

# Residence time structures microbial communities through niche partitioning

Emmi A. Mueller and Jay T. Lennon

2025-02-13

## Initial Setup

### Load packages

```
# devtools::install_github('BlakeRMills/MoMAColors')
library("MoMAColors")

library("vegan")
library("ggplot2")
library("lemon")
library("cowplot")
library("scales")
library("tidyr")
library("BiodiversityR")
library("dplyr")
library("mgcv")
library("tidymv")
library("spaa")
library("lsr")
library("table1")
library("caper")
library("ape")
```

### Set up figure theme

```
my.cols <- RColorBrewer::brewer.pal(n = 4, name = "Greys")[3:4]

# Set theme for figures in the paper
theme_set(theme_classic() + theme(axis.title = element_text(size = 16),
  axis.title.x = element_text(margin = margin(t = 15, b = 15)),
  axis.title.y = element_text(margin = margin(l = 15, r = 15)),
  axis.text = element_text(size = 14), axis.text.x = element_text(margin = margin(t = 5)),
  axis.text.y = element_text(margin = margin(r = 5)), axis.line.x = element_blank(),
  axis.line.y = element_blank(), axis.ticks.x = element_line(linewidth = 1),
  axis.ticks.y = element_line(linewidth = 1), axis.ticks.length = unit(0.1,
    "in"), panel.border = element_rect(color = "black", fill = NA,
    linewidth = 1.5), legend.text = element_text(size = 10),
  strip.text = element_text(size = 14), strip.background = element_blank()))
```

## Define inputs

```
# Tau = general design file for experiment, paired Day 0
# and 20
Tau <- read.csv("data/RTLC/2021_RTLC_Tau_Combined.csv", header = TRUE)
Tau.ctrl <- Tau[50:51, ]
Tau <- Tau[1:49, ]
Tau$Tau <- as.numeric(Tau$Tau)
Tau$Set <- as.character(Tau$Set)
Tau$Day_0_Seq[Tau$Day_0_Seq == ""] <- NA
Tau$Day_20_Seq[Tau$Day_20_Seq == ""] <- NA

# Tau.Seq = Full list of sequenced samples, non-paired Day
# 0 and 20
Tau.Seq <- read.csv("data/RTLC/2021_RTLC_Tau_Seq.csv", header = TRUE)

# Abundance (N) data
Tau <- N.fxn(read.csv("data/RTLC/RTLC_S2/20210628_RTLC_S2_N_M1gate.csv",
  header = TRUE), Tau)
Tau <- N.fxn(read.csv("data/RTLC/RTLC_S3/20210723_RTLC_S3_N_M1gate.csv",
  header = TRUE), Tau)
Tau <- N.fxn(read.csv("data/RTLC/RTLC_S4/20210904_RTLC_S4_N_M1gate.csv",
  header = TRUE), Tau)

# Biofilm data
Tau <- OT.fxn(read.csv("data/RTLC/RTLC_S1/20210602_RTLC_S1_0toole.csv",
  header = TRUE), Tau)
Tau <- OT.fxn(read.csv("data/RTLC/RTLC_S2/20210628_RTLC_S2_0toole.csv",
  header = TRUE), Tau)
Tau <- OT.fxn(read.csv("data/RTLC/RTLC_S3/20210723_RTLC_S3_0toole.csv",
  header = TRUE), Tau)
Tau <- OT.fxn(read.csv("data/RTLC/RTLC_S4/20210904_RTLC_S4_0toole.csv",
  header = TRUE), Tau)

# Bacterial Productivity (BP) data
print("Scintillation counter machine efficiency-")

## [1] "Scintillation counter machine efficiency-"
Tau <- as.data.frame(BP.fxn(read.csv("data/RTLC/RTLC_S1/20210602_RTLC_S1_BP.csv",
  header = FALSE), Tau, 1))

## [1] "Machine efficiency: 0.636787567178765"
Tau <- as.data.frame(BP.fxn(read.csv("data/RTLC/RTLC_S2/20210628_RTLC_S2_BP.csv",
  header = FALSE), Tau, 2))

## [1] "Machine efficiency: 0.638772848217168"
Tau <- as.data.frame(BP.fxn(read.csv("data/RTLC/RTLC_S3/20210723_RTLC_S3_BP.csv",
  header = FALSE), Tau, 3))

## [1] "Machine efficiency: 0.643492469287154"
Tau <- as.data.frame(BP.fxn(read.csv("data/RTLC/RTLC_S4/20210904_RTLC_S4_BP.csv",
  header = FALSE), Tau, 4))
```

```
## [1] "Machine efficiency: 0.63773229695431"

Tau$ind_P <- Tau$uMChr/Tau$N
Tau$ind_P_g <- (Tau$gCLhr/1000)/Tau$N
Tau$turnover_t <- (2.6e-14)/Tau$ind_P_g

# Convert tau from min to h and categorize tau by flow rate
# manipulation method (Pump)
Tau$Set <- as.factor(Tau$Set)
Tau$Tau <- log(((10^Tau$Tau)/60), 10)
Tau$Pump <- Tau$Tau < 1.6

Tau.Seq$Tau <- as.numeric(Tau.Seq$Tau)
Tau.Seq$Tau <- log(((10^Tau.Seq$Tau)/60), 10)
Tau.Seq$Day <- as.factor(Tau.Seq$Day)

# Tau.20 = Tau subset to only samples with Day 20
# sequencing
Tau.20 <- subset(Tau[is.na(Tau$Day_20_Seq) == 0, ])
```

## Load EcoPlate (EP) data (48 hours)

```
EP.S1 <- read.csv("data/RTLC/RTLC_S1/eco.data_rt_S1_48.txt",
  header = TRUE, sep = "\t")
EP.S1 <- cbind(EP.S1[, 1:2], EP.S1[, 4:34])

EP.S2 <- read.csv("data/RTLC/RTLC_S2/eco.data_rt_S2_48.txt",
  header = TRUE, sep = "\t")
EP.S2 <- cbind(EP.S2[, 1:2], EP.S2[, 4:34])

EP.S3 <- read.csv("data/RTLC/RTLC_S3/eco.data_rt_S3_48.txt",
  header = TRUE, sep = "\t")
EP.S3 <- cbind(EP.S3[, 1:2], EP.S3[, 4:34])

EP.S4 <- read.csv("data/RTLC/RTLC_S4/eco.data_rt_S4_48.txt",
  header = TRUE, sep = "\t")
EP.S4 <- cbind(EP.S4[, 1:2], EP.S4[, 4:34])

EP <- rbind(EP.S1, EP.S2, EP.S3, EP.S4)

rm(EP.S1, EP.S2, EP.S3, EP.S4)

ResType <- read.csv("code/resource_type.txt", header = TRUE,
  sep = "\t")

colnames(EP) <- c("Tau", "Set", "2-Hydroxy.Benzoic.Acid", "4-Hydroxy.Benzoic.Acid",
  "alpha-Cyclodextrin", "alpha-D-Lactose", "alpha-Ketobutyric.Acid",
  "beta-Methyl-D-Glucoside", "D-Cellobiose", "D-Galactonic.Acid.gamma-Lactone",
  "D-Galacturonic.Acid", "D-Glucosaminic.Acid", "D-Malic.Acid",
  "D-Mannitol", "D-Xylose", "D,L-alpha-Glycerol.Phosphate",
  "gamma-Hydroxybutyric.Acid", "Glucose-1-Phosphate", "Glycogen",
  "Glycyl-L-Glutamic.Acid", "i-Erythritol", "Itaconic.Acid",
```

```

    "L-Arginine", "L-Asparagine", "L-Phenylalanine", "L-Serine",
    "L-Threonine", "N-Acetyl-D-Glucosamine", "Phenylethylamine",
    "Putrescine", "Pyruvic.Acid.Methyl.Ester", "Tween.40", "Tween.80")

EP$Tau <- log(((10^EP$Tau)/60), 10)

EP.long <- EP %>%
  gather(C_Source, OD_48, "2-Hydroxy.Benzoic.Acid":"Tween.80")

ResType <- distinct(ResType)

EP.long$Type <- NA

for (row in rownames(EP.long)) {
  EP.long[row, "Type"] <- ResType$Type[ResType$Resource ==
    EP.long[row, "C_Source"]]
}

x <- 1
while (x < nrow(EP) + 1) {
  EP[x, "NumRes"] <- sum(EP[x, 3:33] > 0.125)
  EP[x, "AvgRes"] <- mean(unlist(EP[x, 3:33]))
  x <- x + 1
}

Tau$NumRes <- EP$NumRes
Tau$AvgRes <- EP$AvgRes

# EP.env <- subset(EP, select = c(Tau, Set)) EP.env$Set <-
# c(rep(1, 13), rep(2,12), rep(3,10), rep(4,14))
rownames(EP) <- EP$Tau

```

## Clean and rarefy the OTU table

```

# Import Shared Files - OTUs and ASVs
OTUs <- read.otu(shared = "mothur/RTLCA+B/RTLCA.final.shared",
  cutoff = "0.03")
rownames(OTUs)[rownames(OTUs) == "GSF3365_RTLCA"] <- "GSF3365_RTLCA_002"

ASVs <- read.csv("data/RTLCA_ASV.csv", header = TRUE, sep = ",")
ASVs$X <- sub("-", "_", ASVs$X)
rownames(ASVs) <- ASVs$X
ASVs <- ASVs[, -1]

# Import Taxonomy
OTUs.tax <- read.tax(taxonomy = "mothur/RTLCA_A+B/RTLCA.final.taxonomy",
  format = "rdp")

# Remove samples with fewer than 10000 reads
coverage <- rowSums(OTUs)
cutoff <- 10000

```

```

lows <- which(coverage < cutoff)
if (length(lows) > 0) {
  OTUs <- OTUs[-which(coverage < cutoff), ]
}

# Remove OTUs with less than 2 reads across all samples
OTUs <- OTUs[, which(colSums(OTUs) > 2)]

# Generate rarefaction plot
otu.min <- min(rowSums(OTUs))
asv.min <- min(rowSums(ASVs))

# rarefy OTUs and ASVs to the sample with the lowest number
# of reads then remove OTUs with 0 reads after rarefaction
set.seed(47405)
OTUs.r <- rrarefy(OTUs, otu.min)
OTUs.r <- OTUs.r[, -which(colSums(OTUs.r) == 0)]
rm(OTUs)

ASVs.r <- rrarefy(ASVs, asv.min)
ASVs.r <- ASVs.r[, -which(colSums(ASVs.r) == 0)]
rm(ASVs)

# Create dataframes of Day 20 samples rarified
OTUs.r.20 <- OTUs.r[rownames(OTUs.r) %in% unique(Tau$Day_20_Seq),
]
OTUs.r.20 <- OTUs.r.20[, which(colSums(OTUs.r.20) > 0)]
rownames(OTUs.r.20) <- subset(Tau.Seq, Day == 20)$Tau

ASVs.r.20 <- ASVs.r[rownames(ASVs.r) %in% unique(Tau$Day_20_Seq),
]
ASVs.r.20 <- ASVs.r.20[, which(colSums(ASVs.r.20) > 0)]
rownames(ASVs.r.20) <- subset(Tau.Seq, Day == 20)$Tau

# Generate OTU tables with p/a, relative abundance
OTUs.r.REL <- decostand(OTUs.r, method = "total")
OTUs.r.REL.20 <- OTUs.r.REL[rownames(OTUs.r.REL) %in% unique(Tau$Day_20_Seq),
]
OTUs.r.REL.20 <- OTUs.r.REL.20[, which(colSums(OTUs.r.REL.20) >
0)]
rownames(OTUs.r.REL.20) <- subset(Tau.Seq, Day == 20)$Tau

OTUs.r.PA <- decostand(OTUs.r, method = "pa")
OTUs.r.PA.20 <- OTUs.r.PA[rownames(OTUs.r.PA) %in% unique(Tau$Day_20_Seq),
]
OTUs.r.PA.20 <- OTUs.r.PA.20[, which(colSums(OTUs.r.PA.20) >
0)]
rownames(OTUs.r.PA.20) <- subset(Tau.Seq, Day == 20)$Tau

OTUs.tax.20 <- OTUs.tax[OTUs.tax$OTU %in% unique(colnames(OTUs.r.20)),
]

```

```

# Calculate richness (S) and evenness (SimpE) for OTUs (S &
# SimpE) and ASVs (S.ASV & SimpE.ASV)
S <- as.data.frame(cbind(row.names(OTUs.r), unname(S.cal(OTUs.r))))
S.ASV <- as.data.frame(cbind(row.names(ASVs.r), unname(S.cal(ASVs.r))))

SimpE <- as.data.frame(cbind(row.names(OTUs.r), unname(SimpE.cal(OTUs.r))))
SimpE.ASV <- as.data.frame(cbind(row.names(ASVs.r), unname(SimpE.cal(ASVs.r))))

```

**Figure 1. Abundance, productivity, and resource consumption of microbial communities along a residence time gradient.**

## B. Microbial abundance

```

N.gam <- gam(log_N ~ s(Tau), family = gaussian(link = "identity"),
  data = Tau, method = "REML")
N.gam.re <- gam(log_N ~ s(Tau) + s(Set, bs = "re"), family = gaussian(link = "identity"),
  data = Tau, method = "REML")

AIC(N.gam, N.gam.re)

##              df      AIC
## N.gam      6.447302 33.06401
## N.gam.re  9.736133 16.89868

anova(N.gam, N.gam.re, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: log_N ~ s(Tau)
## Model 2: log_N ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev    Df Deviance Pr(>Chi)
## 1      29.684      3.6910
## 2      25.998      1.9624  3.6865    1.7287 3.332e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Random effect of set is significant
rm(N.gam)

summary(N.gam.re)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## log_N ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.115      0.153    46.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

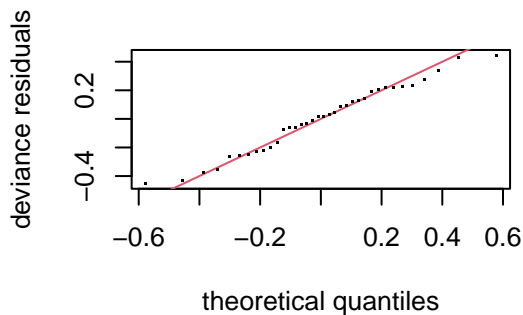
```
## Approximate significance of smooth terms:
##           edf Ref.df      F  p-value
## s(Tau) 4.693  5.771 17.04 < 2e-16 ***
## s(Set) 1.777  2.000  9.70 0.000235 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.759   Deviance explained = 80.3%
## -REML = 13.189   Scale est. = 0.068783   n = 36
mean(summary(N.gam.re)$s.table[, 4])
```

```
## [1] 0.0001177233
```

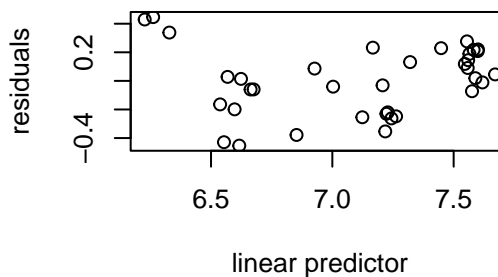
```
k.check(N.gam.re)
```

```
##           k'      edf k-index p-value
## s(Tau)   9 4.693224 1.276555  0.915
## s(Set)   3 1.776724      NA      NA
```

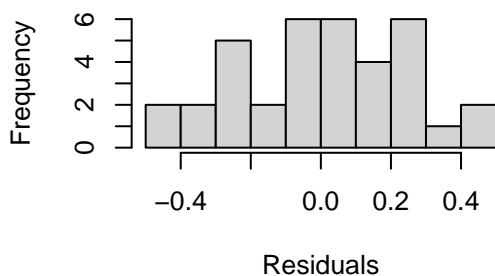
```
gam.check(N.gam.re)
```



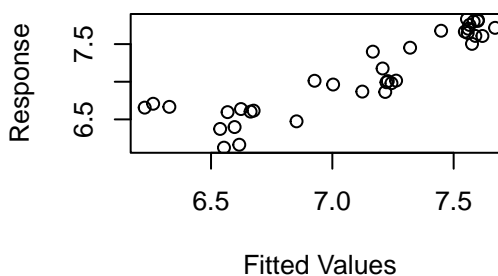
**Resids vs. linear pred.**



**Histogram of residuals**



**Response vs. Fitted Values**

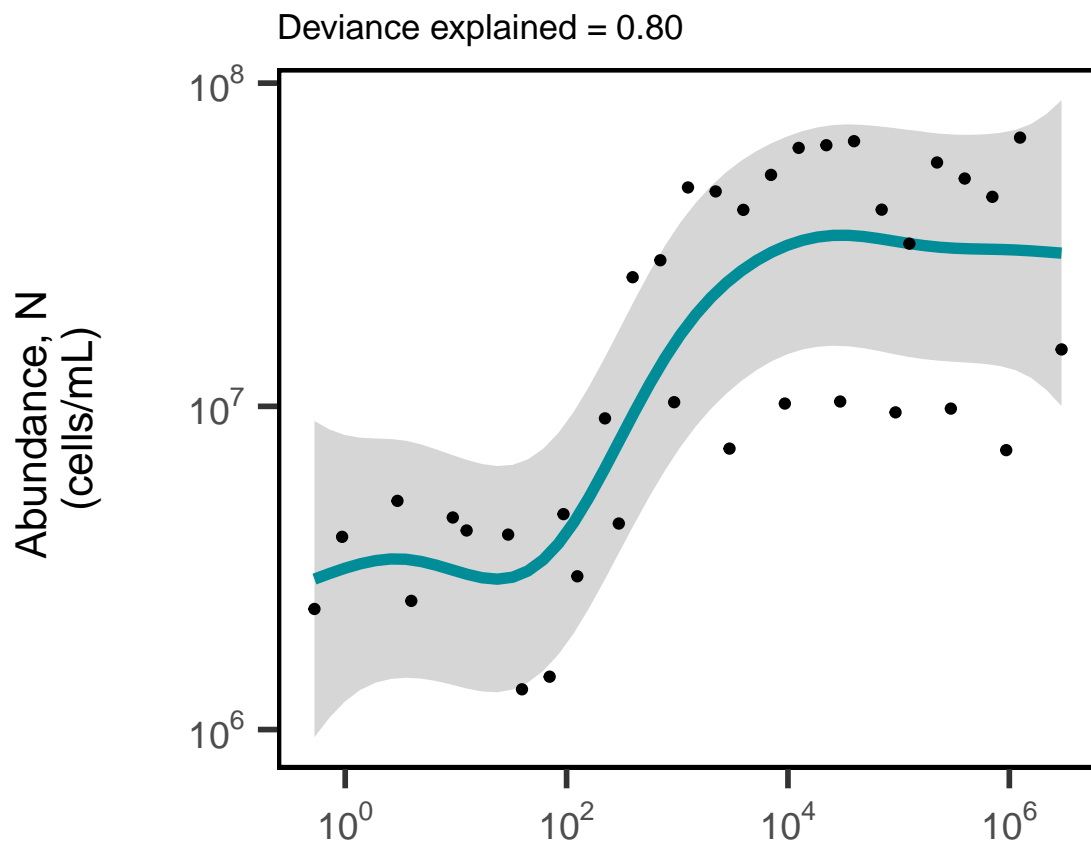


```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-6.707242e-06,7.847398e-07]
## (score 13.18909 & scale 0.06878258).
## Hessian positive definite, eigenvalue range [0.4752446,17.25415].
```

```
## Model rank = 13 / 13
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'   edf k-index p-value
## s(Tau) 9.00 4.69    1.28    0.94
## s(Set) 3.00 1.78      NA      NA

N_Tau <- tidymv::predict_gam(N.gam.re, exclude_terms = "s(Set)") %>%
  filter(Set == "2") %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = Tau, aes(x = Tau,
    y = log_N)) + theme(axis.title.x = element_blank()) + ylab("Abundance, N \n(cells/mL)") +
  scale_x_continuous(labels = label_math(10^.x)) + scale_y_continuous(breaks = c(6,
    7, 8), labels = label_math(10^.x)) + labs(title = bquote("Deviance explained =" ~
    .(round_pad(summary(N.gam.re)$dev, 2)))) + theme(plot.margin = unit(c(0.2,
    0.2, 0.2, 0.8), "cm")) + theme(axis.title.y = element_text(vjust = 6))

N_Tau
```



```
ggsave("./output/N_Tau.pdf")
ggsave("./output/N_Tau.png", width = 7, height = 5)
```



### C. Biomass Production (uM C/hr)

```
BP.gam.re <- gam(log(uMChr, 10) ~ s(Tau) + s(Set, bs = "re"),
  family = gaussian(link = "identity"), data = Tau, method = "REML")
BP.gam <- gam(log(uMChr, 10) ~ s(Tau), family = gaussian(link = "identity"),
  data = Tau, method = "REML")
anova(BP.gam, BP.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: log(uMChr, 10) ~ s(Tau)
## Model 2: log(uMChr, 10) ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev    Df Deviance Pr(>Chi)
## 1      41.310      0.89180
## 2      38.089      0.53545 3.2212  0.35635 8.564e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(BP.gam, BP.gam.re)
```

```
##           df      AIC
## BP.gam      7.597687 -42.05894
## BP.gam.re 10.503073 -61.24511
```

```
anova(BP.gam, BP.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: log(uMChr, 10) ~ s(Tau)
## Model 2: log(uMChr, 10) ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev    Df Deviance Pr(>Chi)
## 1      41.310      0.89180
## 2      38.089      0.53545 3.2212  0.35635 8.564e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Random effect of set is significant
```

```
rm(BP.gam)
```

```
summary(BP.gam.re)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(uMChr, 10) ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.1159      0.0536  -39.48  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df      F  p-value
```

```
## s(Tau) 5.042 6.161 96.966 < 2e-16 ***
## s(Set) 2.666 3.000 8.319 7.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.929   Deviance explained = 94%
## -REML = -22.399   Scale est. = 0.013289   n = 49
```

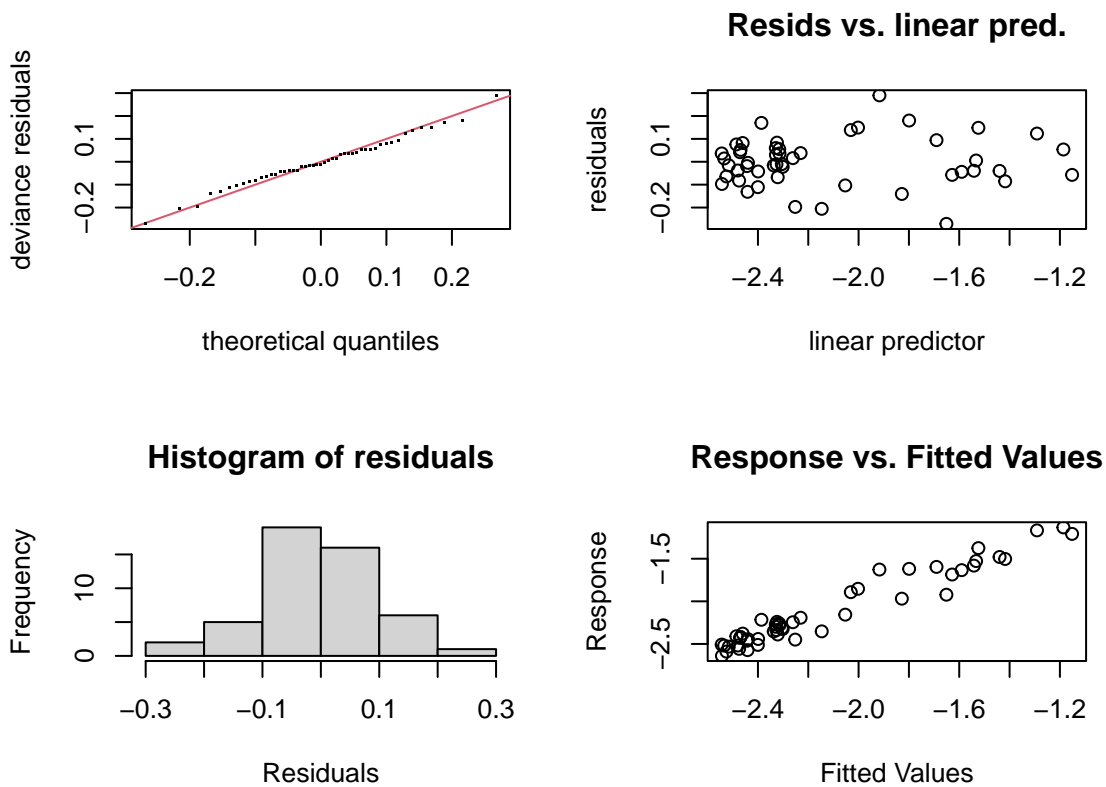
```
mean(summary(BP.gam.re)$s.table[, 4])
```

```
## [1] 3.622457e-05
```

```
k.check(BP.gam.re)
```

```
##          k'      edf  k-index p-value
## s(Tau)   9 5.042344 1.137607  0.765
## s(Set)   4 2.666262      NA      NA
```

```
gam.check(BP.gam.re)
```

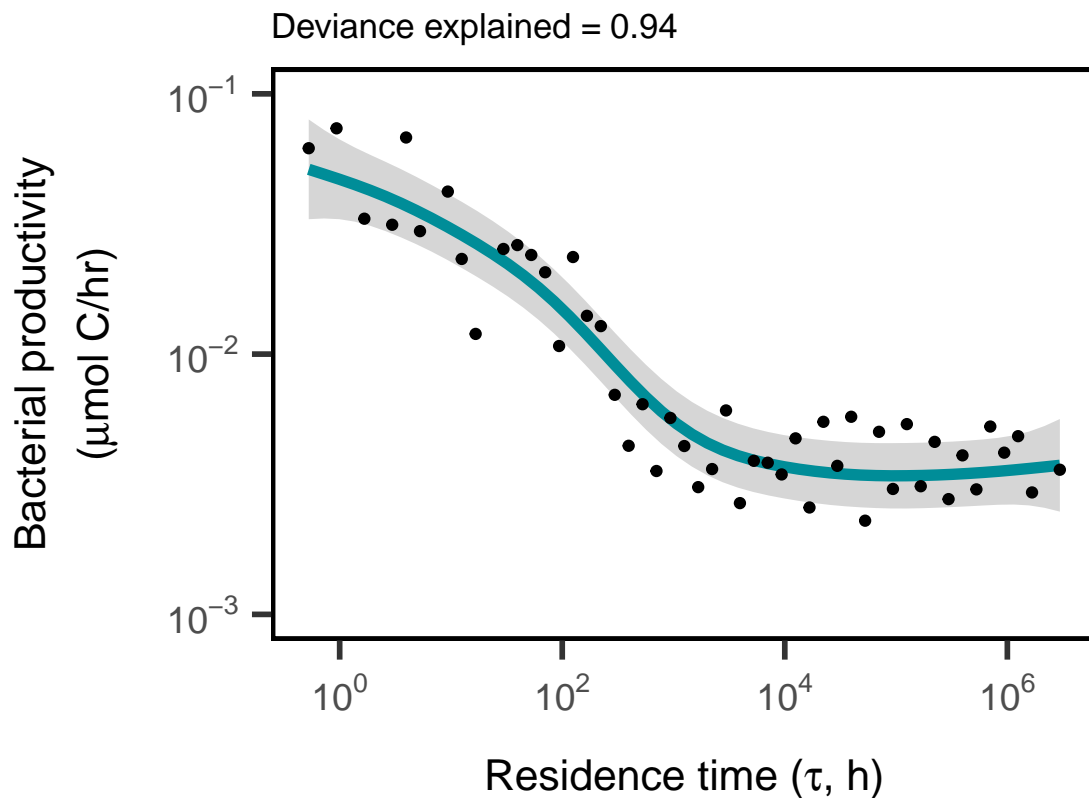


```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-2.109971e-09,7.289529e-10]
## (score -22.39941 & scale 0.01328936).
## Hessian positive definite, eigenvalue range [1.085797,23.76268].
## Model rank = 14 / 14
##
```

```
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'   edf k-index p-value
## s(Tau) 9.00 5.04   1.14   0.80
## s(Set) 4.00 2.67    NA    NA

BP_Tau <- predict_gam(BP.gam.re, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = Tau, aes(x = Tau,
    y = log(uMChr, 10))) + xlab(expression(paste("Residence time (",
    tau, ", h)"))) + ylab(expression(atop("Bacterial productivity",
    paste("(", mu, "mol C/hr)")))) + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + scale_y_continuous(limits = c(-3, -1),
    breaks = c(-1, -2, -3), labels = label_math(expr = 10^.x,
    format = force)) + labs(title = bquote("Deviance explained =" ~
    .(signif(summary(BP.gam.re)$dev, 2))))

BP_Tau
```



```
ggsave("./output/BP_Tau.pdf")
ggsave("./output/BP_Tau.png", width = 7, height = 5)
```

## D. Average well response at 48 hours

```
Avg.gam <- gam(AvgRes ~ s(Tau), data = Tau, family = gaussian(link = "identity"),
  method = "REML")
Avg.gam.re <- gam(AvgRes ~ s(Tau) + s(Set, bs = "re"), data = Tau,
  family = gaussian(link = "identity"), method = "REML")

anova(Avg.gam, Avg.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: AvgRes ~ s(Tau)
## Model 2: AvgRes ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev    Df Deviance Pr(>Chi)
## 1      47.000      1.0175
## 2      43.836      0.5291 3.1639  0.48841 9.064e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(Avg.gam, Avg.gam.re)
```

