Bacterial community assembly differs between benthic and planktonic stream habitats

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.

```r
# 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 occurances across all sites
OTUs <- OTUs[, which(colSums(OTUs) >= 10)]

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.

```r
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 <- geoXY(env$latitude, env$longitude)
#xy <- project(xy, "+proj=utm +zone=10 +ellps=WGS84")
#dist.mat <- as.matrix(dist(xy, method = "euclidean"))
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.

```r
# 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
# OTUs <- rrarefy(OTUs, sample = min(rowSums(OTUs)))
# OTUs <- OTUs[,-which(colSums(OTUs) == 0)]
# saveRDS(OTUs, file = "temp/site_by_species_rarefied.rda")
# OTUs <- readRDS("temp/site_by_species_rarefied.rda")

# 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. This left a total of 49 sites and 18333 bacterial taxa.

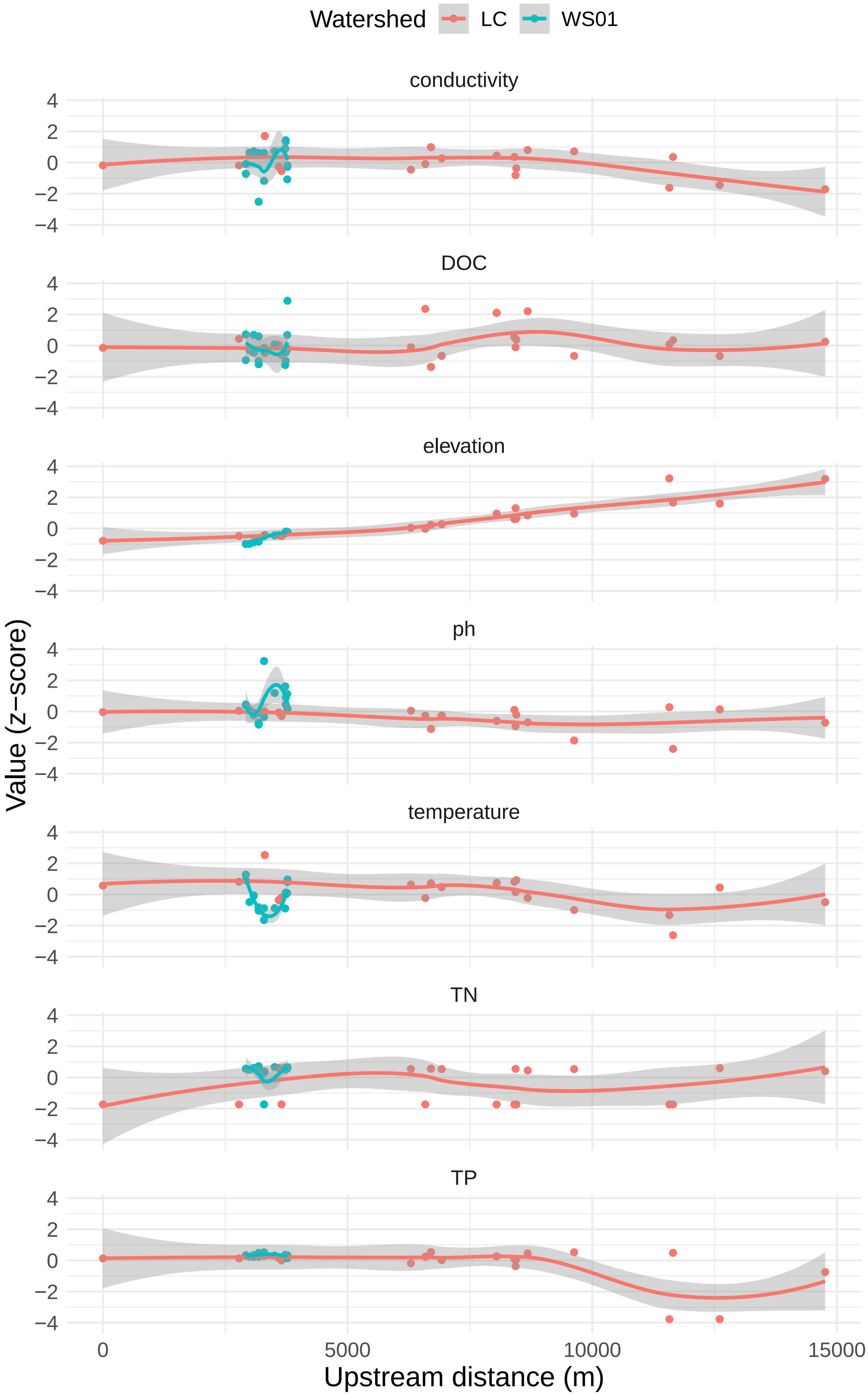
Here, we will read in the phylogenetic tree, root it, and create the unifract distance matrices. I pruned the phylogenetic tree to match only the taxa remaining in the dataset. Then, I rooted the tree using the midpoint method and computed generalized UniFrac distances with a scaling factor of 0.5, along with unweighted and weighted calculations.

