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

```r
# 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)
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')

table_s1 <- readxl::read_excel('Data/LT_CWD_datasheet.xlsx',
                               sheet = 'Wood Morphology')
readr::write_csv(table_s1,
                 'Data/table_s1.csv')
```

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 characeter 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')


# calculate DIN and N:P ratio
chem_sum <- chem_sum %>%
  dplyr::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    134.    193.  37.6  6.22 260.   24.8  231.   3.82
## 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      6.02  164.  52.4  5.73  18.4  24.8  216.  79.4
```

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.
```

```
cwd_coords <- streams_gps %>%
  dplyr::filter(Site %in% sites_long) %>%
```

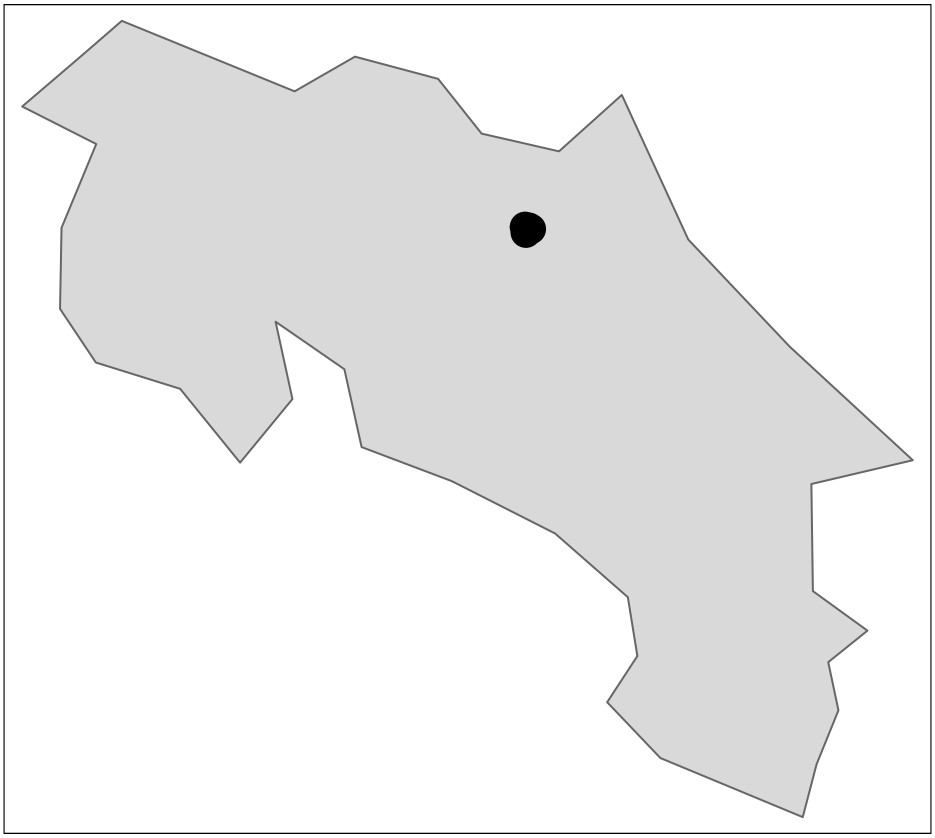
```r
  sf::st_as_sf(.,
               coords = c('Long', 'Lat'),
               crs = st_crs(4326))

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
```

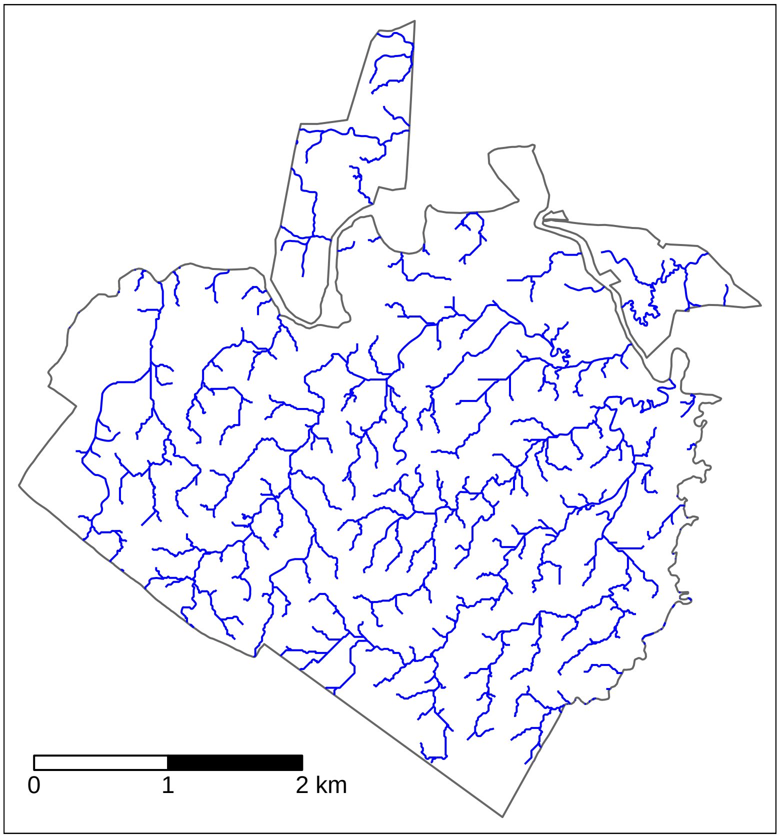
```r
# Streams at La Selva shapefile
streams <- sf::st_read(dsn = 'Data/Spatial/streamsclip.shp')
```

```
## 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
```

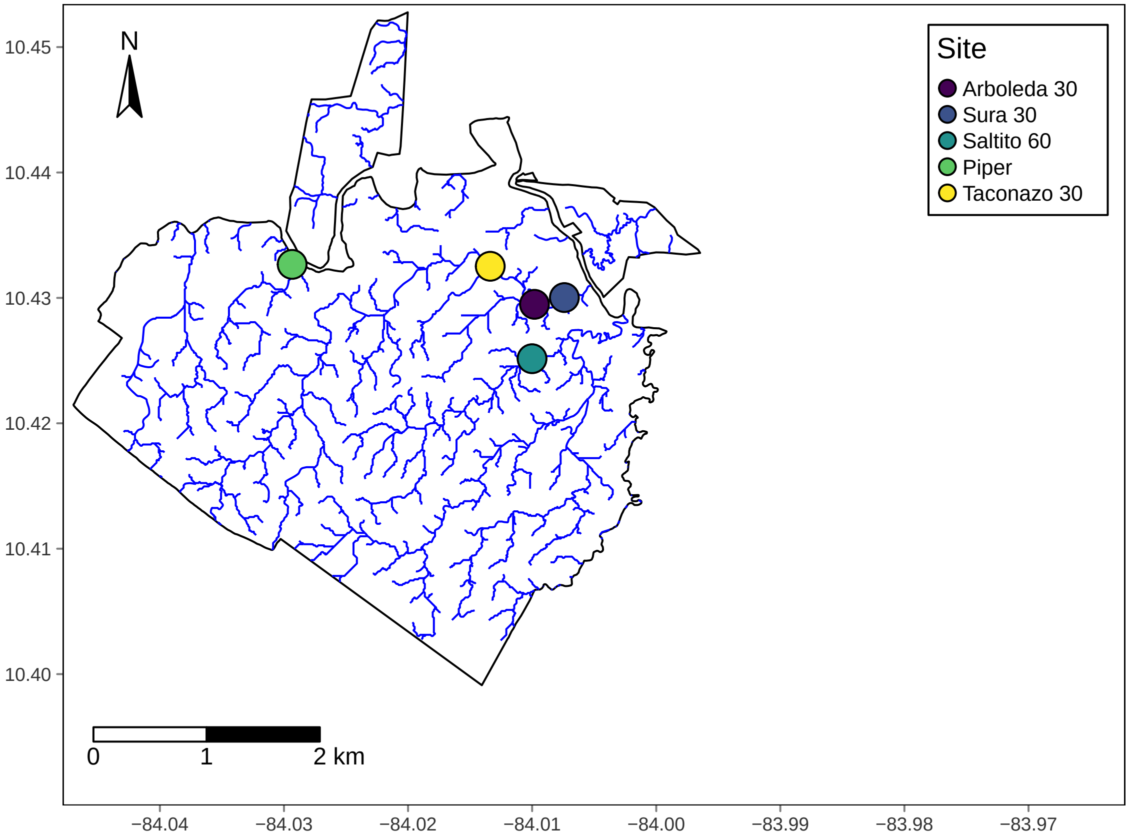
```r
cr <- world %>%
  dplyr::filter(name_long == 'Costa Rica')

# 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
```



```
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
```



```r
tmap_save(tm = fig1,
          filename = 'Figures/fig1.png',
          dpi = 600,
          width = 4, height = 4)
```

```
## Map saved to C:\Users\Nick Marzolf\Desktop\NCSU\STREAMS\Projects\Long-term Wood decomp\LTCWD\Figures
## Resolution: 2400 by 2400 pixels
## Size: 4 by 4 inches (600 dpi)
```

Calculate wood decomposition rates

```r
# 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)
```

```r
# 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
```

```r
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
```

