

Microbial community assembly in a multi-layer dendritic metacommunity

Nathan I. Wisnoski and Jay T. Lennon

2019-01-04

Initial setup

First, we load the data. This includes the site-by-species matrix (generated in Mothur, v. 1.41.1), the RDP taxonomy, the environmental data, and the phylogenetic tree (generated with FastTreeMP).

Next, we will clean up the data. I'll remove any sample that didn't get 10000 reads. Then also cut those samples from the environment and design tables.

```
# Sequencing Coverage
coverage <- rowSums(OTUs)

# Remove Low Coverage Samples
cutoff <- 10000
lows <- which(coverage < cutoff)
OTUs <- OTUs[-which(coverage < cutoff), ]
design <- design.total[-which(coverage < cutoff), ]
env <- env.total[-which(coverage < cutoff), ]

# Remove OTUs with less than 5 occurrences across all sites
OTUs <- OTUs[, which(colSums(OTUs) > 2)]

OTUs <- OTUs[-which(env$sample == "W1_20_W"),]
design <- design[-which(env$sample == "W1_20_W"),]
env <- env[-which(env$sample == "W1_20_W"),]
```

Here, I'll read in the dendritic distances and add a tiny bit of jitter to the spatial distances so nearby sites aren't identical. Then, I'll calculate the earth distance in meters.

```
den.dists <- make.dendritic.dists("data/hja_dendritic-dists.csv")
design$upstreamdist <- as.matrix(den.dists)[1,]

# Read in Distances
# Geo distance Matrix
xy <- cbind(jitter(env$longitude, amount = .0001),
            jitter(env$latitude, amount = .0001))
geo.dists <- SoDA::geoXY(xy[,1], xy[,2])
dist.mat <- fossil::earth.dist(xy) * 1000
```

Next, we will see if any of the environmental variables need to be transformed. I'll then rescale the environmental variables.

```
# Remove orthogonal vectors and make numbers below detection close to zero
env.subs <- env %>% select(habitat, elevation,
```

```

        temperature, conductivity,
        ph, TN, TP, DOC) %>%
mutate(TN = if_else(TN < 0, 0.001, TN),
       TP = if_else(TP < 0, 0.001, TP))

#hist(log(env.subs$TP), breaks = 30)
#hist(log(env.subs$TN), breaks = 30)

env.subs <- env.subs %>% mutate(TN = log(TN), TP = log(TP))

# rescale variables
env.subs <- env.subs %>% mutate_if(is_double, scale_vec)

```

Now, I'll perform some transformations on the abundance data. I'll work with the Hellinger-transformed data for the rest of the analysis.

```

# Rarefy communities
set.seed(47405)
OTUs <- rrarefy(OTUs, min(rowSums(OTUs)))
OTUs <- OTUs[, -which(colSums(OTUs) == 0)]

# Transformations and Standardizations
OTUsREL <- decostand(OTUs, method = "total")
OTUs.PA <- decostand(OTUs, method = "pa")
OTUsREL.log <- decostand(OTUs, method = "log")
OTUsREL.hel <- decostand(OTUs, method = "hellinger")

```

I removed the sites with low coverage, and I removed the OTUs with low abundance across the whole dataset.

Here, we will read in the phylogenetic tree. I pruned the phylogenetic tree to match only the taxa remaining in the dataset. Then, I rooted the tree using the midpoint method.

```

# hja.tree <- read.tree("data/hja_streams.tree")
# matched.phylo <- match.phylo.comm(hja.tree, OTUs)
# hja.tree <- matched.phylo$phy
# is.rooted(hja.tree)
# hja.tree.rooted <- midpoint.root(hja.tree)
# is.rooted(hja.tree.rooted)
# saveRDS(object = hja.tree.rooted, file = "temp/hja_tree_rooted.nwk")
hja.tree.rooted <- readRDS(file = "temp/hja_tree_rooted.nwk")

```

Environmental analysis

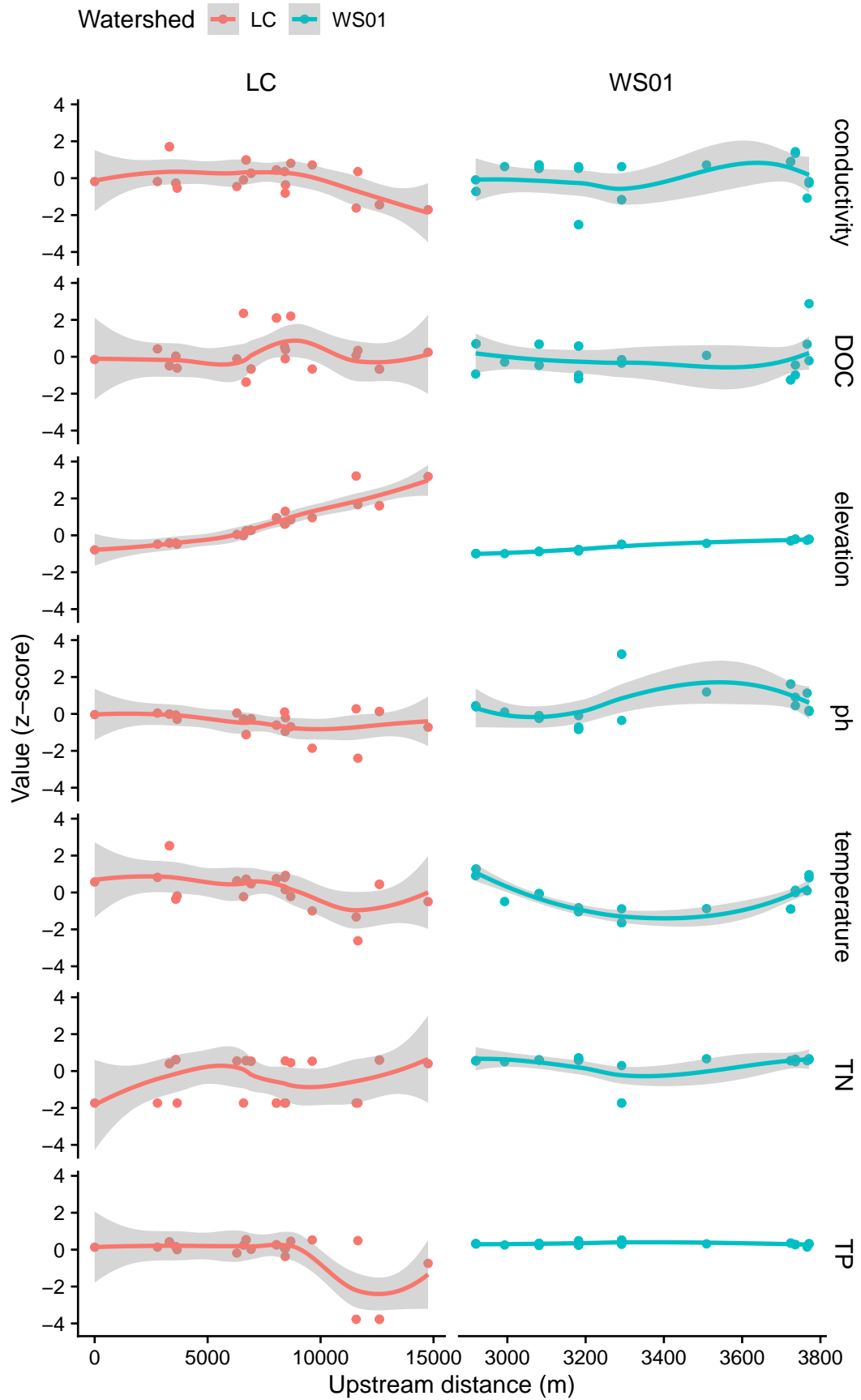
Here, I'll just plot the environmental variables from downstream to upstream across the watershed.

```

env.subs %>% mutate(upstreamdist = design$upstreamdist, watershed = design$watershed) %>%
gather(-upstreamdist, -watershed, -habitat, key = variable, value = measurement) %>%
ggplot(aes(x = upstreamdist, y = measurement, color = watershed)) +
facet_grid(variable ~ watershed, scales = "free_x") +
geom_point() +
geom_smooth() +
theme(legend.position = "top") +
scale_x_continuous(labels = scales::wrap_format(10)) +
labs(x = "Upstream distance (m)",
     y = "Value (z-score)",

```

```
color = "Watershed")
```



alpha Diversity analysis

```
alpha.tbl <- tibble(
  habitat = factor(design$habitat, levels = c("sediment", "water"),
    labels = c("Sediment", "Planktonic")),
  upstream = design$upstreamdist,
  order = design$order,
  sample = rownames(design),
  N0 = rowSums(OTUsREL.hel > 0),
  N1 = exp(diversity(OTUsREL.hel, index = "shannon")),
  N2 = diversity(OTUsREL, index = "invsimpson")
)
summary(lm(N1 ~ habitat, data = alpha.tbl))
```

```
##
## Call:
## lm(formula = N1 ~ habitat, data = alpha.tbl)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1900.9  -284.2   107.3   340.3   945.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1915.0      115.0  16.645 < 2e-16 ***
## habitatPlanktonic    541.3      149.5   3.619  0.00072 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 514.5 on 47 degrees of freedom
## Multiple R-squared:  0.218, Adjusted R-squared:  0.2013
## F-statistic: 13.1 on 1 and 47 DF, p-value: 0.0007205
```

```
alpha.fig <- alpha.tbl %>%
  ggplot(aes(x = habitat, y = N1, fill = habitat, color = habitat)) +
  geom_boxplot(alpha = 0.8) +
  geom_jitter(alpha = 0.2) +
  labs(x = "", y = expression(paste(alpha, "-diversity (species equivalents)"))) +
  scale_fill_manual(values = (my.colors)) +
  scale_color_manual(values = colorspace::darken(my.colors,.4)) +
  guides(fill = FALSE, color = FALSE) +
  scale_y_continuous(limits = c(0, 3500))

summary(lm(N1 ~ habitat * order, data = alpha.tbl))
```

```
##
## Call:
## lm(formula = N1 ~ habitat * order, data = alpha.tbl)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1971.3  -287.7   131.1   336.8   929.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          1946.32      290.10    6.709 2.75e-08 ***
## habitatPlanktonic     634.13      386.76    1.640    0.108
## order                -13.93      117.94   -0.118    0.907
## habitatPlanktonic:order -39.83      156.18   -0.255    0.800
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 524.1 on 45 degrees of freedom
## Multiple R-squared:  0.223, Adjusted R-squared:  0.1712
## F-statistic: 4.304 on 3 and 45 DF,  p-value: 0.009417
```