```
# 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")

# hja.unifrac <- GUniFrac(otu.tab = OTUs, tree = hja.tree.rooted)$unifracs
# saveRDS(hja.unifrac, file = "temp/hja_unifrac.rda")
hja.unifrac <- readRDS(file = "temp/hja_unifrac.rda")
hja.unifrac.dw <- as.dist(hja.unifrac[,,"d_1"])       # Weighted UniFrac
hja.unifrac.du <- as.dist(hja.unifrac[,,"d_UW"])          # Unweighted UniFrac
hja.unifrac.dv <- as.dist(hja.unifrac[,,"d_VAW"])       # Variance adjusted weighted UniFrac
hja.unifrac.d0 <- as.dist(hja.unifrac[,,"d_0"])     # GUniFrac with alpha 0
hja.unifrac.d5 <- as.dist(hja.unifrac[,,"d_0.5"])       # GUniFrac with alpha 0.5
```

Environmental analysis

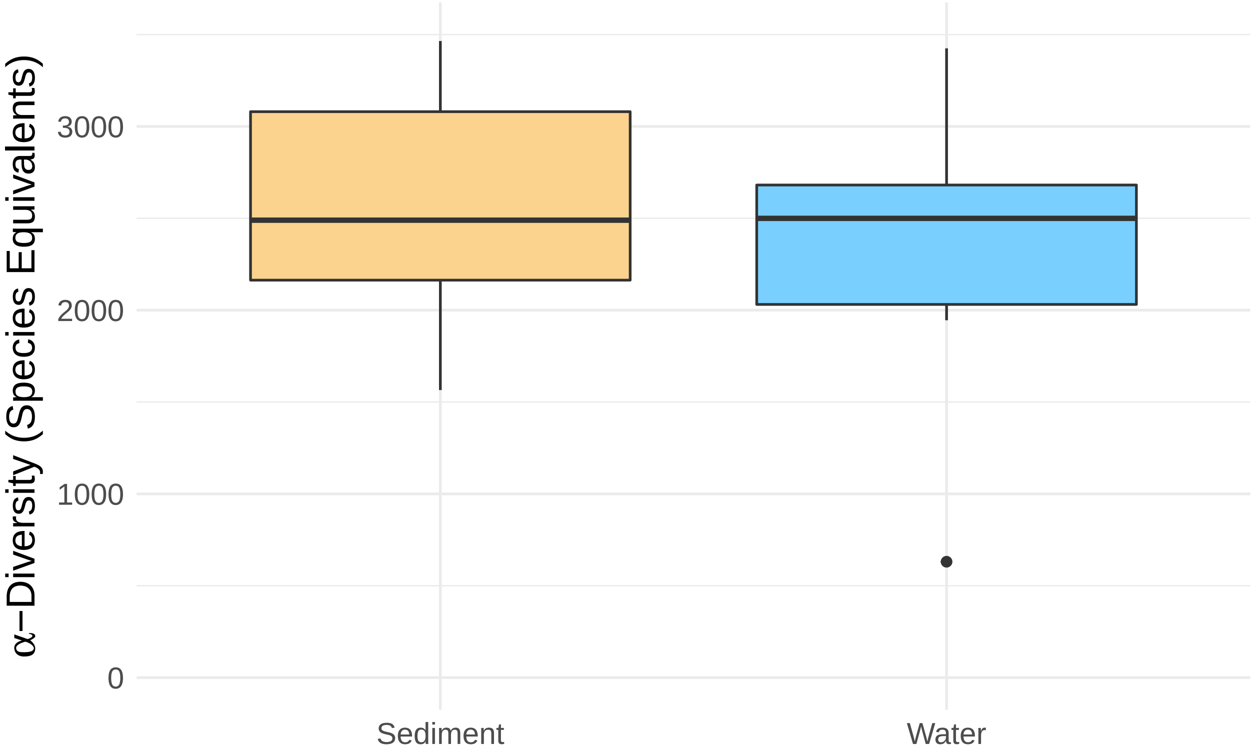
Here, I'll just plot the environmental variables from downstream to upstream across the watershed.

