Residence Time Experiment

Emmi Mueller

July 23, 2019

```
require("png")
```

```
## Loading required package: png
```

```
require("vegan")
```

```
## Loading required package: vegan
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.5-7
```

```
require("ggplot2")
```

```
## Loading required package: ggplot2
```

```
require("ggpubr")
```

```
## Loading required package: ggpubr
```

```
require("cowplot")
```

```
## Loading required package: cowplot
##
## Attaching package: 'cowplot'
## The following object is masked from 'package:ggpubr':
##
##     get_legend
```

```
require("ggpmisc")
```

```
## Loading required package: ggpmisc
## Loading required package: ggpp
##
## Attaching package: 'ggpp'
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
require("scales")
```

```
## Loading required package: scales
```

```r
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_line(size = 1),
        #axis.line.y = element_line(size = 1),
        axis.line.x = element_blank(),
        axis.line.y = element_blank(),
        axis.ticks.x = element_line(size = 1),
        axis.ticks.y = element_line(size = 1),
        axis.ticks.length = unit(.1, "in"),
        panel.border = element_rect(color = "black", fill = NA, size = 1.5),
        legend.title = element_blank(),
        legend.text = element_text(size = 14),
        strip.text = element_text(size = 14),
        strip.background = element_blank()
        ))
```

```r
BP_fxn <- function(CPMs, Tau){
  ##extract whole experiment info from top of csv

  #date experiment was run
  date <- as.Date(as.character(CPMs[1,2]), "%m/%d/%Y")

  #date the standard was produced
  date_std <- as.Date(as.character(CPMs[2,2]), "%m/%d/%Y")

  #DPM of the standard at date of production
  DPM_std <- as.double(as.character(CPMs[3,2]))

  #DPM of the standard based on scintillation counter on experiment date
  DPM_curr <- as.double(as.character(CPMs[4,2]))

  #half life of tritium - 12.346 years
  half_life <- as.double(as.character(CPMs[5,2]))

  #Mols of leucine in each tube based on hot leucine stock concentration
  M_Leu <- as.double(as.character(CPMs[6,2]))

  #CPMs of the voucher on experiment date
  Voucher <- as.double(as.character(CPMs[7,2]))

  ##remove whole experiment info from top of dataframe
  CPMs <- CPMs[-c(1:9),]
  colnames(CPMs) <- c("Sample", "CPM", "Kill")

  ##calculate time from the experiment date to the standard production date
  t <- as.numeric(date - date_std)/365
```

```
  ##calculate the expected DPMs of the standard based on t
  DPM_exp <- (DPM_std)*exp((-0.693/half_life)*t)

  ##calculate scintillation efficiency as DPM ratio
  efficiency <- DPM_curr/DPM_exp

  #divide CPMs into kill and sample dataframes
  Kills <- subset(CPMs, Kill == "T")
  CPMs <- subset(CPMs, Kill == "F")

  #convert CPMs to DPMs, DPMs = CPMs/efficiency
  CPMs$CPM <- as.numeric(as.character(CPMs$CPM))
  CPMs$DPM <- CPMs$CPM / efficiency

  Kills$CPM <- as.numeric(as.character(Kills$CPM))
  Kills$DPM <- Kills$CPM / efficiency

  #average DPMs for each sample and add to Tau
  for(x in unique(CPMs$Sample)){
    Tau[Tau$Tau == x, "DPM"] <- as.numeric(mean(CPMs[CPMs$Sample == x, "DPM"]))
  }

  #for each sample, subtract the corresponding kill DPM
  for (x in unique(Tau$Tau)){
    Tau[Tau$Tau == x, "DPMKills"] <- Tau[Tau$Tau ==x, "DPM"] - (as.numeric(Kills[Kills$Sample == x, "CPI
  }

  #Determine Mols Leucine based on MLeu_sample = MLeu * DPM/voucher
  Tau$MLeu <- (M_Leu * Tau$DPMKills)/Voucher

  #Convert MLeu to ug C/L/hr
  Tau$ugCLhr <- Tau$MLeu * 131.2 * (1/0.073)*0.86*2*1000000


  Tau$uMChr <- Tau$ugCLhr *0.083333

  Tau$log_uMChr <- log(Tau$uMChr, 10)

  return(Tau)
}
```