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

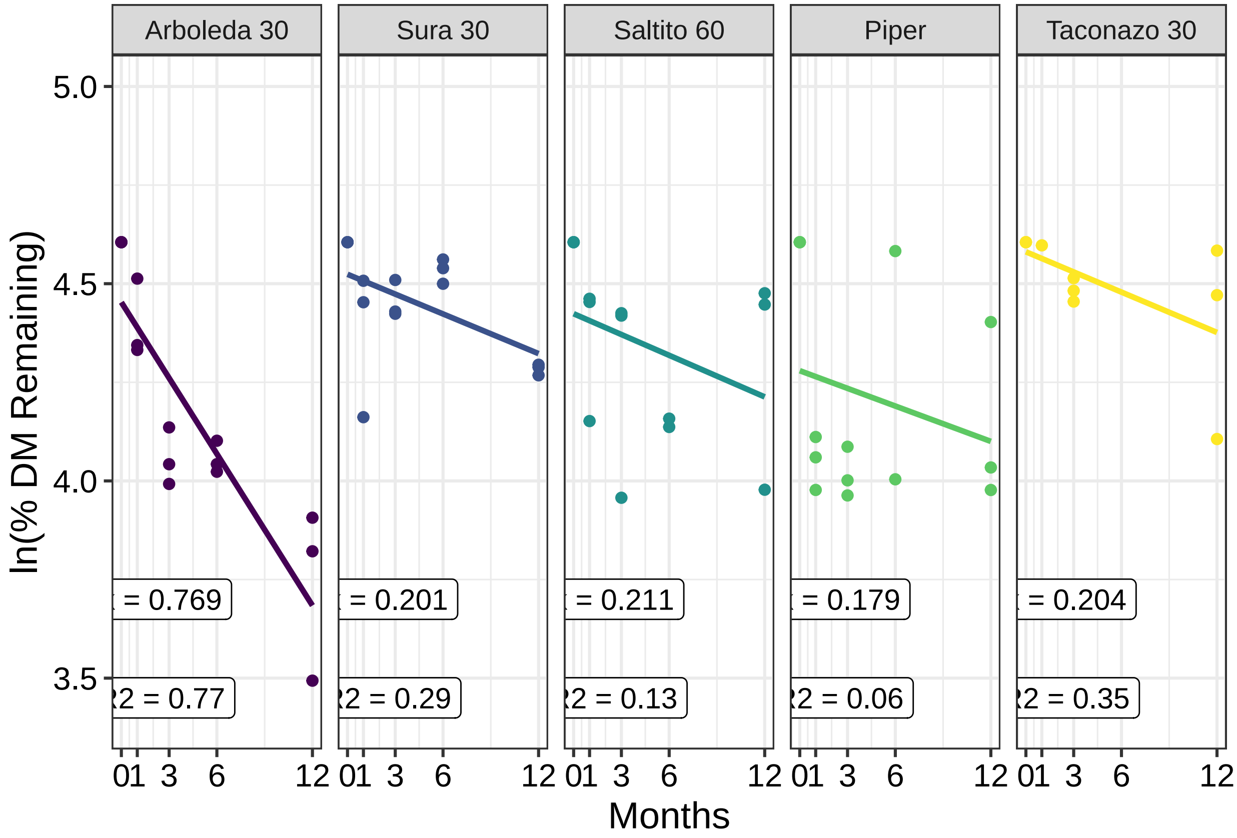
```
## Anova Table (Type III tests)
##
## Response: log(percent_mass)
##              Sum Sq Df   F value    Pr(>F)
## (Intercept) 145.894  1 3915.9413 < 2.2e-16 ***
## month         1.148  1   30.8088 7.408e-07 ***
## site          0.348  4    2.3354   0.06609 .
## month:site    0.503  4    3.3721 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

```
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 = round(summary(lm(log(percent_mass) ~ month))$r.squared, 2),
    p = anova(lm(log(percent_mass) ~ month))$'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  0.77 0.0000151
## 2 Sur30   4.52 0.201 0.0447  0.29 0.0379
## 3 Tito60  4.42 0.211 0.0818  0.13 0.209
## 4 Piper   4.28 0.179 0.103   0.06 0.386
## 5 Tac     4.58 0.204 0.0561  0.35 0.0735
```

```
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)+
  lims(y = c(3.4, 5))+
  geom_label(data = table2 %>%
               dplyr::select(site, k_yr),
             aes(x = 2.5, y = 3.7,
                 label = paste0('k = ',k_yr)))+
  geom_label(data = table2 %>%
               dplyr::select(site, r2),
             aes(x = 2.5, y = 3.45,
                 label = paste0('R2 = ',r2)))+
  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'
```



```
ggsave(plot = fig2,
       'Figures/fig2.png',
       dpi = 600,
       width = 10,height = 3)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
cwd_calc %>%
  group_by(site, month) %>%
  summarise(mean_per_mass = mean(percent_mass, na.rm = TRUE))
```

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

```
## # A tibble: 24 x 3
## # Groups:   site [5]
##    site  month mean_per_mass
##    <fct> <dbl>         <dbl>
##  1 Arb       0         100
##  2 Arb       1          81.4
##  3 Arb       3          57.9
##  4 Arb       6          57.8
##  5 Arb      12          42.8
##  6 Sur30     0         100
##  7 Sur30     1          80.2
##  8 Sur30     3          86.1
##  9 Sur30     6          93.1
## 10 Sur30    12          72.5
## # i 14 more rows
```

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])
}
```

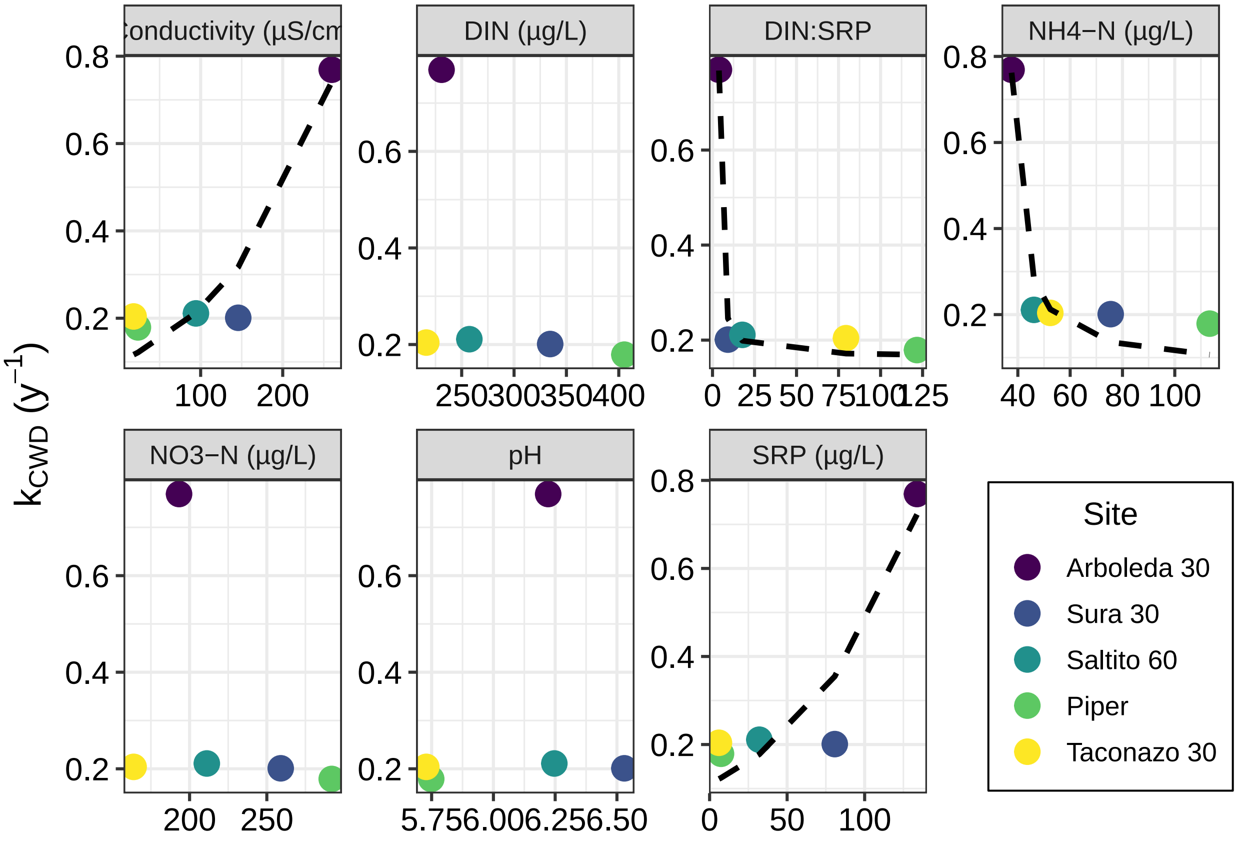
Combine and evaluate model fits

```
# 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          0.659  0.133 0.00508  0.210  0.797  0.114 0.0000622
## 2 Logarithmic       0.0533 0.298 0.00265  0.397  0.0263 0.307 0.0000983
## 3 Linear            0.288  0.316 0.00175  0.392  0.177  0.305 0.0000349
## 4 Michaelis-Menten NA      0.253 0.991    NA     NA      0.274 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)
}

fig3 <- shift_legend2(fig3)
```



```
ggsave(plot = fig3,
       'Figures/fig3.png',
       dpi = 600,
       width = 11,height = 5)
```