```r
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_wrap(~ variable,  ncol = 1) +
  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")
```


Watershed
LC
WS01
conductivity
DOC
elevation
ph
temperature
TN
TP
4
2
0
−2
−4
Value (z−score)
0
5000
10000
15000
Upstream distance (m)

Diversity analysis

```
alpha.tbl <- tibble(
  habitat = str_to_title(design$habitat),
  upstream = design$upstreamdist,
  N0 = rowSums(OTUsREL.hel > 0),
  N1 = exp(diversity(OTUsREL.hel, index = "shannon")),
  N2 = diversity(OTUsREL, index = "invsimpson")
)

alpha.tbl %>%
  ggplot(aes(x = habitat, y = N1, fill = habitat)) +
  geom_boxplot() +
  labs(x = "", y = expression(paste(alpha, "-Diversity (Species Equivalents)"))) +
  scale_fill_manual(values = (my.colors)) +
  guides(fill = FALSE) +
  scale_y_continuous(limits = c(0, 3500))
```



Beta diversity:

Ordination

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

```
## PCoA Axis 1 explains 16.3 percent of total variation.
## PCoA Axis 2 explains 10.6 percent of total variation.
```

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

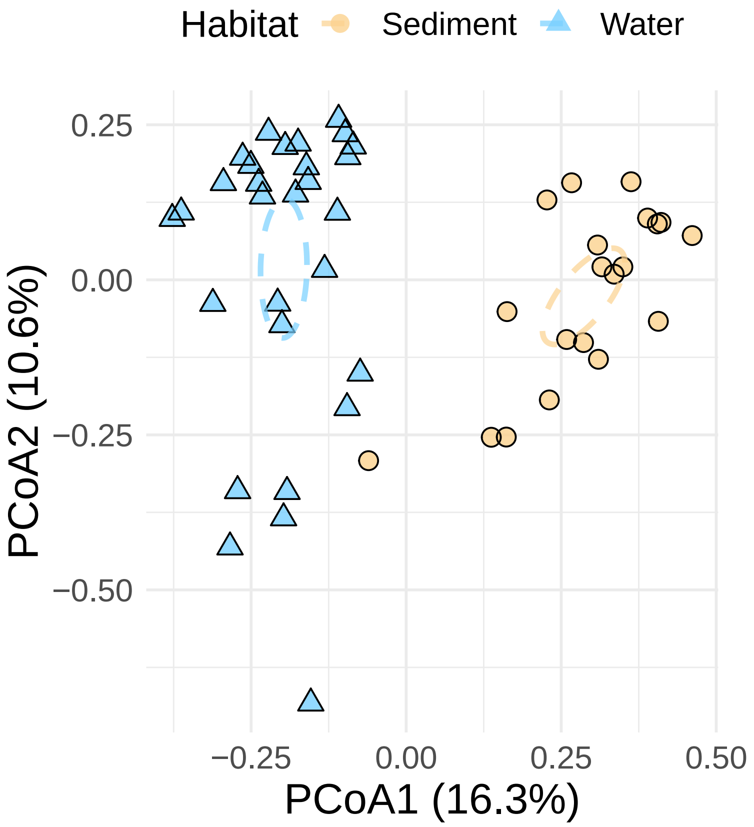
```r
pcoa.plot <- cbind.data.frame(scores(hja.pcoa$pcoa), group = str_to_title(design$habitat))
df_ell <- calc.ellipse(ord = pcoa.plot, ellipse = pcoa.ellipse)