Unique taxa per habitat

```
sediment.only <- OTUs[, which(
  colSums(OTUs[which(design$habitat == "water"),]) == 0)]
water.only <- OTUs[, which(
  colSums(OTUs[which(design$habitat == "sediment"),]) == 0)]
water.sed <- OTUs[, which(
  colSums(OTUs[which(design$habitat == "sediment"),]) > 1 &
  colSums(OTUs[which(design$habitat == "water"),]) > 1)]
in.water <- OTUs[, which(
  colSums(OTUs[which(design$habitat == "water"),]) > 1)]
in.water.pa <- decostand(in.water, method = "pa")
in.sediment <- OTUs[, which(
  colSums(OTUs[which(design$habitat == "sediment"),]) > 1)]
in.sediment.pa <- decostand(in.sediment, method = "pa")

# what proportion of taxa are unique at each site
water.richness <- rowSums(in.water.pa[which(design$habitat == "water"),])
# in water samples, but not in any sediments
water.unique <- rowSums(in.water.pa[
  which(design$habitat == "water"), # count the number of taxa in each water samp
  which(colSums(in.water.pa[which(design$habitat == "sediment"),]) == 0)]) # whose abund in sediments is zero
water.unique.frac <- water.unique / water.richness
se(water.unique.frac)

## [1] 0.009413338

sed.richness <- rowSums(in.sediment.pa[which(design$habitat == "sediment"),])
# in sediments, but not in any water samples
sed.unique <- rowSums(in.sediment.pa[
  which(design$habitat == "sediment"), # count number of taxa in sed samples
  which(colSums(in.sediment.pa[which(design$habitat == "water"),]) == 0)]) #whose abund in water is zero
sed.unique.frac <- sed.unique / sed.richness
se(sed.unique.frac)

## [1] 0.002072766

unique.fracs <- data.frame("unique_frac" = water.unique.frac,
  "habitat" = "Planktonic") %>%
  rbind.data.frame(., data.frame(
    "unique_frac" = sed.unique.frac,
    "habitat" = "Sediment"))
unique.fracs <- unique.fracs %>% rownames_to_column(var = "sample")
```

```

unique.fracs$habitat <- factor(unique.fracs$habitat,
                              levels = c("Sediment", "Planktonic"), ordered = F)
unique.fracs <- left_join(unique.fracs, rownames_to_column(design[,c("order", "upstreamdist")], var = "s",
summary(lm(unique_frac ~ habitat, data = unique.fracs))

##
## Call:
## lm(formula = unique_frac ~ habitat, data = unique.fracs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.126419 -0.013957  0.001765  0.016537  0.085844
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.077675   0.008848   8.779 1.8e-11 ***
## habitatPlanktonic 0.163540   0.011501  14.220 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03957 on 47 degrees of freedom
## Multiple R-squared:  0.8114, Adjusted R-squared:  0.8074
## F-statistic: 202.2 on 1 and 47 DF,  p-value: < 2.2e-16

summary(lm(unique_frac ~ habitat, data = unique.fracs)) %>%
  capture.output(file = "tables/unique_compare.txt")

unique.fig <- unique.fracs %>%
  ggplot(aes(x = habitat, y = unique_frac, color = habitat, fill = habitat)) +
  geom_boxplot(alpha = 0.8) +
  geom_jitter(alpha = 0.2) +
  labs(x = "", y = "Proportion habitat-specific taxa") +
  scale_fill_manual(values = (my.colors)) +
  scale_color_manual(values = colorspace::darken(my.colors,.4)) +
  guides(fill = FALSE, color = FALSE) +
  scale_y_continuous(limits = c(0, .3))

summary(lm(log10(unique_frac) ~ order*habitat, unique.fracs))

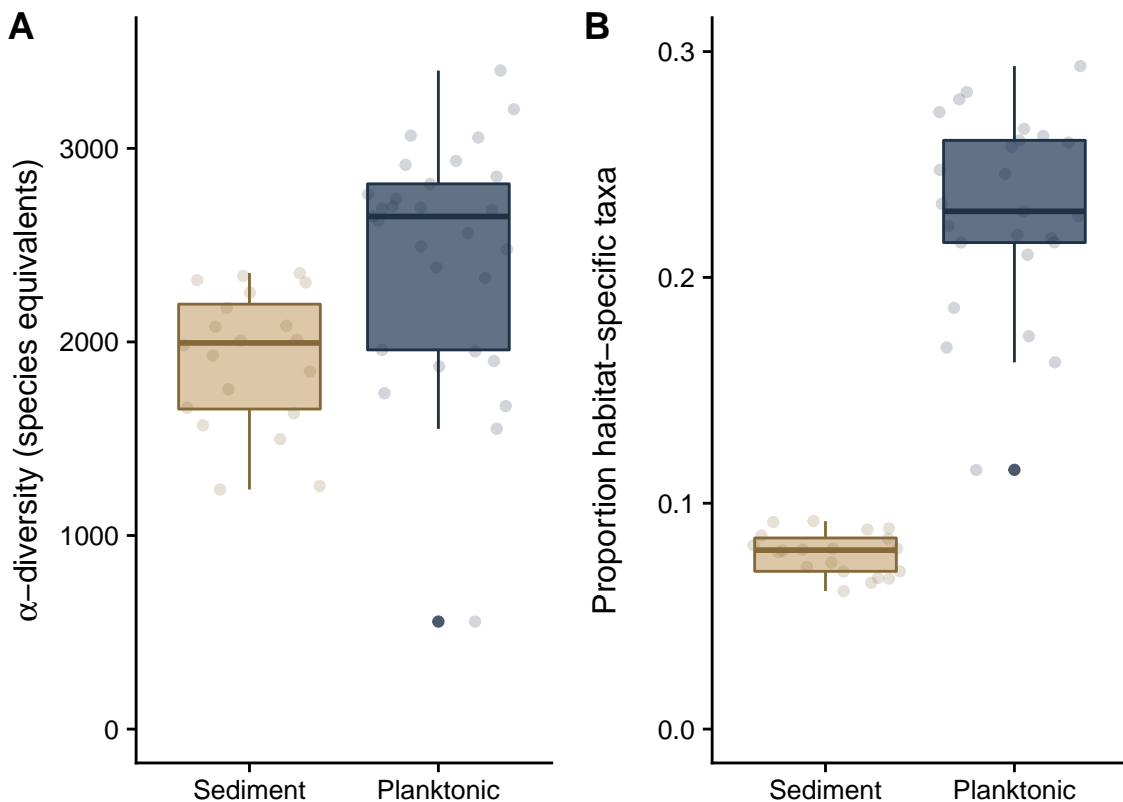
##
## Call:
## lm(formula = log10(unique_frac) ~ order * habitat, data = unique.fracs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25910 -0.02894  0.00473  0.05057  0.18649
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.13679   0.04463 -25.470 < 2e-16 ***
## order           0.01070   0.01815   0.590   0.558
## habitatPlanktonic 0.41544   0.05950   6.982 1.08e-08 ***
## order:habitatPlanktonic 0.02967   0.02403   1.235   0.223
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
##
## Residual standard error: 0.08064 on 45 degrees of freedom
## Multiple R-squared:  0.9061, Adjusted R-squared:  0.8999
## F-statistic: 144.8 on 3 and 45 DF,  p-value: < 2.2e-16

alpha.to.plot <- unique.fracs %>% left_join(alpha.tbl) %>% select(-N0, -N2) %>%
  gather(unique_frac, N1, key = "metric", value = "value")
alpha.to.plot$metric <- factor(alpha.to.plot$metric)
levels(alpha.to.plot$metric) <- c(
  expression(paste(alpha, "-diversity")),
  expression(paste("Proportion habitat-specific taxa"))
)
alpha.plot <- alpha.to.plot %>%
  ggplot(aes(x = habitat, y = value, fill = habitat, color = habitat)) +
  facet_wrap(~ metric, scales = "free_y", strip.position = "left", labeller = label_parsed, ncol = 2) +
  geom_boxplot(width = .5, alpha = 0.8) +
  geom_jitter(alpha = 0.1) +
  scale_fill_manual(values = (my.colors)) +
  scale_color_manual(values = colorspace::darken(my.colors, 0.4)) +
  theme(strip.placement = "outside", legend.position = "none") +
  guides(fill = FALSE) +
  labs(y = "", x = "")

plot_grid(alpha.fig, unique.fig, ncol = 2, align = "hv", labels = c("A", "B"))
```



Beta diversity:

Ordination

```
hja.pcoa <- run.pcoa(comm = OTUsREL.hel, dist.metric = "euclidean", plot = T)

## PCoA Axis 1 explains 13.1 percent of total variation.
## PCoA Axis 2 explains 8.3 percent of total variation.

pcoa.ellipse <- ordiellipse(hja.pcoa$pcoa, str_to_title(design$habitat), display = "sites",
                           kind = "se", conf = 0.95, label = T)

pcoa.plot <- cbind.data.frame(vegan::scores(hja.pcoa$pcoa), group = str_to_title(design$habitat))
df_ell <- calc.ellipse(ord = pcoa.plot, ellipse = pcoa.ellipse)

# Run a PERMANOVA
hja.permanova <- adonis(vegdist(OTUsREL, method = "bray") ~ design$habitat + design$order + design$watershed)
hja.permanova$aov.tab %>% pander::pander()
```

Table 1: Permutation: free

	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
design\$habitat	1	1.676	1.676	8.907	0.1528	0.001
design\$order	1	0.3922	0.3922	2.085	0.03576	0.006
design\$watershed	1	0.4342	0.4342	2.308	0.03959	0.005
Residuals	45	8.465	0.1881	NA	0.7719	NA
Total	48	10.97	NA	NA	1	NA

```
capture.output(hja.permanova$aov.tab, file = "./tables/hja_permanova.txt")
```

Now, we'll run an RDA.

```
#hja.rda <- rda(OTUsREL.hel ~ ., env.subs)
#R2.all.vars <- RsquareAdj(hja.rda)$adj.r.squared
#anova(hja.rda, permutations = how(nperm = 999))
#anova(hja.rda, by = "axis", permutations = how(nperm = 999))

hja.dbrda <- dbrda(vegdist(OTUsREL) ~ ., env.subs)

# overall model and 1st two dbRDA axes significant
#anova(hja.dbrda, permutations = how(nperm = 999))
#anova(hja.dbrda, by = "axis", permutations = how(nperm = 999))
```

We see that the RDA is globally significant and the first two canonical axes are also significant. Because the full model was significant, I'll run a model selection procedure.

```
# forward.sel(OTUsREL.hel,
#             model.matrix(~., env.subs)[-1],
#             adjR2thresh = R2.all.vars,
#             nperm = 999)

# habitat, elevation, and conductivity were selected by Blanchet 2008 method

# reduced model selects habitat, elevation and conductivity. but I'll add back other key env vars that
```

```
hja.rda.reduced <- dbRda(vegdist(OTUsREL) ~ habitat + elevation + conductivity + TP + TN + ph + DOC, en
```

The model selection procedure left us with the model using only habitat, elevation, and conductivity as predictor variables. Now, we have 3 significant RDA axes with the more parsimonious model.

Make figure 2

```
rda.plot <- cbind.data.frame(vegan::scores(hja.rda.reduced)$sites, group = str_to_title(design$habitat))
rda.var1 <- round(eigenvals(hja.rda.reduced)[1] / sum(eigenvals(hja.rda.reduced)) * 100, 1)
rda.var2 <- round(eigenvals(hja.rda.reduced)[2] / sum(eigenvals(hja.rda.reduced)) * 100, 1)
rda.vecs <- as.data.frame(hja.rda.reduced$CCA$biplot)
rda.vecs$predictor <- c("Planktonic\n habitat", "Elevation", "Cond.",
                        "TP", "TN", "pH", "DOC")

rda.vecs$origin <- 0
scale.arrows = 1
hja.rda.fig <- ggplot(data = rda.plot, aes(x = dbRDA1, y = dbRDA2)) +
  geom_hline(aes(yintercept = 0), linetype = "dashed", alpha = 0.25, size = 0.25) +
  geom_vline(aes(xintercept = 0), linetype = "dashed", alpha = 0.25, size = 0.25) +
  geom_point(aes(fill = group, shape = watershed), size = 2, alpha = .8) +
  # geom_point(data = subset(rda.plot, group == "Sediment"), color = "black", size = 2) +
  # geom_point(data = subset(rda.plot, group == "Water"), color = "black", size = 2) +
  # geom_path(data = df_ell,
  #           aes(x = Dim1, y = Dim2, color = group),
  #           size = .8, alpha = 1, linetype = 2) +
  stat_ellipse(data = rda.plot, alpha = 0.7, aes(color = group, linetype = watershed)) +
  labs(x = paste0("dbRDA1 (", rda.var1, "%)"),
       y = paste0("dbRDA2 (", rda.var2, "%)"),
       color = "Habitat", shape = "Watershed") +
  scale_color_manual(values = my.colors) +
  scale_fill_manual(values = my.colors) +
  scale_shape_manual(values = c(21, 24)) +
  coord_fixed() +
  theme(legend.position = "none") +
  geom_segment(data = rda.vecs, size = .3,
              aes(x = origin, y = origin,
                  xend = scale.arrows*dbRDA1,
                  yend = scale.arrows*dbRDA2),
              alpha = .7, color = "black",
              arrow = arrow(angle = 20,
                           length = unit(.1, "inches"),
                           type = "open")) +
  geom_text_repel(data = rda.vecs, size = 1.8,
                  aes(x = (scale.arrows + .1)*dbRDA1,
                      y = (scale.arrows + .1)*dbRDA2, label = predictor),
                  color = "black",
                  segment.alpha = 0,
                  point.padding = 0,
                  nudge_y = c(-.15, #water
                              0, # elev
                              .035, # conduct
                              0, # TP
                              0, #TN
                              -.05, #ph
```

```

        .05)) + # DOC
  annotate("text", x = .8, y = 2.2, label = "Benthic", size = 2.5, color = darken(my.colors[1])) +
  annotate("text", x = -.5, y = 2.2, label = "Planktonic", size = 2.5, color = darken(my.colors[2])) +
  ggsave("figures/Fig2.png", width = 8.4, height = 8.4, units = "cm", dpi = 500) +
  ggsave("figures/Fig2.pdf", width = 8.4, height = 8.4, units = "cm")

```

Subset only flow-connected sites

```

flow.connected <- read.csv("data/flow-connected-matrix.csv", row.names = 1)
den.dists.mat <- as.matrix(den.dists)
flow.connected.dists <- as.dist(
  flow.connected[rownames(den.dists.mat), colnames(den.dists.mat)] * den.dists.mat)

```

Null model analysis

```

# Calculate abundance-weighted Raup-Crick dissimilarities
regional.abunds <- t(as.matrix(colSums(OTUs)))
regional.relabunds <- decostand(regional.abunds, method = "total")
occupancy.probs <- t(as.matrix(colSums(decostand(OTUs, method = "pa")) / nrow(OTUs)))
site.abunds <- rowSums(OTUs)
site.rich <- specnumber(OTUs)
a <- regional.relabunds * occupancy.probs

# Create a null community based on Stegen et al. 2015
set.seed(47405)
#rc.nulls <- nullcom.rcabund(OTUs = OTUs, stand = "total", distance = "bray")
#saveRDS(rc.nulls, "temp/rc_nullmodels_bray.rda")
rc.nulls <- readRDS("temp/rc_nullmodels_bray.rda")

obs.bray <- as.matrix(vegdist(OTUsREL, method = "bray"))
site.compares <- expand.grid(site1 = 1:nrow(obs.bray), site2 = 1:nrow(obs.bray))
site.compares <- site.compares[-which(site.compares[,1] == site.compares[,2]),]
RC.bray <- matrix(NA, nrow = nrow(obs.bray), ncol = nrow(obs.bray))

for(row.i in 1:nrow(site.compares)){
  site1 <- site.compares[row.i,1]
  site2 <- site.compares[row.i,2]
  pairwise.null <- rc.nulls[site1,site2,]
  pairwise.bray <- obs.bray[site1,site2]
  num.greater <- sum(pairwise.null > pairwise.bray)
  num.ties <- sum(pairwise.null == pairwise.bray)
  val <- -1*(((1 * num.greater) + (0.5 * num.ties))/999 - 0.5) * 2)
  RC.bray[site1, site2] <- val
}

rownames(RC.bray) <- rownames(design)
colnames(RC.bray) <- rownames(design)
RC.bray.dist <- as.dist(RC.bray)
range(RC.bray.dist)

```

```
## [1] -1 1
```

```

flow_connected_dists_df <- simba::liste(flow.connected.dists, entry = "den_dists")
dists.df <- simba::liste(RC.bray.dist, entry = "RC_bray") %>% add_column(den_dists = flow_connected_dists)
dists.df$habitat <- NA
dists.df[str_detect(dists.df$NBX, "_W") & str_detect(dists.df$NBY, "_W"),]$habitat <- str_wrap("Planktonic")
dists.df[str_detect(dists.df$NBX, "_W") & str_detect(dists.df$NBY, "_S"),]$habitat <- str_wrap("Planktonic-Benthic")
dists.df[str_detect(dists.df$NBX, "_S") & str_detect(dists.df$NBY, "_W"),]$habitat <- str_wrap("Planktonic-Benthic")
dists.df[str_detect(dists.df$NBX, "_S") & str_detect(dists.df$NBY, "_S"),]$habitat <- str_wrap("Benthic")

dists.df$habitat <- factor(dists.df$habitat, levels = c("Planktonic", "Benthic", "Planktonic-Benthic"))

```

Match phylo

```

matched.phylo <- match.phylo.comm(phy = hja.tree.rooted, comm = OTUs[,which(colSums(OTUs) > 10)])
hja.comm <- matched.phylo$comm
hja.phy <- matched.phylo$phy