Macroinvertebrate analysis

```
data <- readxl::read_excel('Data/Samples_Nick_ana_2024.xlsx',
                           sheet = 'Data')

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

# clean the data
tax_data_clean <- data %>%
  dplyr::mutate(month = as.numeric(gsub(".*?([0-9]+).*", "\\1", Sample)),
                rep = gsub("\\d+", "", Sample)) %>%
  dplyr::rename(ffg = `Functional group`) %>%
  dplyr::filter(QAQC == 0,
                month != 24)

fig_s7 <- tax_data_clean %>%
  filter(!is.na(ffg)) %>%
  group_by(ffg) %>%
  summarise(total_ffg = sum(Total, na.rm = TRUE)) %>%
  ggplot(.,
         aes(x = ffg,
             y = total_ffg))+
  geom_bar(stat = 'identity')+
  geom_label(aes(label = total_ffg))+
  scale_y_log10()+
  labs(x = element_blank(),
       y = 'Total Abundance')+
  theme(axis.text.x = element_text(size = 8))

ggsave(plot = fig_s7,
       'Figures/fig_s7.png',
       dpi = 600,
       height = 4, width = 9)

tot_macros <- sum(tax_data_clean$Total)
threshold <- tot_macros*0.01

# main FFGS: at least 1% of all FFGs identified
main_ffgs <- tax_data_clean %>%
  dplyr::group_by(ffg) %>%
  dplyr::summarise(n = sum(Total)) %>%
  dplyr::filter(n > threshold) %>%
  dplyr::pull(ffg)
```

```
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.
```

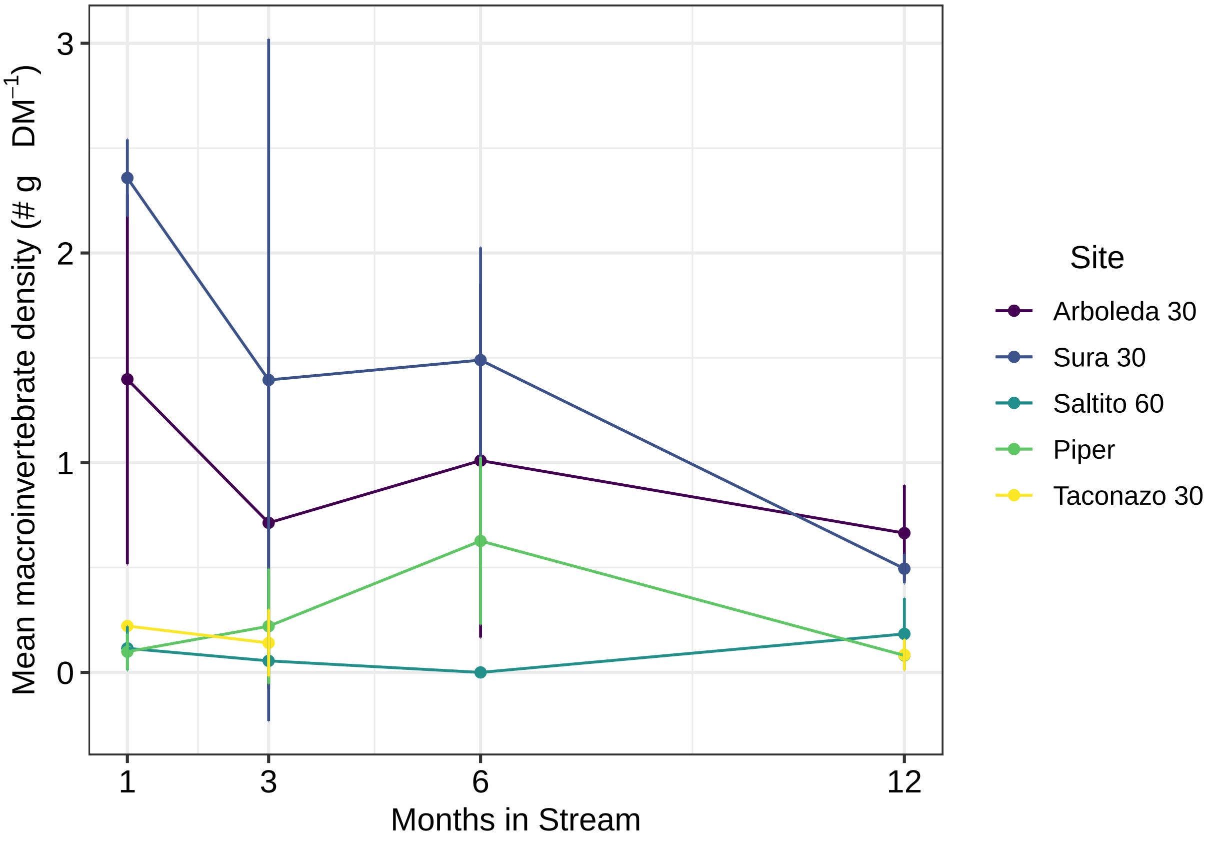
```
final_dry_mass$site <- dplyr::recode_factor(final_dry_mass$site,
                                            Arb = 'Arboleda 30',
                                            Sur30 = 'Sura 30',
                                            Tito60 = 'Saltito 60',
                                    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



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

```
fig4 <- 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 = 'Months in Stream')+
  scale_color_viridis_d(name = 'Site')+
  theme(axis.title = element_text(size = 12))
fig4
```



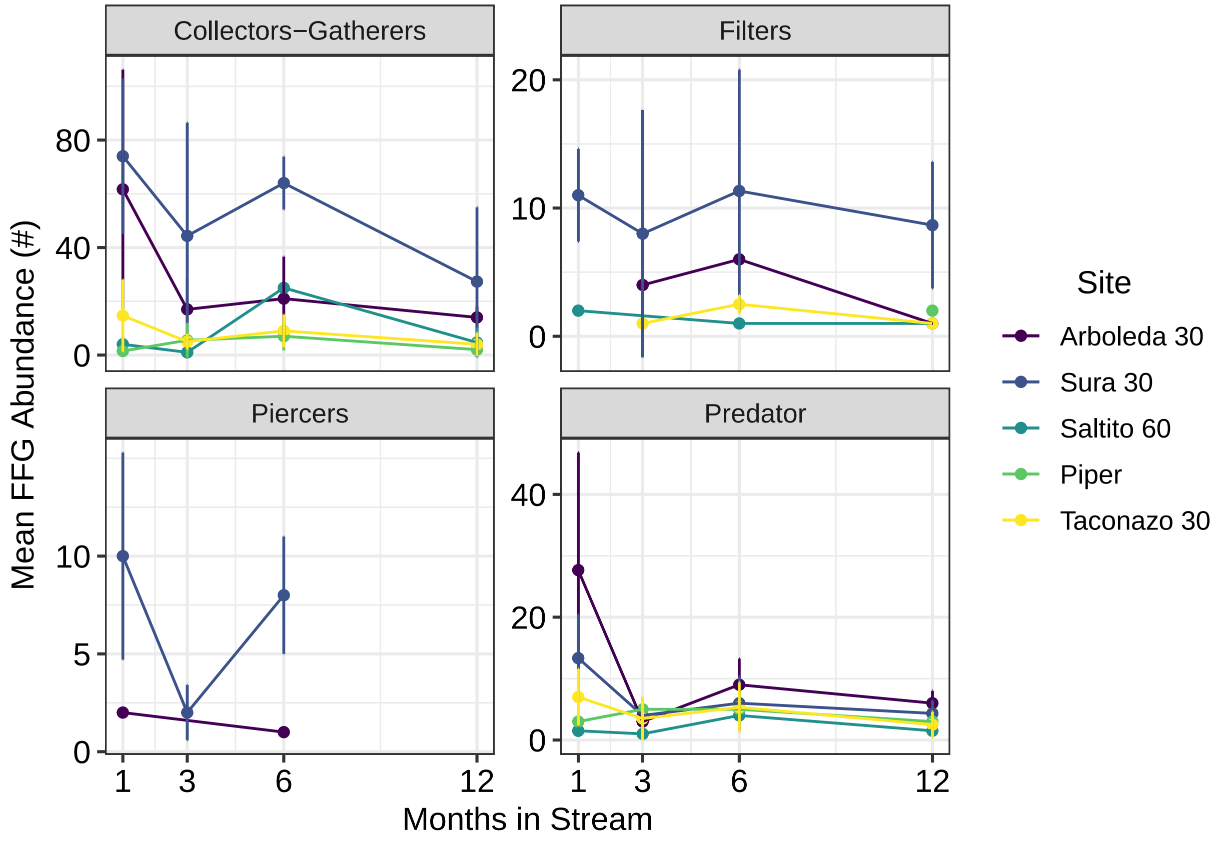
```
macro_density_ffg <- tax_data_clean %>%
  dplyr::filter(!is.na(order)) %>%                          # remove NAs from Order
  dplyr::select(site = Stream, ffg, month, rep, Total) %>%     # get the necessary columns
  dplyr::group_by(site, month, ffg, rep) %>%                     # and do the grouping
  dplyr::summarise(total_family = sum(Total))
```

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

```
macro_density_sum_ffg <- macro_density_ffg %>%
  group_by(site, month, ffg) %>%
  summarise(mean_abund = mean(total_family, na.rm = TRUE),
            se_abund = sd(total_family, na.rm = TRUE)/length(mean_abund))
```

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