# Run a PERMANOVA
hja.permanova <- adonis(hja.pcoa$dist.matrix ~ design$habitat * design$order, permutations = 999)
hja.permanova$aov.tab %>% pander::pander()
```

Table 1: Permutation: free (continued below)

	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
design$habitat	1	2.859	2.859	8.122	0.1438	0.001
design$order	1	0.7355	0.7355	2.09	0.037	0.006
design$habitat : design$**order**	1	0.4464	0.4464	1.268	0.02246	0.176
Residuals	45	15.84	0.352	NA	0.7967	NA
Total	48	19.88	NA	NA	1	NA

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

```r
ggplot(data = pcoa.plot, aes(Dim1, Dim2)) +
  geom_point(aes(color = group, shape = group), size = 3, alpha = .8) +
  geom_point(data = subset(pcoa.plot, group == "Sediment"), shape = 1, color = "black", size = 3) +
  geom_point(data = subset(pcoa.plot, group == "Water"), shape = 2, color = "black", size = 3) +
  geom_path(data = df_ell,
            aes(x = Dim1, y = Dim2, color = group),
            size = 1, alpha = 0.7, linetype = 2) +
  labs(x = paste0("PCoA1 (", hja.pcoa$var1, "%)"),
    y = paste0("PCoA2 (", hja.pcoa$var2, "%)"),
    color = "Habitat", shape = "Habitat") +
  scale_color_manual(values = my.colors) +
  coord_fixed()
```



LCBD and SCBD

Now, I'm going calculate the total beta diversity in the samples, and calcluate the local contributions to beta diversity (LCBD) and species contributions to beta diversity (SCBD). LCBD may be highest in more isolated reaches of the stream network