```
##              df      AIC
## Avg.gam      3.000094 -44.79265
## Avg.gam.re  5.946984 -70.94147
```

```
# Random effect of set is significant
rm(Avg.gam)
```

```
summary(Avg.gam.re)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## AvgRes ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.60253    0.05653   10.66 8.42e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(Tau) 1.000      1 13.00 0.000792 ***
## s(Set) 2.764      3 12.69 2.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.509   Deviance explained = 54.7%
## -REML = -29.608   Scale est. = 0.011961   n = 49
```

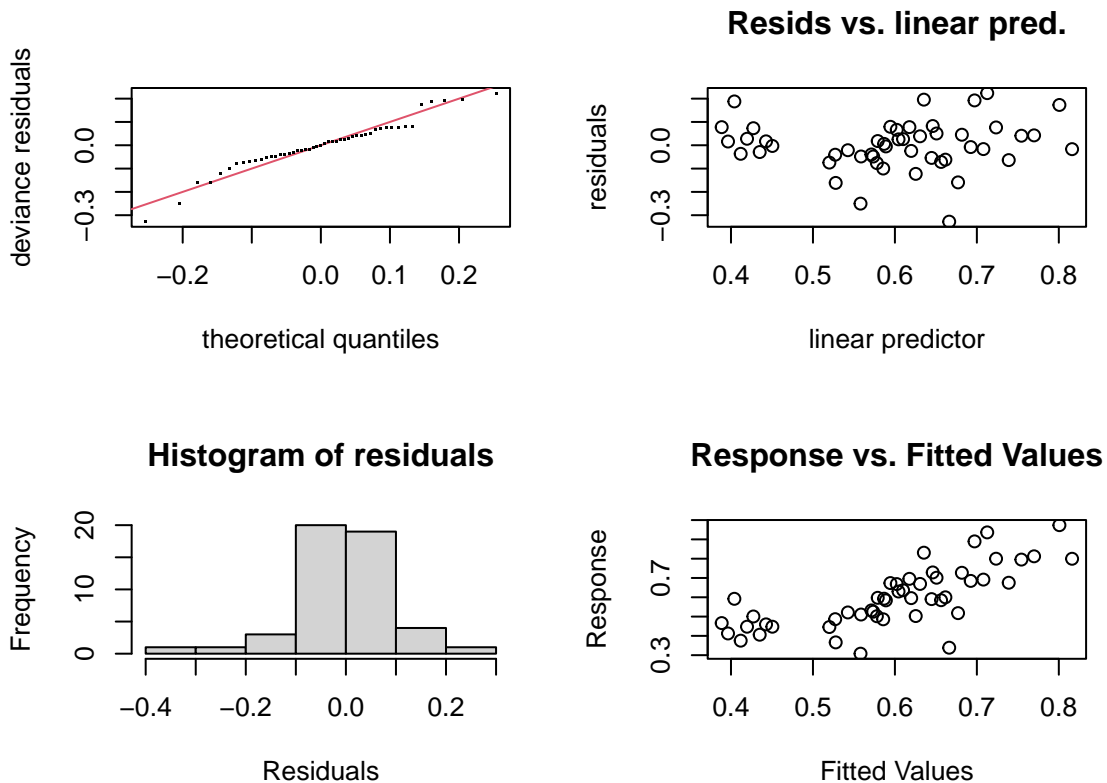
```
mean(summary(Avg.gam.re)$s.table[, 4])
```

```
## [1] 0.0003971892
```

```
k.check(Avg.gam.re)
```

```
##          k'      edf   k-index p-value
## s(Tau)   9 1.000047 0.9029773 0.1925
## s(Set)   4 2.764202          NA      NA
```

```
gam.check(Avg.gam.re)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 10 iterations.
## Gradient range [-1.938184e-05,4.092839e-06]
## (score -29.60753 & scale 0.01196094).
## Hessian positive definite, eigenvalue range [1.938115e-05,23.58567].
## Model rank = 14 / 14
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'   edf k-index p-value
## s(Tau) 9.00 1.00    0.9  0.24
## s(Set) 4.00 2.76    NA   NA
```

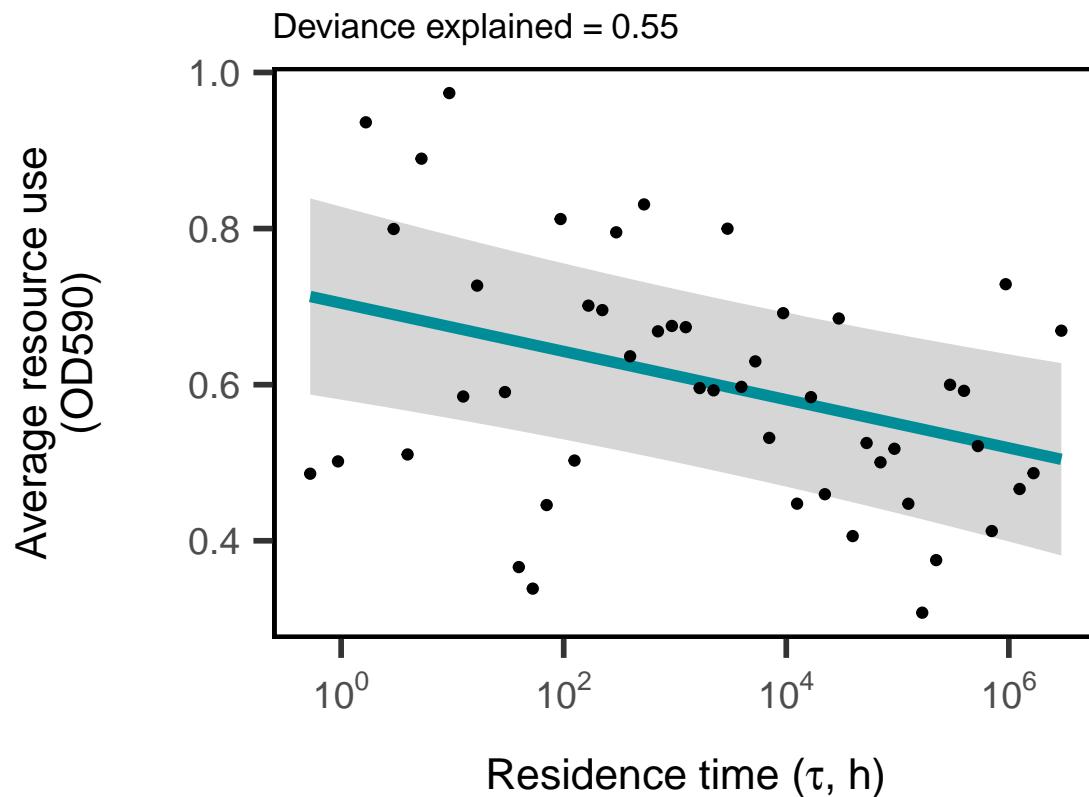
```
AvgRes_Tau <- predict_gam(Avg.gam.re, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = Tau, aes(x = Tau,
```

```

y = AvgRes)) + xlab(expression(paste("Residence time (",
tau, ", h)")))) + ylab("Average resource use\n(OD590)") +
scale_x_continuous(labels = label_math(expr = 10^.x, format = force)) +
labs(title = bquote("Deviance explained =" ~ .(signif(summary(Avg.gam.re)$dev,
2)))) + theme(plot.margin = unit(c(0.2, 0.2, 0.2, 0.8),
"cm")) + theme(axis.title.y = element_text(vjust = 6))

```

AvgRes\_Tau



```

ggsave("./output/AvgRes_Tau.pdf")
ggsave("./output/AvgRes_Tau.png", width = 7, height = 5)

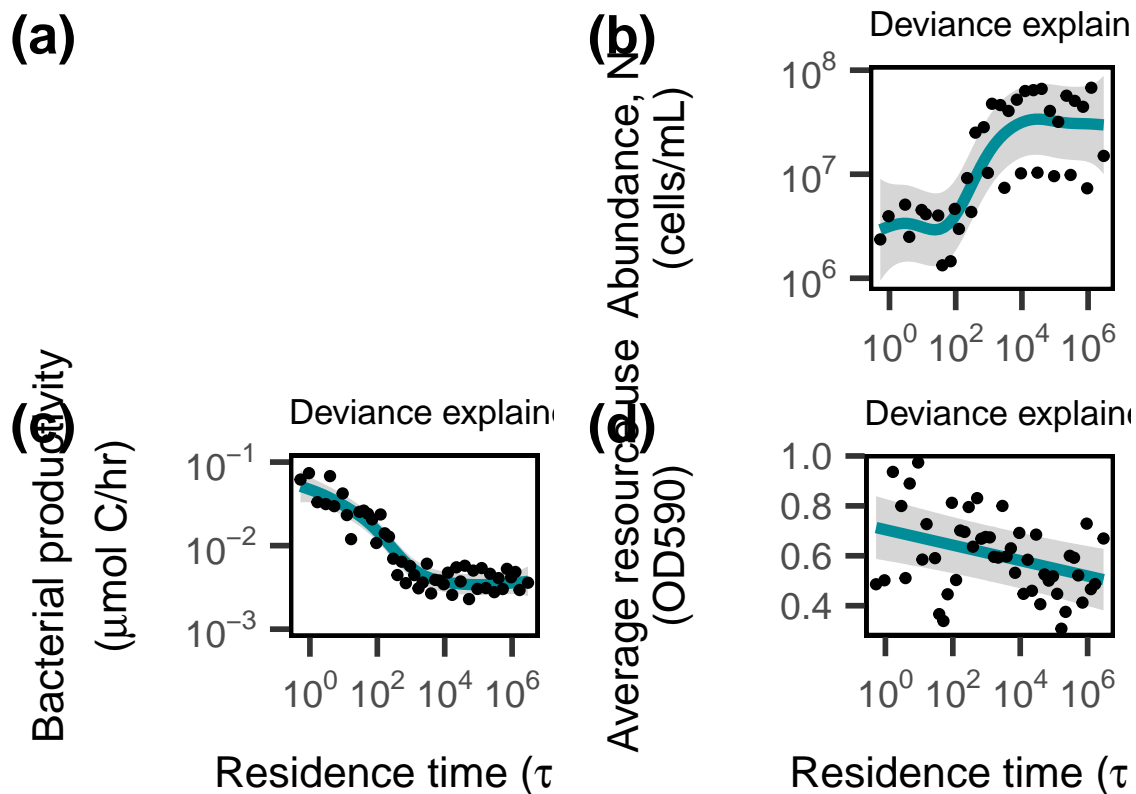
```

## Draw Figure

```

ggdraw() + draw_plot(N_Tau, x = 0.5, y = 0.55, width = 0.5, height = 0.45) +
draw_plot(BP_Tau, x = 0, y = 0, width = 0.5, height = 0.55) +
draw_plot(AvgRes_Tau, x = 0.5, y = 0, width = 0.5, height = 0.55) +
draw_plot(label = c("(a)", "(b)", "(c)", "(d)"), size = 20,
x = c(0, 0.5, 0, 0.5), y = c(1, 1, 0.55, 0.55)) + theme(plot.background = element_rect(fill = "
color = NA))

```



```
ggsave("./output/RTLC_Fig1.pdf")
ggsave("./output/RTLC_Fig1.png", width = 12, height = 8, dpi = 800)
```

Figure 2. Resource use changed with residence time

Generate individual carbon source GAMs

```
csource.sig <- data.frame()
for (c in unique(EP.long$C_Source)) {
  type <- unique(subset(EP.long, EP.long$C_Source == c)$Type)
  csource.gam <- gam(data = subset(EP.long, EP.long$C_Source ==
    c), OD_48 ~ s(Tau), family = gaussian(link = "identity"),
    method = "REML")
  csource.sig <- rbind(csource.sig, c(c, type, mean(summary(csource.gam)$s.table[,
    4])))
  assign(paste("gam.", gsub("-", "_", c), sep = "."), csource.gam)
  rm(csource.gam)
}

colnames(csource.sig) <- c("Resource", "Type", "P_gam")
csource.sig$BH_gam <- p.adjust(csource.sig$P_gam, "BH")

csource.sig <- subset(csource.sig, csource.sig$BH_gam < 0.05)

csource.gams <- data.frame()
```

```

for (c in as.list(csource.sig$Resource)) {
  csource.gams <- rbind(csource.gams, cbind(rep(c, 50), rep(subset(csource.sig,
    csource.sig$Resource == c)$Type, 50), predict_gam(gam(data = subset(EP.long,
    EP.long$C_Source == c), OD_48 ~ s(Tau), family = gaussian(link = "identity"),
    method = "REML"))))
}
colnames(csource.gams) <- c("C_Source", "Type", "Tau", "fit",
  "se.fit")

csource.gams$C_Source <- as.factor(csource.gams$C_Source)

```

## A. Bray-Curtis distance Resource use

```

EP.rbys <- subset(EP, select = -c(Tau, NumRes, AvgRes, Set))
EP.db <- vegdist(EP.rbys, method = "bray")
EP.pcoa <- cmdscale(EP.db, eig = TRUE)

EP.REL <- EP.rbys
for (i in 1:nrow(EP.rbys)) {
  EP.REL[i, ] = EP.rbys[i, ]/sum(EP.rbys[i, ])
}

EP.pcoa <- add.spec.scores(EP.pcoa, EP.REL, method = "pcoa.scores")

explainvar1.ep <- round(EP.pcoa$eig[1]/sum(EP.pcoa$eig), 3) *
  100
explainvar2.ep <- round(EP.pcoa$eig[2]/sum(EP.pcoa$eig), 3) *
  100
explainvar3.ep <- round(EP.pcoa$eig[3]/sum(EP.pcoa$eig), 3) *
  100
sum.eig.ep <- sum(explainvar3.ep, explainvar2.ep, explainvar1.ep)

types <- read.csv("data/RTL/C/EcoPlate/Csourcetypes.csv", header = TRUE,
  sep = ",")

EP.plot <- as.data.frame(EP.pcoa$points)
EP.plot <- cbind(EP.plot, Tau$Tau, Tau$Pump, Tau$Set)
colnames(EP.plot) <- c("V1", "V2", "Tau", "Pump", "Set")
EP.plot$Pump <- factor(EP.plot$Pump, levels = c("TRUE", "FALSE"))

EP.cproj <- as.data.frame(EP.pcoa$cproj)
EP.cproj <- cbind(EP.cproj, as.data.frame(row.names(EP.pcoa$cproj)))
spe.corr.ep <- add.spec.scores(EP.pcoa, EP.REL, method = "cor.scores")$cproj
spe.corr.ep <- as.data.frame(cbind(spe.corr.ep, types$Type))
corr.cut.ep <- 0.7
imp.spp.ep <- as.data.frame(spe.corr.ep[abs(as.numeric(spe.corr.ep[,
  1])) >= corr.cut.ep | abs(as.numeric(spe.corr.ep[, 2])) >=
  corr.cut.ep, ])

Dim1.ep <- c()
Dim2.ep <- c()
for (x in unique(spe.corr.ep$V3)) {
  Dim1.ep <- c(Dim1.ep, mean(as.numeric(subset(spe.corr.ep,

```



```

    spe.corr.ep$V3 == x)$Dim1)))
Dim2.ep <- c(Dim2.ep, mean(as.numeric(subset(spe.corr.ep,
    spe.corr.ep$V3 == x)$Dim2)))
}

spe.corr.ep.means <- as.data.frame(unique(spe.corr.ep$V3))
spe.corr.ep.means <- cbind(spe.corr.ep.means, as.vector(Dim1.ep),
    as.vector(Dim2.ep))
colnames(spe.corr.ep.means) <- c("C_types", "Dim1", "Dim2")

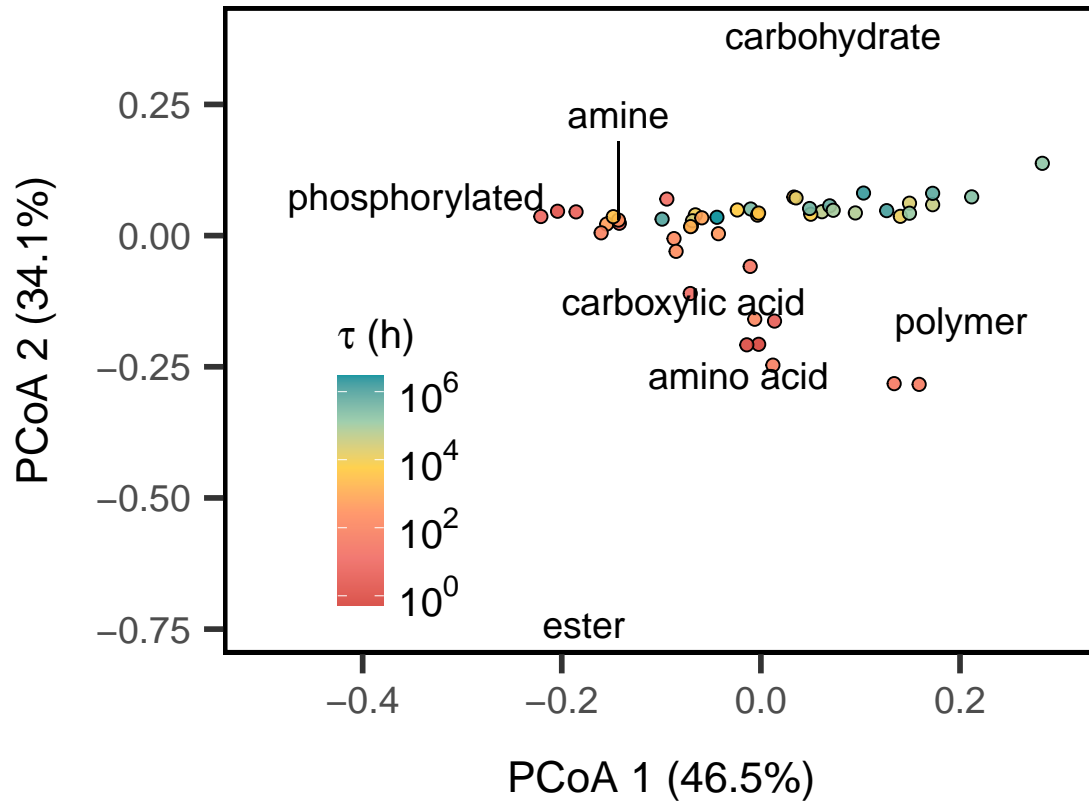
EP.plot$Tau <- as.numeric(EP.plot$Tau)

spe.corr.ep <- spe.corr.ep[spe.corr.ep$C_source %in% c(gsub("-",
    ".", csource.sig$Resource), "X4.Hydroxy.Benzoic.Acid"), ]
spe.corr.ep$Dim1 <- as.numeric(spe.corr.ep$Dim1)
spe.corr.ep$Dim2 <- as.numeric(spe.corr.ep$Dim2)
spe.corr.ep$C_source <- rownames(spe.corr.ep)

ResUse_PCoA <- ggplot(EP.plot, aes(x = V1, y = V2)) + geom_point(shape = 21,
    cex = 2, aes(fill = as.numeric(Tau)), color = "black") +
    xlab(paste("PCoA 1 (", explainvar1.ep, "%)", sep = "")) +
    ylab(paste("PCoA 2 (", explainvar2.ep, "%)", sep = "")) +
    xlim(c(-0.5, 0.3)) + labs(fill = expression(paste(tau, " (h)"))) +
    scale_fill_gradientn(colors = alpha(moma.colors("ustwo"),
    0.9), labels = label_math(expr = 10^.x, format = force)) +
    theme(legend.title = element_text(size = 14)) + geom_text(size = 5,
    data = subset(spe.corr.ep.means, spe.corr.ep.means$C_types !=
    "amine"), aes(x = Dim1, y = Dim2, label = C_types), color = "black") +
    geom_text(size = 5, data = subset(spe.corr.ep.means, spe.corr.ep.means$C_types ==
    "amine"), aes(x = Dim1, y = Dim2 + 0.2, label = C_types),
    color = "black") + geom_segment(aes(x = as.numeric(spe.corr.ep.means[6,
    "Dim1"]), y = as.numeric(spe.corr.ep.means[6, "Dim2"]), xend = as.numeric(spe.corr.ep.means[6,
    "Dim1"]), yend = as.numeric(spe.corr.ep.means[6, "Dim2"])) +
    0.15)) + theme(legend.title = element_text(size = 15), legend.position = c(0.2,
    0.3), legend.text = element_text(size = 15))

ResUse_PCoA

```



```
ggsave("./output/ResUse_PCoA.pdf")
ggsave("./output/ResUse_PCoA.png", width = 8, height = 5)
```

## B. Resource Use PCoA1

```
ResUse.PCoA1.gam <- gam(data = EP.plot, V1 ~ s(Tau), family = gaussian(link = "identity"),
  method = "REML")
ResUse.PCoA1.gam.re <- gam(data = EP.plot, V1 ~ s(Tau) + s(Set,
  bs = "re"), family = gaussian(link = "identity"), method = "REML")

anova(ResUse.PCoA1.gam, ResUse.PCoA1.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: V1 ~ s(Tau)
## Model 2: V1 ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1      42.564    0.39463
## 2      43.859    0.22261 -1.2951  0.17202
```

```
AIC(ResUse.PCoA1.gam, ResUse.PCoA1.gam.re)
```

```
##           df      AIC
## ResUse.PCoA1.gam  6.549076 -84.10591
## ResUse.PCoA1.gam.re 5.956827 -113.34358
```

```

# Random effect of set is not significant
rm(ResUse.PCoA1.gam.re)

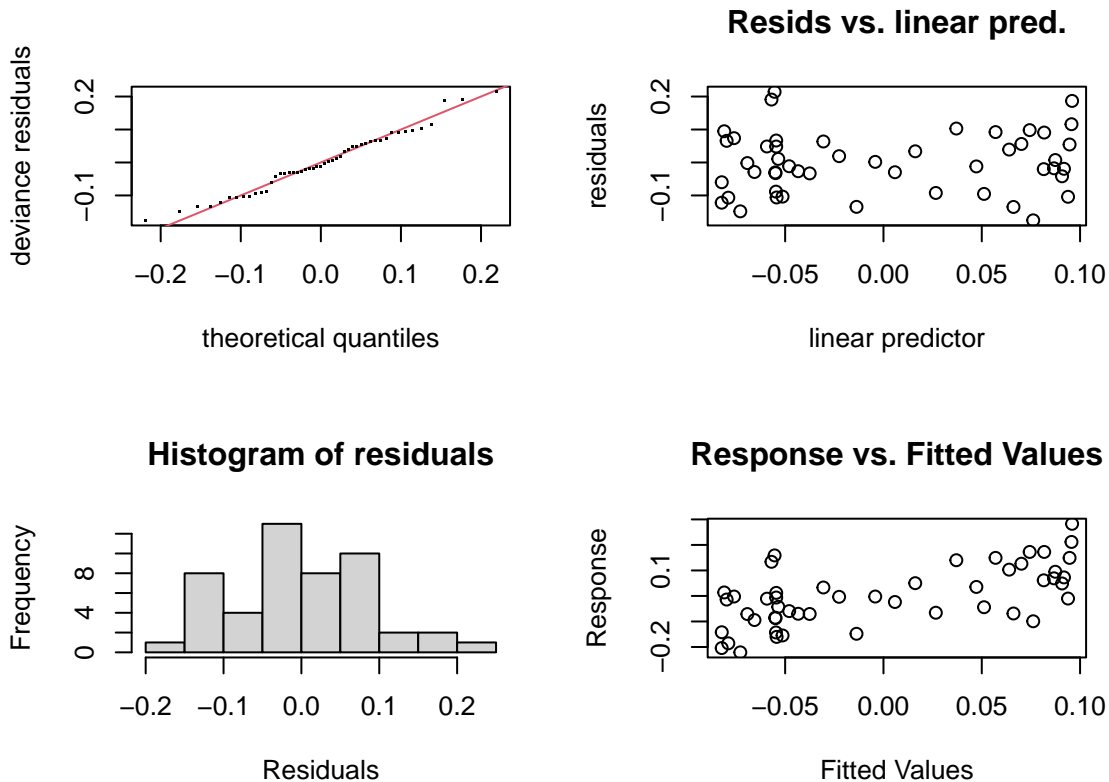
summary(ResUse.PCoA1.gam)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## V1 ~ s(Tau)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.078e-17  1.348e-02      0      1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## s(Tau) 3.662  4.549 5.446 0.000706 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.345   Deviance explained = 39.5%
## -REML = -37.513   Scale est. = 0.0089005   n = 49
mean(summary(ResUse.PCoA1.gam)$s.table[, 4])

## [1] 0.0007055848
k.check(ResUse.PCoA1.gam)

##           k'           edf  k-index p-value
## s(Tau)  9 3.662264 0.989424  0.3875
gam.check(ResUse.PCoA1.gam)

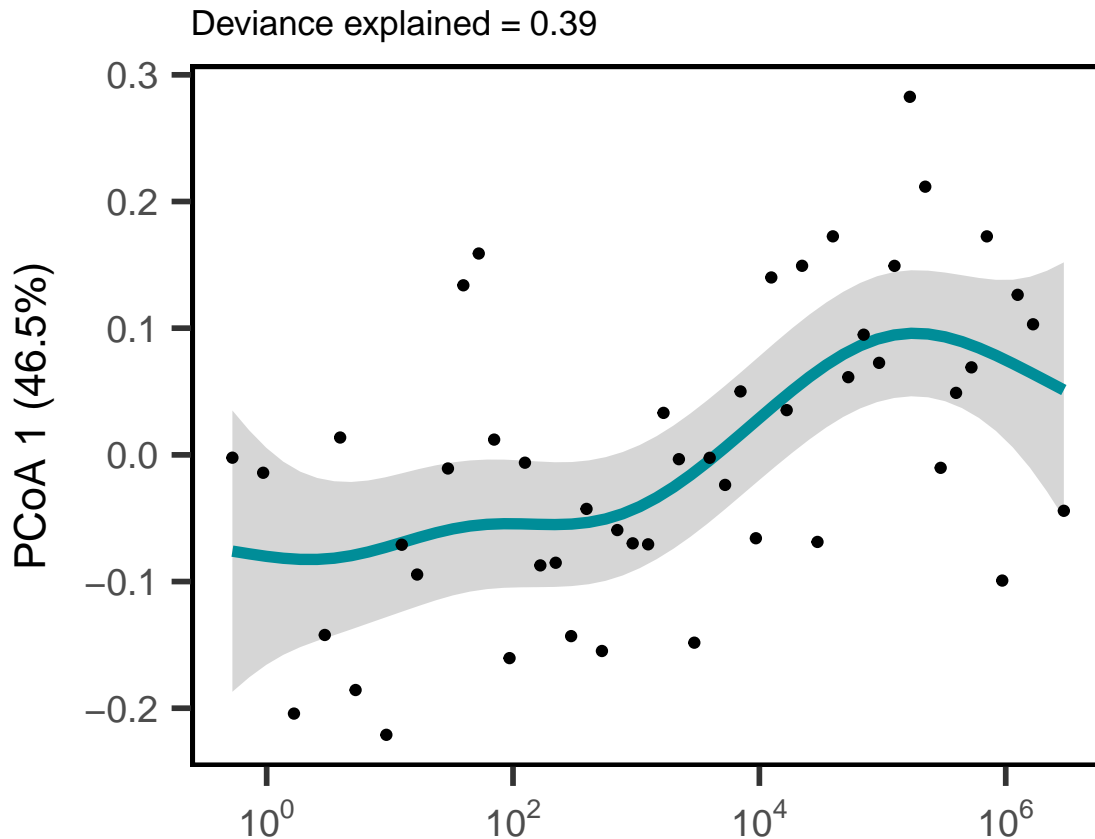
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [-3.325556e-05,3.54571e-05]
## (score -37.51287 & scale 0.008900487).
## Hessian positive definite, eigenvalue range [0.1398772,23.57581].
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'   edf k-index p-value
## s(Tau) 9.00 3.66   0.99   0.38

ResUse_PCoA1 <- predict_gam(ResUse.PCoA1.gam) %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = EP.plot, aes(x = Tau,
    y = V1)) + ylab(paste("PCoA 1 (", explainvar1.ep, "%)\n",
    sep = "")) + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + theme(axis.title.y = element_text(margin = margin(t = 0,
    r = -6, b = 0, l = 12))) + labs(title = bquote("Deviance explained = " ~
    .(signif(summary(ResUse.PCoA1.gam)$dev, 2)))) + theme(axis.title.x = element_blank())

ResUse_PCoA1
```



```
ggsave("./output/ResUse_PCoA1.pdf")
ggsave("./output/ResUse_PCoA1.png", width = 6.5, height = 5)
```

### C. Resource Use PCoA2

```
ResUse.PCoA2.gam <- gam(data = EP.plot, V2 ~ s(Tau), family = gaussian(link = "identity"),
  method = "REML")
ResUse.PCoA2.gam.re <- gam(data = EP.plot, V2 ~ s(Tau) + s(Set,
  bs = "re"), family = gaussian(link = "identity"), method = "REML")
```

```
anova(ResUse.PCoA2.gam, ResUse.PCoA2.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model 1: V2 ~ s(Tau)
```

```
## Model 2: V2 ~ s(Tau) + s(Set, bs = "re")
```

```
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
```

```
## 1      45.890      0.30490
```

```
## 2      35.967      0.15484  9.9228  0.15006 3.269e-05 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(ResUse.PCoA2.gam, ResUse.PCoA2.gam.re)
```