```

Make figure 4

```

#mtd.hja <- comdistnt.par(hja.comm, cophenetic(hja.phy), abundance.weighted = T, cores = 32)

#hja.mtd.ses <- ses.comdistnt2(
#  samp = hja.comm,
#  dis = cophenetic(hja.phy),
#  method = "quasiswap",
#  fixedmar = "both",
#  shuffle = "both",
#  strata = NULL,
#  mtype = "count",
#  burnin = 0,
#  thin = 1,
#  abundance.weighted = TRUE,
#  exclude.conspecifics = FALSE,
#  runs = 999,
#  cores = 32)
# saveRDS(mtd.hja, file = "data/mtds.rda")

# # Create null comms
# mtd.null <- array(NA, c(50, 50, 999))
# for(i in 1:999){
#   if(i == 1) pb <- progress_bar$new(total = 999, force = T)
#   pb$update(ratio = i/999)
#   #print(paste("creating null community ", i, " of 999"))
#   temp.mtd <- comdistnt(hja.comm,
#                         cophenetic(tipShuffle(hja.phy)),
#                         abundance.weighted=T)
#   mtd.null[, ,i] <- as.matrix(temp.mtd)
#   if(i %% 50 == 0 | i == 999) saveRDS(mtd.null, file = "data/mtds-null-dist.rda")
# }

# read null dists
#mtd.hja <- readRDS(file = "data/mtds.rda")

```

```

#mntds.null <- readRDS(file = "data/mntds-null-dist.rda")

# obs.mntds <- as.matrix(mntd.hja)
#site.compares <- expand.grid(site1 = 1:ncol(obs.mntds), site2 = 1:ncol(obs.mntds))
#bNTI <- matrix(NA, nrow = nrow(obs.mntds), ncol = ncol(obs.mntds))
#for(row.i in 1:nrow(site.compares)){
#  site1 <- site.compares[row.i,1]
#  site2 <- site.compares[row.i,2]
#  pairwise.null <- mntds.null[site1,site2,]
#  pairwise.mntd <- obs.mntds[site1,site2]
#  null.mean <- mean(pairwise.null, na.rm = TRUE)
#  null.sd <- sd(pairwise.null, na.rm = TRUE)
#  val <- (pairwise.mntd - null.mean) / null.sd
#  bNTI[site1, site2] <- val
#}
#colnames(bNTI) <- rownames(hja.comm)
#rownames(bNTI) <- rownames(hja.comm)

hja.mntd.ses <- readRDS("temp/hja.mntd.ses.rda")
bNTI.dist <- as.dist(hja.mntd.ses$comdistnt.obs.z)
sum(bNTI.dist < 2 & bNTI.dist > -2) / length(bNTI.dist) # undom

## [1] 0.1403061

sum(bNTI.dist > 2) / length(bNTI.dist) # variable selection

## [1] 0.6122449

sum(bNTI.dist < -2) / length(bNTI.dist) # homogeneous selection

## [1] 0.247449

hja.bnti.dist.ls <- simba::liste(bNTI.dist, entry = "bNTI")
hja.rcbray.dist.ls <- simba::liste(RC.bray.dist, entry = "RCbray")
hja.euclid.dist.ls <- simba::liste(dist(SoDA::geoXY(latitude = xy[,2], longitude = xy[,1])), entry = "euclid")
hja.assembly <- full_join(dists.df, hja.bnti.dist.ls)
hja.assembly <- cbind(hja.assembly, env = simba::liste(dist(env.subs[,-1]))[,3], euclid = hja.euclid.dist.ls)

hja.assembly.plot <- hja.assembly %>% filter(den_dists > 0) %>%
  mutate(process = ifelse(bNTI < -2, "Homogeneous selection",
    ifelse(bNTI > 2, "Variable selection",
      ifelse(RC_bray > 0.95, "Dispersal limitation",
        ifelse(RC_bray < -0.95, "Mass effects", "Undominated"))))) %>%
  mutate(signif = ifelse(abs(RC_bray) < 0.95 & abs(bNTI) < 2, FALSE, TRUE)) %>%
  gather(RC_bray, bNTI, key = "metric", value = "value")
hja.assembly.plot$metric <- factor(hja.assembly.plot$metric)
levels(hja.assembly.plot$metric) <- c(
  expression(paste(beta,"NTI")),
  expression(beta["RC, Bray-Curtis"])
)

hja.assembly.rounded <- hja.assembly.plot %>%
  mutate(den_dists_log10 = log10(den_dists)) %>%

```

```

mutate(den_dists_log10_rounded = round(den_dists_log10,0)) %>%
mutate(den_dists_rounded = 10^den_dists_log10_rounded)

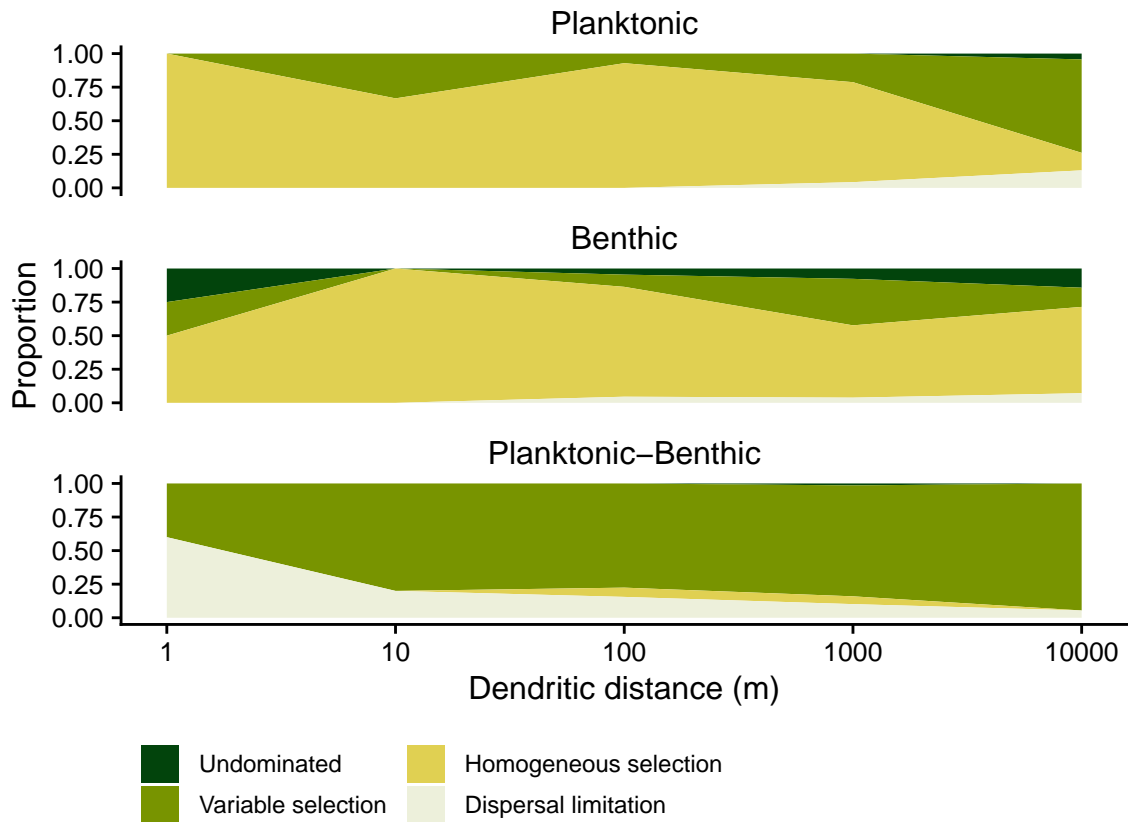
hja.assembly.grouped <- hja.assembly.rounded %>%
  group_by(habitat, den_dists_rounded) %>%
  count(process) %>%
  mutate(proportion = n/sum(n))
hja.assembly.grouped <- hja.assembly.grouped %>%
  full_join(unique(expand.grid(hja.assembly.grouped[,1:3])), fill = 0) %>%
  mutate(proportion = ifelse(is.na(proportion),0, proportion))

write_csv(hja.assembly.grouped, "tables/assembly_counts_by_scale.csv")

hja.assembly.grouped %>%
  arrange(process, den_dists_rounded) %>%
  mutate(process = factor(process, levels = c("Undominated",
                                             "Variable selection",
                                             "Homogeneous selection",
                                             "Mass effects",
                                             "Dispersal limitation")) %>%

  ggplot(aes(x = den_dists_rounded, y = proportion, fill = process)) +
  geom_area(alpha = 1, position = "stack", color = NA) +
  scale_x_log10() +
  facet_wrap(~ habitat, ncol = 1) +
  scale_fill_manual(values = c("#02440c", "#789400", "#e0d052", "#edf0db", "#40becd")) +
  labs(x = "Dendritic distance (m)", y = "Proportion") +
  theme(legend.position = "bottom",
        legend.title = element_blank(),
        legend.text = element_text(size = 9)) +
  guides(fill = guide_legend(nrow = 2)) +
  ggsave("figures/Fig4.png", height = 8, width = 4, dpi = 600) +
  ggsave("figures/Fig4.pdf", height = 8, width = 4)