```
fig5 <- ggplot(macro_density_sum_ffg %>%
                   dplyr::filter(!is.na(ffg),
                                 ffg %in% main_ffgs),
               aes(x = month,
                   y = mean_abund,
                   color = site))+
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin = mean_abund - se_abund,
                    ymax = mean_abund + se_abund),
                width = 0)+
  # geom_boxplot()+
  facet_wrap(ffg ~ .,
             scales = 'free_y')+
  scale_color_viridis_d(name = 'Site')+
  scale_x_continuous(breaks = c(1, 3, 6, 12))+
  scale_y_continuous(breaks = function(y) {
    max_value <- max(y, na.rm = TRUE)
    tick_marks <- seq(0, round(max_value*.75, -1), length.out = 3)
    return(tick_marks)
  })+
  labs(y = expression(paste('Mean FFG Abundance (#)')),
       x = 'Months in Stream')+
  theme(axis.title = element_text(size = 12))
fig5
```



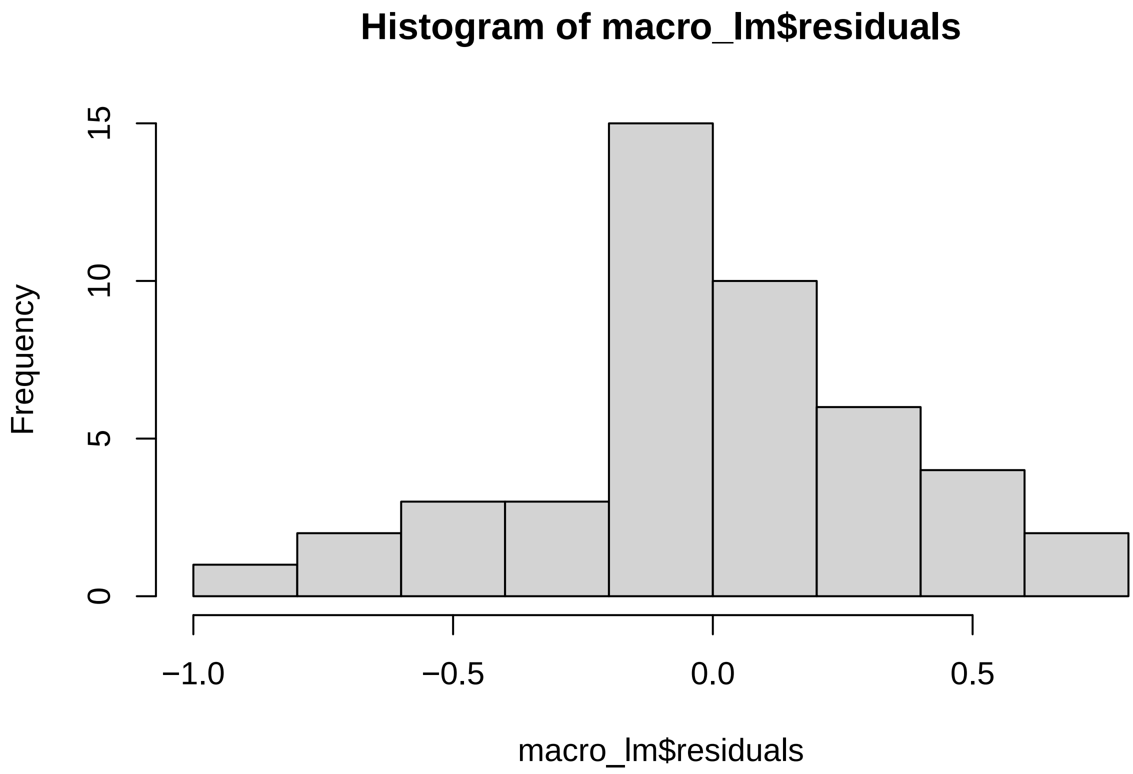
```r
# fig4_b <- shift_legend2(fig4_b)

ggsave(plot = fig5,
       'Figures/fig5.png',
       dpi = 600,
       width = 6, height = 4)

macro_density_mod <- macro_density %>%
  dplyr::mutate(log_abund = log10(abund_per_dm)) %>%
  dplyr::filter(is.finite(log_abund))

macro_lm <- lm(data = macro_density_mod %>%
                 dplyr::filter(is.finite(abund_per_dm)),
               log_abund ~ factor(month) * site)

summary(macro_lm)
```

```
##
## Call:
## lm(formula = log_abund ~ factor(month) * site, data = macro_density_mod %>%
##     dplyr::filter(is.finite(abund_per_dm)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.89963 -0.13689 -0.00345  0.20023  0.62367
##
## Coefficients: (2 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      0.093209   0.263905   0.353  0.72659
## factor(month)3                  -0.467030   0.373218  -1.251  0.22116
## factor(month)6                  -0.376372   0.373218  -1.008  0.32187
## factor(month)12                 -0.287686   0.373218  -0.771  0.44727
## siteSura 30                      0.278458   0.373218   0.746  0.46182
## siteSaltito 60                  -0.864065   0.417270  -2.071  0.04771 *
## sitePiper                       -1.200646   0.417270  -2.877  0.00759 **
## siteTaconazo 30                 -0.748172   0.527809  -1.418  0.16737
## factor(month)3:siteSura 30       0.002746   0.527809   0.005  0.99589
## factor(month)6:siteSura 30       0.159771   0.527809   0.303  0.76435
## factor(month)12:siteSura 30     -0.392632   0.527809  -0.744  0.46314
## factor(month)3:siteSaltito 60    0.123994   0.590109   0.210  0.83510
## factor(month)6:siteSaltito 60          NA         NA      NA       NA
## factor(month)12:siteSaltito 60   0.198692   0.590109   0.337  0.73885
## factor(month)3:sitePiper         0.593267   0.590109   1.005  0.32334
## factor(month)6:sitePiper         1.231915   0.590109   2.088  0.04605 *
## factor(month)12:sitePiper        0.297927   0.559826   0.532  0.59880
## factor(month)3:siteTaconazo 30   0.004342   0.646432   0.007  0.99469
## factor(month)6:siteTaconazo 30         NA         NA      NA       NA
## factor(month)12:siteTaconazo 30 -0.372137   0.646432  -0.576  0.56943
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.4571 on 28 degrees of freedom
## Multiple R-squared:  0.6796, Adjusted R-squared:  0.4851
## F-statistic: 3.494 on 17 and 28 DF,  p-value: 0.001663
```

```
anova(macro_lm)
```

```
## Analysis of Variance Table
##
## Response: log_abund
##                    Df Sum Sq Mean Sq F value    Pr(>F)
## factor(month)       3 3.0854 1.02846  4.9223  0.007177 **
## site                4 7.8834 1.97085  9.4327 5.854e-05 ***
## factor(month):site 10 1.4418 0.14418  0.6901  0.724905
## Residuals          28 5.8502 0.20894
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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



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

```
## $statistics
##     MSerror Df       Mean        CV
##   0.2089371 28 -0.5067934 -90.19384
##
## $parameters
##    test name.t ntr StudentizedRange alpha
##   Tukey   site   5           4.1203  0.05
##
## $means
##                log_abund       std  r        se        Min         Max
## Arboleda 30 -0.18956334 0.4697270 12 0.1319524 -1.1827949  0.38288383
## Piper       -0.88585069 0.5085039  9 0.1523655 -1.5797836 -0.04139269
## Saltito 60  -0.91486584 0.2944634  6 0.1866088 -1.2833012 -0.51648452
## Sura 30      0.03136597 0.3699442 12 0.1319524 -0.6345892  0.51224243
## Taconazo 30 -1.13603949 0.5886647  7 0.1727662 -2.0813473 -0.49398056
##                    Q25         Q50        Q75
## Arboleda 30 -0.3458852 -0.05760492  0.1306942
## Piper       -1.1949766 -1.02015403 -0.4623980
## Saltito 60  -1.1385314 -0.89888760 -0.7296380
## Sura 30     -0.2647593  0.07216143  0.3516841
## Taconazo 30 -1.4294868 -1.05744874 -0.7302632
##
## $comparison
## NULL
##
## $groups
##                log_abund groups
## Sura 30      0.03136597      a
## Arboleda 30 -0.18956334      a
## Piper       -0.88585069      b
## Saltito 60  -0.91486584      b
## Taconazo 30 -1.13603949      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) %>%
  tidyr::pivot_wider(names_from = Family,                         # pivot data
                     values_from = log_total,
                     values_fill = 0)
```