```
##                df      AIC
```

```
## ResUse.PCoA2.gam      3.783514 -102.2766
```

```
## ResUse.PCoA2.gam.re 12.293564 -118.4579
```

```

# Random effect of set is significant
rm(ResUse.PCoA2.gam)

summary(ResUse.PCoA2.gam.re)

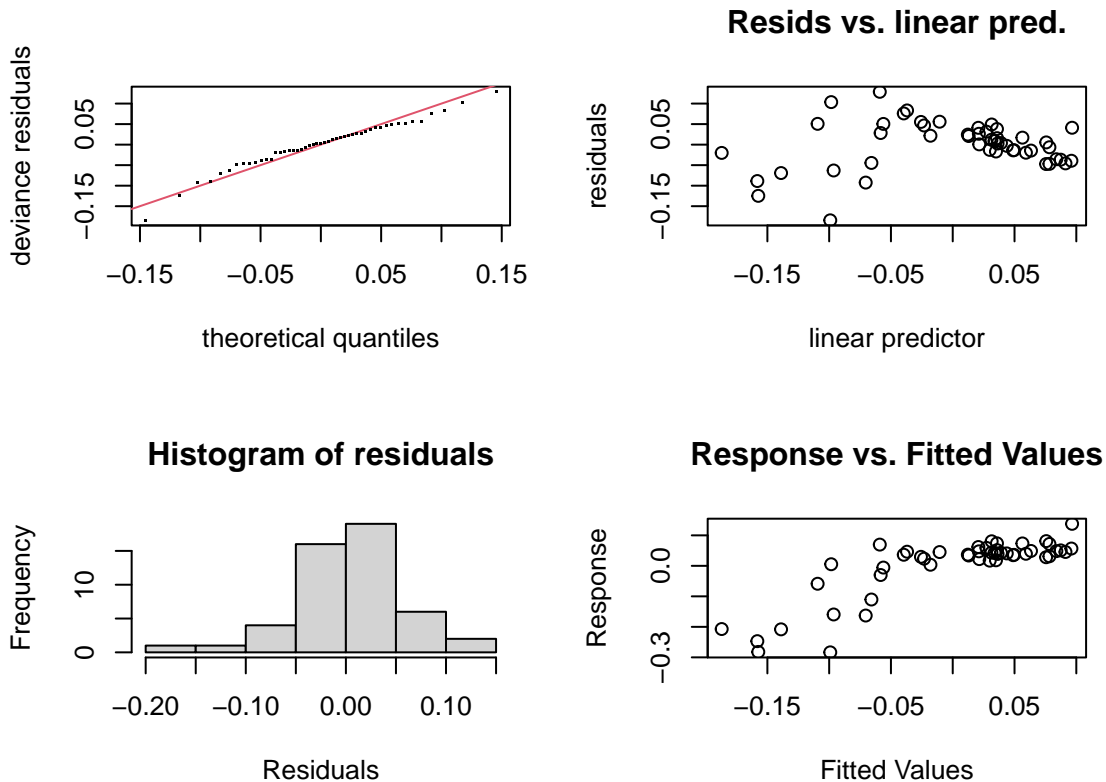
##
## Family: gaussian
## Link function: identity
##
## Formula:
## V2 ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.001121   0.018245   0.061   0.951
##
## Approximate significance of smooth terms:
##             edf Ref.df    F  p-value
## s(Tau) 6.389   7.542 8.536 2.29e-06 ***
## s(Set) 2.165   3.000 2.844  0.0144 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.606   Deviance explained = 67.6%
## -REML = -49.709   Scale est. = 0.0039255   n = 49
mean(summary(ResUse.PCoA2.gam.re)$s.table[, 4])

## [1] 0.007196484

k.check(ResUse.PCoA2.gam.re)

##           k'           edf  k-index p-value
## s(Tau)    9 6.388785 1.041467  0.5525
## s(Set)    4 2.165336          NA        NA
gam.check(ResUse.PCoA2.gam.re)

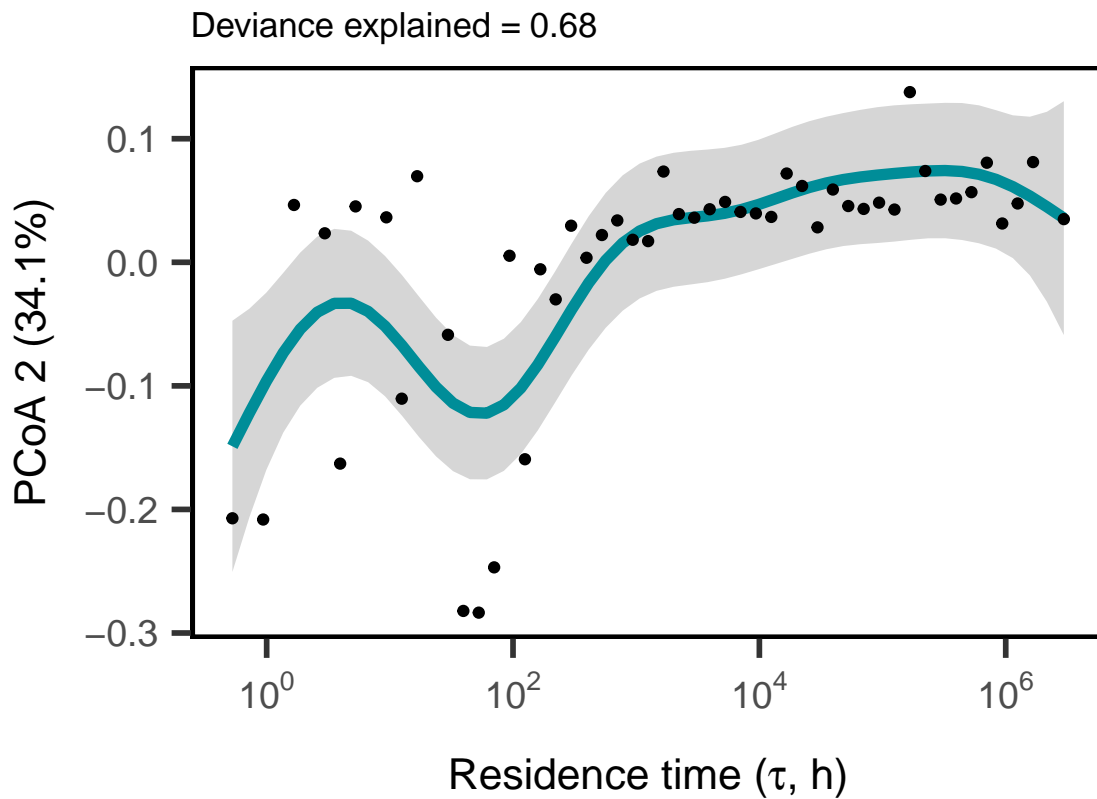
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 8 iterations.
## Gradient range [-7.662219e-07,1.587988e-08]
## (score -49.70925 & scale 0.003925458).
## Hessian positive definite, eigenvalue range [0.6970872,23.87046].
## Model rank = 14 / 14
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k'   edf k-index p-value
## s(Tau) 9.00 6.39   1.04   0.51
## s(Set) 4.00 2.17    NA     NA

ResUse_PCoA2 <- predict_gam(ResUse.PCoA2.gam.re, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = EP.plot, aes(x = Tau,
    y = V2)) + ylab(paste("PCoA 2 (", explainvar2.ep, "%)\n",
    sep = "")) + xlab(expression(paste("Residence time (", tau,
    ", h)"))) + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + theme(axis.title.y = element_text(margin = margin(t = 0,
    r = -6, b = 0, l = 12))) + labs(title = bquote("Deviance explained =" ~
    .(signif(summary(ResUse.PCoA2.gam.re)$dev, 2))))

ResUse_PCoA2
```

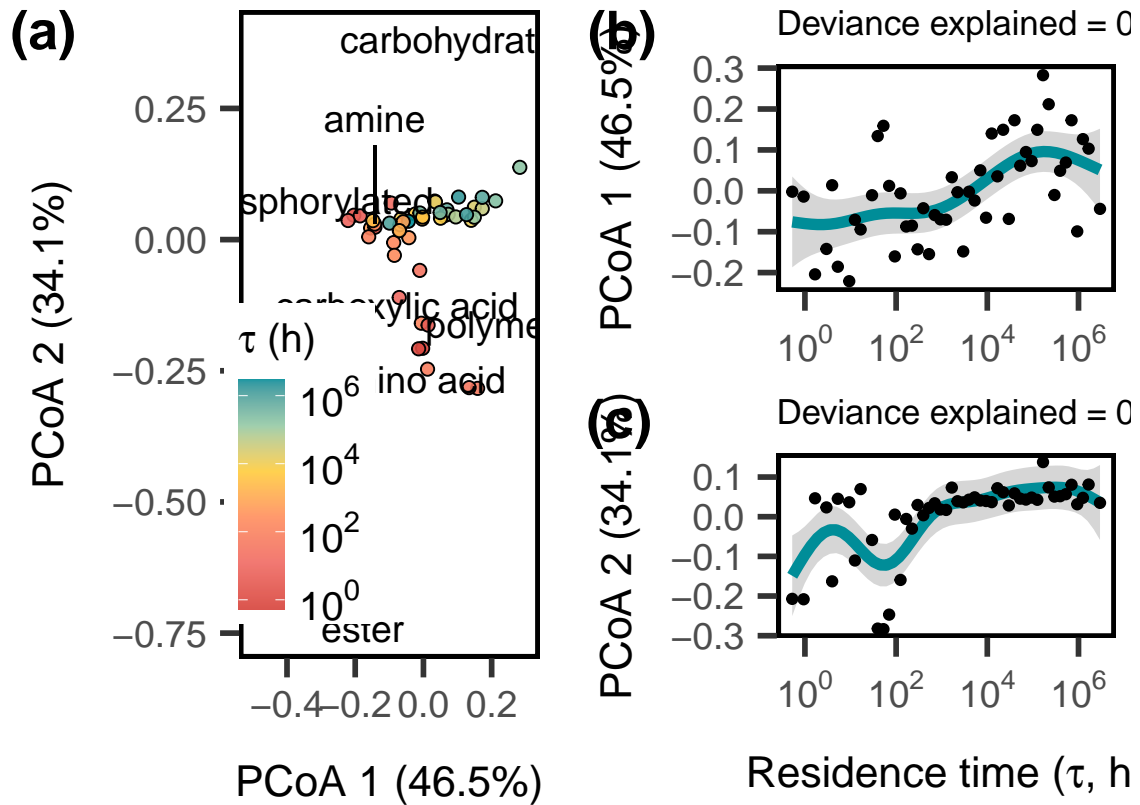


```
ggsave("./output/ResUse_PCoA2.pdf")
ggsave("./output/ResUse_PCoA2.png", width = 6.5, height = 5)
```

## Draw Figure

```
ggdraw() + draw_plot(ResUse_PCoA, x = 0, y = 0, width = 0.5,
  height = 1) + draw_plot(ResUse_PCoA1, x = 0.5, y = 0.55,
  width = 0.5, height = 0.45) + draw_plot(ResUse_PCoA2, x = 0.5,
  y = 0, width = 0.5, height = 0.55) + draw_plot_label(label = c("(a)",
  "(b)", "(c)"), size = 20, x = c(0, 0.5, 0.5), y = c(1.005,
  1.005, 0.555)) + theme(plot.background = element_rect(fill = "white",
  color = NA))
```





```
ggsave("./output/RTLC_Fig2.pdf")
ggsave("./output/RTLC_Fig2.png", width = 12, height = 6, dpi = 800)
```

**Figure 3. Microbial diversity increases with residence time**

Add OTU S and SimpE to Tau

```
Tau <- S.fxn(S, Tau)
Tau$Day_20.S <- as.numeric(Tau$Day_20.S)
Tau$Day_0.S <- as.numeric(Tau$Day_0.S)

Tau <- SimpE.fxn(SimpE, Tau)
Tau$Day_0.SimpE <- as.numeric(Tau$Day_0.SimpE)
Tau$Day_20.SimpE <- as.numeric(Tau$Day_20.SimpE)
```

#### A. Observed Richness - OTUs

```
S.OTU.gam <- gam(Day_20.S ~ s(Tau, k = 14), family = gaussian(link = "identity"),
  data = Tau, method = "REML")
S.OTU.gam.re <- gam(Day_20.S ~ s(Tau) + s(Set, bs = "re"), family = gaussian(link = "identity"),
  data = Tau, method = "REML")

AIC(S.OTU.gam, S.OTU.gam.re)
```

```
##          df      AIC
```

```

## S.OTU.gam      8.101273 541.8116
## S.OTU.gam.re 10.954306 535.4213
anova(S.OTU.gam, S.OTU.gam.re, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: Day_20.S ~ s(Tau, k = 14)
## Model 2: Day_20.S ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1      26.740      6816611
## 2      23.449      4824718 3.2906  1991893  0.01517 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Random effect of set is significant
rm(S.OTU.gam)

summary(S.OTU.gam.re)

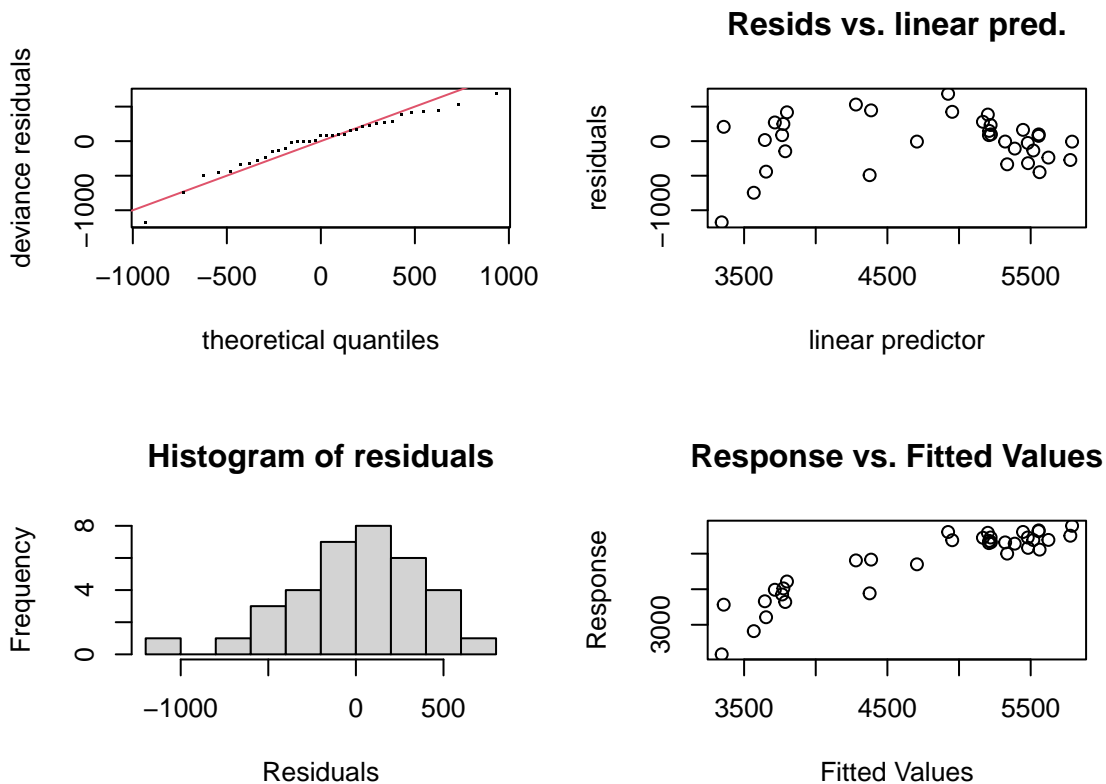
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Day_20.S ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4790.2      161.7    29.62  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(Tau) 5.064  6.138 18.225  <2e-16 ***
## s(Set) 2.294  3.000  2.947  0.0186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.792   Deviance explained = 83.7%
## -REML = 257.71   Scale est. = 1.8109e+05   n = 35

k.check(S.OTU.gam.re)

##           k'      edf  k-index p-value
## s(Tau)    9 5.063830 1.204698    0.84
## s(Set)    4 2.294089      NA      NA

gam.check(S.OTU.gam.re)

```



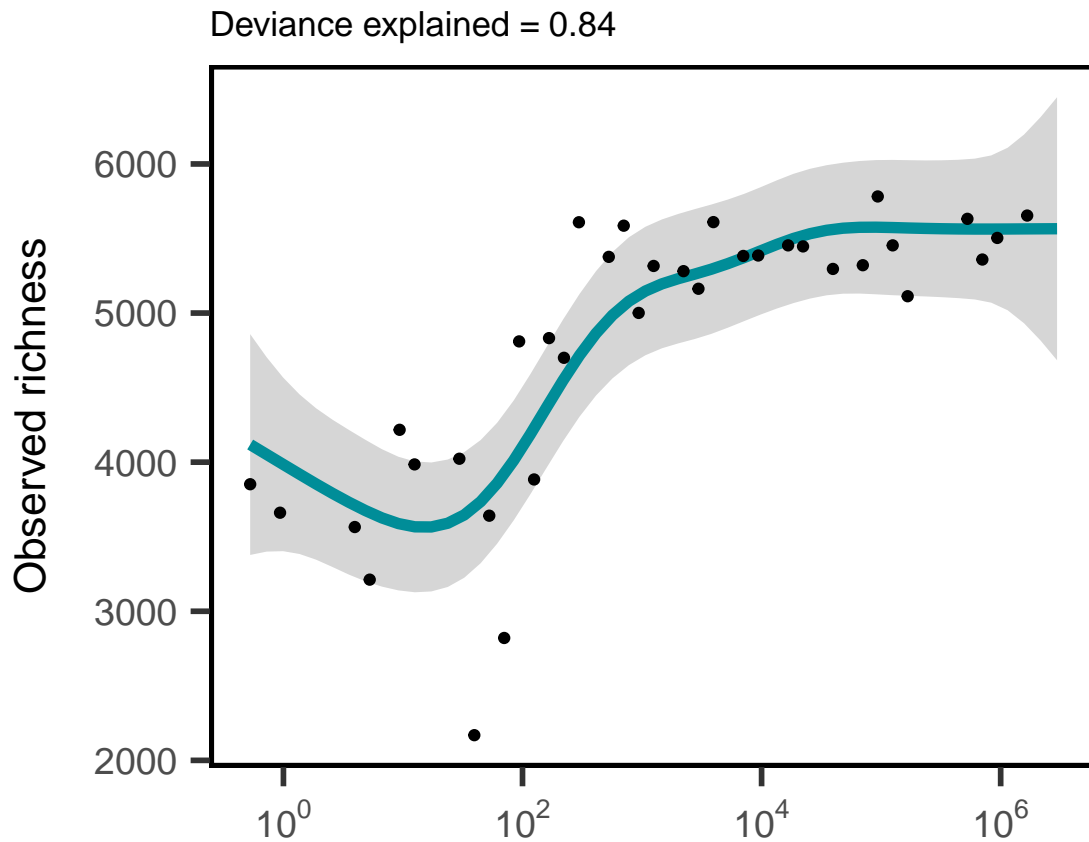
```
##
## Method: REML   Optimizer: outer newton
## full convergence after 4 iterations.
## Gradient range [-0.0001428268,1.919717e-05]
## (score 257.706 & scale 181093.9).
## Hessian positive definite, eigenvalue range [0.5611962,16.84203].
## Model rank = 14 / 14
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k'   edf k-index p-value
## s(Tau) 9.00 5.06    1.2   0.86
## s(Set) 4.00 2.29    NA     NA

mean(summary(S.OTU.gam.re)$s.table[, 4])

## [1] 0.009281618

S_OTU_Tau <- predict_gam(S.OTU.gam.re, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = Tau, aes(x = Tau,
    y = Day_20.S)) + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + ylab("Observed richness") + theme(axis.title.x = element_blank()) +
  labs(title = bquote("Deviance explained = " ~ .(signif(summary(S.OTU.gam.re)$dev,
    2))))
```

S\_OTU\_Tau



```
ggsave("./output/S_OTU_Tau.pdf")
ggsave("./output/S_OTU_Tau.png", width = 7, height = 5)
```

## B. Simpson's Evenness ( $E[1/D]$ ) - OTUs

```
SimpE.OTU.gam <- gam(Day_20.SimpE ~ s(Tau, k = 15), family = gaussian(link = "identity"),
  data = Tau, method = "REML")
SimpE.OTU.gam.re <- gam(Day_20.SimpE ~ s(Tau) + s(Set, bs = "re"),
  family = gaussian(link = "identity"), data = Tau, method = "REML")
```

```
AIC(SimpE.OTU.gam, SimpE.OTU.gam.re)
```

```
##           df      AIC
## SimpE.OTU.gam  8.147650 -195.5583
## SimpE.OTU.gam.re 7.715705 -195.1028
```

```
anova(SimpE.OTU.gam, SimpE.OTU.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model 1: Day_20.SimpE ~ s(Tau, k = 15)
```

```
## Model 2: Day_20.SimpE ~ s(Tau) + s(Set, bs = "re")
```

```
##   Resid. Df Resid. Dev      Df    Deviance Pr(>Chi)
```

```

## 1    26.677  0.0048176
## 2    27.253  0.0050026 -0.57592 -0.00018506  0.1612
# Random effect of set is not significant
rm(SimpE.OTU.gam.re)

summary(SimpE.OTU.gam)

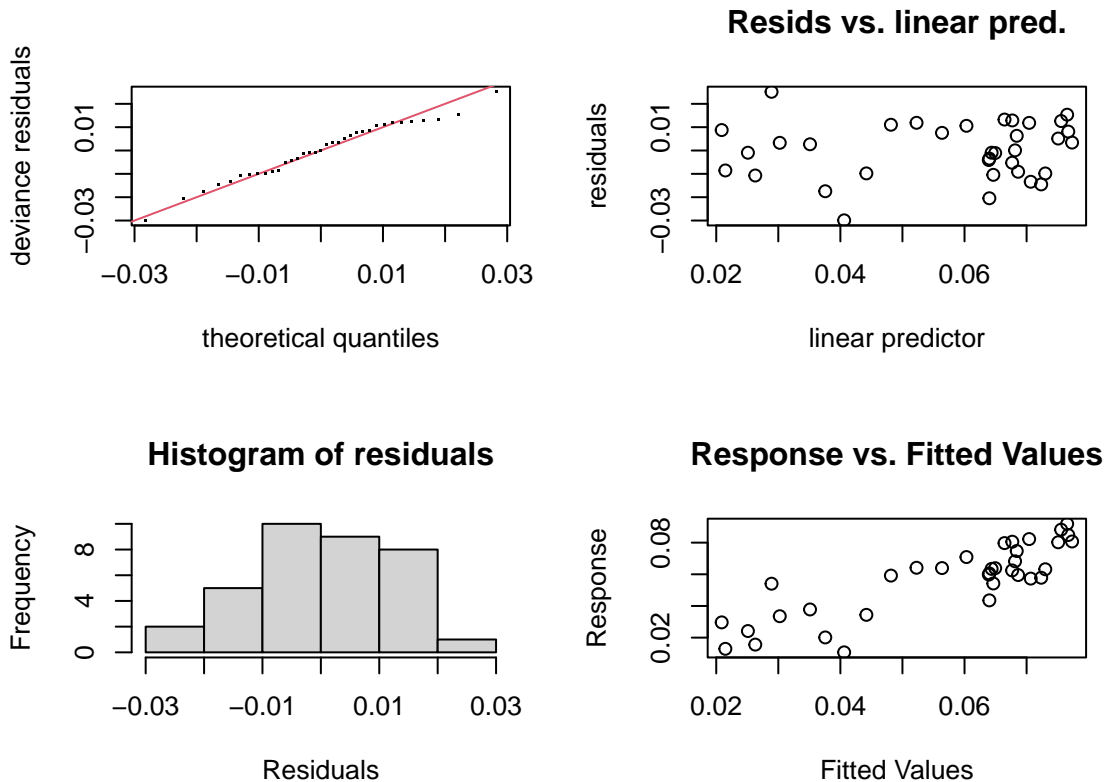
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Day_20.SimpE ~ s(Tau, k = 15)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.056801   0.002178   26.09  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F  p-value
## s(Tau)  4.972  6.148 11.51 1.83e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.674   Deviance explained = 72.2%
## -REML = -88.306   Scale est. = 0.00016596   n = 35

k.check(SimpE.OTU.gam)

##           k'           edf  k-index p-value
## s(Tau) 14 4.972393 0.963992  0.355

gam.check(SimpE.OTU.gam)

```

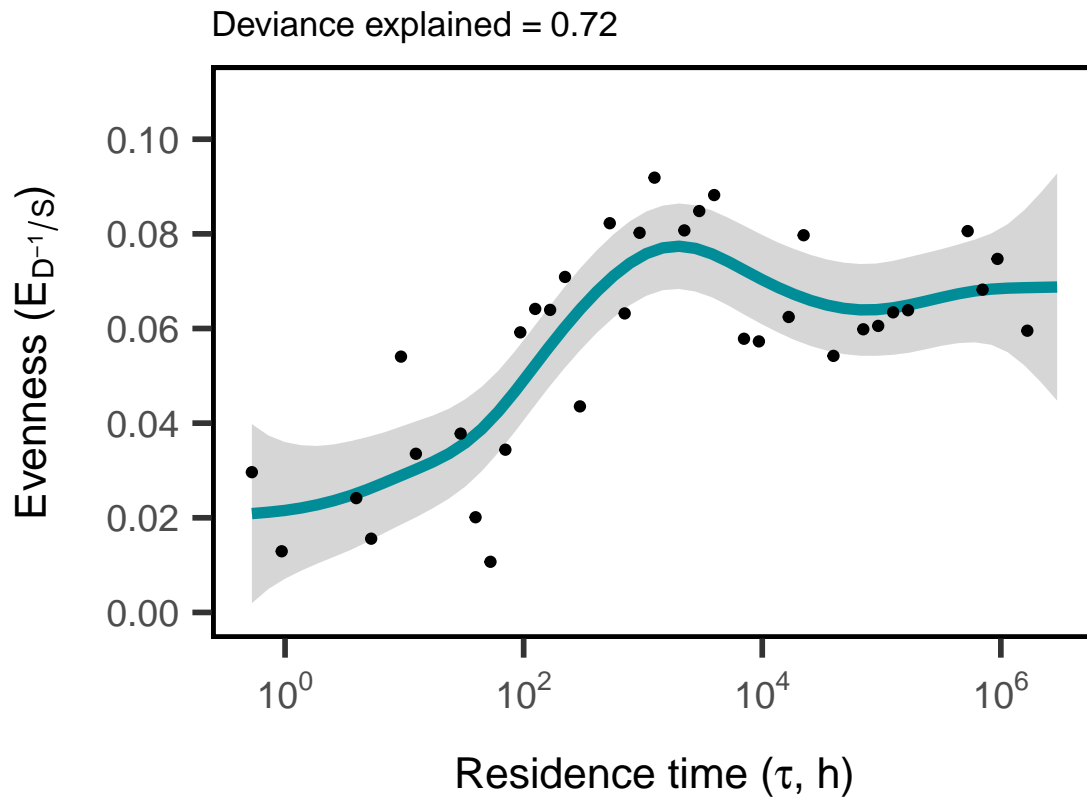


```
##
## Method: REML   Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-1.075916e-08,-8.738233e-09]
## (score -88.30582 & scale 0.0001659648).
## Hessian positive definite, eigenvalue range [0.7307918,16.75017].
## Model rank = 15 / 15
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'   edf k-index p-value
## s(Tau) 14.00  4.97   0.96   0.34
mean(summary(SimpE.OTU.gam)$s.table[, 4])