```



```
hja.assembly.grouped %>%
  ungroup() %>%
  group_by(den_dists_rounded) %>%
  count()
```

```
## # A tibble: 5 x 2
## # Groups:   den_dists_rounded [5]
##   den_dists_rounded    n
##             <dbl> <int>
## 1                1    12
## 2               10    12
## 3              100    12
## 4             1000    12
## 5            10000    12
```

```
hja.assembly.grouped %>%
  count()
```

```
## # A tibble: 15 x 3
## # Groups:   habitat, den_dists_rounded [15]
##   habitat      den_dists_rounded    n
##   <fct>             <dbl> <int>
## 1 Planktonic          1         4
## 2 Planktonic         10         4
## 3 Planktonic        100         4
## 4 Planktonic       1000         4
## 5 Planktonic      10000         4
## 6 Benthic             1         4
## 7 Benthic            10         4
```

```
## 8 Benthic 100 4
## 9 Benthic 1000 4
## 10 Benthic 10000 4
## 11 Planktonic-Benthic 1 4
## 12 Planktonic-Benthic 10 4
## 13 Planktonic-Benthic 100 4
## 14 Planktonic-Benthic 1000 4
## 15 Planktonic-Benthic 10000 4
```

```
hja.assembly.plot %>%
  group_by(habitat) %>%
  count(process)
```

```
## # A tibble: 12 x 3
## # Groups:   habitat [3]
##   habitat      process      n
##   <fct>      <chr>    <int>
## 1 Planktonic Dispersal limitation    10
## 2 Planktonic Homogeneous selection   142
## 3 Planktonic Undominated         2
## 4 Planktonic Variable selection    60
## 5 Benthic    Dispersal limitation     6
## 6 Benthic    Homogeneous selection    88
## 7 Benthic    Undominated        12
## 8 Benthic    Variable selection     28
## 9 Planktonic-Benthic Dispersal limitation    44
## 10 Planktonic-Benthic Homogeneous selection    16
## 11 Planktonic-Benthic Undominated         2
## 12 Planktonic-Benthic Variable selection   286
```

```
hja.assembly.plot %>% count(process)
```

```
##           process      n
## 1 Dispersal limitation    60
## 2 Homogeneous selection  246
## 3 Undominated           16
## 4 Variable selection    374
```

Make figure 5

```
headwaters.dist <- as.dist(as.matrix(bNTI.dist)[which(design$order == 1), which(design$order == 1)])
mainstem.dist <- as.dist(as.matrix(bNTI.dist)[which(design$order != 1), which(design$order != 1)])

headwaters.dist.sed.tax <- as.dist(as.matrix(RC.bray.dist)[
  which(design$order == 1 & design$habitat != "water"),
  which(design$order == 1 & design$habitat != "water")])
mainstem.dist.sed.tax <- as.dist(as.matrix(RC.bray.dist)[
  which(design$order != 1 & design$habitat != "water"),
  which(design$order != 1 & design$habitat != "water")])
headwaters.dist.water.tax <- as.dist(as.matrix(RC.bray.dist)[
  which(design$order == 1 & design$habitat == "water"),
  which(design$order == 1 & design$habitat == "water")])
mainstem.dist.water.tax <- as.dist(as.matrix(RC.bray.dist)[
  which(design$order != 1 & design$habitat == "water"),
```



```

which(design$order != 1 & design$habitat == "water"))

headwaters.dist.sed.phy <- as.dist(as.matrix(bNTI.dist)[
  which(design$order == 1 & design$habitat != "water"),
  which(design$order == 1 & design$habitat != "water")])
mainstem.dist.sed.phy <- as.dist(as.matrix(bNTI.dist)[
  which(design$order != 1 & design$habitat != "water"),
  which(design$order != 1 & design$habitat != "water")])
headwaters.dist.water.phy <- as.dist(as.matrix(bNTI.dist)[
  which(design$order == 1 & design$habitat == "water"),
  which(design$order == 1 & design$habitat == "water")])
mainstem.dist.water.phy <- as.dist(as.matrix(bNTI.dist)[
  which(design$order != 1 & design$habitat == "water"),
  which(design$order != 1 & design$habitat == "water")])

# now cross-habitat comparisons, not square matrix
headwaters.dist.sed.water.phy <- as.matrix(bNTI.dist)[
  which(design$order == 1 & design$habitat != "water"),
  which(design$order == 1 & design$habitat == "water")]
downstream.dist.sed.water.phy <- as.matrix(bNTI.dist)[
  which(design$order != 1 & design$habitat != "water"),
  which(design$order != 1 & design$habitat == "water")]

headwaters.dist.sed.water.tax <- as.matrix(RC.bray.dist)[
  which(design$order == 1 & design$habitat != "water"),
  which(design$order == 1 & design$habitat == "water")]
downstream.dist.sed.water.tax <- as.matrix(RC.bray.dist)[
  which(design$order != 1 & design$habitat != "water"),
  which(design$order != 1 & design$habitat == "water")]

# construct dfs for headwater downstream comparison
hwsp <- simba::liste(headwaters.dist.sed.phy, entry = "bNTI")
hwwp <- simba::liste(headwaters.dist.water.phy, entry = "bNTI")
dssp <- simba::liste(mainstem.dist.sed.phy, entry = "bNTI")
dswp <- simba::liste(mainstem.dist.water.phy, entry = "bNTI")
hwsdp <- simba::liste(headwaters.dist.sed.water.phy, entry = "bNTI")
dsswp <- simba::liste(downstream.dist.sed.water.phy, entry = "bNTI")

hwsst <- simba::liste(headwaters.dist.sed.water.tax, entry = "RC_hel")
dssst <- simba::liste(downstream.dist.sed.water.tax, entry = "RC_hel")
hwst <- simba::liste(headwaters.dist.sed.tax, entry = "RC_hel")
hwwt <- simba::liste(headwaters.dist.water.tax, entry = "RC_hel")
dsst <- simba::liste(mainstem.dist.sed.tax, entry = "RC_hel")
dswt <- simba::liste(mainstem.dist.water.tax, entry = "RC_hel")

hwsdp$position <- "Headwater"
hwwp$position <- "Headwater"
hwsdp$position <- "Headwater"
hwst$position <- "Headwater"
hwwt$position <- "Headwater"
hwsst$position <- "Headwater"

dssp$position <- "Downstream"

```

```

dswp$position <- "Downstream"
dsswp$position <- "Downstream"
dsst$position <- "Downstream"
dswt$position <- "Downstream"
dsswt$position <- "Downstream"

hwsdp$habitat <- "Benthic"
hwwp$habitat <- "Planktonic"
hwsdp$habitat <- "Planktonic-Benthic"
hwst$habitat <- "Benthic"
hwwt$habitat <- "Planktonic"
hwswt$habitat <- "Planktonic-Benthic"

dssp$habitat <- "Benthic"
dswp$habitat <- "Planktonic"
dsswp$habitat <- "Planktonic-Benthic"
dsst$habitat <- "Benthic"
dswt$habitat <- "Planktonic"
dsswt$habitat <- "Planktonic-Benthic"

assembly.by.position <- bind_rows(
  full_join(hwsdp, hwst),
  full_join(hwwp, hwwt),
  full_join(hwsdp, hwswt),
  full_join(dssp, dsst),
  full_join(dswp, dswt),
  full_join(dsswp, dsswt)
)

assembly.by.position.process <- assembly.by.position %>%
  mutate(process = ifelse(bNTI < -2, "Homogeneous selection",
    ifelse(bNTI > 2, "Variable selection",
      ifelse(RC_hel > 0.95, "Dispersal limitation",
        ifelse(RC_hel < -0.95, "Mass effects", "Undominated"))))) %>%
  mutate(signif = ifelse(abs(RC_hel) < 0.95 & abs(bNTI) < 2, FALSE, TRUE))

assembly.position.grouped <- assembly.by.position.process %>%
  group_by(habitat, position) %>%
  count(process) %>%
  mutate(proportion = n/sum(n))

assembly.position.grouped <- assembly.position.grouped %>%
  full_join(unique(expand.grid(assembly.position.grouped[,1:3])), fill = 0) %>%
  mutate(proportion = ifelse(is.na(proportion), 0, proportion))

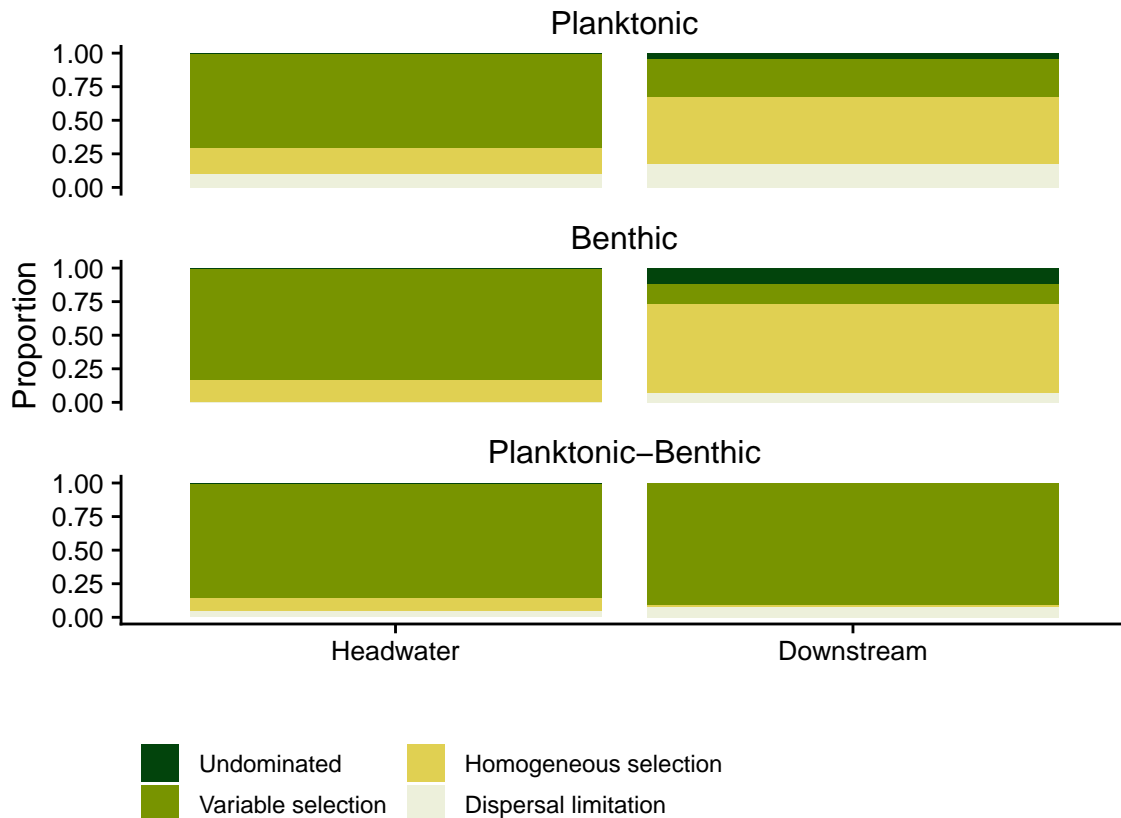
assembly.position.grouped$habitat <- factor(assembly.position.grouped$habitat, levels = c("Planktonic",
assembly.position.grouped %>%
  mutate(position = factor(position, levels = c("Headwater", "Downstream"))) %>%
  mutate(process = factor(process, levels = c("Undominated",
    "Variable selection",
    "Homogeneous selection",