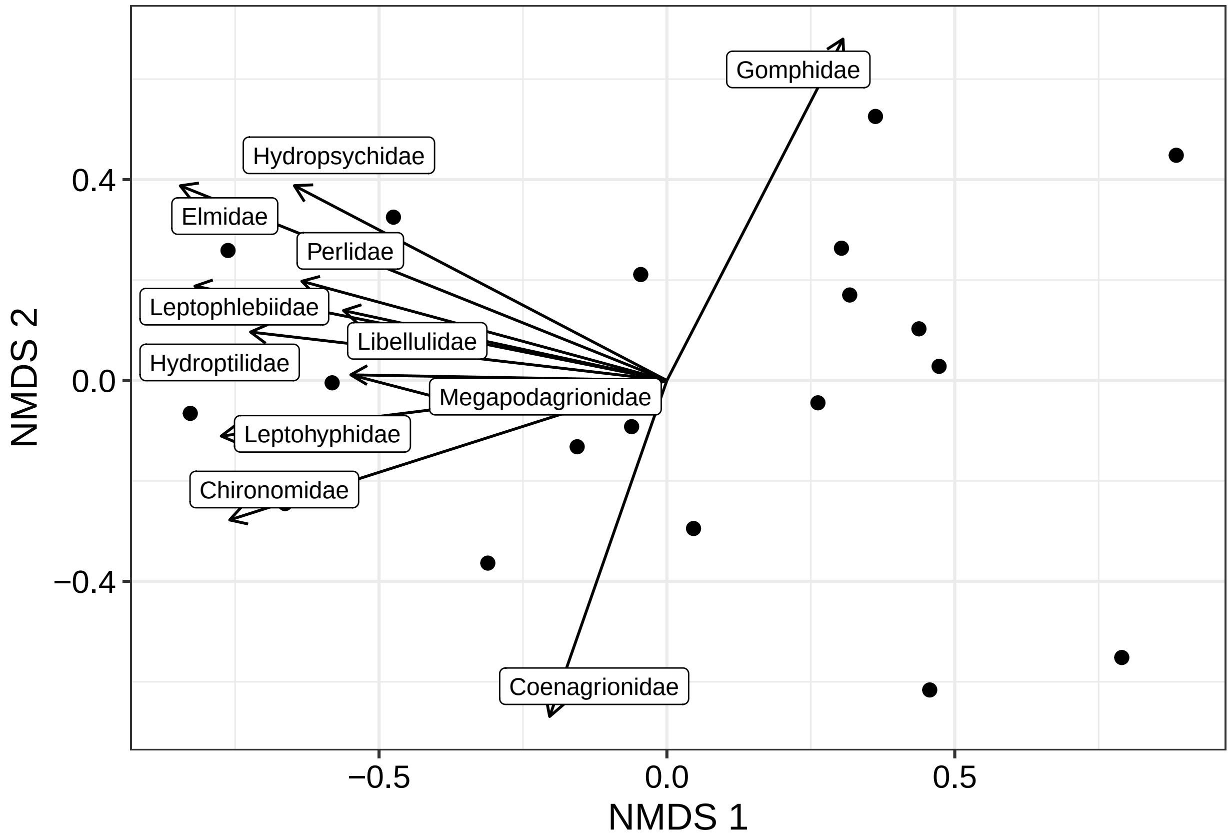
```
## ‘summarise()‘ has grouped output by ’Stream’, ’month’. You can override using
## the ‘.groups‘ argument.
```

```
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)
```

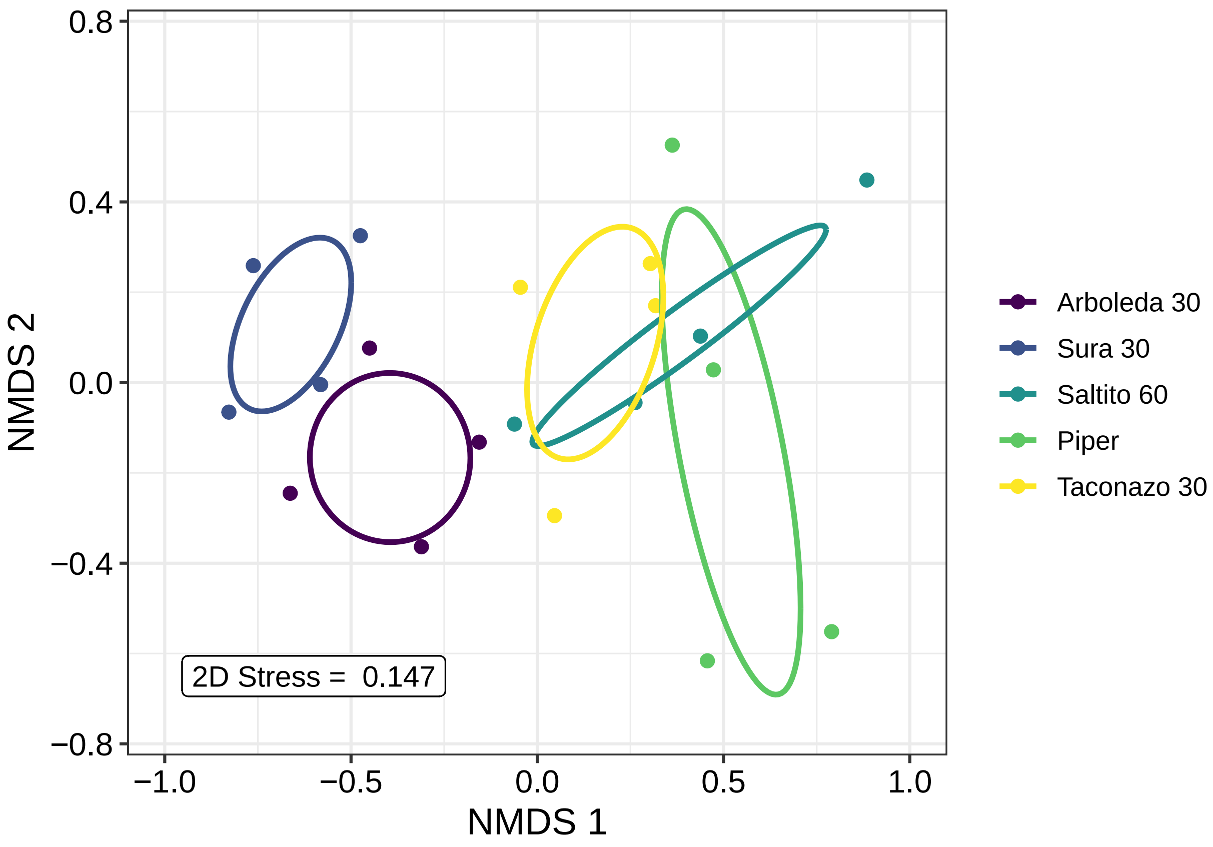
```
## Run 0 stress 0.1466003
## Run 1 stress 0.1540781
## Run 2 stress 0.1513751
## Run 3 stress 0.173021
## Run 4 stress 0.1513751
## Run 5 stress 0.1466002
## ... New best solution
## ... Procrustes: rmse 0.000117617  max resid 0.0004263929
## ... Similar to previous best
## Run 6 stress 0.1466582
## ... Procrustes: rmse 0.05843379  max resid 0.2110857
## Run 7 stress 0.1566809
## Run 8 stress 0.1466008
## ... Procrustes: rmse 0.0005925832  max resid 0.002149753
## ... Similar to previous best
## Run 9 stress 0.1650556
## Run 10 stress 0.1537933
## Run 11 stress 0.160739
## Run 12 stress 0.1466582
## ... Procrustes: rmse 0.05837108  max resid 0.2110195
## Run 13 stress 0.1513751
## Run 14 stress 0.146601
## ... Procrustes: rmse 0.0007554404  max resid 0.002733676
## ... Similar to previous best
## Run 15 stress 0.1567762
## Run 16 stress 0.155067
## Run 17 stress 0.1513752
## Run 18 stress 0.1534708
## Run 19 stress 0.160678
## Run 20 stress 0.1576684
## *** Best solution repeated 3 times
```

```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)
```