## [1] 1.832464e-06

SimpE_OTU_Tau <- predict_gam(SimpE.OTU.gam) %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = Tau, aes(x = Tau,
    y = Day_20.SimpE)) + scale_y_continuous(breaks = c(0, 0.02,
    0.04, 0.06, 0.08, 0.1), limits = c(0, 0.11)) + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + ylab(expression(paste("Evenness", " (E"[D^-1]/S),
    ")")) + xlab(expression(paste("Residence time (", tau, ", h)")) +
    labs(title = bquote("Deviance explained = " ~ .(signif(summary(SimpE.OTU.gam)$dev,
    2))))
```

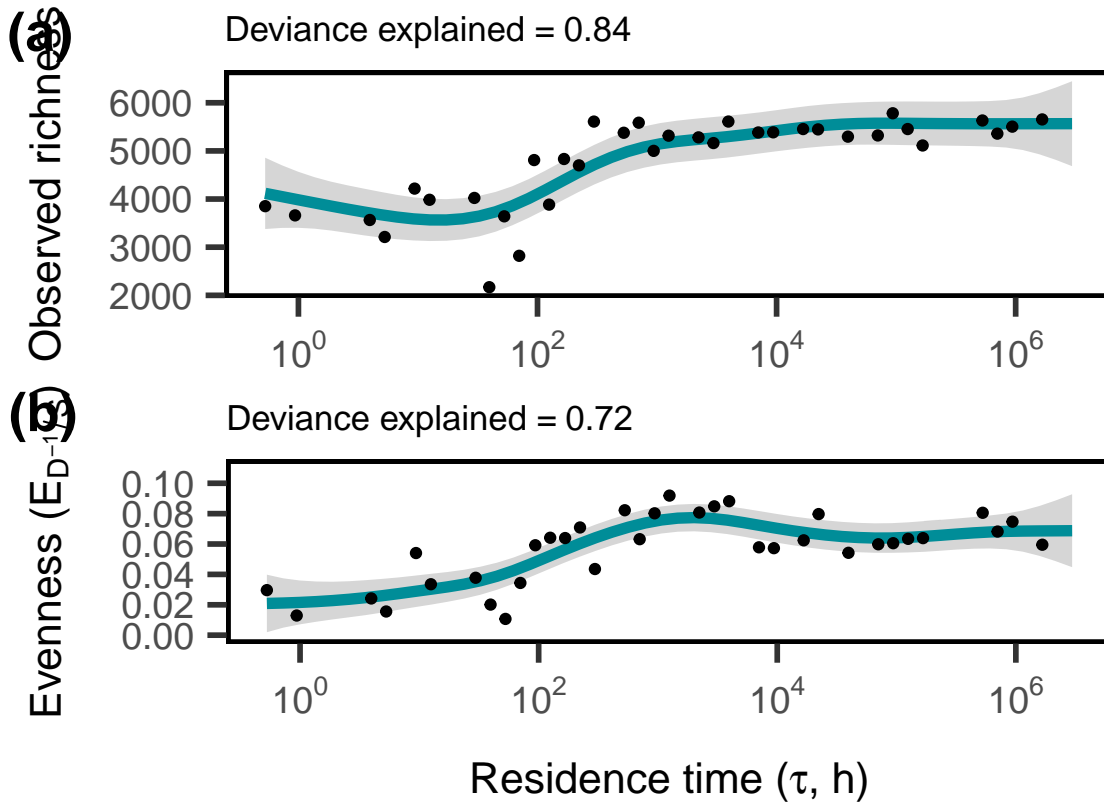
SimpE\_OTU\_Tau



```
ggsave("./output/SimpE_OTU_Tau.pdf")
ggsave("./output/SimpE_OTU_Tau.png", width = 7, height = 5)
```

Draw figure

```
ggdraw() + draw_plot(S_OTU_Tau, x = 0, y = 0.55, width = 1, height = 0.45) +
  draw_plot(SimpE_OTU_Tau, x = 0, y = 0, width = 1, height = 0.55) +
  draw_plot_label(label = c("(a)", "(b)"), size = 20, x = c(0,
    0), y = c(1.01, 0.57)) + theme(plot.background = element_rect(fill = "white",
  color = NA))
```



```
ggsave("./output/RTLC_Fig3.pdf")
ggsave("./output/RTLC_Fig3.png", width = 5, height = 6, dpi = 800)
```

Figure 4. Community assembly along a residence time gradient

```
CC.db <- vegdist(OTUs.r.REL.20, method = "bray", upper = TRUE,
  diag = TRUE)

CC.pcoa <- cmdscale(CC.db, eig = TRUE, k = 3)
explainvar1 <- round(CC.pcoa$eig[1]/sum(CC.pcoa$eig), 3) * 100
explainvar2 <- round(CC.pcoa$eig[2]/sum(CC.pcoa$eig), 3) * 100
explainvar3 <- round(CC.pcoa$eig[3]/sum(CC.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar3, explainvar2, explainvar1)

CC.plot <- as.data.frame(CC.pcoa$points)
CC.plot <- cbind(CC.plot, Tau.20)
CC.plot$Pump <- factor(CC.plot$Pump, levels = c("TRUE", "FALSE"))
CC.plot$V2 <- CC.plot$V2 * -1

CC.plot.order <- CC.plot[order(as.numeric(CC.plot$Tau)), ]
```

## A. Bray-Curtis Distance Community Composition PCoA

```
cor.test(x = CC.plot.order$Tau, y = CC.plot.order$V1)
```



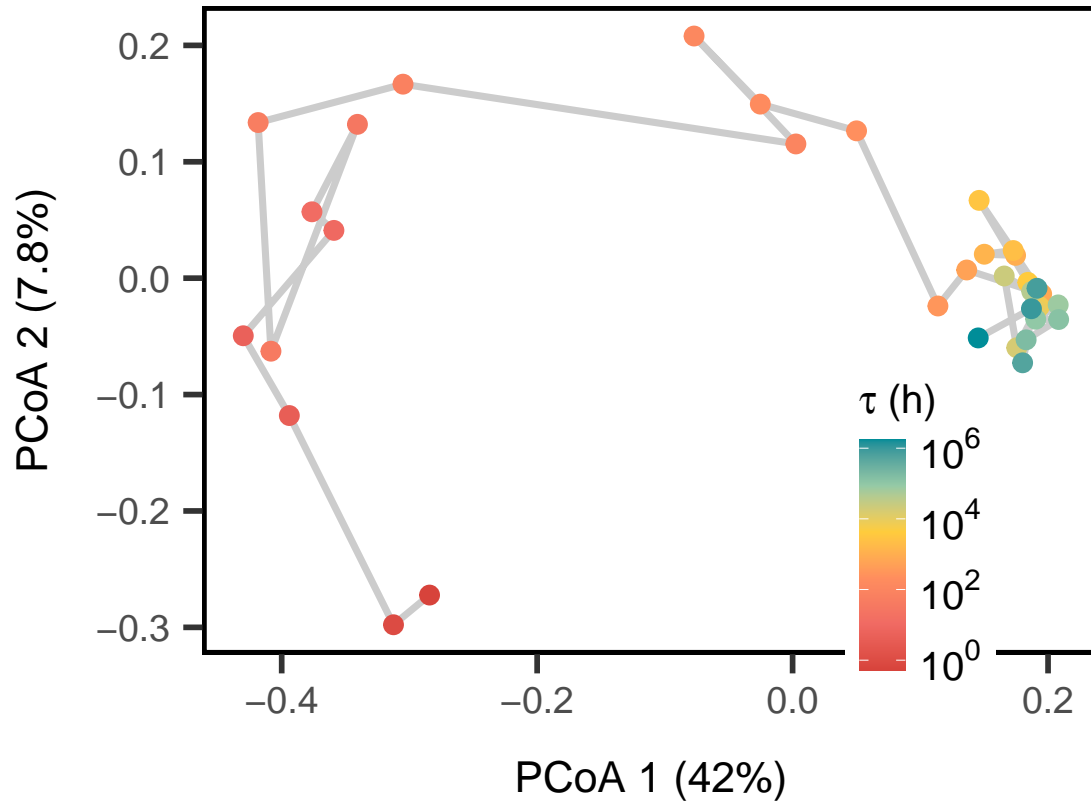
```
##
## Pearson's product-moment correlation
##
## data: CC.plot.order$Tau and CC.plot.order$V1
## t = 8.3414, df = 33, p-value = 1.233e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.6758255 0.9077065
## sample estimates:
## cor
## 0.8235867
```

```
cor.test(x = CC.plot.order$Tau, y = CC.plot.order$V2)
```

```
##
## Pearson's product-moment correlation
##
## data: CC.plot.order$Tau and CC.plot.order$V2
## t = 0.14656, df = 33, p-value = 0.8844
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.3103807 0.3557270
## sample estimates:
## cor
## 0.02550377
```

```
CommComp_PCoA <- ggplot(CC.plot.order, aes(x = V1, y = V2)) +
  geom_path(linewidth = 1.25, color = alpha("black", 0.2)) +
  geom_point(cex = 3, aes(color = as.numeric(Tau))) + xlab(paste("PCoA 1 (",
  explainvar1, "%)", sep = "")) + ylab(paste("PCoA 2 (", explainvar2,
  "%)", sep = "")) + labs(color = expression(paste(tau, " (h)"))) +
  scale_color_gradientn(colors = moma.colors("ustwo"), labels = label_math(expr = 10^.x,
  format = force)) + guides(color = guide_colorbar(position = "inside")) +
  theme(legend.title = element_text(size = 15), legend.position.inside = c(0.8,
  0.2), legend.text = element_text(size = 15))
```

```
CommComp_PCoA
```



```
ggsave("./output/CommComp_PCoA.pdf")
ggsave("./output/CommComp_PCoA.png", width = 10, height = 10)
```

## B. Community Composition - PCoA1

```
CommComp.PCoA1.gam <- gam(V1 ~ s(Tau), family = gaussian(link = "identity"),
  data = CC.plot, method = "REML")
CommComp.PCoA1.gam.re <- gam(V1 ~ s(Tau) + s(Set, bs = "re"),
  family = gaussian(link = "identity"), data = CC.plot, method = "REML")
```

```
AIC(CommComp.PCoA1.gam, CommComp.PCoA1.gam.re)
```

```
##              df      AIC
## CommComp.PCoA1.gam    9.822328 -98.60803
## CommComp.PCoA1.gam.re 13.273815 -109.02901
```

```
anova(CommComp.PCoA1.gam, CommComp.PCoA1.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model 1: V1 ~ s(Tau)
```

```
## Model 2: V1 ~ s(Tau) + s(Set, bs = "re")
```

```
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
```

```
## 1      25.222   0.069866
```

```
## 2      21.471   0.042590 3.7504 0.027276 0.00347 **
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

# Random effect of set is significant
rm(CommComp.PCoA1.gam)

summary(CommComp.PCoA1.gam.re)

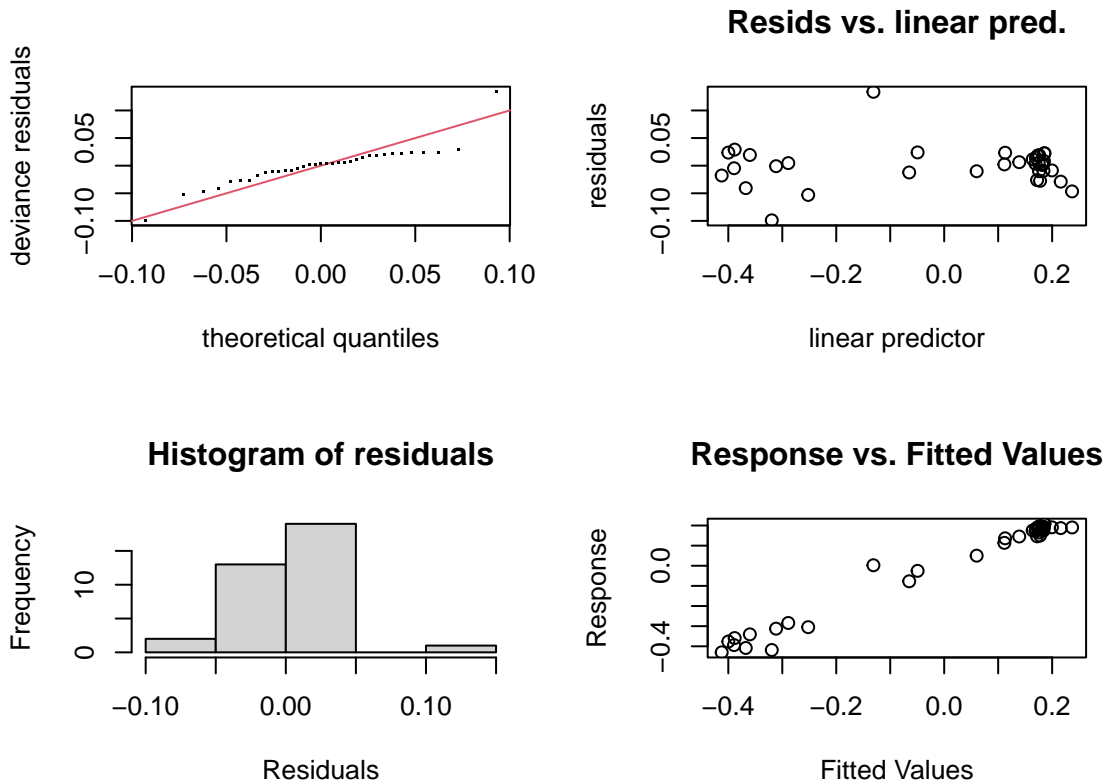
##
## Family: gaussian
## Link function: identity
##
## Formula:
## V1 ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0006728  0.0180653   0.037    0.971
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## s(Tau)  7.881  8.668 119.009 <2e-16 ***
## s(Set)  2.407  3.000   3.668  0.011 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.97   Deviance explained = 97.9%
## -REML = -38.669   Scale est. = 0.0017961   n = 35

k.check(CommComp.PCoA1.gam.re)

##           k'           edf  k-index p-value
## s(Tau)    9 7.880743 1.098318  0.6725
## s(Set)    4 2.407351          NA        NA

gam.check(CommComp.PCoA1.gam.re)

```



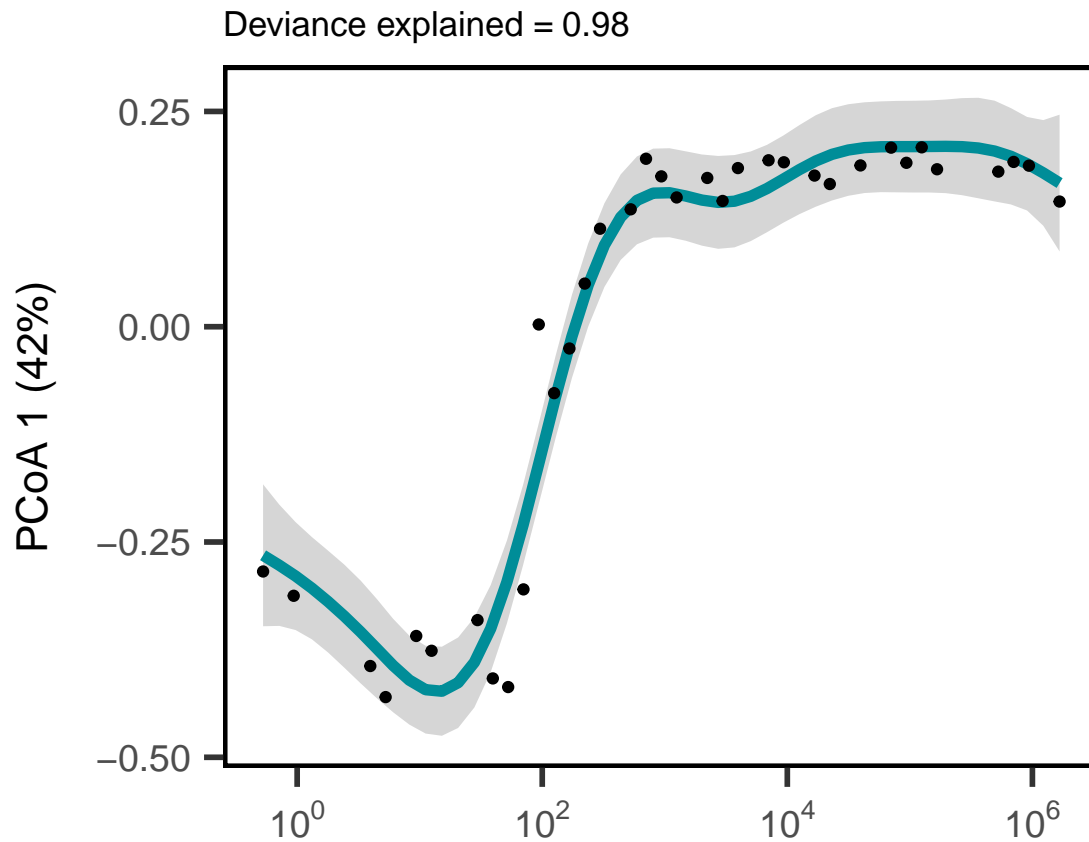
```
##
## Method: REML   Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-1.815658e-06,1.885777e-07]
## (score -38.66944 & scale 0.001796123).
## Hessian positive definite, eigenvalue range [0.7279977,17.40389].
## Model rank = 14 / 14
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k'   edf k-index p-value
## s(Tau) 9.00 7.88    1.1   0.72
## s(Set) 4.00 2.41    NA     NA

mean(summary(CommComp.PCoA1.gam.re)$s.table[, 4])

## [1] 0.005504149

CommComp_PCoA1 <- predict_gam(CommComp.PCoA1.gam.re, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = CC.plot, aes(x = Tau,
    y = V1)) + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + ylab(paste("PCoA 1 (", explainvar1, "%)",
    sep = "")) + labs(title = bquote("Deviance explained =" ~
    .(signif(summary(CommComp.PCoA1.gam.re)$dev, 2)))) + theme(axis.title.x = element_blank())
```

CommComp\_PCoA1



```
ggsave("./output/CommComp_PCoA1.pdf")
ggsave("./output/CommComp_PCoA1.png", width = 6.5, height = 5)
```

### C. Community composition - PCoA2

```
CommComp.PCoA2.gam <- gam(V2 ~ s(Tau), family = gaussian(link = "identity"),
  data = CC.plot, method = "REML")
CommComp.PCoA2.gam.re <- gam(V2 ~ s(Tau) + s(Set, bs = "re"),
  family = gaussian(link = "identity"), data = CC.plot, method = "REML")
```

```
AIC(CommComp.PCoA2.gam.re, CommComp.PCoA2.gam)
```

```
##              df      AIC
## CommComp.PCoA2.gam.re 7.714415 -98.27226
## CommComp.PCoA2.gam    7.713969 -98.27299
anova(CommComp.PCoA2.gam, CommComp.PCoA2.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: V2 ~ s(Tau)
## Model 2: V2 ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev      Df   Deviance Pr(>Chi)
## 1      27.185    0.079569
```

```
## 2      27.185    0.079569 0.00054824 3.5117e-07 0.002489 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Random effect of set is not significant (no difference in
# AIC)
rm(CommComp.PCoA2.gam.re)

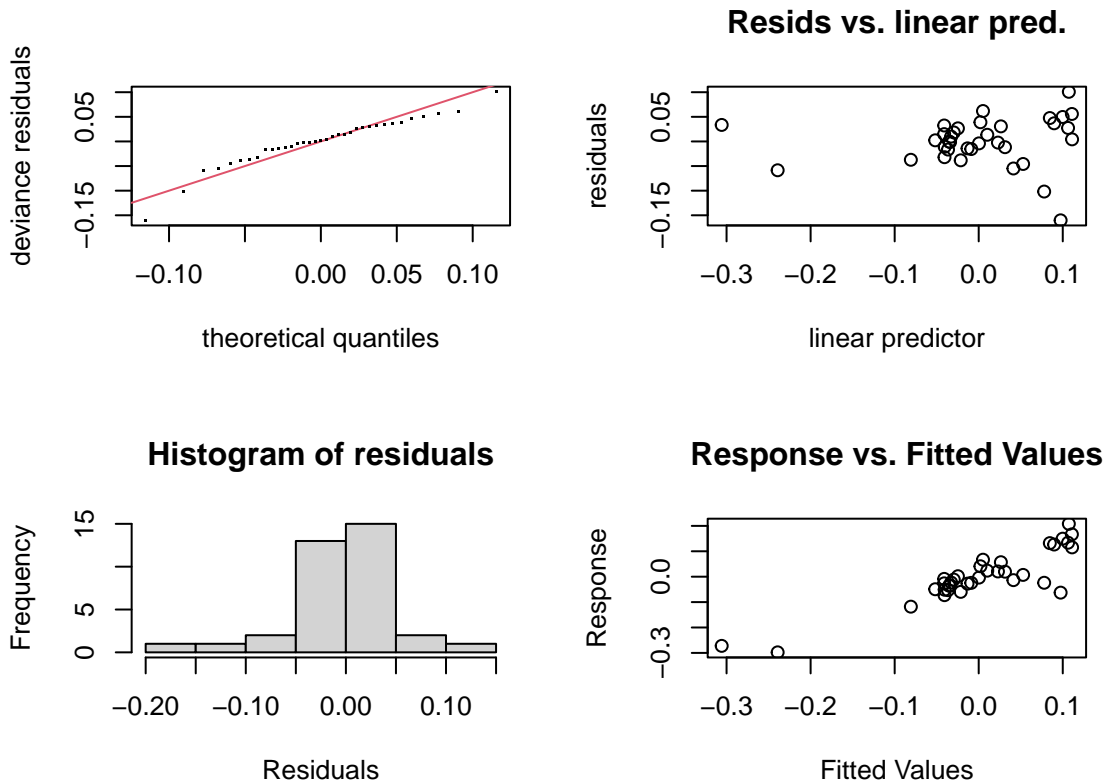
summary(CommComp.PCoA2.gam)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## V2 ~ s(Tau)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.772e-17  8.908e-03      0        1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(Tau) 5.351  6.452 15.66  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.746   Deviance explained = 78.6%
## -REML = -40.918   Scale est. = 0.0027774   n = 35

k.check(CommComp.PCoA2.gam)

##          k'          edf    k-index p-value
## s(Tau)   9 5.351009 0.9590669    0.37

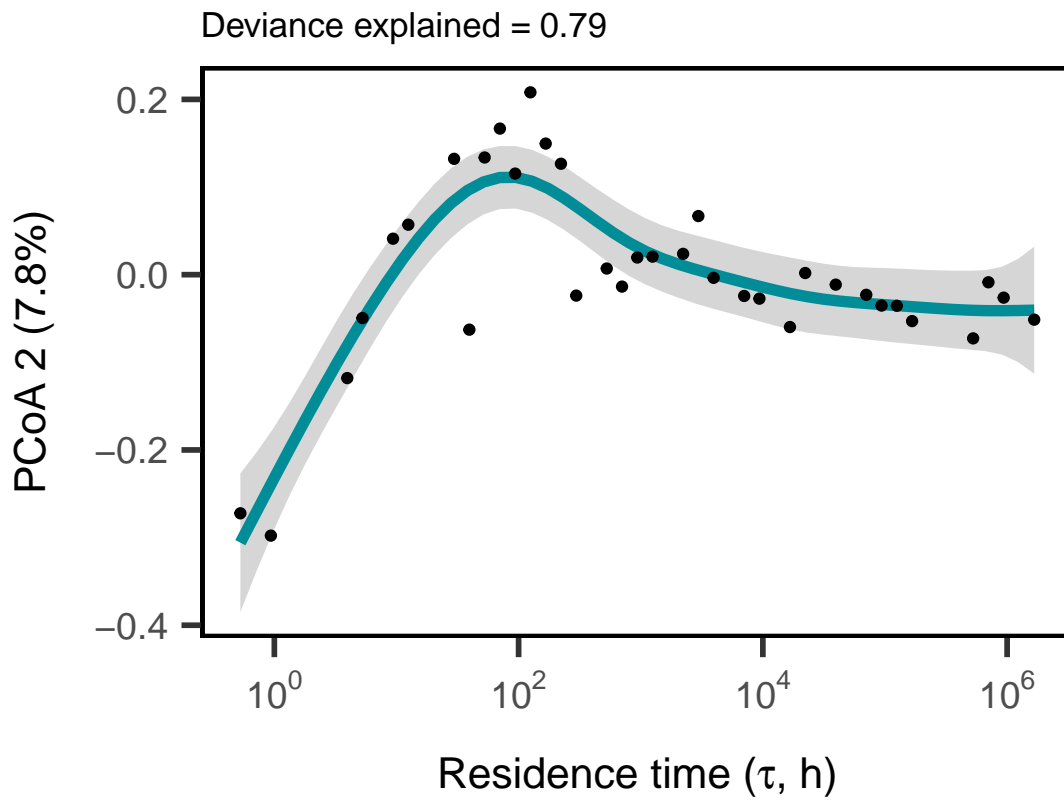
gam.check(CommComp.PCoA2.gam)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [-4.364615e-10,4.034817e-11]
## (score -40.91816 & scale 0.002777386).
## Hessian positive definite, eigenvalue range [1.626398,16.8182].
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'   edf k-index p-value
## s(Tau) 9.00 5.35   0.96   0.29
mean(summary(CommComp.PCoA2.gam)$s.table[, 4])

## [1] 0

CommComp_PCoA2 <- predict_gam(CommComp.PCoA2.gam) %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = CC.plot, aes(x = Tau,
    y = V2)) + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + xlab(expression(paste("Residence time (",
    tau, ", h)"))) + ylab(paste("PCoA 2 (", explainvar2, "%)",
    sep = "")) + labs(title = bquote("Deviance explained =" ~
    .(signif(summary(CommComp.PCoA2.gam)$dev, 2)))) + theme(plot.margin = unit(c(0.2,
    0.2, 0.2, 0.5), "cm"))
```

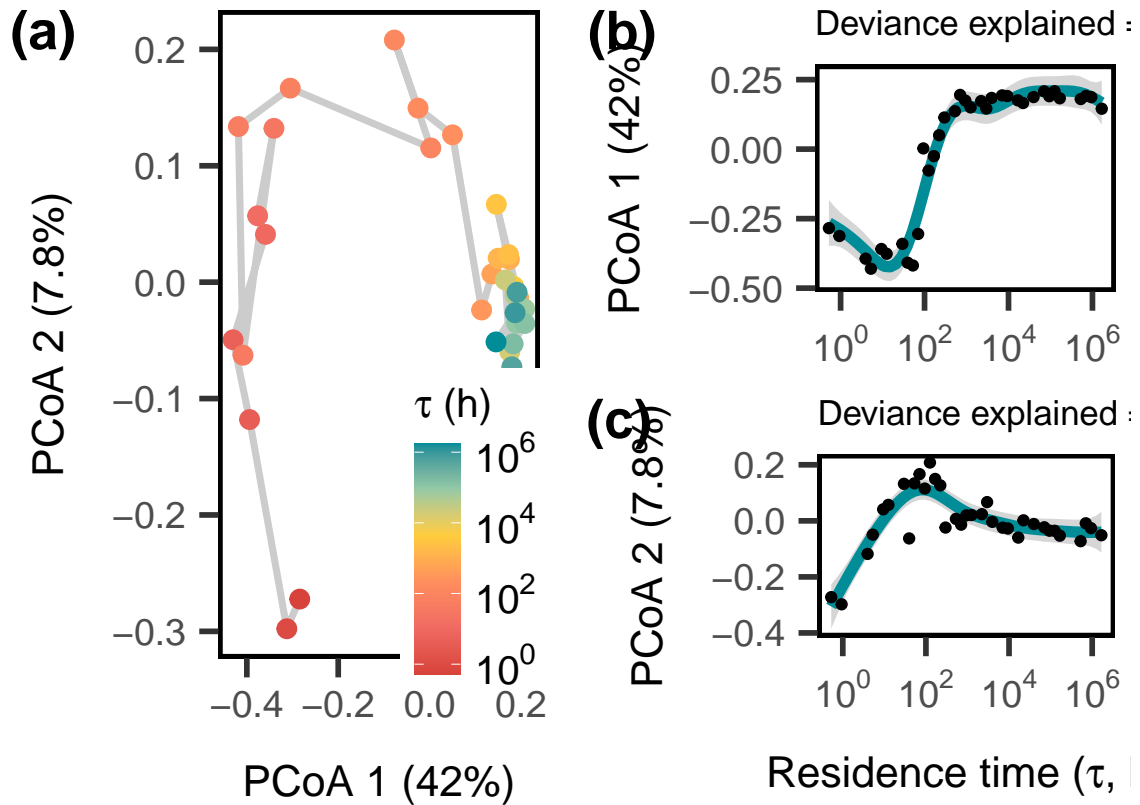


```
ggsave("../output/CommComp_PCoA2.pdf")
ggsave("../output/CommComp_PCoA2.png", width = 6.5, height = 5)
```

### Draw Figure

```
ggdraw() + draw_plot(CommComp_PCoA, x = 0, y = 0, width = 0.5,
  height = 1) + draw_plot(CommComp_PCoA1, x = 0.5, y = 0.55,
  width = 0.5, height = 0.45) + draw_plot(CommComp_PCoA2, x = 0.5,
  y = 0, width = 0.5, height = 0.55) + draw_plot_label(label = c("(a)",
  "(b)", "(c)"), size = 20, x = c(0, 0.5, 0.5), y = c(1.005,
  1.005, 0.555)) + theme(plot.background = element_rect(fill = "white",
  color = NA))
```





```
ggsave("./output/RTL_C_Fig4.pdf")
ggsave("./output/RTL_C_Fig4.png", width = 12, height = 6, dpi = 800)
```

Figure 5. Niche partitioning along a residence time gradient.

#### Generate niche clusters

```
# Generate OTU distance matrix using Spearman's correlation
# coefficient
OTUs.70 <- as.data.frame(cbind(Tau.20$Tau, OTUs.r.REL.20[, colnames(OTUs.r.REL.20) %in%
  colnames(as.data.frame(OTUs.r.PA.20) %>%
    select_if(colSums(OTUs.r.PA.20) >= 25)))]), [-1]

OTUs.70 <- OTUs.70[order(as.numeric(row.names(OTUs.70))), ]

# Pairwise correlation of OTU trajectory across residence
# time gradient and distance matrix - time consuming,
# commented out to save time

# OTUs.70.corr <- as.data.frame(colnames(OTUs.70))
# colnames(OTUs.70.corr) <- c('OTU') for (otu1 in
# OTUs.70.corr$OTU){ corr <- c() for(otu2 in
# OTUs.70.corr$OTU){ corr <- c(corr,
# cor(as.numeric(OTUs.70[,paste0(otu1)]),
# as.numeric(OTUs.70[,paste0(otu2)]), method = 'pearson'))
```