```

```

    "Mass effects",
    "Dispersal limitation")))) %>%
ggplot(aes(x = position, y = proportion, fill = process)) +
geom_bar(alpha = 1, position = "stack", stat = "identity") +
facet_wrap(~ habitat, ncol = 1) +
scale_fill_manual(values = c("#02440c", "#789400", "#e0d052", "#edf0db", "#40becd")) +
labs(x = "", y = "Proportion") +
theme(legend.position = "bottom",
      legend.title = element_blank(),
      legend.text = element_text(size = 9)) +
guides(fill = guide_legend(nrow = 2)) +
ggsave("figures/Fig5.png", height = 8, width = 4, dpi = 600) +
ggsave("figures/Fig5.pdf", height = 8, width = 4)

```



```
assembly.position.grouped %>% count()
```

```

## # A tibble: 6 x 3
## # Groups:   habitat, position [6]
##   habitat      position      n
##   <fct>         <chr>    <int>
## 1 Planktonic    Downstream     4
## 2 Planktonic    Headwater      4
## 3 Benthic       Downstream     4
## 4 Benthic       Headwater      4
## 5 Planktonic-Benthic Downstream     4
## 6 Planktonic-Benthic Headwater      4

```

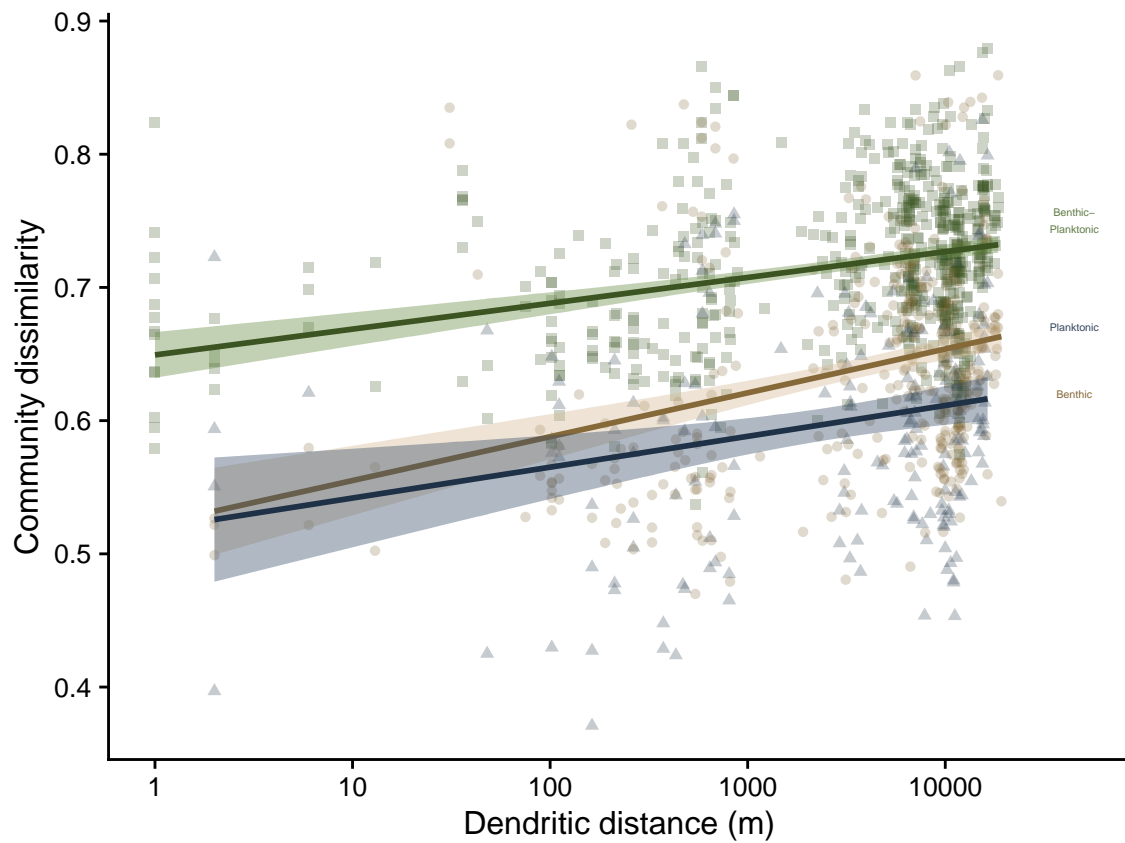
```
assembly.by.position.process %>%
  group_by(position, habitat) %>%
  count(process)
```

```
## # A tibble: 20 x 4
## # Groups:   position, habitat [6]
##   position habitat process n
##   <chr>      <chr>      <chr> <int>
## 1 Downstream Benthic Dispersal limitation 9
## 2 Downstream Benthic Homogeneous selection 79
## 3 Downstream Benthic Undominated 15
## 4 Downstream Benthic Variable selection 17
## 5 Downstream Planktonic Dispersal limitation 49
## 6 Downstream Planktonic Homogeneous selection 137
## 7 Downstream Planktonic Undominated 12
## 8 Downstream Planktonic Variable selection 78
## 9 Downstream Planktonic-Benthic Dispersal limitation 29
## 10 Downstream Planktonic-Benthic Homogeneous selection 8
## 11 Downstream Planktonic-Benthic Undominated 1
## 12 Downstream Planktonic-Benthic Variable selection 346
## 13 Headwater Benthic Homogeneous selection 1
## 14 Headwater Benthic Variable selection 5
## 15 Headwater Planktonic Dispersal limitation 1
## 16 Headwater Planktonic Homogeneous selection 2
## 17 Headwater Planktonic Variable selection 7
## 18 Headwater Planktonic-Benthic Dispersal limitation 1
## 19 Headwater Planktonic-Benthic Homogeneous selection 2
## 20 Headwater Planktonic-Benthic Variable selection 17
```

Make figure 3

```
hja.assembly$hel <- vegdist(OTUsREL.hel, method = "euc")
hja.assembly$bray <- vegdist(OTUsREL, method = "bray")

hja.assembly %>% left_join(simba::liste(den.dists, entry = "unconnected_dists")) %>%
  ggplot(aes(x = unconnected_dists + 1, y = bray, color = habitat, fill = habitat, shape = habitat)) +
  geom_point(alpha = 0.25) +
  geom_smooth(method = "lm") +
  scale_x_log10(lim = c(1, 50000)) +
  scale_color_manual(values=colorspace::darken(c(my.colors, "#688c45"),0.4)) +
  scale_fill_manual(values=c(my.colors, "#688c45", my.colors, "#688c45")) +
  labs(x = "Dendritic distance (m)",
       y = "Community dissimilarity") +
  theme(legend.position = "none") +
  annotate("text", x = 45000, y = 0.75, label = "Benthic-\nPlanktonic", size = 1.5, color = darken("#688c45",.4)) +
  annotate("text", x = 45000, y = 0.67, label = "Planktonic", size = 1.5, color = darken("#3a4f6a",.4)) +
  annotate("text", x = 45000, y = 0.62, label = "Benthic", size = 1.5, color = darken("#d5ba94",.4)) +
  ggsave("figures/Fig3.pdf", width = 8, height = 8*3/4, units = "cm") +
  ggsave("figures/Fig3.png", width = 8, height = 8*3/4, units = "cm", dpi = 600)
```