```
otu.beta <- beta.div(OTUs, method = "hellinger", nperm = 9999)
```

```
otu.beta$beta # max is 1
```

```
##    SStotal    BDtotal
## 19.8790475  0.4141468
```

```
# which taxa contribute most to beta diversity?
OTU.tax[order(otu.beta$SCBD[otu.beta$SCBD > mean(otu.beta$SCBD)],
              decreasing = T)[1:10],-c(1,2)] %>%
  remove_rownames() %>% pander()
```

Table 3: Table continues below

Phylum	Class	Order
Proteobacteria	Gammaproteobacteria	Pseudomonadales
Proteobacteria	Alphaproteobacteria	Sphingomonadales
Proteobacteria	Proteobacteria_unclassified	Proteobacteria_unclassified
Actinobacteria	Actinobacteria	Actinomycetales
Proteobacteria	Gammaproteobacteria	Pseudomonadales
Proteobacteria	Betaproteobacteria	Burkholderiales
Proteobacteria	Alphaproteobacteria	Rhizobiales

Phylum	Class	Order
Proteobacteria	Gammaproteobacteria	Enterobacteriales
Actinobacteria	Actinobacteria	Actinomycetales
Proteobacteria	Betaproteobacteria	Burkholderiales

Family	Genus
Pseudomonadaceae	Pseudomonas
Sphingomonadaceae	Sphingomonas
Proteobacteria_unclassified	Proteobacteria_unclassified
Micrococcaceae	Arthrobacter
Moraxellaceae	Acinetobacter
Oxalobacteraceae	Massilia
Methylobacteriaceae	Methylobacterium
Enterobacteriaceae	Yersinia
Micrococcaceae	Kocuria
Comamonadaceae	Rhodoferax

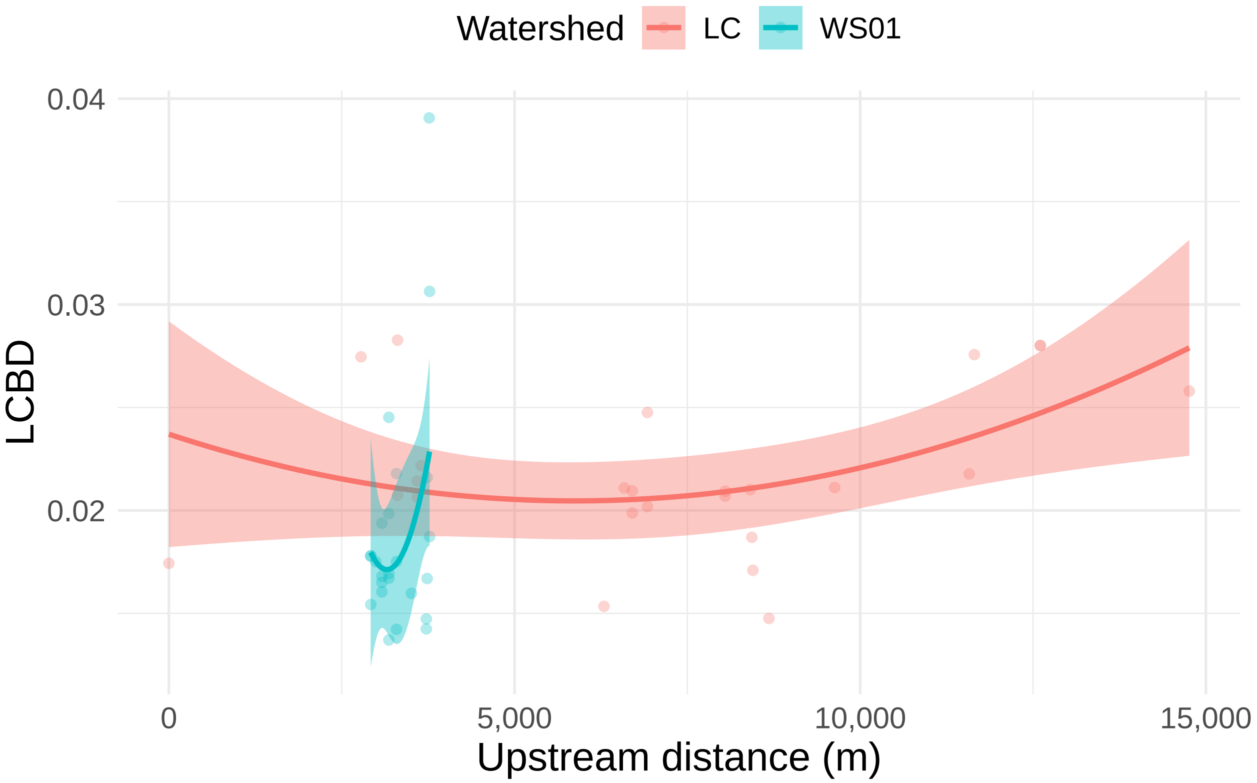
```r
row.names(OTUs[which(otu.beta$p.adj <= 0.05),])
```

```
##  [1] "LC_03_W" "LC_10_W" "LC_16_W" "LC_18_S" "LC_18_W" "LC_19_S" "LC_20_W"
##  [8] "W1_06_S" "W1_17_W" "W1_19_S"
```

```r
design[which(otu.beta$p.adj <= 0.05),]
```

```
##         watershed  site  habitat elev order   flow upstreamdist
## LC_03_W        LC LC_03    water  542     5   <NA>         2780
## LC_10_W        LC LC_10    water  680     3   <NA>         6922
## LC_16_W        LC LC_16    water  554     3   <NA>         3308
## LC_18_S        LC LC_18 sediment  922     3   <NA>        12605
## LC_18_W        LC LC_18    water  922     3   <NA>        12605
## LC_19_S        LC LC_19 sediment  932     2   <NA>        11651
## LC_20_W        LC LC_20    water 1210     1   <NA>        14760
## W1_06_S      WS01 W1_06 sediment  489     2   pool         3182
## W1_17_W      WS01 W1_17    water  581     1 riffle         3766
## W1_19_S      WS01 W1_19 sediment  591     1 riffle         3771
```

```r
beta.tbl <- cbind.data.frame(
  design,
  LCBD = otu.beta$LCBD,
  pval = otu.beta$p.adj)
beta.tbl %>%
  ggplot(aes(x = upstreamdist, y = LCBD, color = watershed, fill = watershed)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2)) +
  scale_x_continuous(labels = scales::comma) +
  labs(x = "Upstream distance (m)", color = "Watershed", fill = "Watershed")
```



We observed $BD_{total} = 0.414$ out of 1.

Appendix: Session Info