```

# } OTUs.70.corr[,paste0(otu1)] <- corr } OTUs.70.dist <-
# ((1-OTUs.70.corr[,2:2205])/2) rownames(OTUs.70.dist) <-
# colnames(OTUs.70.corr[,2:2205]) OTUs.70.dist <-
# as.matrix(OTUs.70.dist) OTUs.70.dist <-
# as.dist(OTUs.70.dist, upper = TRUE, diag = TRUE)
# saveRDS(OTUs.70.dist, './data/OTUs.70.dist')

# Read in saved distance matrix
OTUs.70.dist <- readRDS("./data/OTUs.70.dist")

# Perform clustering with 6 clusters
set.seed(47401)
kmeans.clust <- kmeans(OTUs.70.dist, centers = 5)

# Calculate total abundance of each niche at each residence
# time
OTUs.niches <- data.frame()
OTUs.niches.sum <- as.data.frame(as.numeric(rownames(OTUs.70)))
n.cluster <- 1
while (n.cluster <= 5) {
  cond <- sapply(kmeans.clust$cluster, function(x) x == n.cluster)
  cluster <- names(kmeans.clust$cluster[cond])
  OTUs.cluster <- cbind(as.numeric(rownames(OTUs.70)), rep(n.cluster,
    35))
  colnames(OTUs.cluster) <- c("Tau", "Cluster")
  OTUs.cluster <- cbind(OTUs.cluster, OTUs.70[, colnames(OTUs.70) %in%
    cluster])
  OTUs.niches.sum <- cbind(OTUs.niches.sum, rowSums(OTUs.70[,
    colnames(OTUs.70) %in% cluster]))
  OTUs.cluster <- as.data.frame(OTUs.cluster) %>%
    gather(OTU, REL, cluster[1]:cluster[length(cluster)])
  colnames(OTUs.cluster) <- c("Tau", "Cluster", "OTU", "REL")
  OTUs.niches <- rbind(OTUs.niches, OTUs.cluster)
  n.cluster <- n.cluster + 1
}

colnames(OTUs.niches.sum) <- c("Tau", 1:(n.cluster - 1))
OTUs.niches.sum <- as.data.frame(OTUs.niches.sum) %>%
  gather(Cluster, REL, "1":paste(n.cluster - 1))

OTUs.niches.sum$Cluster <- as.numeric(OTUs.niches.sum$Cluster)
OTUs.niches.sum$Cluster <- as.factor(OTUs.niches.sum$Cluster)

```

## Classify taxa by niche and determine proportion of each niche

```

niche.taxa <- data.frame()
for (n in 1:5) {
  num <- nrow(subset(OTUs.tax, OTUs.tax$OTU %in% names(kmeans.clust$cluster[sapply(kmeans.clust$cluster,
    function(x) x == n)])))
  niche.taxa <- rbind(niche.taxa, cbind(rep(n, num), rep(1/num,
    num), subset(OTUs.tax, OTUs.tax$OTU %in% names(kmeans.clust$cluster[sapply(kmeans.clust$cluster,
    function(x) x == n)]))))
}

```

```

colnames(niche.taxa) <- c("Niche", "Prop_num", "OTU", "Domain",
  "Phylum", "Class", "Order", "Family", "Genus")
niche.taxa$Niche <- as.factor(niche.taxa$Niche)
niche.taxa$Prop_num <- as.numeric(niche.taxa$Prop_num)

for (x in 1:5) {
  print(paste("Niche: ", x, "    N:", nrow(subset(niche.taxa,
    niche.taxa$Niche == x)), "    %:", round(nrow(subset(niche.taxa,
    niche.taxa$Niche == x))/nrow(niche.taxa), digits = 4)))
}

```

```

## [1] "Niche:  1      N: 673      %: 0.3054"
## [1] "Niche:  2      N: 427      %: 0.1937"
## [1] "Niche:  3      N: 336      %: 0.1525"
## [1] "Niche:  4      N: 287      %: 0.1302"
## [1] "Niche:  5      N: 481      %: 0.2182"

```

## Fit total abundance of niches with gams and draw figure

```

niche.gams <- data.frame()
for (n in 1:5) {
  niche.gams <- rbind(niche.gams, cbind(rep(n, 50), predict_gam(gam(REL ~
    s(Tau), family = gaussian(link = "identity"), data = subset(OTUs.niches.sum,
    OTUs.niches.sum$Cluster == n), method = "REML"))))
  print(summary(gam(REL ~ s(Tau), family = gaussian(link = "identity"),
    data = subset(OTUs.niches.sum, OTUs.niches.sum$Cluster ==
    n), method = "REML")))
  print(mean(summary(gam(REL ~ s(Tau), family = gaussian(link = "identity"),
    data = subset(OTUs.niches.sum, OTUs.niches.sum$Cluster ==
    n), method = "REML"))$s.table[, 4]))
}

```

```

##
## Family: gaussian
## Link function: identity
##
## Formula:
## REL ~ s(Tau)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.335576   0.006688   50.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(Tau) 7.04   8.086 61.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.936   Deviance explained = 94.9%
## -REML = -46.42   Scale est. = 0.0015655   n = 35
## [1] 0

```

```

##
## Family: gaussian
## Link function: identity
##
## Formula:
## REL ~ s(Tau)
##
## Parametric coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.104320   0.005555  18.78  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df    F  p-value
## s(Tau) 3.346  4.153 11.8 5.57e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.587   Deviance explained = 62.8%
## -REML = -59.924   Scale est. = 0.00108    n = 35
## [1] 5.568932e-06
##
## Family: gaussian
## Link function: identity
##
## Formula:
## REL ~ s(Tau)
##
## Parametric coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.08128   0.00248  32.78  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df    F  p-value
## s(Tau) 5.676  6.797 5.224 0.000724 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.512   Deviance explained = 59.4%
## -REML = -82.46   Scale est. = 0.00021521    n = 35
## [1] 0.000724011
##
## Family: gaussian
## Link function: identity
##
## Formula:
## REL ~ s(Tau)
##
## Parametric coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.220047   0.008547  25.75  <2e-16 ***

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##      edf Ref.df      F p-value
## s(Tau) 6.911   7.98 49.65  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.921   Deviance explained = 93.7%
## -REML = -38.689   Scale est. = 0.0025567   n = 35
## [1] 0
##
## Family: gaussian
## Link function: identity
##
## Formula:
## REL ~ s(Tau)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.074524   0.002968   25.11  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##      edf Ref.df      F p-value
## s(Tau) 4.111   5.057 9.058 2.39e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.571   Deviance explained = 62.3%
## -REML = -79.423   Scale est. = 0.00030832   n = 35
## [1] 2.390613e-05

colnames(niche.gams) <- c("Cluster", "Tau", "fit", "se.fit")

niche.gams$Cluster <- as.numeric(niche.gams$Cluster)
niche.gams$Cluster <- as.factor(niche.gams$Cluster)

niche.gams$umin <- (log(2)/(1/((10^niche.gams$Tau) * 60)))
niche.gams$umin <- na_if(niche.gams$umin, niche.gams$Tau < 3)
scale.factor <- max(subset(niche.gams, niche.gams$Cluster %in%
  c(1, 4))$fit)/max(niche.gams$umin)

tau.1 <- log10(1/(log(2)))
tau.100 <- log10(100/(log(2)))
tau.12000 <- log10(12000/(log(2)))
tau.876000 <- log10(876000/(log(2)))

OTUs_major_niches <- ggplot(data = subset(niche.gams, niche.gams$Cluster %in%
  c(1, 4)), aes(x = Tau, y = fit, group = Cluster)) + geom_smooth_ci(cex = 2,
  ci_alpha = 0.2, color = "grey") + geom_line(aes(color = Cluster),

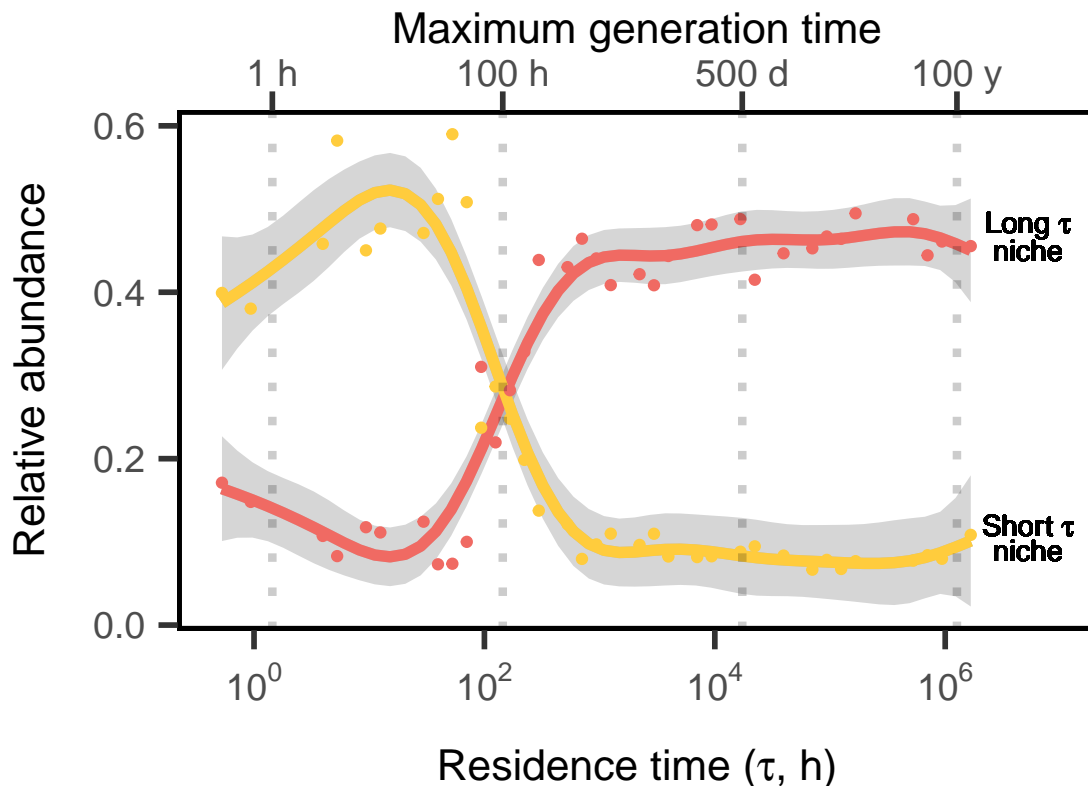
```

```

linewidth = 2) + scale_y_continuous(name = "Relative abundance") +
geom_point(data = subset(OTUs.niches.sum, OTUs.niches.sum$Cluster %in%
  c(1, 4)), aes(x = Tau, y = REL, color = Cluster)) + geom_vline(xintercept = c(tau.1,
tau.100, tau.12000, tau.876000), color = alpha("black", 0.2),
linetype = "13", linewidth = 1.5) + scale_x_continuous(labels = label_math(expr = 10^.x,
format = force), limits = c(-0.3, 6.95), sec.axis = sec_axis(~log10(log(2)/(1/(10^.))),
name = "Maximum generation time", breaks = c(0, 2, log10(12000),
  log10(876000)), labels = c("1 h", "100 h", "500 d", "100 y"))) +
scale_color_manual(values = as.vector(c(moma.colors("ustwo",
5)[1], moma.colors("ustwo", 5)[3]))) + xlab(expression(paste("Residence time (",
tau, ", h)"))) + theme(legend.position = "none") + geom_text(x = 6.72,
y = 0.48, label = expression(paste("Long ", tau))) + geom_text(x = 6.72,
y = 0.45, label = "niche") + geom_text(x = 6.72, y = 0.12,
label = expression(paste("Short ", tau))) + geom_text(x = 6.72,
y = 0.09, label = "niche")

```

OTUs\_major\_niches



```

ggsave("./output/RTLC_Fig5.pdf")
ggsave("./output/RTLC_Fig5.png", width = 6, height = 4, dpi = 800)

```

Test phylogenetic clustering of Niche 1 and 4 and look at unique taxa in each

```

# Generate phylogenetic tree with niche as tip labels
niche1.otus <- subset(niche.taxa, niche.taxa$Niche == 1)$OTU

```

```

niche4.otus <- subset(niche.taxa, niche.taxa$Niche == 4)$OTU
niche3.otus <- subset(niche.taxa, niche.taxa$Niche == 3)$OTU
niche2.otus <- subset(niche.taxa, niche.taxa$Niche == 2)$OTU
niche5.otus <- subset(niche.taxa, niche.taxa$Niche == 5)$OTU

niche_tree <- read.tree("./data/niche_oturep.phylip.tre")
tip_otus <- read.csv("./data/seq_to_otu.csv", header = FALSE)
colnames(tip_otus) <- c("Tips", "Otu")
tip_otus$Tips <- factor(tip_otus$Tips, levels = unique(as.character(niche_tree$tip.label)))
tip_otus <- tip_otus[order(tip_otus$Tips), ]
niche_tree$tip.label <- tip_otus$Otu

niche1 <- sapply(niche1.otus, grep, niche_tree$tip.label)
niche4 <- sapply(niche4.otus, grep, niche_tree$tip.label)
niche3 <- sapply(niche3.otus, grep, niche_tree$tip.label)
niche2 <- sapply(niche2.otus, grep, niche_tree$tip.label)
niche5 <- sapply(niche5.otus, grep, niche_tree$tip.label)

niche_tree$edge.length[1] <- niche_tree$edge.length[296]

# Calculate Phylogenetic D and Cohen's D for Niche 1 and 4
niche.list <- as.data.frame(niche.taxa[, "Niche"])
rownames(niche.list) <- niche.taxa$OTU
colnames(niche.list) <- c("Niche")
niche.list$Niche <- as.numeric(niche.list$Niche)
niche.list <- as.data.frame((niche.list == 4) * 1)
niche.list$name <- rownames(niche.list)

rtlc.niche <- comparative.data(niche_tree, niche.list, "name")
D.otu.4 <- phylo.d(rtlc.niche, binvar = Niche)

niche.4.D <- D.otu.4$Permutations[["random"]]
cohensD(x = as.numeric(D.otu.4$Parameters[["Observed"]]), y = niche.4.D)

## [1] 19.14312

niche.list <- as.data.frame(niche.taxa[, "Niche"])
rownames(niche.list) <- niche.taxa$OTU
colnames(niche.list) <- c("Niche")
niche.list$Niche <- as.numeric(niche.list$Niche)
niche.list <- as.data.frame((niche.list == 1) * 1)
niche.list$name <- rownames(niche.list)

rtlc.niche <- comparative.data(niche_tree, niche.list, "name")
D.otu.1 <- phylo.d(rtlc.niche, binvar = Niche)

niche.1.D <- D.otu.1$Permutations[["random"]]
cohensD(x = as.numeric(D.otu.1$Parameters[["Observed"]]), y = niche.1.D)

## [1] 15.36044

# Look at unique taxonomic orders between two major niches
n4.unique <- subset(niche.taxa, niche.taxa$Niche == 4)

```

```
n4.unique <- subset(n4.unique, !(n4.unique$Order %in% intersect(unique(subset(niche.taxa,
  niche.taxa$Niche == 1)$Order), unique(subset(niche.taxa,
  niche.taxa$Niche == 4)$Order))))

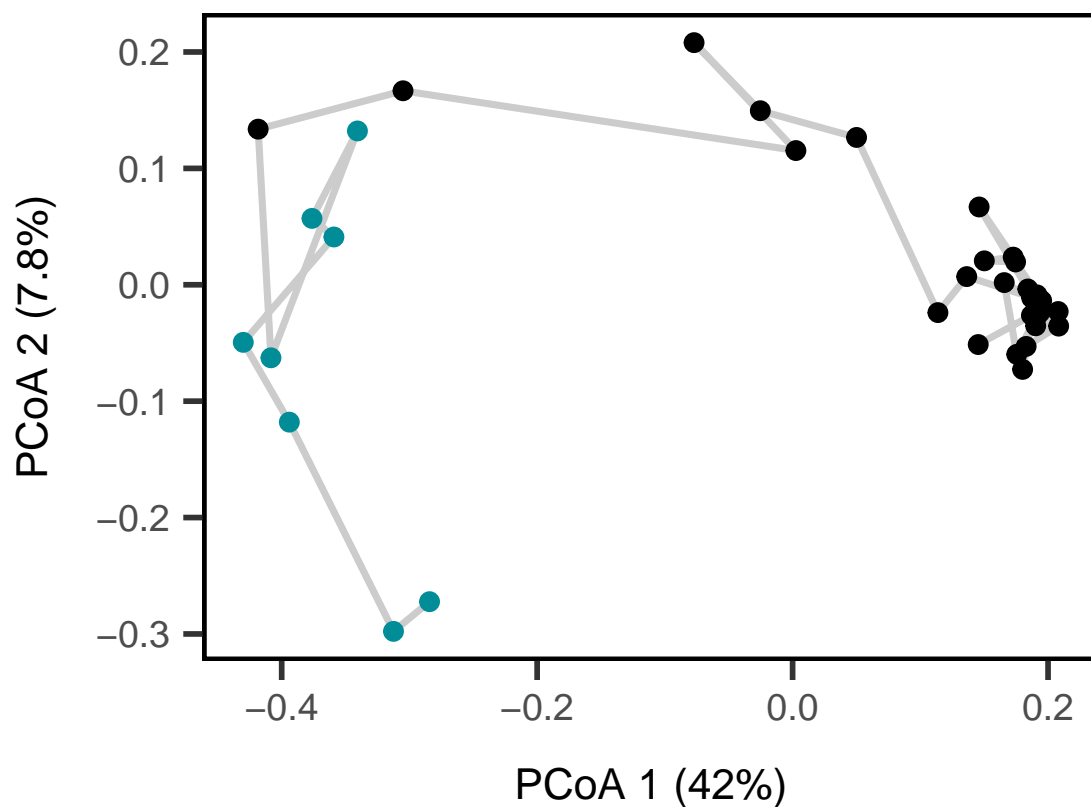
n1.unique <- subset(niche.taxa, niche.taxa$Niche == 1)
n1.unique <- subset(n1.unique, !(n1.unique$Order %in% intersect(unique(subset(niche.taxa,
  niche.taxa$Niche == 1)$Order), unique(subset(niche.taxa,
  niche.taxa$Niche == 4)$Order))))
```

**Figure S1. Community composition and resource use do not reflect mode of flow-rate manipulation.**

#### A. Community composition colored by pump/pipette

```
CommComp_pump <- ggplot(CC.plot.order, aes(x = V1, y = V2)) +
  geom_path(linewidth = 1.25, color = alpha("black", 0.2)) +
  xlab(paste("PCoA 1 (", explainvar1, "%)", sep = "")) + ylab(paste("PCoA 2 (",
  explainvar2, "%)", sep = "")) + geom_point(cex = 3, aes(color = Pump)) +
  labs(color = "Flow rate\nmanipulation") + scale_color_manual(values = c("#008d98",
  "black"), labels = c("Peristaltic \nPump", "Manual \nPipette")) +
  theme(legend.title = element_text(size = 14), legend.position = "none")
```

CommComp\_pump



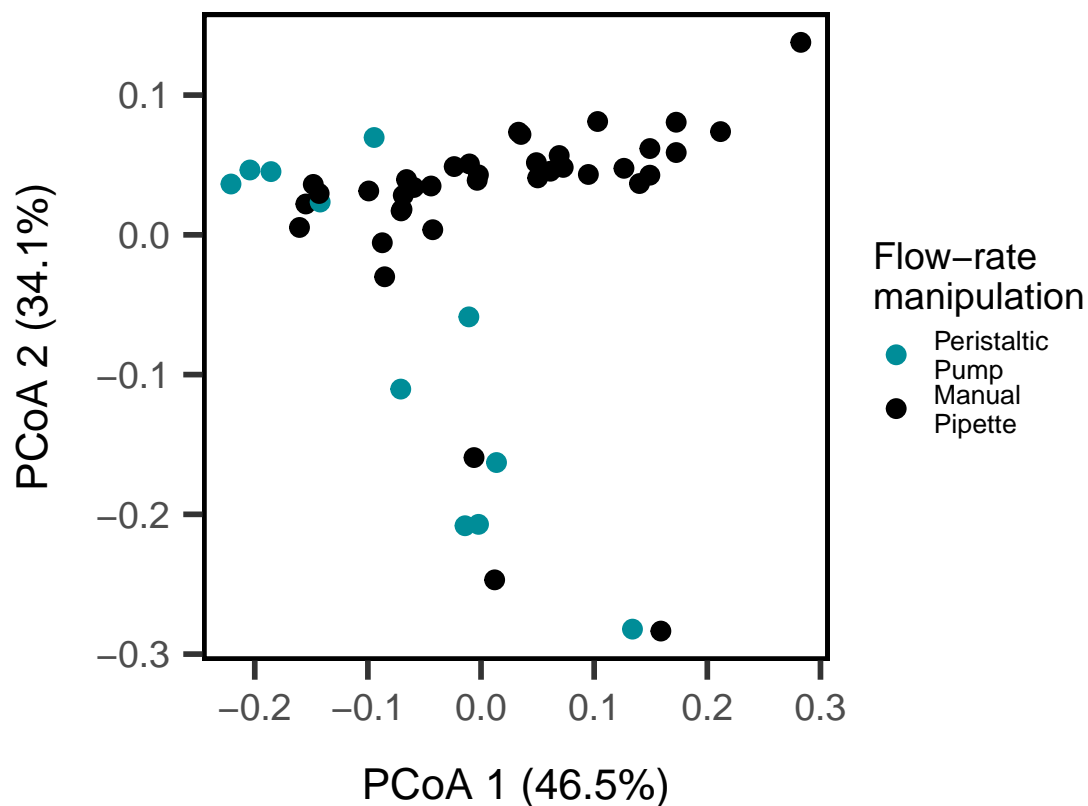


```
ggsave("./output/CommComp_PCoA_Pump.pdf")
ggsave("./output/CommComp_PCoA_Pump.png", width = 8, height = 5)
```

## B. Resource use colored by pump/pipette

```
ResUse_pump <- ggplot(EP.plot, aes(x = V1, y = V2)) + xlab(paste("PCoA 1 (",
  explainvar1.ep, "%)", sep = "")) + ylab(paste("PCoA 2 (",
  explainvar2.ep, "%)", sep = "")) + geom_point(cex = 3, aes(color = Pump)) +
  labs(color = "Flow-rate\nmanipulation") + scale_color_manual(values = c("#008d98",
  "black"), labels = c("Peristaltic \nPump", "Manual \nPipette")) +
  theme(legend.title = element_text(size = 14))
```

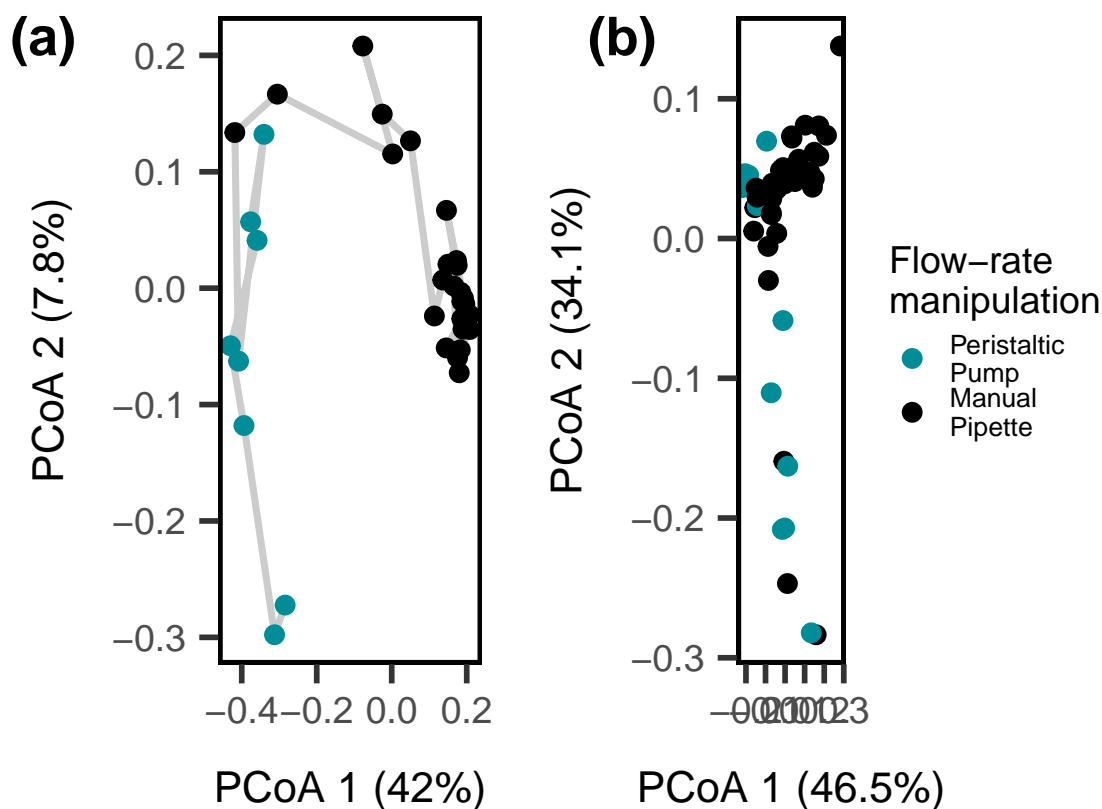
ResUse\_pump



```
ggsave("./output/ResUse_PCoA_Pump.pdf")
ggsave("./output/ResUse_PCoA_Pump.png", width = 8, height = 5)
```

## Draw Figure

```
ggdraw() + draw_plot(CommComp_pump, x = 0, y = 0, width = 0.45,
  height = 1) + draw_plot(ResUse_pump, x = 0.45, y = 0, width = 0.55,
  height = 1) + draw_plot_label(label = c("(a)", "(b)"), size = 20,
  x = c(0, 0.5), y = c(1.005, 1.005)) + theme(plot.background = element_rect(fill = "white",
  color = NA))
```



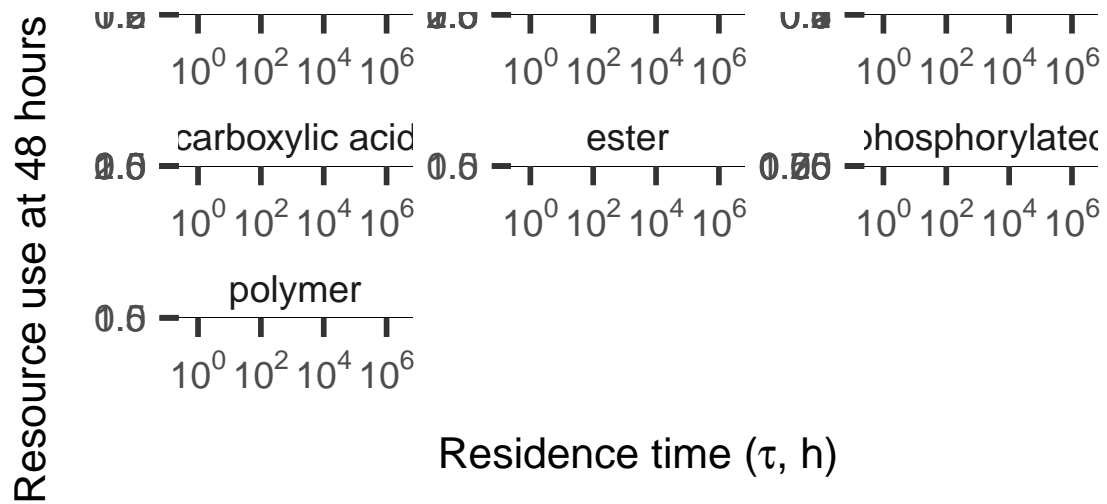
```
ggsave("./output/RTL_C_FigS1.pdf")
ggsave("./output/RTL_C_FigS1.png", width = 12, height = 6, dpi = 800)
```

Figure S3. Use of individual carbon sources mainly decreases with increased residence time.