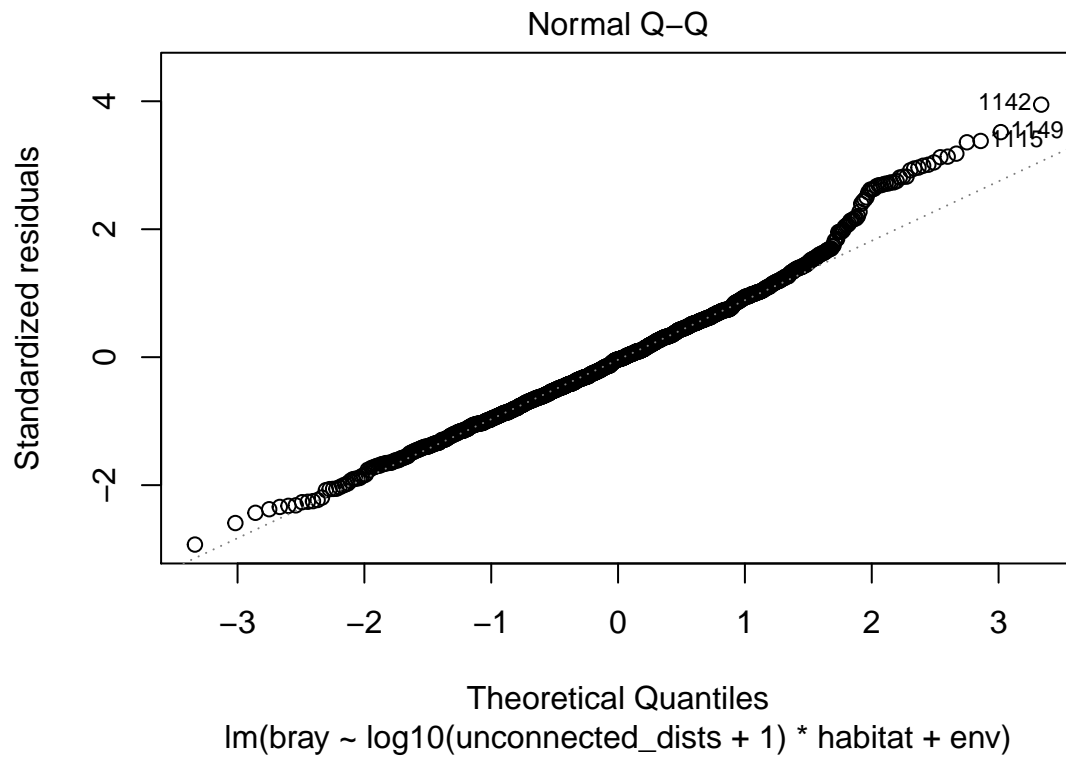
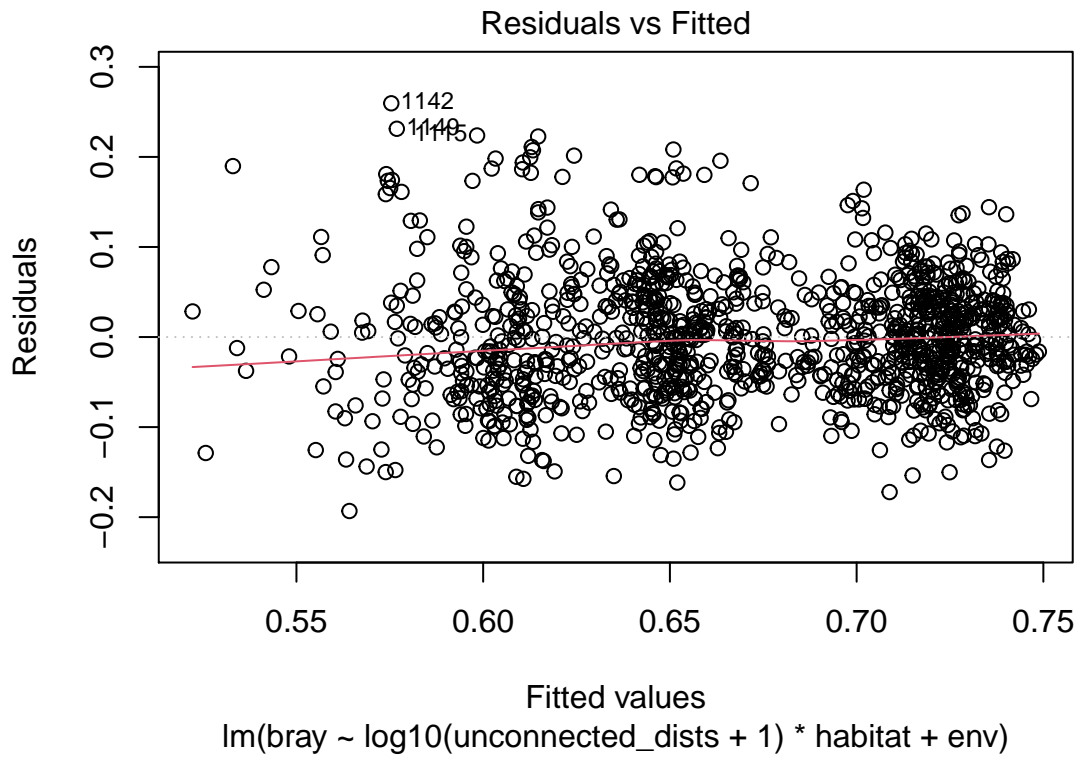


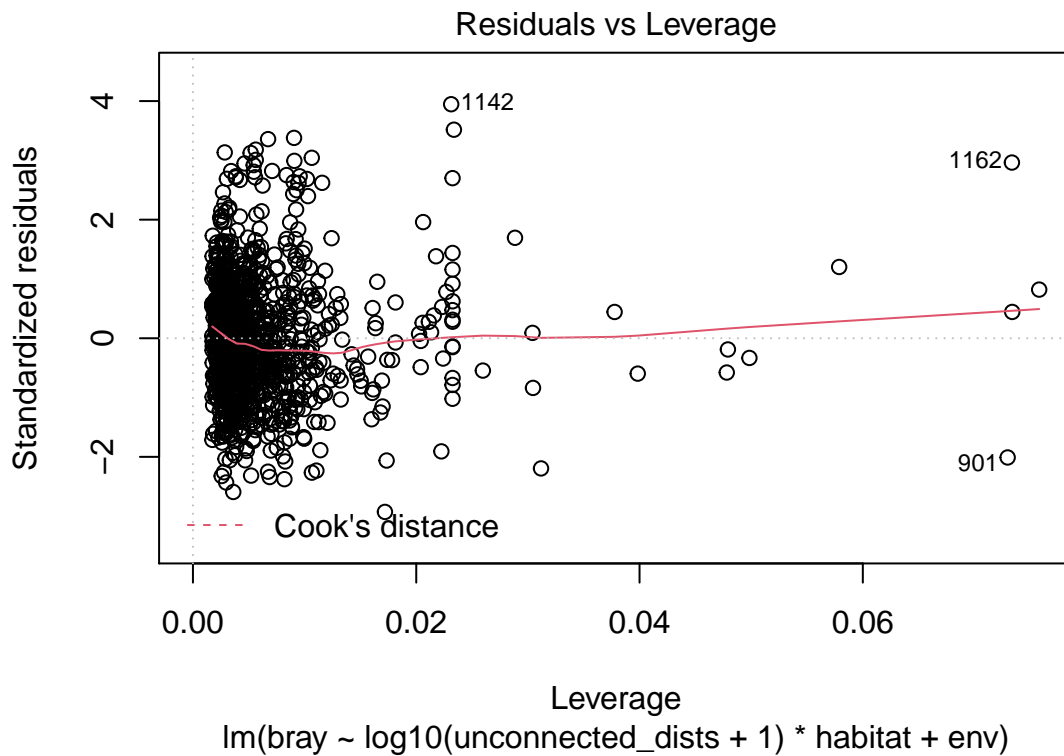
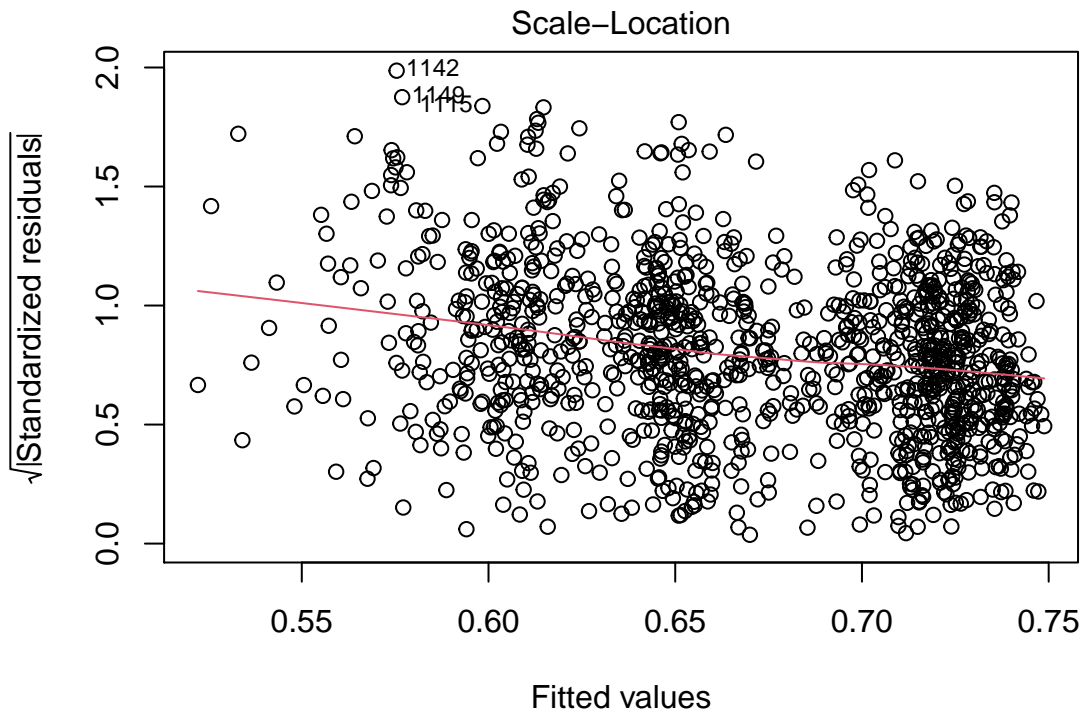
```

ddr_formod <- hja.assembly %>% left_join(simba::liste(den.dists, entry = "unconnected_dists")) %>% mutate(
  ddr_envdist_mod <- lm(bray ~ log10(unconnected_dists+1) * habitat + env, data = ddr_formod)
  AIC(ddr_envdist_mod)

## [1] -3027.704
plot(ddr_envdist_mod)