```r
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.2
## 
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/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] vegetarian_1.2     bindrcpp_0.2.2     forcats_0.3.0     
##  [4] dplyr_0.7.8        purrr_0.2.5        readr_1.3.1       
##  [7] tidyr_0.8.2        tibble_1.4.2       ggplot2_3.1.0     
## [10] tidyverse_1.2.1    GUniFrac_1.1       matrixStats_0.54.0
## [13] phytools_0.6-60    maps_3.3.0         picante_1.7       
## [16] nlme_3.1-137       ape_5.2            stringr_1.3.1     
## [19] pander_0.6.3       adespatial_0.3-2   vegan_2.5-3       
## [22] lattice_0.20-38    permute_0.9-4     
## 
## loaded via a namespace (and not attached):
##   [1] colorspace_1.3-2        seqinr_3.4-5           
##   [3] deldir_0.1-15           rstudioapi_0.8         
##   [5] lubridate_1.7.4         xml2_1.2.0             
##   [7] codetools_0.2-15        splines_3.5.2          
##   [9] mnormt_1.5-5            knitr_1.21             
##  [11] ade4_1.7-13             jsonlite_1.6           
##  [13] broom_0.5.1             phylobase_0.8.4        
##  [15] cluster_2.0.7-1         shiny_1.2.0            
##  [17] compiler_3.5.2          httr_1.4.0             
##  [19] adegraphics_1.0-15      backports_1.1.3        
##  [21] assertthat_0.2.0        Matrix_1.2-15          
##  [23] lazyeval_0.2.1          cli_1.0.1              
##  [25] later_0.7.5             htmltools_0.3.6        
##  [27] prettyunits_1.0.2       tools_3.5.2            
##  [29] igraph_1.2.2            coda_0.19-2            
##  [31] gtable_0.2.0            glue_1.3.0             
##  [33] reshape2_1.4.3          clusterGeneration_1.3.4
##  [35] gmodels_2.18.1          fastmatch_1.1-0        
##  [37] Rcpp_1.0.0              cellranger_1.1.0       
##  [39] spdep_0.8-1             gdata_2.18.0           
##  [41] xfun_0.4                adephylo_1.1-11        
##  [43] rvest_0.3.2             mime_0.6               
##  [45] phangorn_2.4.0          gtools_3.8.1           
##  [47] XML_3.98-1.16           LearnBayes_2.15.1      
##  [49] MASS_7.3-51.1           scales_1.0.0
##  [51] simba_0.3-5             hms_0.4.2
##  [53] promises_1.0.1          parallel_3.5.2
##  [55] expm_0.999-3            animation_2.6
##  [57] RColorBrewer_1.1-2      yaml_2.2.0
##  [59] latticeExtra_0.6-28     stringi_1.2.4
##  [61] plotrix_3.7-4           boot_1.3-20
##  [63] spData_0.2.9.6          rlang_0.3.0.1
##  [65] pkgconfig_2.0.2         rncl_0.8.3
##  [67] evaluate_0.12           bindr_0.1.1
##  [69] labeling_0.3            tidyselect_0.2.5
##  [71] plyr_1.8.4              magrittr_1.5
##  [73] R6_2.3.0                generics_0.0.2
##  [75] fossil_0.3.7            combinat_0.0-8
##  [77] foreign_0.8-71          withr_2.1.2
##  [79] pillar_1.3.1            haven_2.0.0
##  [81] mgcv_1.8-26             shapefiles_0.7
##  [83] scatterplot3d_0.3-41    sp_1.3-1
##  [85] modelr_0.1.2            crayon_1.3.4
##  [87] uuid_0.1-2              KernSmooth_2.23-15
##  [89] rmarkdown_1.11          progress_1.2.0
##  [91] RNeXML_2.2.0            adegenet_2.1.1
##  [93] grid_3.5.2              readxl_1.2.0
##  [95] data.table_1.11.8       digest_0.6.18
##  [97] xtable_1.8-3            httpuv_1.4.5.1
##  [99] numDeriv_2016.8-1       munsell_0.5.0
## [101] quadprog_1.5-5
```