BP[[i]]$Leucine < -((BP[[i]]$DPM / 2.2e12) / 153) / 1000 # DPM$1Ci/2.2e12DPM$ 1mmolLeu/153Ci * 1molLeu/1000mmol BP[[i]]$Leucine.per < -(BP[[i]]$Leucine * (1/1) * (1/0.0015)) # Leu incorporated * 1/time (hrs) * 1/vol (L) BP[[i]]$Protein < -(BP[[i]]$Leucine.per / 0.073) * 131.2 # mol leu * 1 mol protein/0.073 mol leu * 131.2 g protein/1mol protein BP[[i]]$Carbon < -BP[[i]]$Protein * (1/0.63) * (0.54/1) * (10^6) # g protein * 1 g DW/0.63 g Pro * 0.54 g C/1g DW = gC/l/hr * 10^6 = ugC/L/Hr

```
Abundance_fxn <- function(N, Tau){
  for(x in unique(N$Sample)){
    Tau$N[Tau$Sample == x] <- mean(N$N[N$Sample == x])
  }
  Tau$log_N <- log(Tau$N, 10)

  return(Tau)
```

```
}
```

```
OT_fxn <- function(OT, Tau){
  for(x in unique(OT$Tau)){
    Tau$OT[Tau$Tau == x] <- mean(OT$OT[OT$Tau == x])-mean(OT$OT[OT$Tau == 0])
  }

  return(Tau)
}
```

Read in data

```
#Tau read in
Tau <- read.csv("data/RTLC/2021_RTLC_Tau.csv", header = TRUE)

#Abundance data
#N <- read.csv("data/RTLC/RTLC_S1/20210602_RTLC_S1_N.csv", header = TRUE)
#Tau <- Abundance_fxn(N, Tau)


#Biofilm data
OT <- read.csv("data/RTLC/RTLC_S1/20210602_RTLC_S1_Otoole.csv", header = TRUE)
#OT <- cbind(OT, read.csv("data/RTLC/RTLC_S2/20210628_RTLC_S2_Otoole.csv"))
Tau <- OT_fxn(OT, Tau)


#BP data
CPM_S1 <- read.csv("data/RTLC/RTLC_S1/20210602_RTLC_S1_BP.csv", header = FALSE)
Tau <- as.data.frame(BP_fxn(CPM_S1, Tau))
#Tau$ind_P <- Tau$ugCLhr/Tau$N


#Biolog data
```

```
Tau$Tau <- as.numeric(Tau$Tau)
OT_lm <- lm(OT ~ Tau, data = Tau)
OT_lm_2 <- lm(OT ~ poly(Tau, 2, raw = TRUE), data = Tau)
summary(OT_lm)
```

```
##
## Call:
## lm(formula = OT ~ Tau, data = Tau)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09236 -0.02402  0.01558  0.03941  0.05550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.011486   0.036965   0.311   0.7613
## Tau         0.034725   0.007164   4.847   0.0004 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05403 on 12 degrees of freedom
```

```
## Multiple R-squared:  0.6619, Adjusted R-squared:  0.6338
## F-statistic:  23.5 on 1 and 12 DF,  p-value: 0.0004001
```

```
summary(OT_lm_2)
```

```
##
## Call:
## lm(formula = OT ~ poly(Tau, 2, raw = TRUE), data = Tau)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.101849 -0.013515  0.005875  0.038519  0.058757
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.061650   0.084810   0.727    0.482
## poly(Tau, 2, raw = TRUE)1 0.008965   0.039655   0.226    0.825
## poly(Tau, 2, raw = TRUE)2 0.002712   0.004102   0.661    0.522
##
## Residual standard error: 0.05534 on 11 degrees of freedom
## Multiple R-squared:  0.6748, Adjusted R-squared:  0.6157
## F-statistic: 11.42 on 2 and 11 DF,  p-value: 0.002072
```

```
AIC(OT_lm)
```

