1.7-1)*, *reprex(v.2.0.2)*, *stringi(v.1.7.12)*, *highr(v.0.10)*, *classInt(v.0.4-8)*, *vctrs(v.0.5.2)*, *pillar(v.1.8.1)*, *lifecycle(v.1.0.3)*, *combinat(v.0.0-8)*, *cowplot(v.1.1.1)*, *httpuv(v.1.6.8)*, *agricolae(v.1.3-5)*, *R6(v.2.5.1)*, *promises(v.1.2.0.1)*, *KernSmooth(v.2.23-20)*, *gridExtra(v.2.3)*, *codetools(v.0.2-18)*, *dichromat(v.2.0-0.1)*, *gtools(v.3.9.4)*, *assertthat(v.0.2.1)*, *withr(v.2.5.0)*, *multcomp(v.1.4-22)*, *mgcv(v.1.8-40)*, *parallel(v.4.2.1)*, *hms(v.1.1.2)*, *terra(v.1.7-3)*, *labelled(v.2.10.0)*, *class(v.7.3-20)*, *rmarkdown(v.2.20)*, *googledrive(v.2.0.0)*, *shiny(v.1.7.4)*, *lubridate(v.1.9.1)* and *base64enc(v.0.1-3)*