```
Ind_ResUse <- ggplot(data = csource.gams, aes(x = Tau, y = fit,
group = C_Source)) + geom_smooth_ci(cex = 2, ci_alpha = 0.2,
color = "grey") + geom_line(cex = 2, aes(color = C_Source)) +
facet_wrap(~Type, scales = "free") + geom_point(data = subset(EP.long,
EP.long$C_Source %in% csource.sig$Resource), aes(x = Tau,
y = OD_48, color = C_Source, group = C_Source)) + scale_x_continuous(labels = label_math(expr = 10^
format = force)) + ylab("Resource use at 48 hours (OD 590)") +
scale_color_discrete(type = moma.colors("ustwo", 16, type = "continuous"),
labels = c("4-Hydroxy Benzoic Acid", expression(paste("D-Galactonic Acid ",
gamma, "-Lactone")), "D-Galacturonic Acid", "D-Glucosaminic Acid",
"D-Xylose", expression(paste(gamma, "-Hydroxybutyric Acid")),
"Glucose-1-Phosphate", "Glycyl-L-Glutamic Acid",
"L-Arginine", "L-Asparagine", "L-Serine", "L-Threonine",
"Putrescine", "Pyruvic Acid Methyl Ester", "Tween 40",
"Tween 80")) + xlab(expression(paste("Residence time (",
tau, ", h)"))) + guides(color = guide_legend(nrow = 6)) +
theme(legend.position = "bottom", legend.title = element_blank(),
legend.box.background = element_rect(color = "black",
```

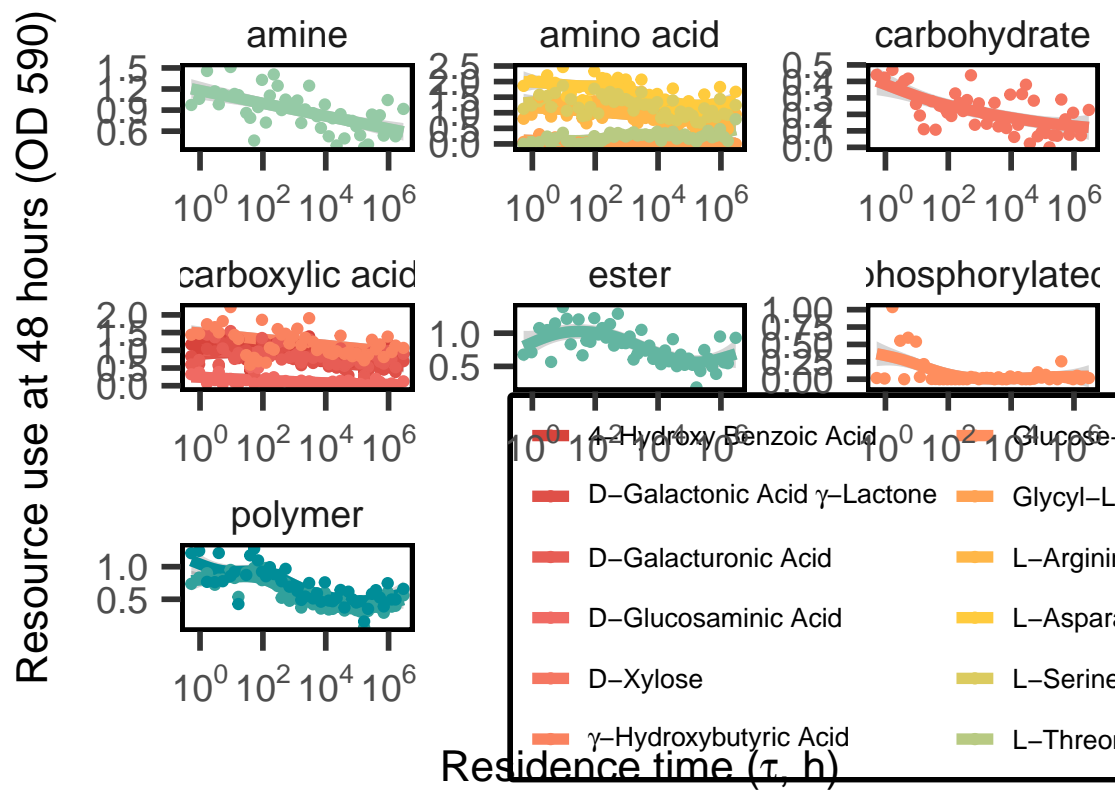
```
size = 2))
```

Ind\_ResUse



- |                                     |                        |                         |
|-------------------------------------|------------------------|-------------------------|
| 4-Hydroxy Benzoic Acid              | Glucose-1-Phosphate    | Putrescine              |
| D-Galactonic Acid $\gamma$ -Lactone | Glycyl-L-Glutamic Acid | Pyruvic Acid Metabolite |
| D-Galacturonic Acid                 | L-Arginine             | Tween 40                |
| D-Glucosaminic Acid                 | L-Asparagine           | Tween 80                |
| D-Xylose                            | L-Serine               |                         |

```
ggsave("../output/RTL_C_FigS3.pdf", plot = reposition_legend(Ind_ResUse,
  "left", panel = "panel-3-2"))
```



```
ggsave("../output/RTL_C_FigS3.png", plot = reposition_legend(Ind_ResUse,
  "left", panel = "panel-3-2"), width = 12, height = 10, dpi = 800)
```

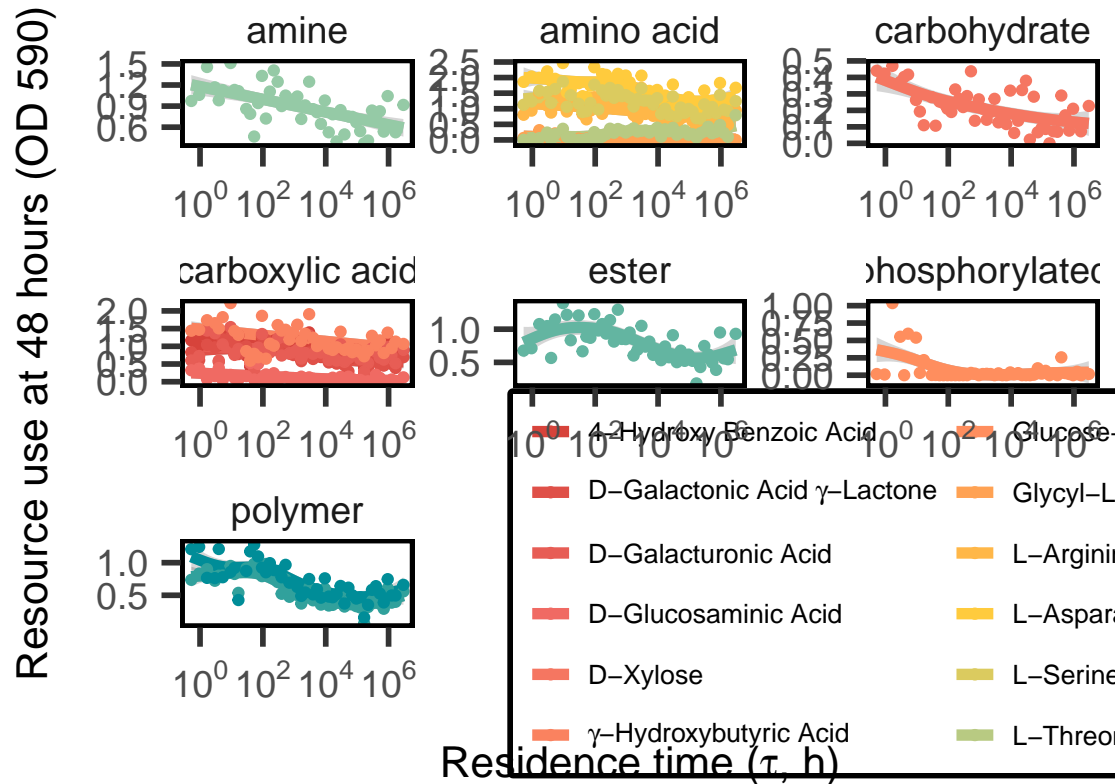


Figure S4. Phylum level changes across the residence time gradient.

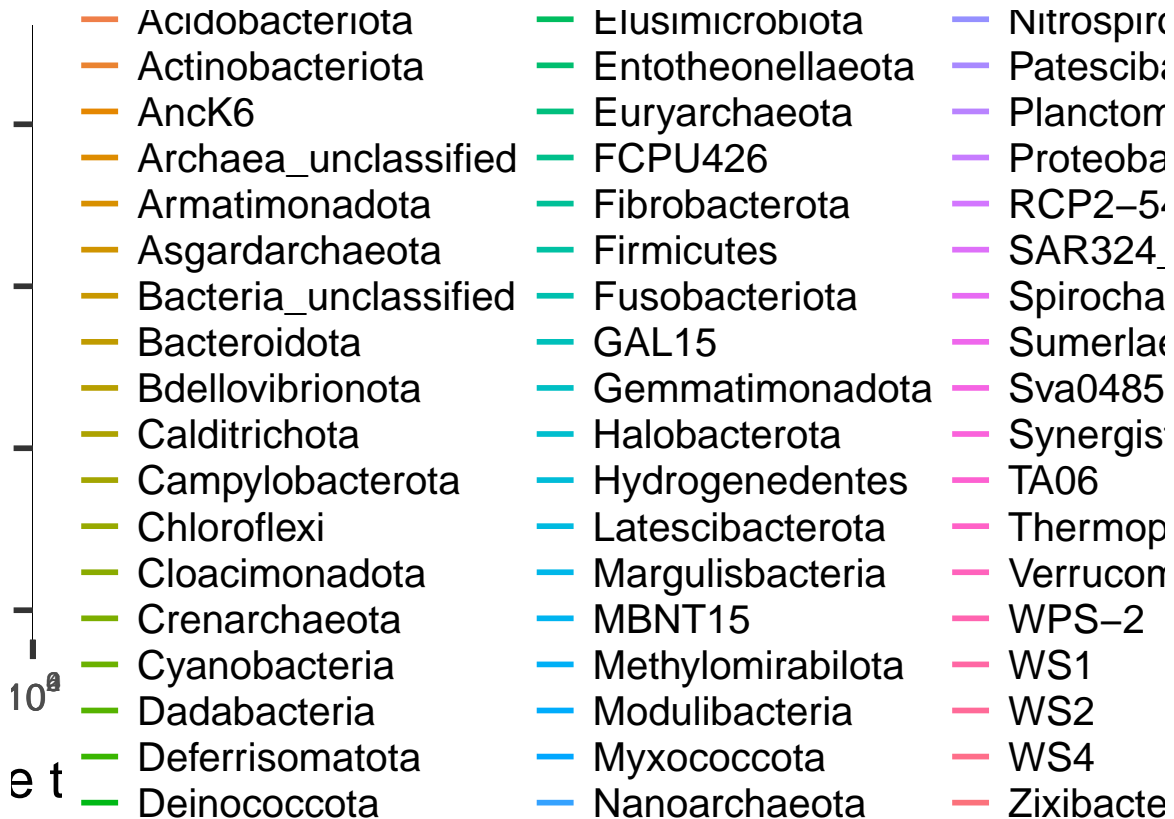
```
# Generate total relative abundance for each phylum - time
# consuming, commented out to save time

# taxa <- as.data.frame(Tau.20$Tau) for(x in
# unique(OTUs.tax.20$Phylum)){ list <- data.frame(rep(0,
# 35)) colnames(list) <- c(as.character(x)) for(y in
# OTUs.tax.20$OTU){ if(subset(OTUs.tax.20, OTUs.tax.20$OTU
# == y)$Phylum == x){ list[,x] <- list[,x] +
# as.data.frame(OTUs.r.20[,y]) } } taxa[,as.character(x)]
# <- list[,x] } rownames(taxa) <- taxa$`Tau.20$Tau` taxa.20
# <- taxa[, -1] write.csv(taxa.20, './data/phyla.20.csv')

# Read in phylum abundance data generated above
phyla.20 <- read.csv("./data/phyla.20.csv", header = TRUE)
rownames(phyla.20) <- phyla.20$X
phyla.20 <- phyla.20[, -1]
phyla.20.rel <- decostand(phyla.20, method = "total")
phyla.20.rel <- cbind(Tau.20$Tau, phyla.20.rel)
colnames(phyla.20.rel) <- c("Tau", unique(OTUs.tax.20$Phylum))
phyla.20.long <- gather(phyla.20.rel, phylum, REL, Proteobacteria:Modulibacteria)

phyla.20.long$Tau <- as.numeric(phyla.20.long$Tau)
```

```
Phylum_Tau <- ggplot(phyla.20.long, aes(x = Tau, y = REL, group = phylum)) +
  geom_smooth(method = "loess", se = F, aes(color = phylum,
    group = phylum)) + ylab("Relative abundance") + scale_x_continuous(labels = label_math(expr = 10^h),
    format = force)) + labs(color = "Phylum") + xlab(expression(paste("Residence time (",
    tau, ", h)"))) + theme(legend.title = element_text(size = 20),
    legend.text = element_text(size = 15), axis.title = element_text(size = 20)) +
    guides(color = guide_legend(ncol = 3))
Phylum_Tau
```



```
ggsave("./output/RTLC_FigS4.pdf")
ggsave("./output/RTLC_FigS4.png", width = 12, height = 8, dpi = 800)
```

**Figure S5.** Amplicon sequence variant richness and evenness reflect operational taxonomic unit richness and evenness.

Add ASV S and SimpE to Tau

```
Tau <- ASV.S.fxn(S.ASV, Tau)
Tau$ASV_Day_20.S <- as.numeric(Tau$ASV_Day_20.S)
Tau$ASV_Day_0.S <- as.numeric(Tau$ASV_Day_0.S)

Tau <- ASV.SimpE.fxn(SimpE.ASV, Tau)
Tau$ASV_Day_0.SimpE <- as.numeric(Tau$ASV_Day_0.SimpE)
Tau$ASV_Day_20.SimpE <- as.numeric(Tau$ASV_Day_20.SimpE)
```

## A. Species Richness - ASVs

```
S.ASV.gam <- gam(ASV_Day_20.S ~ s(Tau, k = 14), family = gaussian(link = "identity"),
  data = Tau, method = "REML")
S.ASV.gam.re <- gam(ASV_Day_20.S ~ s(Tau) + s(Set, bs = "re"),
  family = gaussian(link = "identity"), data = Tau, method = "REML")
```

```
AIC(S.ASV.gam, S.ASV.gam.re)
```

```
##              df      AIC
## S.ASV.gam    8.390120 488.3867
## S.ASV.gam.re 8.155298 488.2937
```

```
anova(S.ASV.gam, S.ASV.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: ASV_Day_20.S ~ s(Tau, k = 14)
## Model 2: ASV_Day_20.S ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1      26.340    1457077
## 2      26.744    1472838 -0.40446   -15762    0.272
```

```
# Random effect of set is not significant
```

```
rm(S.ASV.gam.re)
```

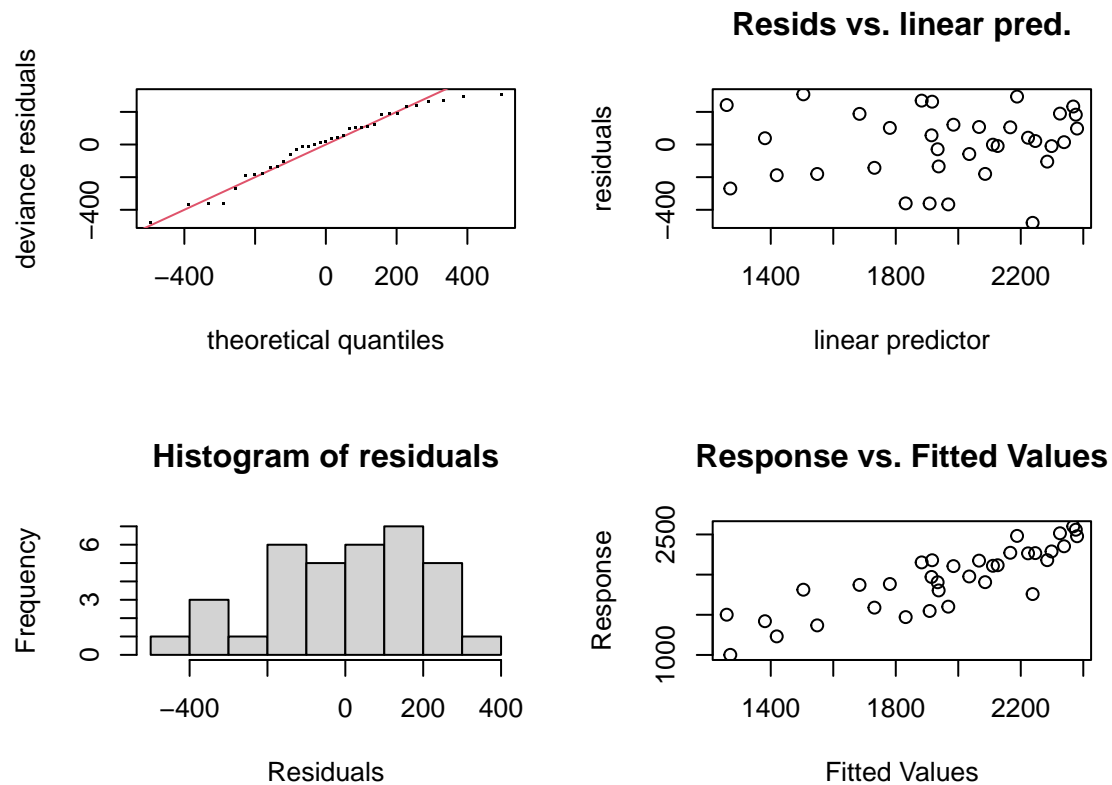
```
summary(S.ASV.gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## ASV_Day_20.S ~ s(Tau, k = 14)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1963.31      38.23   51.35  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(Tau)  5.52    6.79 10.96 2.32e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.684   Deviance explained = 73.5%
## -REML = 235.18   Scale est. = 51162      n = 35
```

```
k.check(S.ASV.gam)
```

```
##           k'      edf k-index p-value
## s(Tau) 13 5.520123 1.510897      1
```

```
gam.check(S.ASV.gam)
```



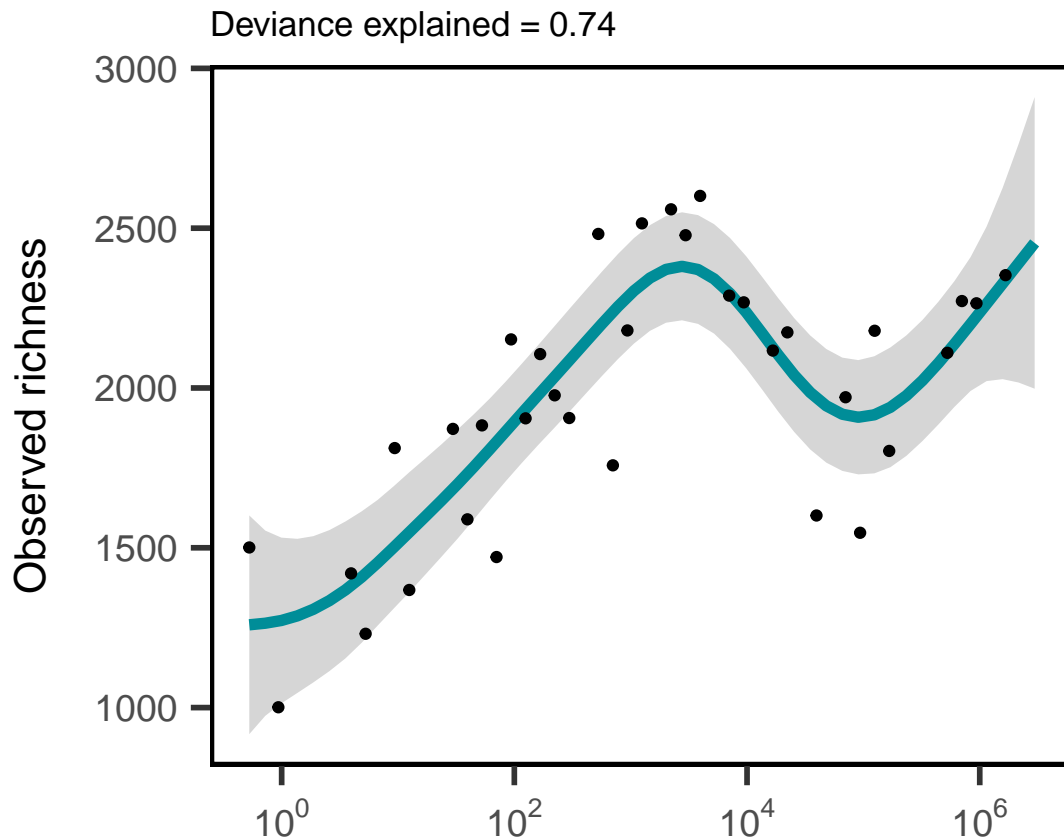
```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-4.287254e-06,-6.357866e-07]
## (score 235.1819 & scale 51161.62).
## Hessian positive definite, eigenvalue range [1.108322,16.83186].
## Model rank = 14 / 14
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'   edf k-index p-value
## s(Tau) 13.00  5.52   1.51      1
mean(summary(S.ASV.gam)$s.table[, 4])

## [1] 2.31524e-06
S.ob_ASV_Tau <- predict_gam(S.ASV.gam) %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = Tau, aes(x = Tau,
    y = ASV_Day_20.S)) + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + ylab("Observed richness") + theme(axis.title.x = element_blank()) +
  labs(title = bquote("Deviance explained =" ~ .(signif(summary(S.ASV.gam)$dev,
    2)))) + theme(plot.margin = unit(c(0.2, 0.2, 0.2, 0.5),
```



```
"cm"))
```

```
S.ob_ASV_Tau
```



```
ggsave("./output/S.ob_ASV_Tau.pdf")  
ggsave("./output/S.ob_ASV_Tau.png", width = 7, height = 5)
```

## B. Species evenness - ASVs

```
SimpE.ASV.gam <- gam(ASV_Day_20.SimpE ~ s(Tau, k = 15), family = gaussian(link = "identity"),  
  data = Tau, method = "REML")  
SimpE.ASV.gam.re <- gam(ASV_Day_20.SimpE ~ s(Tau) + s(Set, bs = "re"),  
  family = gaussian(link = "identity"), data = Tau, method = "REML")
```

```
AIC(SimpE.ASV.gam, SimpE.ASV.gam.re)
```

```
##           df      AIC  
## SimpE.ASV.gam  8.034365 -91.4506  
## SimpE.ASV.gam.re 11.527062 -100.9649
```

```
anova(SimpE.ASV.gam, SimpE.ASV.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model 1: ASV_Day_20.SimpE ~ s(Tau, k = 15)
```

```
## Model 2: ASV_Day_20.SimpE ~ s(Tau) + s(Set, bs = "re")
```

```

##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1    26.810   0.094940
## 2    22.883   0.059254 3.9269 0.035686 0.003237 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Random effect of set is significant

rm(SimpE.ASV.gam)

summary(SimpE.ASV.gam.re)

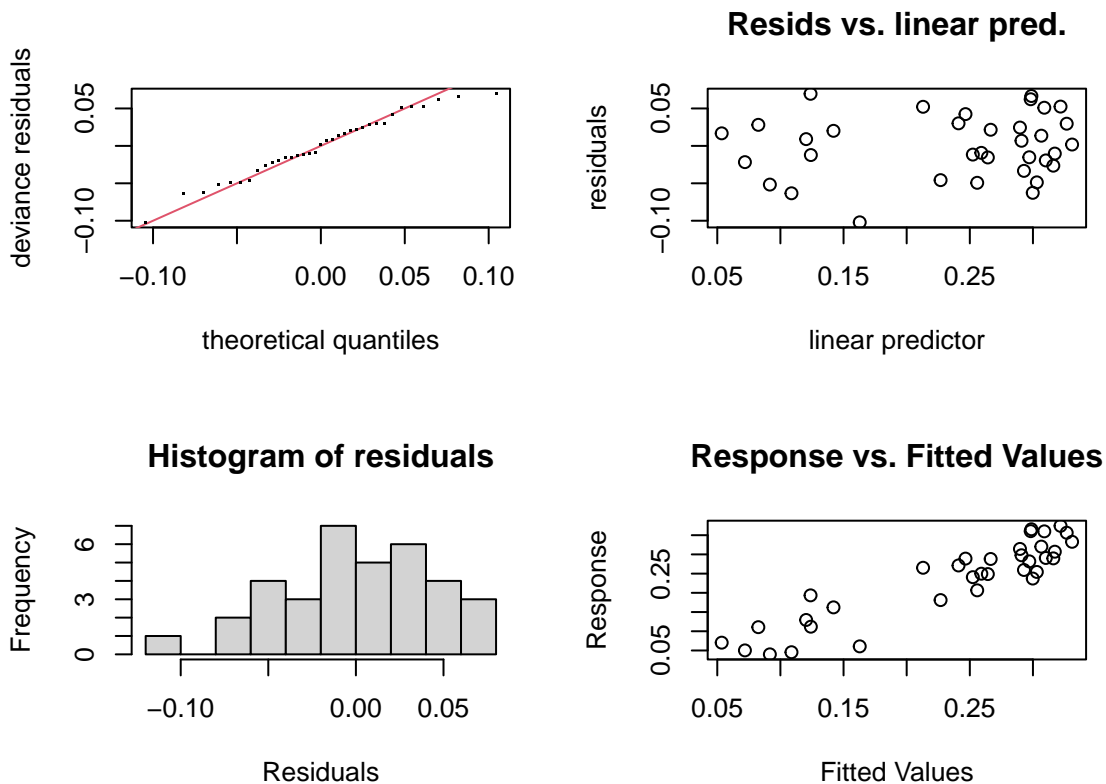
##
## Family: gaussian
## Link function: identity
##
## Formula:
## ASV_Day_20.SimpE ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.23700    0.01959   12.1 3.37e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df      F  p-value
## s(Tau) 5.551  6.668 13.59 9.53e-07 ***
## s(Set) 2.386  3.000  3.56  0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.78   Deviance explained = 83.1%
## -REML = -41.309   Scale est. = 0.0022735   n = 35

k.check(SimpE.ASV.gam.re)

##           k'           edf  k-index p-value
## s(Tau)  9 5.551311 1.055007  0.5625
## s(Set)  4 2.385561      NA      NA

gam.check(SimpE.ASV.gam.re)

```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-3.039985e-08,1.816935e-08]
## (score -41.30925 & scale 0.002273465).
## Hessian positive definite, eigenvalue range [0.3848146,16.90965].
## Model rank = 14 / 14
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k'   edf k-index p-value
## s(Tau) 9.00 5.55   1.06   0.5
## s(Set) 4.00 2.39    NA    NA

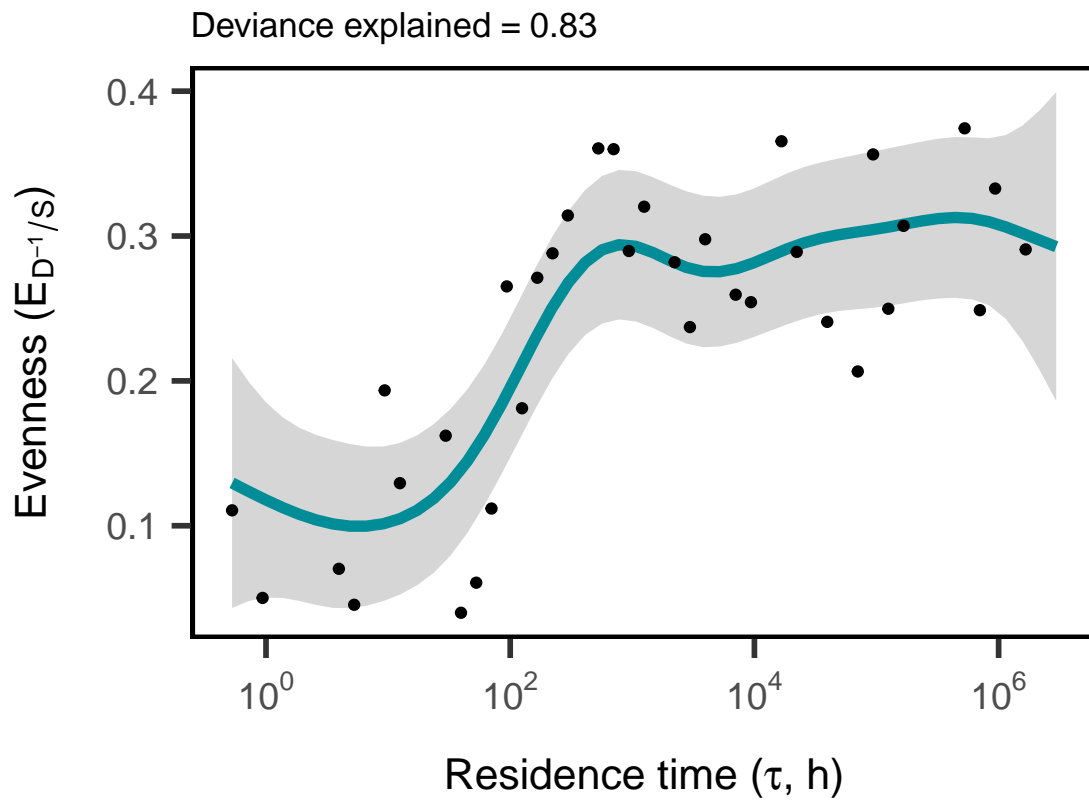
mean(summary(SimpE.ASV.gam.re)$s.table[, 4])

## [1] 0.005267034

SimpE_ASV_Tau <- predict_gam(SimpE.ASV.gam.re, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = Tau, aes(x = Tau,
    y = ASV_Day_20.SimpE)) + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + ylab(expression(paste("Evenness", " (E"[D^-1]/S],
    ")")))) + xlab(expression(paste("Residence time (", tau, ", h)")))) +
  labs(title = bquote("Deviance explained = " ~ .(signif(summary(SimpE.ASV.gam.re)$dev,
```

2))))

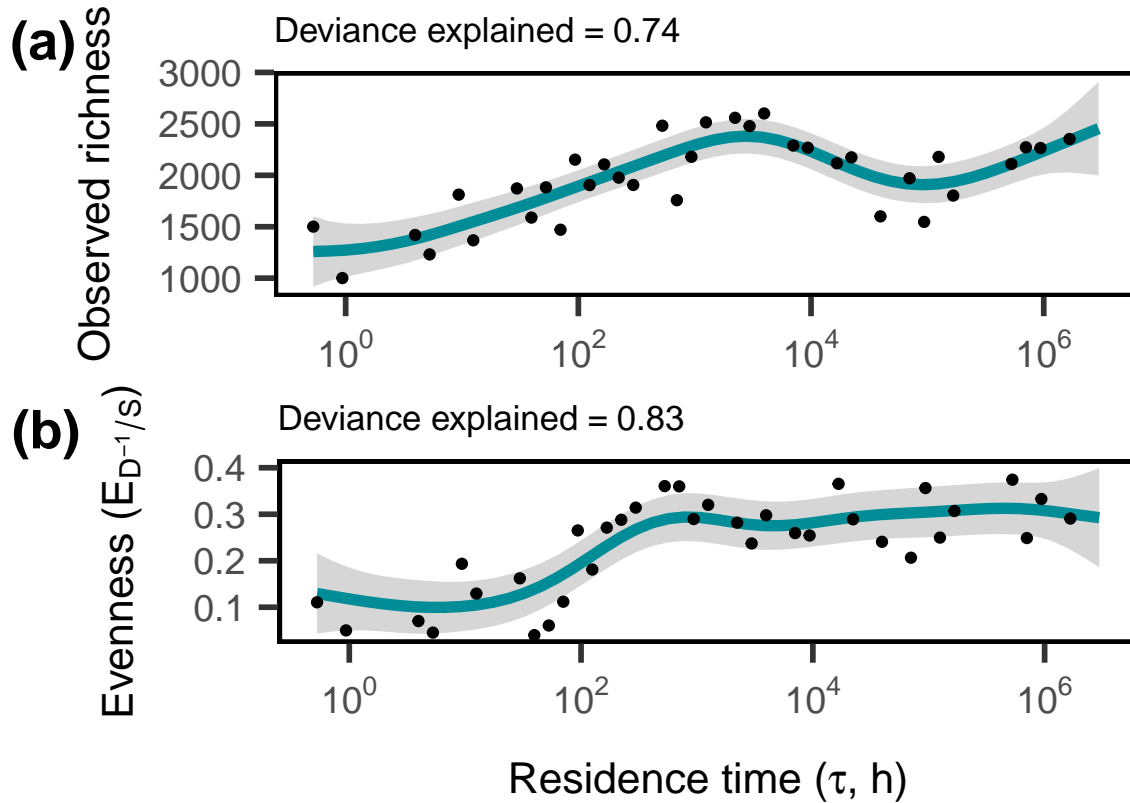
SimpE\_ASV\_Tau



```
ggsave("./output/SimpE_ASV_Tau.pdf")
ggsave("./output/SimpE_ASV_Tau.png", width = 7, height = 5)
```

Draw figure

```
ggdraw() + draw_plot(S.ob_ASV_Tau, x = 0, y = 0.55, width = 1,
  height = 0.45) + draw_plot(SimpE_ASV_Tau, x = 0.04, y = 0,
  width = 0.96, height = 0.55) + draw_plot_label(label = c("(a)",
  "(b)"), size = 20, x = c(-0.02, -0.02), y = c(1, 0.55)) +
  theme(plot.background = element_rect(fill = "white", color = NA))
```



```
ggsave("./output/RTLC_FigS5.pdf")
ggsave("./output/RTLC_FigS5.png", width = 5, height = 6, dpi = 800)
```

Figure S6. Community composition over time.