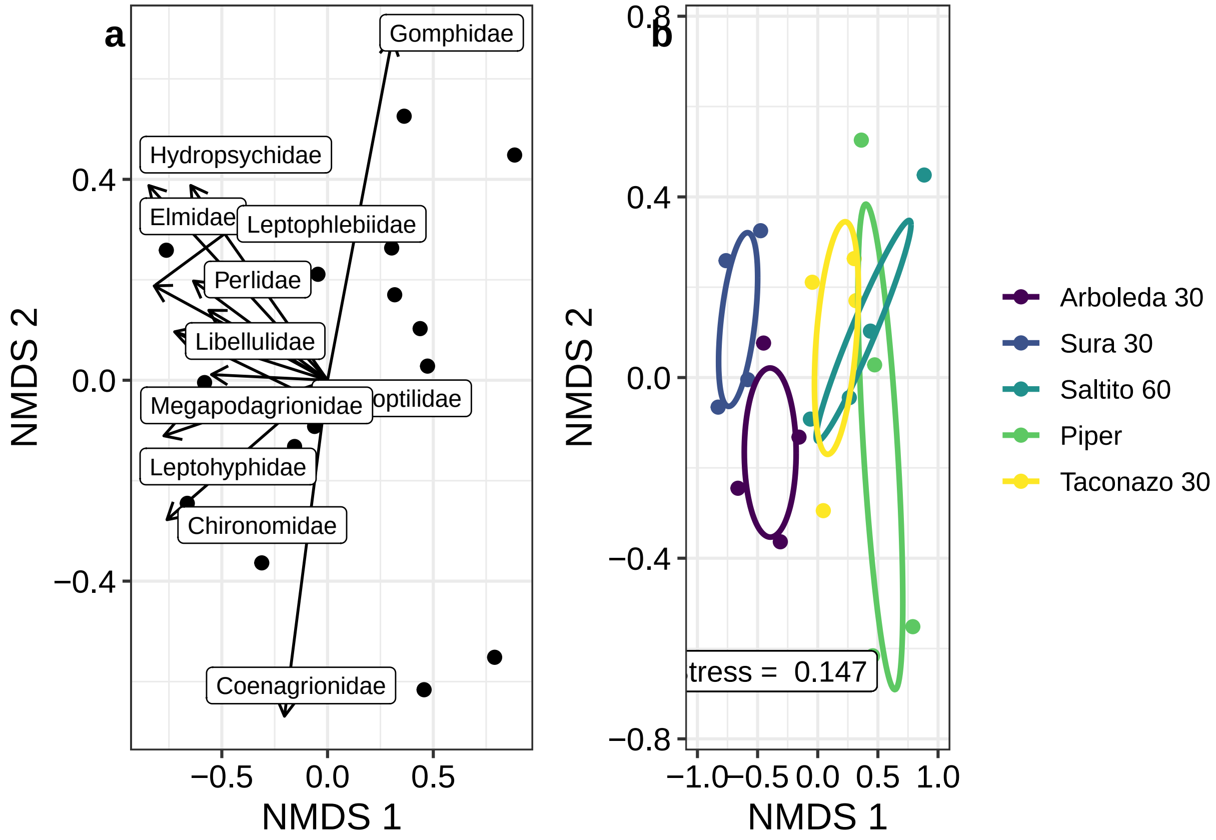
```
plot_nmds_family_scrs <- ggplot(nmds_family_out,
                                aes(x = x, y = y))+
  geom_point(size = 2)+
  geom_segment(data = scrs_fam_sig,
               aes(x = 0, xend = NMDS1,
                   y = 0, yend = NMDS2),
               arrow = arrow(length = unit(0.25, 'cm')),
               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))+
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())+
  geom_label(label = paste('2D Stress = ', stress_fam_log_2d),
             x = -0.6,
             y = -0.65)
plot_nmds_streams
```



```
fig6 <- ggpubr::ggarrange(plot_nmds_family_scrs,
                          plot_nmds_streams,
                          align = 'h',
                          widths = c(1,1.25),
                          labels = 'auto',
                          label.x = c(0.18,0.13),
                          label.y = 0.98)
fig6
```



```
ggsave(plot = fig6,
       'Figures/fig6.png',
       dpi = 600,
       width = 11.5,height = 4)

# 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, perm
##               Df SumOfSqs      R2      F Pr(>F)
## Stream         4  1.56809 0.52929 4.1208 0.0001 ***
## month          1  0.13925 0.04700 1.4637 0.1873
## Stream:month   4  0.30394 0.10259 0.7987 0.7161
## Residual      10  0.95133 0.32111
## Total        19  2.96261 1.00000
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

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

summary(sim)
```

```
##
## Contrast: Arboleda 30_Sura 30
##
##                   average      sd   ratio     ava     avb cumsum     p
## Hydroptilidae     0.04465 0.03069 1.45510 0.19450 0.61820  0.104 0.109
## Perlidae          0.03581 0.02526 1.41810 0.07530 0.45040  0.187 0.081 .
## Hydropsychidae    0.03418 0.01836 1.86130 0.34510 0.74720  0.267 0.924
## Leptohyphidae     0.02984 0.01978 1.50890 0.50550 0.72550  0.336 0.992
## Chironomidae      0.02956 0.01523 1.94090 0.77320 0.87940  0.405 0.985
## Leptophlebiidae   0.02352 0.01574 1.49400 0.47580 0.67660  0.460 0.993
## Ceratopogonidae   0.02224 0.01647 1.35050 0.22580 0.28650  0.511 0.904
## Caenidae          0.02172 0.01577 1.37770 0.28650 0.07530  0.562 0.538
## Polycentropodidae 0.02159 0.02297 0.94020 0.00000 0.23860  0.612 0.059 .
## Elmidae           0.02049 0.01095 1.87160 0.83920 1.03930  0.660 1.000
## Libellulidae      0.02037 0.02188 0.93080 0.07530 0.25000  0.707 0.146
## Simuliidae        0.01795 0.01907 0.94160 0.00000 0.20600  0.749 0.067 .
## Baetidae          0.01705 0.01981 0.86070 0.19450 0.00000  0.789 0.338
## Coenagrionidae    0.01705 0.01409 1.20960 0.22580 0.07530  0.828 0.747
## Gyrinidae         0.01450 0.01517 0.95540 0.15050 0.00000  0.862 0.137
## Megapodagrionidae 0.01384 0.01469 0.94250 0.07530 0.15050  0.894 0.187
## Empididae         0.01344 0.01409 0.95390 0.00000 0.15050  0.925 0.463
## Scirtidae         0.00983 0.01342 0.73250 0.07530 0.07530  0.948 0.133
## Leptoceridae      0.00775 0.01398 0.55430 0.00000 0.07530  0.967 0.619
## Calamoceratidae   0.00723 0.01306 0.55410 0.07530 0.00000  0.983 0.412
## Planariidae       0.00719 0.01295 0.55480 0.00000 0.07530  1.000 0.364
## Gomphidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.942
## Gordiidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.659
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Contrast: Arboleda 30_Saltito 60
##
##                   average      sd   ratio     ava     avb cumsum     p
## Elmidae           0.07900 0.03646 2.16650 0.83920 0.36930  0.140 0.135
## Leptophlebiidae   0.07653 0.03517 2.17620 0.47580 0.00000  0.275 0.034 *
## Chironomidae      0.05650 0.02820 2.00340 0.77320 0.51340  0.375 0.253
## Caenidae          0.04161 0.02883 1.44320 0.28650 0.11930  0.449 0.040 *
## Leptohyphidae     0.04114 0.03879 1.06060 0.50550 0.22580  0.521 0.835
## Hydropsychidae    0.03934 0.03466 1.13510 0.34510 0.22580  0.591 0.837
## Coenagrionidae    0.03661 0.02385 1.53460 0.22580 0.00000  0.656 0.055 .
## Ceratopogonidae   0.03354 0.02877 1.16590 0.22580 0.15050  0.715 0.568
## Baetidae          0.03076 0.03740 0.82260 0.19450 0.00000  0.770 0.004 **
## Gyrinidae         0.02775 0.02900 0.95720 0.15050 0.00000  0.819 0.004 **
## Hydroptilidae     0.02695 0.02808 0.95990 0.19450 0.00000  0.866 0.596
## Gomphidae         0.02628 0.02819 0.93220 0.00000 0.15050  0.913 0.379
## Calamoceratidae   0.01383 0.02492 0.55480 0.07530 0.00000  0.937 0.016 *
## Libellulidae      0.00885 0.01588 0.55730 0.07530 0.00000  0.953 0.522
## Megapodagrionidae 0.00885 0.01588 0.55730 0.07530 0.00000  0.969 0.468
## Perlidae          0.00885 0.01588 0.55730 0.07530 0.00000  0.984 0.904
## Scirtidae         0.00885 0.01588 0.55730 0.07530 0.00000  1.000 0.261
## Polycentropodidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.658
## Simuliidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.658
## Empididae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.795
## Planariidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.423
## Leptoceridae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.633
## Gordiidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.697
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Contrast: Arboleda 30_Piper
##
##                   average      sd   ratio     ava     avb cumsum     p
## Elmidae           0.11845 0.04991 2.37350 0.83920 0.15050  0.179 0.001 ***
## Leptophlebiidae   0.06748 0.04382 1.53990 0.47580 0.07530  0.282 0.089 .
## Leptohyphidae     0.06456 0.03297 1.95820 0.50550 0.07530  0.379 0.082 .
## Hydropsychidae    0.05495 0.04068 1.35080 0.34510 0.07530  0.462 0.225
## Chironomidae      0.04841 0.03522 1.37470 0.77320 0.46110  0.536 0.557
## Caenidae          0.04474 0.02722 1.64350 0.28650 0.00000  0.604 0.017 *
## Ceratopogonidae   0.04351 0.02789 1.55980 0.22580 0.36930  0.669 0.259
## Baetidae          0.03148 0.03827 0.82270 0.19450 0.00000  0.717 0.001 ***
## Coenagrionidae    0.03107 0.02624 1.18370 0.22580 0.07530  0.764 0.145
## Gyrinidae         0.02848 0.02966 0.96010 0.15050 0.00000  0.807 0.003 **
## Hydroptilidae     0.02750 0.02860 0.96170 0.19450 0.00000  0.849 0.570
## Leptoceridae      0.02518 0.04572 0.55080 0.00000 0.17470  0.887 0.390
## Calamoceratidae   0.01418 0.02552 0.55590 0.07530 0.00000  0.909 0.007 **
## Empididae         0.01351 0.02468 0.54750 0.00000 0.07530  0.929 0.452
## Gomphidae         0.01084 0.01969 0.55080 0.00000 0.07530  0.946 0.915
## Libellulidae      0.00900 0.01614 0.55760 0.07530 0.00000  0.959 0.491
## Megapodagrionidae 0.00900 0.01614 0.55760 0.07530 0.00000  0.973 0.446
## Perlidae          0.00900 0.01614 0.55760 0.07530 0.00000  0.986 0.906
## Scirtidae         0.00900 0.01614 0.55760 0.07530 0.00000  1.000 0.228
## Polycentropodidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.637
## Simuliidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.637
## Planariidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.407
## Gordiidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.662
## ---
## 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     0.06749 0.02101 3.21300 0.50550 0.00000  0.130 0.054 .
## Elmidae           0.05222 0.02824 1.84900 0.83920 0.47970  0.230 0.747
## Leptophlebiidae   0.05000 0.03250 1.53800 0.47580 0.28260  0.326 0.582
## Ceratopogonidae   0.04086 0.03070 1.33100 0.22580 0.40050  0.404 0.321
## Caenidae          0.03864 0.02328 1.66000 0.28650 0.00000  0.478 0.073 .
## Chironomidae      0.03675 0.02222 1.65400 0.77320 0.63870  0.549 0.902
## Hydropsychidae    0.03294 0.02603 1.26600 0.34510 0.25700  0.612 0.943
## Gordiidae         0.02882 0.03161 0.91200 0.00000 0.19450  0.667 0.115
## Baetidae          0.02704 0.03223 0.83900 0.19450 0.00000  0.719 0.022 *
## Coenagrionidae    0.02688 0.02235 1.20200 0.22580 0.07530  0.771 0.253
## Hydroptilidae     0.02408 0.02501 0.96300 0.19450 0.00000  0.817 0.670
## Gyrinidae         0.02402 0.02488 0.96600 0.15050 0.00000  0.863 0.026 *
## Gomphidae         0.02096 0.02215 0.94700 0.00000 0.15050  0.903 0.604
## Perlidae          0.01431 0.01939 0.73800 0.07530 0.07530  0.931 0.812
## Calamoceratidae   0.01197 0.02146 0.55800 0.07530 0.00000  0.954 0.127
## Libellulidae      0.00807 0.01444 0.55900 0.07530 0.00000  0.969 0.591
## Megapodagrionidae 0.00807 0.01444 0.55900 0.07530 0.00000  0.985 0.555
## Scirtidae         0.00807 0.01444 0.55900 0.07530 0.00000  1.000 0.435
## Polycentropodidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.647
## Simuliidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.647
## Empididae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.794
## Planariidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.396
## 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   0.07984 0.01426 5.60000 0.67660 0.00000  0.120 0.030 *
## Elmidae           0.07939 0.01541 5.15000 1.03930 0.36930  0.240 0.145
## Hydroptilidae     0.06702 0.04391 1.52600 0.61820 0.00000  0.341 0.004 **
## Hydropsychidae    0.06303 0.02445 2.57700 0.74720 0.22580  0.436 0.073 .