```





```
summary(DDR_envdist_mod)
```

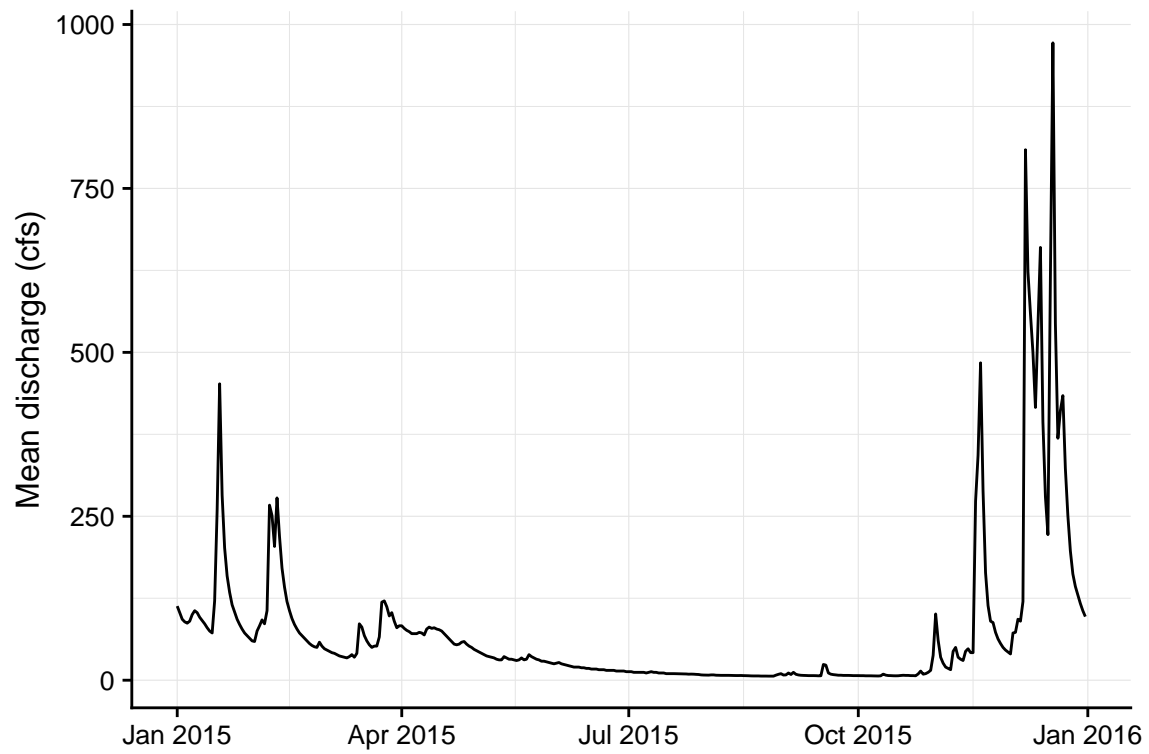
```
##
## Call:
## lm(formula = bray ~ log10(unconnected_dists + 1) * habitat +
##     env, data = ddr_formod)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.193150 -0.044140 -0.002145  0.038978  0.259542
##
## Coefficients:
##                                     Estimate Std. Error
## (Intercept)                       0.516084   0.016717
## log10(unconnected_dists + 1)       0.029122   0.004681
## habitatBenthic                     -0.010557   0.026598
## habitatPlanktonic-Benthic          0.125375   0.019924
## env                                0.005600   0.001475
## log10(unconnected_dists + 1):habitatBenthic -0.007878   0.007418
## log10(unconnected_dists + 1):habitatPlanktonic-Benthic -0.013038   0.005488
##                                     t value Pr(>|t|)
## (Intercept)                      30.873  < 2e-16 ***
## log10(unconnected_dists + 1)       6.221 6.86e-10 ***
## habitatBenthic                     -0.397 0.691505
## habitatPlanktonic-Benthic          6.293 4.40e-10 ***
## env                                3.796 0.000155 ***
## log10(unconnected_dists + 1):habitatBenthic -1.062 0.288440
## log10(unconnected_dists + 1):habitatPlanktonic-Benthic -2.376 0.017680 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06653 on 1169 degrees of freedom
## Multiple R-squared:  0.3764, Adjusted R-squared:  0.3732
## F-statistic: 117.6 on 6 and 1169 DF, p-value: < 2.2e-16
```

```
HF00402_v12 <- read_csv("data/HF00402_v12.csv")
```

```
HF00402_v12 %>%
  filter(SITECODE == "GSLOOK") %>%
  filter(lubridate::year(DATE) == "2015") %>%
  ggplot(aes(x = DATE, y = MEAN_Q)) +
  geom_line() +
  theme(panel.grid.major = element_line(size = .2, color = "gray90"),
        panel.grid.minor = element_line(size = .1, color = "gray90")) +
  labs(x="", y = "Mean discharge (cfs)", subtitle = "Annual hydrograph for Lookout Creek watershed, H.J
```


Annual hydrograph for Lookout Creek watershed, H.J. Andrews LTER



Appendix: Session Info

```
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] colorspace_1.4-1 doSNOW_1.0.18 snow_0.4-3 iterators_1.0.12
## [5] foreach_1.5.0 progress_1.2.2 ggrepel_0.8.2 cowplot_1.1.0
## [9] forcats_0.5.0 dplyr_1.0.2 purrr_0.3.4 readr_1.3.1
## [13] tidyr_1.1.2 tibble_3.0.3 ggplot2_3.3.2 tidyverse_1.3.0
## [17] phytools_0.7-70 maps_3.3.0 picante_1.8.2 nlme_3.1-149
## [21] vegan_2.5-6 lattice_0.20-41 permute_0.9-5 ape_5.4-1
## [25] stringr_1.4.0 pander_0.6.3 adespacial_0.3-8
##
## loaded via a namespace (and not attached):
## [1] readxl_1.3.1 uuid_0.1-4 backports_1.1.10
## [4] fastmatch_1.1-0 plyr_1.8.6 igraph_1.2.5
## [7] lazyeval_0.2.2 sp_1.4-2 splines_4.0.2
## [10] rnc1_0.8.4 digest_0.6.25 htmltools_0.5.0
## [13] gdata_2.18.0 fansi_0.4.1 magrittr_1.5
## [16] cluster_2.1.0 modelr_0.1.8 gmodels_2.18.1
## [19] prettyunits_1.1.1 jpeg_0.1-8.1 blob_1.2.1
## [22] rvest_0.3.6 haven_2.3.1 xfun_0.17
## [25] crayon_1.3.4 jsonlite_1.7.1 simba_0.3-5
## [28] phylobase_0.8.10 phangorn_2.5.5 glue_1.4.2
## [31] gtable_0.3.0 seqinr_3.6-1 adegraphics_1.0-15
## [34] scales_1.1.1 DBI_1.1.0 shapefiles_0.7
## [37] Rcpp_1.0.5.2 plotrix_3.7-8 xtable_1.8-4
## [40] spData_0.3.8 tmvnsim_1.0-2 units_0.6-7
## [43] foreign_0.8-80 spdep_1.1-5 httr_1.4.2
## [46] RColorBrewer_1.1-2 ellipsis_0.3.1 farver_2.0.3
## [49] pkgconfig_2.0.3 XML_3.99-0.5 dbplyr_1.4.4
## [52] deldir_0.1-29 utf8_1.1.4 labeling_0.3
## [55] tidyselect_1.1.0 rlang_0.4.7 reshape2_1.4.4
## [58] later_1.1.0.1 munsell_0.5.0 adephylo_1.1-11
## [61] cellranger_1.1.0 tools_4.0.2 cli_2.0.2
## [64] generics_0.0.2 ade4_1.7-15 SoDA_1.0-6
## [67] broom_0.7.0 evaluate_0.14 fastmap_1.0.1
## [70] yaml_2.2.1 knitr_1.30 fs_1.5.0
## [73] fossil_0.4.0 mime_0.9 adegenet_2.1.3
## [76] xml2_1.3.2 compiler_4.0.2 rstudioapi_0.11
```

## [79]	png_0.1-7	e1071_1.7-3	reprex_0.3.0
## [82]	clusterGeneration_1.3.4	RNeXML_2.4.5	stringi_1.5.3
## [85]	Matrix_1.2-18	classInt_0.4-3	vctrs_0.3.4
## [88]	pillar_1.4.6	LearnBayes_2.15.1	lifecycle_0.2.0
## [91]	combinat_0.0-8	data.table_1.13.0	raster_3.3-13
## [94]	httpuv_1.5.4	R6_2.4.1	latticeExtra_0.6-29
## [97]	promises_1.1.1	KernSmooth_2.23-17	codetools_0.2-16
## [100]	boot_1.3-25	MASS_7.3-53	gtools_3.8.2
## [103]	assertthat_0.2.1	withr_2.3.0	mnormt_2.0.2
## [106]	mgcv_1.8-33	expm_0.999-5	parallel_4.0.2
## [109]	hms_0.5.3	quadprog_1.5-8	grid_4.0.2
## [112]	coda_0.19-3	class_7.3-17	rmarkdown_2.3
## [115]	sf_0.9-6	numDeriv_2016.8-1.1	scatterplot3d_0.3-41
## [118]	shiny_1.5.0	lubridate_1.7.9	