#### A. Community composition data with Day 0 samples

```
tau.db.d0 <- vegdist(OTUs.r.REL, method = "bray", upper = TRUE,
  diag = TRUE)
adonis2(tau.db.d0 ~ Tau.Seq$Day)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = tau.db.d0 ~ Tau.Seq$Day)
##          Df SumOfSqs      R2      F Pr(>F)
## Tau.Seq$Day 1   2.6197 0.27559 26.25  0.001 ***
## Residual    69   6.8861 0.72441
## Total      70   9.5058 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

tau.pcoa.d0 <- cmdscale(tau.db.d0, eig = TRUE, k = 3)
explainvar1.d0 <- round(tau.pcoa.d0$eig[1]/sum(tau.pcoa.d0$eig),
```

```

    3) * 100
explainvar2.d0 <- round(tau.pcoa.d0$eig[2]/sum(tau.pcoa.d0$eig),
    3) * 100
explainvar3.d0 <- round(tau.pcoa.d0$eig[3]/sum(tau.pcoa.d0$eig),
    3) * 100
sum.eig.d0 <- sum(explainvar3.d0, explainvar2.d0, explainvar1.d0)

tau.plot.d0 <- as.data.frame(tau.pcoa.d0$points)
tau.plot.d0 <- cbind(tau.plot.d0, Tau.Seq)

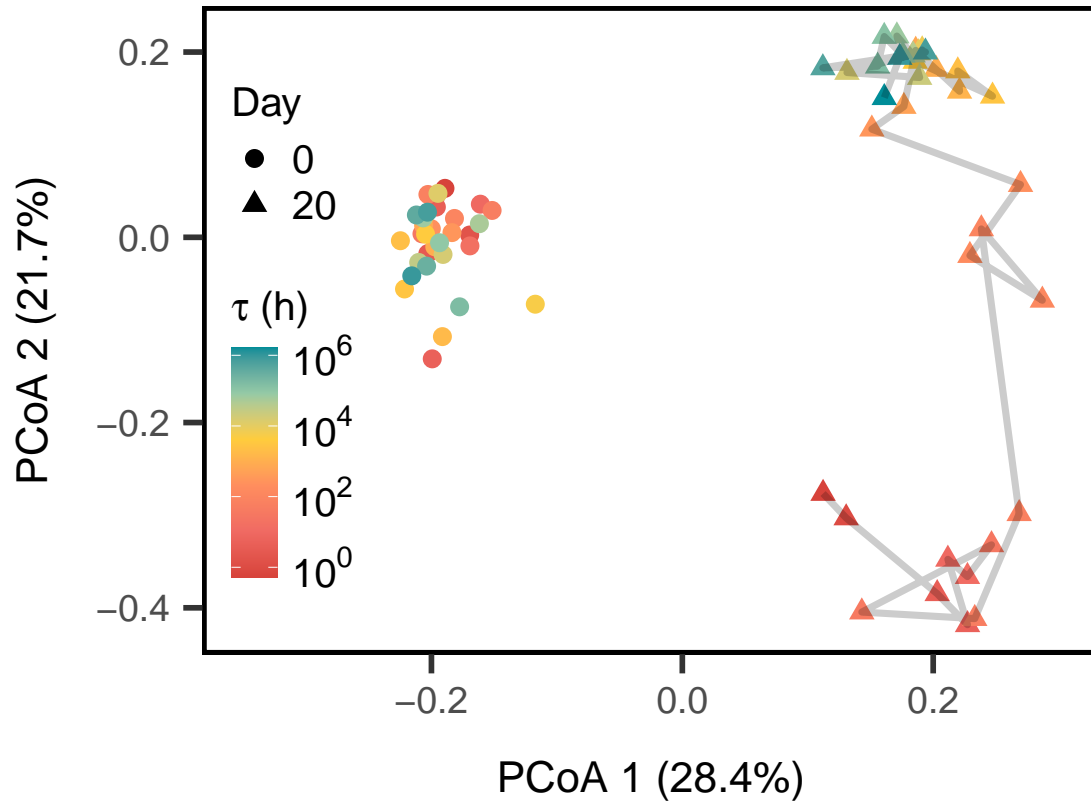
tau.plot.d0.order <- tau.plot.d0[order(as.numeric(tau.plot.d0$Tau)),
]

tau.plot.d0.order <- subset(tau.plot.d0.order, is.na(tau.plot.d0.order$Tau) ==
    FALSE)

CommComp_D0_PCoA <- ggplot(tau.plot.d0.order, aes(x = V1, y = V2)) +
  geom_point(cex = 3, aes(color = as.numeric(Tau), shape = Day)) +
  geom_path(data = subset(tau.plot.d0.order, tau.plot.d0.order$Day ==
    20), aes(x = V1, y = V2), linewidth = 1.25, color = alpha("black",
    0.2)) + xlab(paste("PCoA 1 (", explainvar1.d0, "%)",
    sep = "")) + ylab(paste("PCoA 2 (", explainvar2.d0, "%)",
    sep = "")) + labs(color = expression(paste(tau, " (h)"))) +
  scale_color_gradientn(colors = moma.colors("ustwo"), labels = label_math(expr = 10^.x,
    format = force)) + xlim(c(-0.35, 0.3)) + guides(color = guide_colorbar(position = "inside"),
    shape = guide_legend(position = "inside")) + theme(legend.title = element_text(size = 15),
    legend.position.inside = c(0.1, 0.5), legend.text = element_text(size = 15))

CommComp_D0_PCoA

```



```
ggsave("./output/CommComp_D0_PCoA.pdf")
ggsave("./output/CommComp_D0_PCoA.png", width = 6, height = 6)
```

## B. Paired Day 0 - Day 20 Bray-curtis distances

```
Tau.paired <- Tau

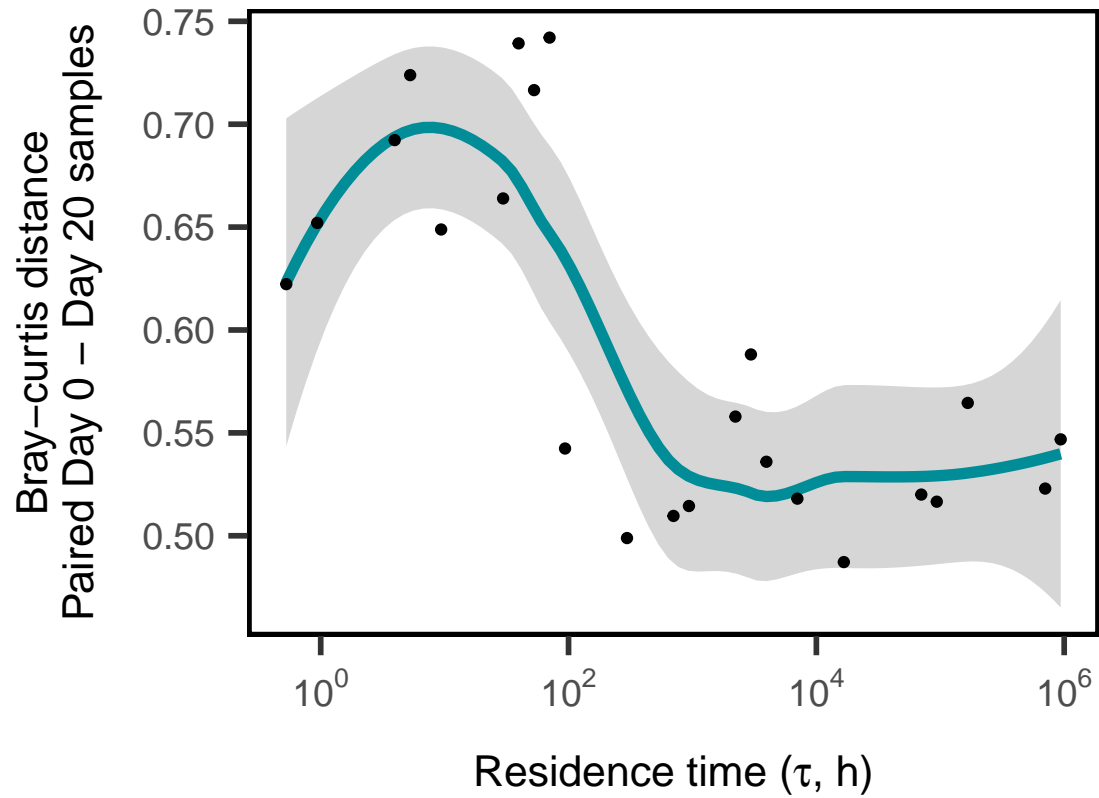
Tau.paired <- subset(Tau.paired, !is.na(Tau.paired$Day_0_Seq) &
  !is.na(Tau.paired$Day_20_Seq))

Tau.paired$BC <- NA

tau <- 1
while (tau <= nrow(Tau.paired)) {
  pair <- subset(OTUs.r.REL, rownames(OTUs.r.REL) %in% c(Tau.paired[tau,
    "Day_0_Seq"], Tau.paired[tau, "Day_20_Seq"]))
  Tau.paired[tau, "BC"] <- as.numeric(vegdist(pair, method = "bray"))
  tau <- tau + 1
}

PairedBC_Tau <- ggplot(data = Tau.paired, aes(x = Tau, y = BC)) +
  geom_smooth(method = "loess", color = "#008d98", linewidth = 2) +
  geom_point() + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + xlab(expression(paste("Residence time (",
    tau, ", h)"))) + ylab("Bray-curtis distance\nPaired Day 0 - Day 20 samples")
```

PairedBC\_Tau

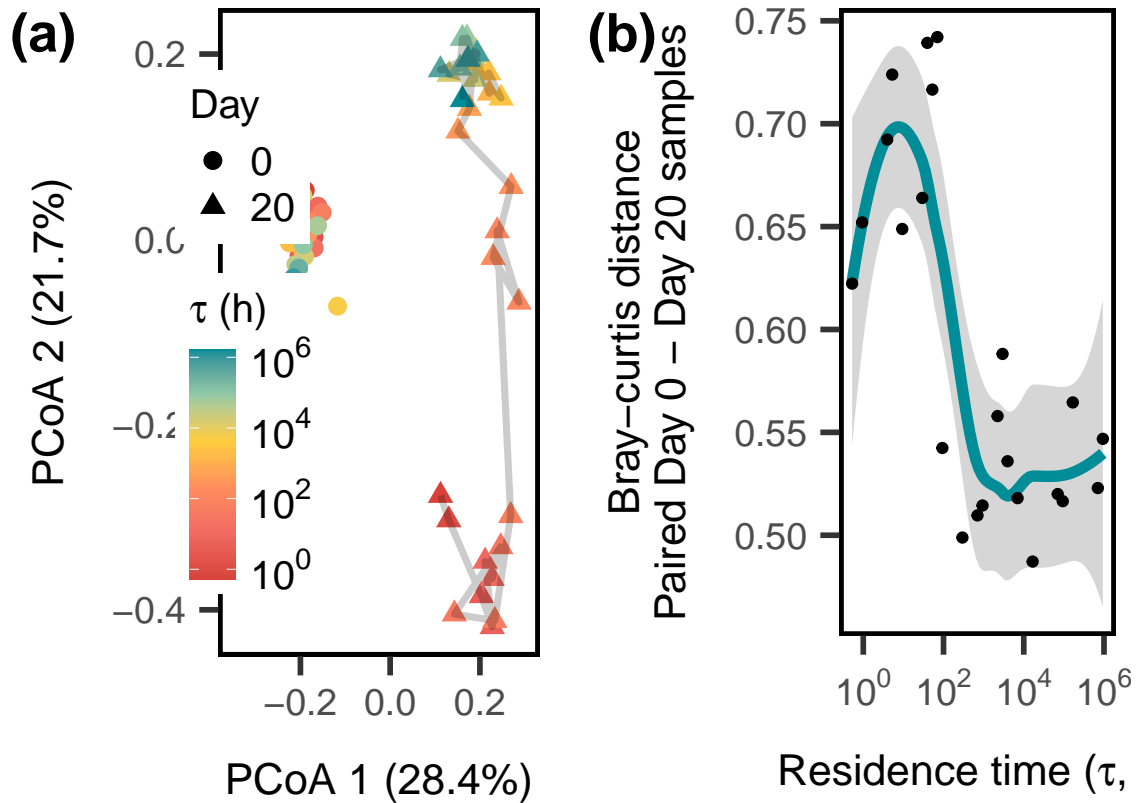


```
ggsave("./output/PairedBC_Tau.pdf")  
ggsave("./output/PairedBC_Tau.png", width = 7, height = 5)
```

Draw figure

```
ggdraw() + draw_plot(CommComp_D0_PCoA, x = 0, y = 0, width = 0.5,  
  height = 1) + draw_plot(PairedBC_Tau, x = 0.5, y = 0, width = 0.5,  
  height = 1) + draw_plot_label(label = c("(a)", "(b)"), size = 20,  
  x = c(0, 0.5), y = c(1, 1)) + theme(plot.background = element_rect(fill = "white",  
  color = NA))
```





```
ggsave("./output/RTLC_FigS6.pdf")
ggsave("./output/RTLC_FigS6.png", width = 12, height = 5, dpi = 800)
```

#Figure S7. Observed niche overlap was lower than expected from null models. ## Null model 1 - Uniform distribution, no fixed zeros

```
# A.niche.overlap <- c() x <- 1 while(x < 100){ for(col in
# 1:ncol(OTUs.70)){ OTUs.70[,col] <- runif(35, min = 0, max
# = 1) } OTUs.70 <- decostand(OTUs.70, method = 'total') A.p
# <- mean(niche.overlap(OTUs.70, method = 'pianka'))
# print(paste(x, ' : ', A.p)) A.niche.overlap <-
# c(A.niche.overlap, A.p) x = x + 1 } all.niche.overlap <-
# as.data.frame(A.niche.overlap)
# write.csv(all.niche.overlap, file =
# './data/uniform_nicheoverlap.csv')
```

#### Null model 2 - Uniform distribution with fixed zeros

```
# A.niche.overlap <- c() x <- 1 while(x < 100){ for(col in
# 1:ncol(OTUs.70)){ n.zeros <- colSums(OTUs.70 == 0)[[col]]
# + 1 OTUs.70 <- OTUs.70[order(OTUs.70[,col]),]
# OTUs.70[,col] <- c(rep(0, n.zeros-1), runif(36-n.zeros,
# min = 0, max = 1)) } OTUs.70 <- decostand(OTUs.70, method
# = 'total') A.p <- mean(niche.overlap(OTUs.70, method =
# 'pianka')) print(paste(x, ' : ', A.p)) A.niche.overlap <-
# c(A.niche.overlap, A.p) x = x + 1 } all.niche.overlap <-
```

```
# as.data.frame(A.niche.overlap)
# write.csv(all.niche.overlap, file =
# './data/uniform_fixedzeros_nicheoverlap.csv')
```

Calculate actual niche overlap, Cohen's D for null models, and draw figure

```
initial.overlap <- mean(niche.overlap(as.data.frame(OTUs.70),
  method = "pianka"))

# Read in previously generated null model data
uniform.niche.overlap <- data.frame(Pianka = read.csv("./data/uniform_nicheoverlap.csv",
  header = TRUE)[, -1])
uniform.fixedzero.niche.overlap <- data.frame(Pianka = read.csv("./data/uniform_fixedzeros_nicheoverlap",
  header = TRUE)[, -1])

cohensD(x = initial.overlap, y = uniform.fixedzero.niche.overlap$Pianka)

## [1] 45.439

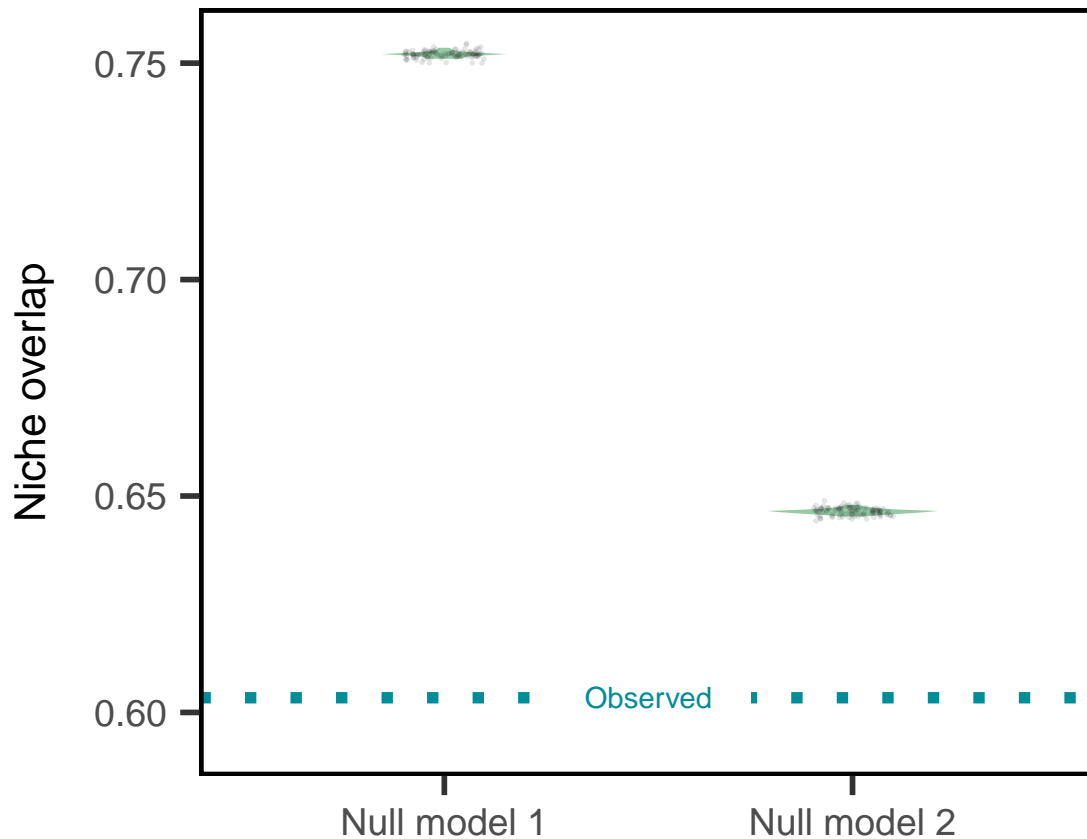
cohensD(x = initial.overlap, y = uniform.niche.overlap$Pianka)

## [1] 175.1867

niche.overlap.uniform.long <- as.data.frame(uniform.niche.overlap$Pianka)
niche.overlap.uniform.long <- cbind(niche.overlap.uniform.long,
  uniform.fixedzero.niche.overlap$Pianka)
colnames(niche.overlap.uniform.long) <- c("uniform", "uniform.fixedzero")
niche.overlap.uniform.long <- gather(niche.overlap.uniform.long,
  key = "Null_model", value = "Pianka")

niche_uniform_nullmodels <- ggplot() + geom_hline(yintercept = initial.overlap,
  linetype = "dotted", color = "#008d98", cex = 2) + annotate("rect",
  xmin = 1.25, xmax = 1.75, ymin = initial.overlap - 0.01,
  ymax = initial.overlap + 0.01, fill = "white") + geom_violin(data = niche.overlap.uniform.long,
  aes(x = Null_model, y = Pianka), fill = "#95caa6", color = "white",
  cex = 1.5) + geom_jitter(data = niche.overlap.uniform.long,
  width = 0.1, color = alpha("black", 0.1), aes(x = Null_model,
  y = Pianka), cex = 0.3) + ylab("Niche overlap") + scale_x_discrete(labels = c("Null model 1",
  "Null model 2")) + theme(legend.position = "none", axis.title.x = element_blank()) +
  annotate("text", x = 1.5, y = initial.overlap, label = "Observed",
  color = "#008d98")

niche_uniform_nullmodels
```



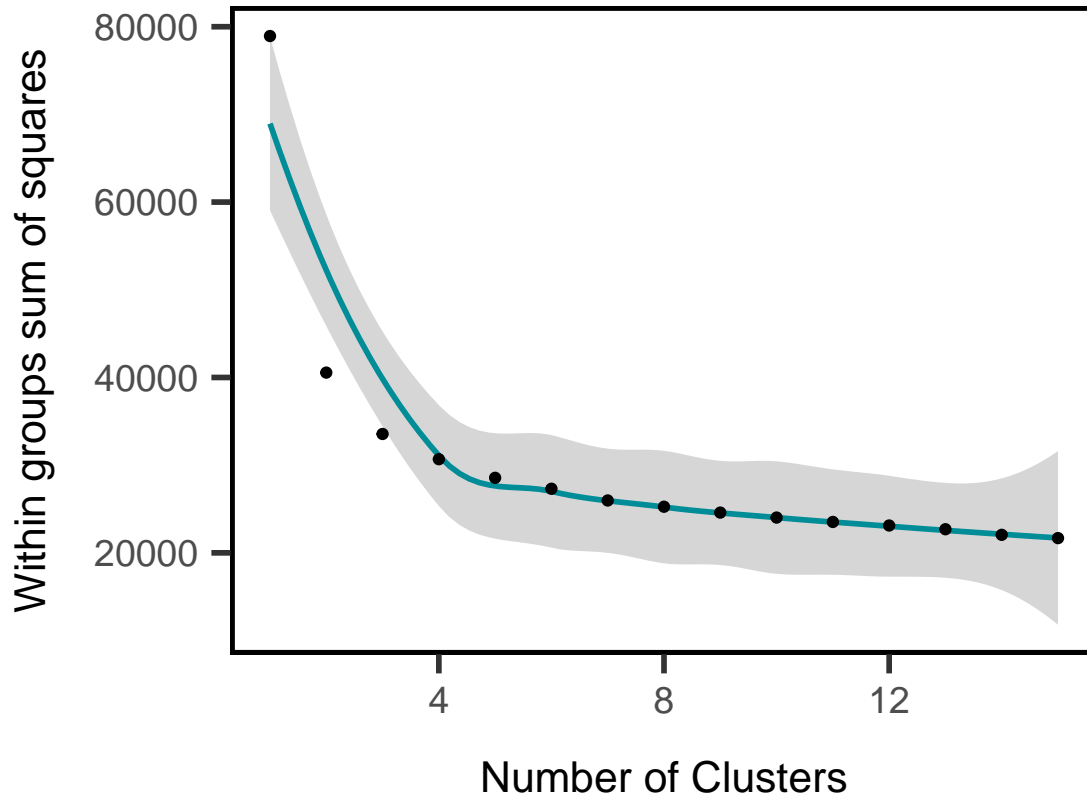
```
ggsave("./output/RTLC_FigS7.pdf")
ggsave("./output/RTLC_FigS7.png", width = 5, height = 5, dpi = 800)
```

**Figure S8.** Five niches were identified across the residence time gradient using k-means clustering.

#### A. Within-group sum of squares for number of clusters

```
wss <- (nrow(OTUs.70.dist) - 1) * sum(apply(OTUs.70.dist, 2,
  var))
for (i in 2:15) wss[i] <- sum(kmeans(OTUs.70.dist, centers = i)$withinss)
wss.df <- as.data.frame(1:15)
wss.df$wss <- wss
colnames(wss.df) <- c("clusters", "wss")
Nclust_Niches <- ggplot(data = wss.df, aes(x = clusters, y = wss)) +
  geom_smooth(method = "loess", color = "#008d98") + geom_point() +
  xlab("Number of Clusters") + ylab("Within groups sum of squares")

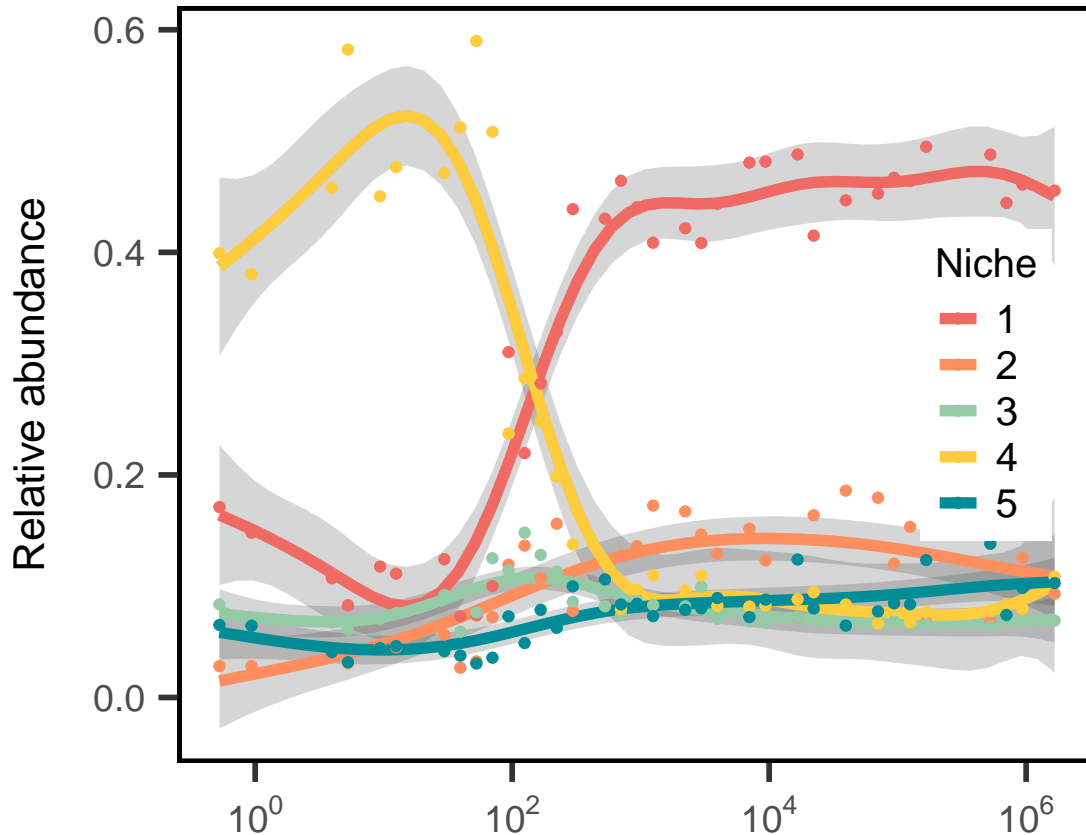
Nclust_Niches
```



```
ggsave("./output/Nclust_Niches.pdf")
ggsave("./output/Nclust_niches.png", width = 6, height = 4)
```