## Leptohyphidae     0.06076 0.03319 1.83100 0.72550 0.22580  0.527 0.137
## Perlidae          0.05012 0.03376 1.48400 0.45040 0.00000  0.603 0.009 **
## Chironomidae      0.04977 0.03313 1.50200 0.87940 0.51340  0.678 0.471
## Polycentropodidae 0.02897 0.03106 0.93300 0.23860 0.00000  0.721 0.003 **
## Libellulidae      0.02548 0.02969 0.85800 0.25000 0.00000  0.760 0.017 *
## Ceratopogonidae   0.02425 0.02136 1.13500 0.28650 0.15050  0.796 0.865
## Simuliidae        0.02380 0.02490 0.95600 0.20600 0.00000  0.832 0.003 **
## Megapodagrionidae 0.01890 0.02002 0.94400 0.15050 0.00000  0.861 0.019 *
## Gomphidae         0.01885 0.02000 0.94300 0.00000 0.15050  0.889 0.703
## Empididae         0.01784 0.01867 0.95600 0.15050 0.00000  0.916 0.347
## Caenidae          0.01745 0.02260 0.77200 0.07530 0.11930  0.942 0.680
## Leptoceridae      0.01079 0.01940 0.55600 0.07530 0.00000  0.959 0.411
## Scirtidae         0.00974 0.01748 0.55700 0.07530 0.00000  0.973 0.157
## Planariidae       0.00974 0.01748 0.55700 0.07530 0.00000  0.988 0.016 *
## Coenagrionidae    0.00810 0.01453 0.55800 0.07530 0.00000  1.000 0.924
## Baetidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.653
## Gyrinidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.666
## Calamoceratidae   0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.400
## Gordiidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.631
## ---
## 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 0.004 **
## Hydropsychidae    0.08135 0.02132 3.81600 0.74720 0.07530  0.262 0.005 **
## Leptohyphidae     0.07875 0.03093 2.54600 0.72550 0.07530  0.370 0.010 **
## Leptophlebiidae   0.07303 0.02391 3.05400 0.67660 0.07530  0.471 0.054 .
## 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 0.395
## Perlidae          0.05094 0.03427 1.48600 0.45040 0.00000  0.708 0.005 **
## Polycentropodidae 0.02950 0.03161 0.93300 0.23860 0.00000  0.749 0.002 **
## Libellulidae      0.02586 0.03011 0.85900 0.25000 0.00000  0.785 0.010 **
## Leptoceridae      0.02524 0.03090 0.81700 0.07530 0.17470  0.820 0.397
## Simuliidae        0.02421 0.02530 0.95700 0.20600 0.00000  0.853 0.002 **
## 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 0.947
## Empididae         0.01828 0.01927 0.94800 0.15050 0.07530  0.931 0.330
## Coenagrionidae    0.01355 0.01844 0.73500 0.07530 0.07530  0.950 0.829
## Scirtidae         0.00992 0.01779 0.55700 0.07530 0.00000  0.964 0.163
## 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 0.899
## Gomphidae         0.00820 0.01484 0.55300 0.00000 0.07530  1.000 0.958
## Baetidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.630
## Gyrinidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.644
## Calamoceratidae   0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.357
## Gordiidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.662
## ---
## 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     0.07887 0.02362 3.33900 0.72550 0.00000  0.135 0.009 **
## Hydroptilidae     0.06147 0.04031 1.52500 0.61820 0.00000  0.240 0.010 **
## Elmidae           0.05963 0.01146 5.20300 1.03930 0.47970  0.341 0.609
## Hydropsychidae    0.05319 0.02071 2.56800 0.74720 0.25700  0.432 0.282
## Leptophlebiidae   0.04507 0.03241 1.39100 0.67660 0.28260  0.509 0.746
## Perlidae          0.04262 0.02874 1.48300 0.45040 0.07530  0.582 0.021 *
## Chironomidae      0.03568 0.02244 1.59000 0.87940 0.63870  0.643 0.912
## Polycentropodidae 0.02625 0.02788 0.94200 0.23860 0.00000  0.688 0.026 *
## Libellulidae      0.02351 0.02738 0.85900 0.25000 0.00000  0.728 0.085 .
## Simuliidae        0.02166 0.02267 0.95600 0.20600 0.00000  0.765 0.026 *
## Gordiidae         0.02165 0.02368 0.91400 0.00000 0.19450  0.802 0.280
## Ceratopogonidae   0.02025 0.01831 1.10600 0.28650 0.40050  0.836 0.946
## Megapodagrionidae 0.01709 0.01796 0.95100 0.15050 0.00000  0.865 0.063 .
## Empididae         0.01624 0.01691 0.96000 0.15050 0.00000  0.893 0.361
## Gomphidae         0.01598 0.01679 0.95200 0.00000 0.15050  0.920 0.738
## Coenagrionidae    0.01193 0.01610 0.74100 0.07530 0.07530  0.941 0.875
## Leptoceridae      0.00964 0.01727 0.55800 0.07530 0.00000  0.957 0.476
## Scirtidae         0.00879 0.01575 0.55800 0.07530 0.00000  0.972 0.259
## Planariidae       0.00879 0.01575 0.55800 0.07530 0.00000  0.987 0.120
## Caenidae          0.00744 0.01333 0.55900 0.07530 0.00000  1.000 0.930
## Baetidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.674
## Gyrinidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.668
## Calamoceratidae   0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.413
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Contrast: Saltito 60_Piper
##
```

```
##                    average      sd   ratio     ava     avb cumsum     p
## Elmidae            0.07103 0.05900 1.20380 0.36930 0.15050  0.140 0.339
## Ceratopogonidae    0.06988 0.06205 1.12630 0.15050 0.36930  0.277 0.009 **
## Chironomidae       0.06391 0.05177 1.23440 0.51340 0.46110  0.403 0.097 .
## Hydropsychidae     0.05534 0.04699 1.17780 0.22580 0.07530  0.512 0.211
## Leptohyphidae      0.05502 0.04763 1.15510 0.22580 0.07530  0.620 0.329
## Gomphidae          0.05153 0.05732 0.89900 0.15050 0.07530  0.721 0.010 **
## Leptoceridae       0.04146 0.07514 0.55180 0.00000 0.17470  0.803 0.050 *
## Caenidae           0.02967 0.05356 0.55390 0.11930 0.00000  0.861 0.263
## Empididae          0.02650 0.04881 0.54280 0.00000 0.07530  0.913 0.080 .
## Coenagrionidae     0.02616 0.04816 0.54320 0.00000 0.07530  0.965 0.308
## Leptophlebiidae    0.01786 0.03236 0.55180 0.00000 0.07530  1.000 1.000
## Baetidae           0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.656
## 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 0.772
## Megapodagrionidae  0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.797
## Perlidae           0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.937
## Scirtidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.646
## Gyrinidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.639
## Calamoceratidae    0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.395
## Polycentropodidae  0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.640
## Simuliidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.640
## Planariidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.391
## Gordiidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.669
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Contrast: Saltito 60_Taconazo 30
##
##                    average      sd   ratio     ava     avb cumsum     p
## Leptophlebiidae    0.06402 0.07033 0.91020 0.00000 0.28260  0.142 0.126
## Chironomidae       0.06359 0.04595 1.38370 0.51340 0.63870  0.283 0.108
## Ceratopogonidae    0.06261 0.05573 1.12350 0.15050 0.40050  0.422 0.024 *
## Leptohyphidae      0.04965 0.03018 1.64500 0.22580 0.00000  0.532 0.556
## Gordiidae          0.04838 0.05262 0.91950 0.00000 0.19450  0.639 0.002 **
## Gomphidae          0.03616 0.03871 0.93400 0.15050 0.15050  0.719 0.091 .
## Hydropsychidae     0.03549 0.04011 0.88470 0.22580 0.25700  0.798 0.898
## Elmidae            0.03348 0.02569 1.30330 0.36930 0.47970  0.872 0.994
## Caenidae           0.02381 0.04272 0.55740 0.11930 0.00000  0.925 0.483
## Coenagrionidae     0.01793 0.03249 0.55180 0.00000 0.07530  0.965 0.657
## Perlidae           0.01584 0.02863 0.55340 0.00000 0.07530  1.000 0.743
## Baetidae           0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.664
## Hydroptilidae      0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.954
## Libellulidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.791
## Megapodagrionidae  0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.807
## Scirtidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.664
## Gyrinidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.666
## Calamoceratidae    0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.405
## Polycentropodidae  0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.650
## Simuliidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.650
## Empididae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.786
## Planariidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.410
## Leptoceridae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.648
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Contrast: Piper_Taconazo 30
##
##                     average      sd   ratio     ava     avb cumsum     p
## Elmidae             0.08479 0.05310 1.59670 0.15050 0.47970  0.171 0.084 .
## Leptophlebiidae     0.06670 0.06650 1.00310 0.07530 0.28260  0.305 0.113
## Chironomidae        0.05850 0.03975 1.47180 0.46110 0.63870  0.423 0.208
## Hydropsychidae      0.05275 0.04041 1.30550 0.07530 0.25700  0.530 0.289
## Gordiidae           0.05001 0.05401 0.92600 0.00000 0.19450  0.630 0.002 **
## Gomphidae           0.03556 0.03710 0.95850 0.07530 0.15050  0.702 0.123
## Leptoceridae        0.03446 0.06181 0.55750 0.17470 0.00000  0.772 0.211
## Coenagrionidae      0.02850 0.03846 0.74090 0.07530 0.07530  0.829 0.193
## Ceratopogonidae     0.02816 0.02893 0.97310 0.36930 0.40050  0.886 0.769
## Leptohyphidae       0.02018 0.03628 0.55620 0.07530 0.00000  0.926 0.997
## Empididae           0.02018 0.03628 0.55620 0.07530 0.00000  0.967 0.296
## Perlidae            0.01631 0.02938 0.55500 0.00000 0.07530  1.000 0.750
## Baetidae            0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.625
## Caenidae            0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.933
## Hydroptilidae       0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.937
## Libellulidae        0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.778
## Megapodagrionidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.774
## Scirtidae           0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.659
## Gyrinidae           0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.628
## Calamoceratidae     0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.381
## Polycentropodidae 0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.666
## Simuliidae          0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.666
## Planariidae         0.00000 0.00000     NaN 0.00000 0.00000  1.000 0.414
## ---
## 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_s2 <- simper_use %>%
  dplyr::filter(p <= 0.05) %>%
  dplyr::group_by(comparison, family)
```

```
table_s2[,-c(8:9)] <- round(table_s2[,-c(8:9)],
                            digits = 2)

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

```
##
## Call:
## vegan::anosim(x = tax_data_log_matrix_fam, grouping = tax_data_log_meta_fam$Stream,      permutation
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.5483
##       Significance: 1e-04
##
## Permutation: free
## Number of permutations: 9999
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.139 0.188 0.229 0.277
##
## Dissimilarity ranks between and within classes:
##             0%   25%   50%    75% 100%   N
## Between      2 60.25 107.5 149.25  190 160
## Arboleda 30  6 20.75  46.5  58.00   76   6
## Sura 30      7 11.25  17.0  36.25   47   6
## Saltito 60   1 18.25  32.5  86.50  136   6
## Piper       59 76.50  93.5 109.75  154   6
## Taconazo 30  4 14.25  38.0  68.50   89   6
```

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

Session Info

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

R version 4.2.3 (2023-03-15 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.2.2.1)*, *tmap(v.3.3-4)*, *spData(v.2.3.0)*, *raster(v.3.6-26)*, *sp(v.2.1-1)*, *sf(v.1.0-14)*, *qpcR(v.1.4-1)*, *Matrix(v.1.6-3)*, *robustbase(v.0.99-0)*, *rgl(v.1.2.1)*, *minpack.lm(v.1.2-4)*, *nlstools(v.2.0-1)*, *nlme(v.3.1-162)*, *drc(v.3.0-1)*, *MASS(v.7.3-58.2)*, *car(v.3.1-2)*, *carData(v.3.0-5)*, *lemon(v.0.4.7)*, *ggrepel(v.0.9.4)*, *ggpubr(v.0.6.0)*, *ggeffects(v.1.3.2)*, *readxl(v.1.4.3)*, *lubridate(v.1.9.3)*, *forcats(v.1.0.0)*, *stringr(v.1.5.1)*, *dplyr(v.1.1.3)*, *purrr(v.1.0.2)*, *readr(v.2.1.4)*, *tidyr(v.1.3.0)*, *tibble(v.3.2.1)*, *ggplot2(v.3.4.4)* and *tidyverse(v.2.0.0)*

loaded via a namespace (and not attached): *backports(v.1.4.1)*, *systemfonts(v.1.0.5)*, *lwgeom(v.0.2-13)*, *plyr(v.1.8.9)*, *splines(v.4.2.3)*, *AlgDesign(v.1.2.1)*, *crosstalk(v.1.2.0)*, *TH.data(v.1.1-2)*, *digest(v.0.6.33)*, *htmltools(v.0.5.7)*, *leaflet.providers(v.2.0.0)*, *fansi(v.1.0.5)*, *magrittr(v.2.0.3)*, *cluster(v.2.1.4)*, *tzdb(v.0.4.0)*, *vroom(v.1.6.4)*, *sandwich(v.3.0-2)*, *timechange(v.0.2.0)*, *colorspace(v.2.1-0)*, *textshaping(v.0.3.7)*, *xfun(v.0.41)*, *leafem(v.0.2.3)*, *crayon(v.1.5.2)*, *jsonlite(v.1.8.7)*, *survival(v.3.5-3)*, *zoo(v.1.8-12)*, *glue(v.1.6.2)*, *stars(v.0.6-4)*, *gtable(v.0.3.4)*, *DEoptimR(v.1.1-3)*, *abind(v.1.4-5)*, *scales(v.1.2.1)*, *mvtnorm(v.1.2-3)*, *DBI(v.1.1.3)*, *rstatix(v.0.7.2)*, *Rcpp(v.1.0.11)*, *plotrix(v.3.8-4)*, *viridisLite(v.0.4.2)*, *units(v.0.8-4)*, *bit(v.4.0.5)*, *proxy(v.0.4-27)*, *htmlwidgets(v.1.6.2)*, *RColorBrewer(v.1.1-3)*, *wk(v.0.9.0)*, *pkgconfig(v.2.0.3)*, *XML(v.3.99-0.15)*, *farver(v.2.1.1)*, *utf8(v.1.2.4)*, *tidyselect(v.1.2.0)*, *labeling(v.0.4.3)*, *rlang(v.1.1.2)*, *tmaptools(v.3.1-1)*, *munsell(v.0.5.0)*, *cellranger(v.1.1.0)*, *tools(v.4.2.3)*, *cli(v.3.6.1)*, *generics(v.0.1.3)*, *broom(v.1.0.5)*, *evaluate(v.0.23)*, *fastmap(v.1.1.1)*, *yaml(v.2.3.7)*, *ragg(v.1.2.6)*, *leafsync(v.0.1.0)*, *knitr(v.1.45)*, *bit64(v.4.0.5)*, *s2(v.1.1.4)*, *compiler(v.4.2.3)*, *rstudioapi(v.0.15.0)*, *png(v.0.1-8)*, *e1071(v.1.7-13)*, *ggsignif(v.0.6.4)*, *stringi(v.1.8.1)*, *highr(v.0.10)*, *classInt(v.0.4-9)*, *vctrs(v.0.6.4)*, *pillar(v.1.9.0)*, *lifecycle(v.1.0.4)*, *cowplot(v.1.1.1)*, *agricolae(v.1.3-7)*, *R6(v.2.5.1)*, *KernSmooth(v.2.23-20)*, *gridExtra(v.2.3)*, *codetools(v.0.2-19)*, *dichromat(v.2.0-0.1)*, *gtools(v.3.9.4)*, *withr(v.2.5.2)*, *multcomp(v.1.4-25)*, *mgcv(v.1.8-42)*, *parallel(v.4.2.3)*, *hms(v.1.1.3)*, *terra(v.1.7-55)*, *class(v.7.3-21)*, *rmarkdown(v.2.25)* and *base64enc(v.0.1-3)*