## B. Total relative abundance of all niches

```
OTUs_all_niches <- ggplot(data = niche.gams, aes(x = Tau, y = fit,
  group = Cluster)) + geom_smooth_ci(cex = 2, ci_alpha = 0.2,
  color = "grey") + geom_line(linewidth = 2, aes(color = Cluster)) +
  geom_point(data = OTUs.niches.sum, aes(x = Tau, y = REL,
    color = Cluster)) + scale_x_continuous(labels = label_math(expr = 10^.x,
  format = force)) + ylab("Relative abundance") + scale_color_manual(values = as.vector(c(moma.colors
  5)[1], moma.colors("ustwo", 5)[2], moma.colors("ustwo", 5)[4],
  moma.colors("ustwo", 5)[3], moma.colors("ustwo", 5)[5]))) +
  xlab(expression(paste("Residence time (", tau, ", h)"))) +
  labs(color = "Niche") + theme(legend.position = c(0.88, 0.5),
  axis.title.x = element_blank(), legend.text = element_text(size = 15),
  legend.title = element_text(size = 15))
OTUs_all_niches
```



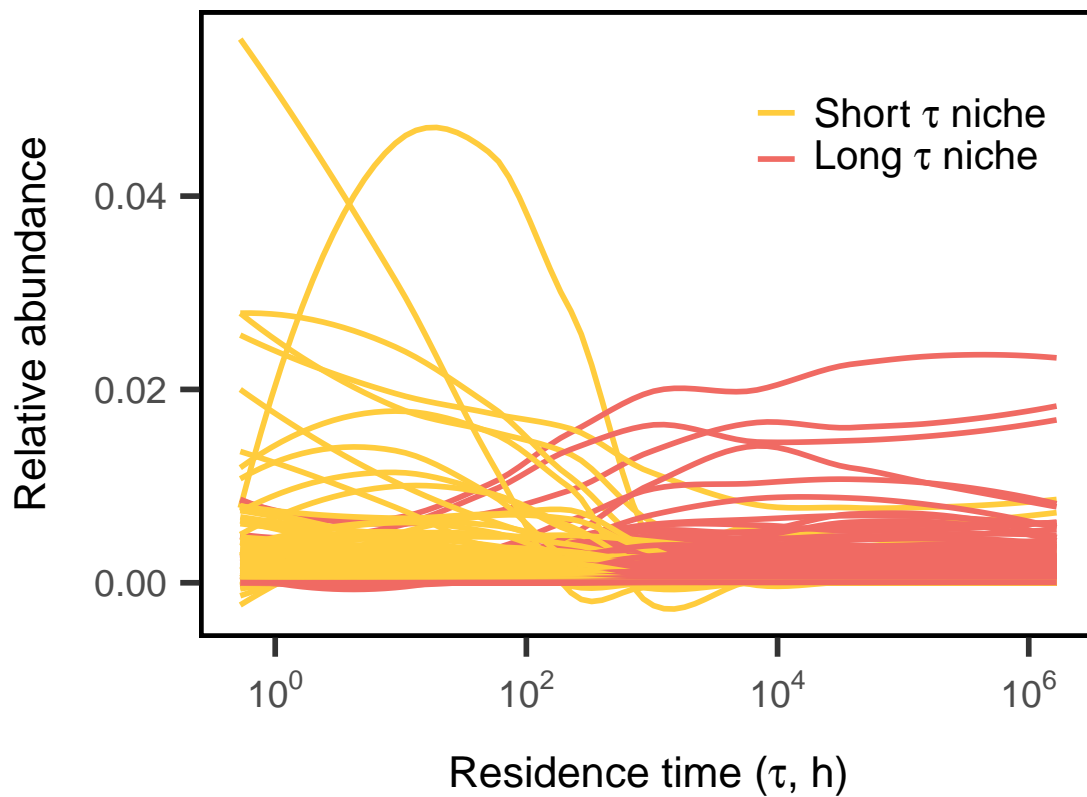
```
ggsave("./output/OTUs_all_niches.pdf")
ggsave("./output/OTUs_all_niches.png", width = 6, height = 4)
```

### C. Individual relative abundance of OTUs in Niche 1 and 4

```
OTUs.niches$Cluster <- as.factor(OTUs.niches$Cluster)
OTUs.niches$Cluster <- ordered(OTUs.niches$Cluster, levels = c("4",
  "1", "2", "3", "5"))

OTUs_ind_major_niches <- ggplot(data = subset(OTUs.niches, OTUs.niches$Cluster %in%
  c(1, 4)), aes(x = Tau, y = REL, group = OTU, color = Cluster)) +
  geom_smooth(se = FALSE) + scale_color_discrete(label = c(expression(paste("Short ",
    tau, " niche")), expression(paste("Long ", tau, " niche"))),
  type = c(moma.colors("ustwo", 5)[3], moma.colors("ustwo",
    5)[1])) + xlab(expression(paste("Residence time (", tau,
    ", h)"))) + ylab("Relative abundance") + scale_x_continuous(labels = label_math(expr = 10^.x,
    format = force)) + theme(legend.title = element_blank(),
    legend.position = c(0.78, 0.8), legend.text = element_text(size = 15))

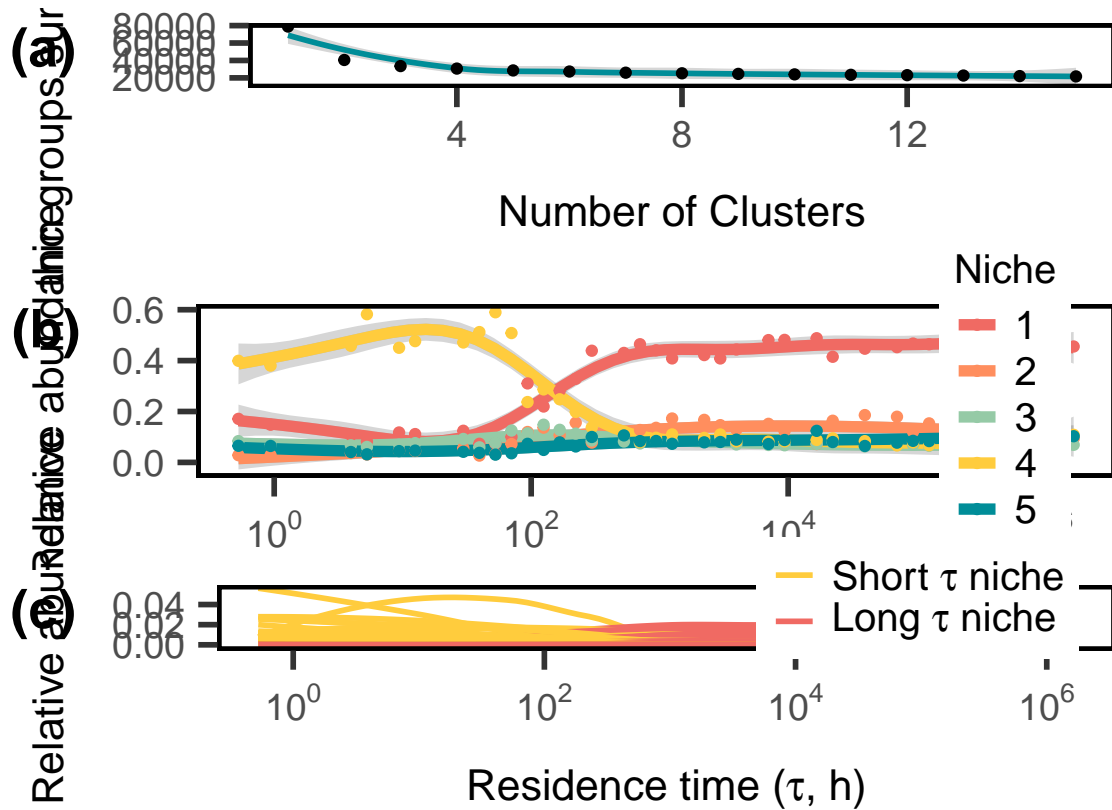
OTUs_ind_major_niches
```



```
ggsave("./output/OTUs_ind_major_niches.pdf")
ggsave("./output/OTUs_ind_major_niches.png", width = 6, height = 4)
```

## Draw Figure

```
ggdraw() + draw_plot(Nclust_Niches, x = 0, y = 0.675, width = 1,
  height = 0.325) + draw_plot(OTUs_all_niches, x = 0, y = 0.35,
  width = 1, height = 0.325) + draw_plot(OTUs_ind_major_niches,
  x = 0, y = 0, width = 1, height = 0.35) + draw_plot_label(label = c("(a)",
  "(b)", "(c)"), size = 20, x = c(0, 0, 0), y = c(1, 0.675,
  0.35)) + theme(plot.background = element_rect(fill = "white",
  color = NA))
```



```
ggsave("./output/RTLC_FigS8.pdf")
ggsave("./output/RTLC_FigS8.png", width = 10, height = 15, dpi = 800)
```

Figure S9. Biofilm production across the residence time gradient.

#### A. Total biofilm production

```
TotOT.gam.re <- gam(OT ~ s(Tau) + s(Set, bs = "re"), data = Tau,
  family = gaussian(link = "identity"), method = "REML")
TotOT.gam <- gam(OT ~ s(Tau), data = Tau, family = gaussian(link = "identity"),
  method = "REML")
```

```
AIC(TotOT.gam, TotOT.gam.re)
```

```
##              df      AIC
## TotOT.gam    3.000080 -114.2994
## TotOT.gam.re 9.102045 -161.7488
```

```
anova(TotOT.gam, TotOT.gam.re, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: OT ~ s(Tau)
## Model 2: OT ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1      47.000    0.246314
```

```
## 2      39.968    0.072907 7.0317  0.17341 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# Random effect of set is significant

rm(TotOT.gam)

summary(TotOT.gam.re)

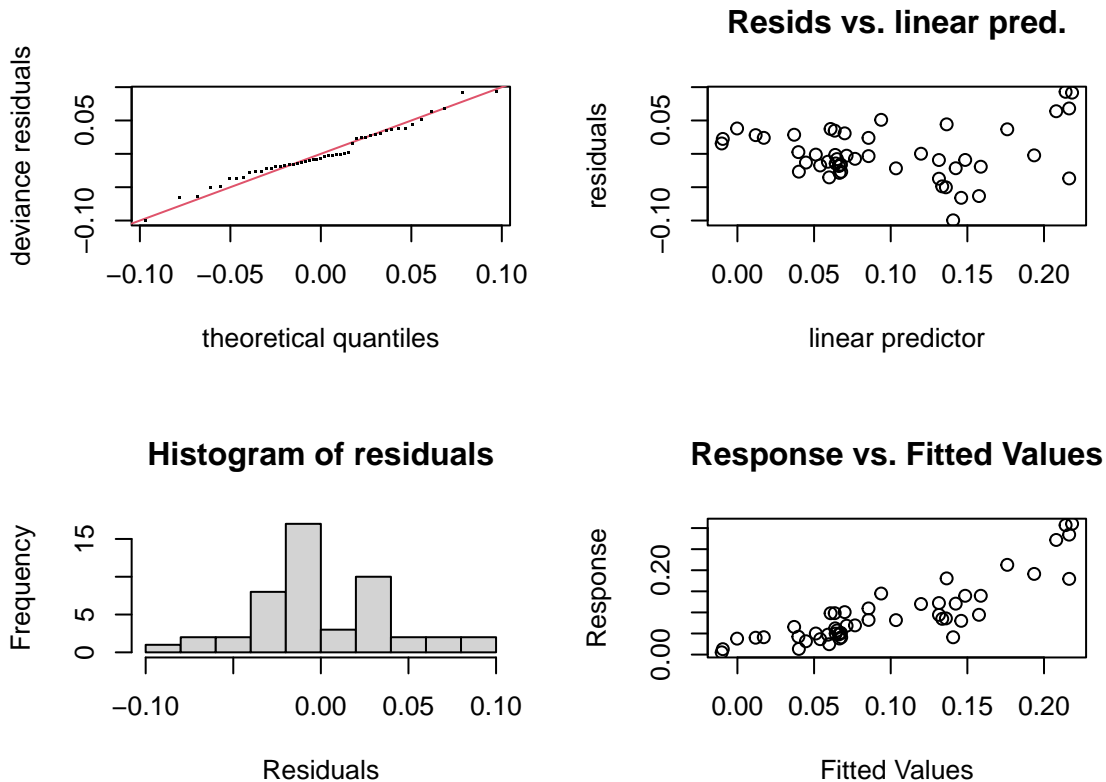
##
## Family: gaussian
## Link function: identity
##
## Formula:
## OT ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.09732    0.03270   2.976  0.00484 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(Tau) 3.287  4.109  6.142 0.000519 ***
## s(Set) 2.885  3.000 30.374 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.698   Deviance explained = 73.7%
## -REML = -71.365   Scale est. = 0.001743   n = 49

k.check(TotOT.gam.re)

##           k'           edf k-index p-value
## s(Tau)  9 3.287071 1.24012  0.935
## s(Set)  4 2.885161      NA      NA

gam.check(TotOT.gam.re)
```





```
##
## Method: REML   Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-7.128326e-07,6.345747e-07]
## (score -71.36517 & scale 0.00174304).
## Hessian positive definite, eigenvalue range [0.6636435,23.65044].
## Model rank = 14 / 14
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k'   edf k-index p-value
## s(Tau) 9.00 3.29   1.24   0.96
## s(Set) 4.00 2.89    NA    NA

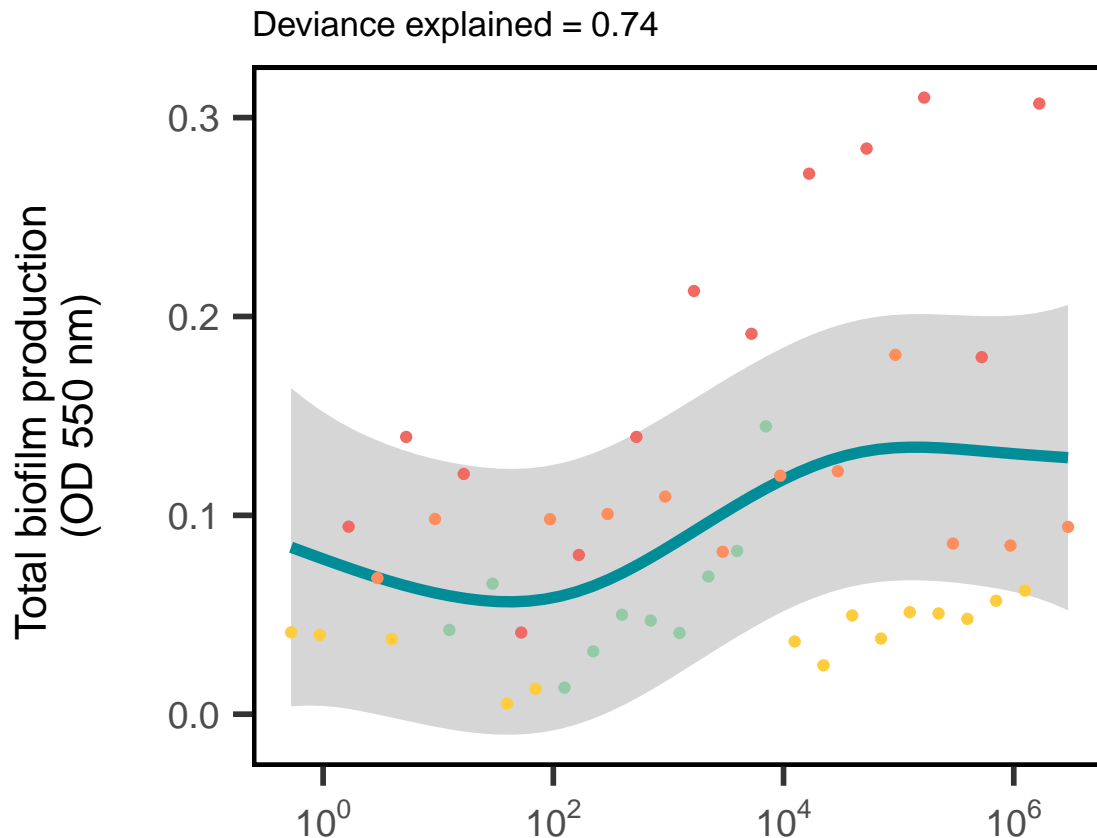
mean(summary(TotOT.gam.re)$s.table[, 4])

## [1] 0.0002595414

Total_OT_Tau <- predict_gam(TotOT.gam.re, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = Tau, aes(x = Tau,
    y = OT, color = Set)) + xlab(expression(paste("Residence time (",
    tau, ", h)"))) + ylab("Total biofilm production \n(OD 550 nm)" +
    scale_color_manual(values = as.vector(c(moma.colors("ustwo",
    5)[1], moma.colors("ustwo", 5)[2], moma.colors("ustwo",
```

```
5)[4], moma.colors("ustwo", 5)[3])) + scale_x_continuous(labels = label_math(expr = 10^.x,
format = force)) + labs(title = bquote("Deviance explained = " ~
.(signif(summary(TotOT.gam.re)$dev, 2)))) + theme(axis.title.x = element_blank(),
legend.position = "none", plot.margin = unit(c(0.2, 0.2,
0.2, 0.6), "cm"), axis.title.y = element_text(margin = margin(r = 25)))
```

Total\_OT\_Tau



```
ggsave("./output/Total_OT_Tau.pdf")
ggsave("./output/Total_OT_Tau.png")
```

## B. Per cell biofilm production

```
PC.OT.gam.re <- gam(log(OT/(N * 0.005), 10) ~ s(Tau) + s(Set,
bs = "re"), data = Tau, family = gaussian(link = "identity"),
method = "REML")
PC.OT.gam <- gam(log(OT/(N * 0.005), 10) ~ s(Tau), data = Tau,
family = gaussian(link = "identity"), method = "REML")
AIC(PC.OT.gam, PC.OT.gam.re)
```

```
##           df      AIC
## PC.OT.gam  4.099905 53.24869
## PC.OT.gam.re 7.431379 11.24138
```

```

anova(PC.OT.gam, PC.OT.gam.re, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: log(OT/(N * 0.005), 10) ~ s(Tau)
## Model 2: log(OT/(N * 0.005), 10) ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev      Df Deviance  Pr(>Chi)
## 1      32.483      7.3669
## 2      28.808      1.9061 3.6754    5.4608 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Random effect of set is significant

rm(PC.OT.gam)

summary(PC.OT.gam.re)

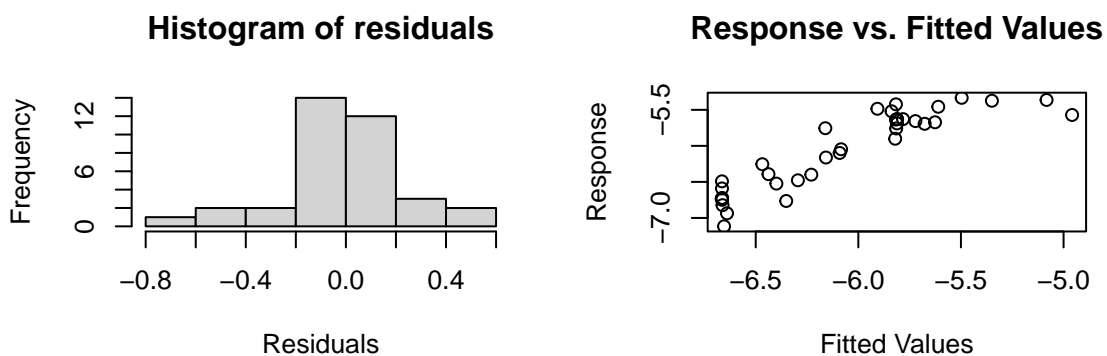
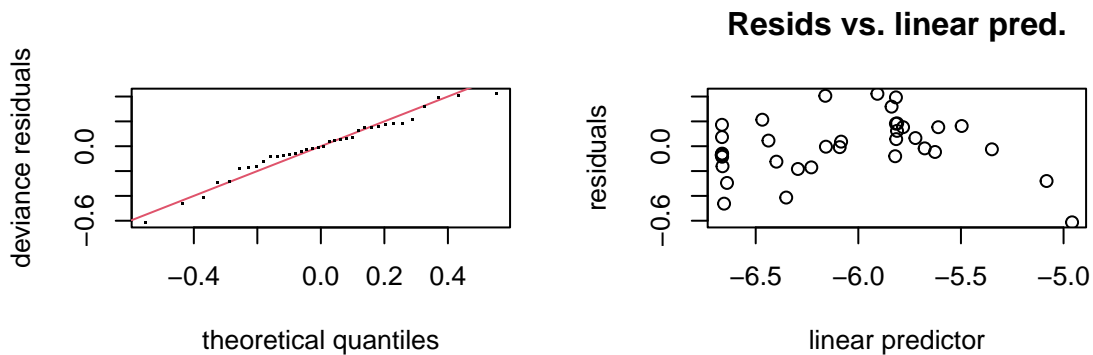
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(OT/(N * 0.005), 10) ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -6.0626      0.2687  -22.56  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(Tau)  2.732   3.437  17.66 9.69e-07 ***
## s(Set)  1.939   2.000  40.26 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.788   Deviance explained = 81.6%
## -REML = 9.9783   Scale est. = 0.062846   n = 36

k.check(PC.OT.gam.re)

##           k'      edf k-index p-value
## s(Tau)  9 2.731798 1.100471    0.66
## s(Set)  3 1.938628      NA      NA

gam.check(PC.OT.gam.re)

```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [-1.494689e-06,1.392979e-06]
## (score 9.97834 & scale 0.06284554).
## Hessian positive definite, eigenvalue range [0.4618004,17.10335].
## Model rank = 13 / 13
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k'   edf k-index p-value
## s(Tau) 9.00 2.73     1.1   0.68
## s(Set) 3.00 1.94     NA     NA

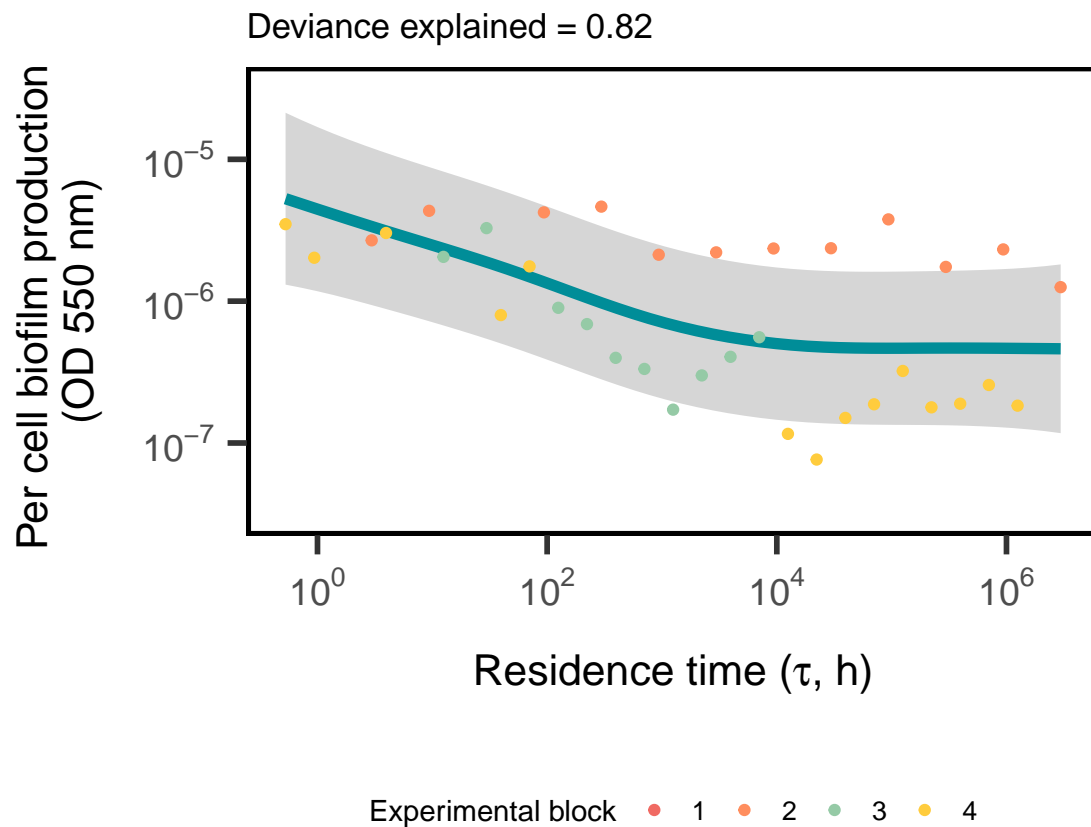
mean(summary(PC.OT.gam.re)$s.table[, 4])

## [1] 4.845389e-07

PC_OT_Tau <- predict_gam(PC.OT.gam.re, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit)) + geom_smooth_ci(color = "#008d98",
    cex = 2, ci_alpha = 0.2) + geom_point(data = Tau, aes(x = Tau,
    y = log(OT/(N * 0.005), 10), color = Set)) + xlab(expression(paste("Residence time (",
    tau, ", h)"))) + ylab("Per cell biofilm production \n(OD 550 nm)") +
  scale_color_manual(values = as.vector(c(moma.colors("ustwo",
    5)[1], moma.colors("ustwo", 5)[2], moma.colors("ustwo",
```

```
5)[4], moma.colors("ustwo", 5)[3])), ) + scale_x_continuous(labels = label_math(expr = 10^.x,
format = force)) + scale_y_continuous(limits = c(-7.5, -4.5),
breaks = c(-7, -6, -5), labels = label_math(expr = 10^.x,
format = force)) + labs(title = bquote("Deviance explained =" ~
.(signif(summary(PC.OT.gam.re)$dev, 2)))) + theme(legend.position = "bottom") +
guides(color = guide_legend(title = "Experimental block"))
```

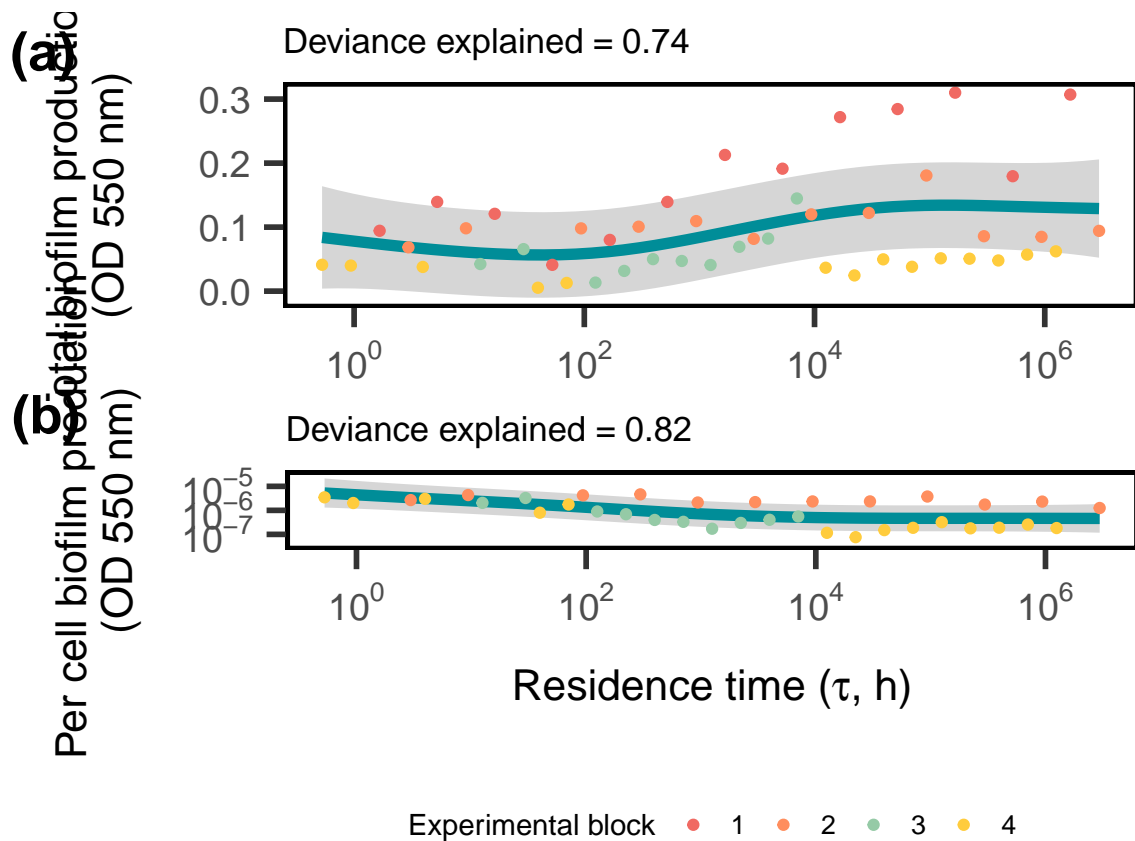
PC\_OT\_Tau



```
ggsave("../output/PC_OT_Tau.pdf")
ggsave("../output/PC_OT_Tau.png")
```

## Draw Figure

```
ggdraw() + draw_plot(Total_OT_Tau, x = 0, y = 0.55, width = 1,
height = 0.45) + draw_plot(PC_OT_Tau, x = 0, y = 0, width = 1,
height = 0.55) + draw_plot_label(label = c("(a)", "(b)"),
size = 20, x = c(-0.02, -0.02), y = c(1, 0.58)) + theme(plot.background = element_rect(fill = "white",
color = NA))
```



```
ggsave("./output/RTLC_FigS9.pdf")
ggsave("./output/RTLC_FigS9.png", width = 5, height = 8, dpi = 800)
```