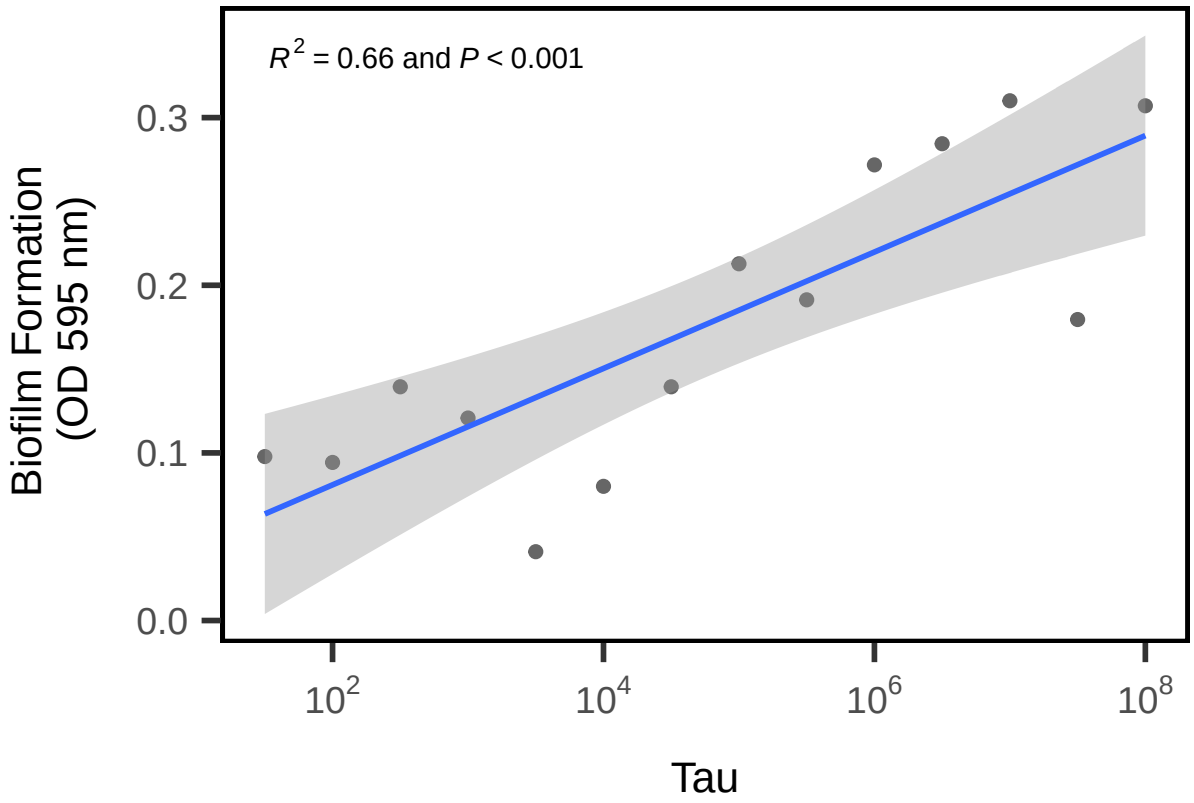
```
## [1] -38.13986
```

```
AIC(OT_lm_2)
```

```
## [1] -36.68521
```

```
OT <- ggplot(Tau, aes(x = Tau, y = OT))+
  geom_point(size = 2, alpha = 0.6)+
  xlab("Tau")+
  ylab("Biofilm Formation \n (OD 595 nm)")+
  geom_smooth(method = "lm", formula = y~x)+
  stat_poly_eq(aes(label =  paste(stat(rr.label), "*\" and \"*", stat(p.value.label), sep = "")),
               formula = y~x, parse = TRUE, size = 4)+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))

OT
```




```
ggsave("./output/RTLC_OT.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_OT.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
BP_lm <- lm(uMChr ~ Tau, data = Tau)
BP_lm_2 <- lm(uMChr ~ poly(Tau, 2, raw = TRUE), data = Tau)
summary(BP_lm)
```

```
##
## Call:
## lm(formula = uMChr ~ Tau, data = Tau)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54492 -0.41006 -0.08064  0.39481  0.72400
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.42677    0.32194   7.538 6.88e-06 ***
## Tau         -0.33003    0.06239  -5.290 0.000192 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4705 on 12 degrees of freedom
```

```
## Multiple R-squared:  0.6998, Adjusted R-squared:  0.6748
## F-statistic: 27.98 on 1 and 12 DF,  p-value: 0.0001916
```

```
summary(BP_lm_2)
```

```
##
## Call:
## lm(formula = uMChr ~ poly(Tau, 2, raw = TRUE), data = Tau)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77834 -0.22265  0.00378  0.12173  0.66541
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   3.38868    0.67986   4.984 0.000413 ***
## poly(Tau, 2, raw = TRUE)1 -0.82398    0.31788  -2.592 0.025044 *
## poly(Tau, 2, raw = TRUE)2  0.05200    0.03288   1.581 0.142137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4436 on 11 degrees of freedom
## Multiple R-squared:  0.7554, Adjusted R-squared:  0.711
## F-statistic: 16.99 on 2 and 11 DF,  p-value: 0.0004327
```

```
AIC(BP_lm)
```

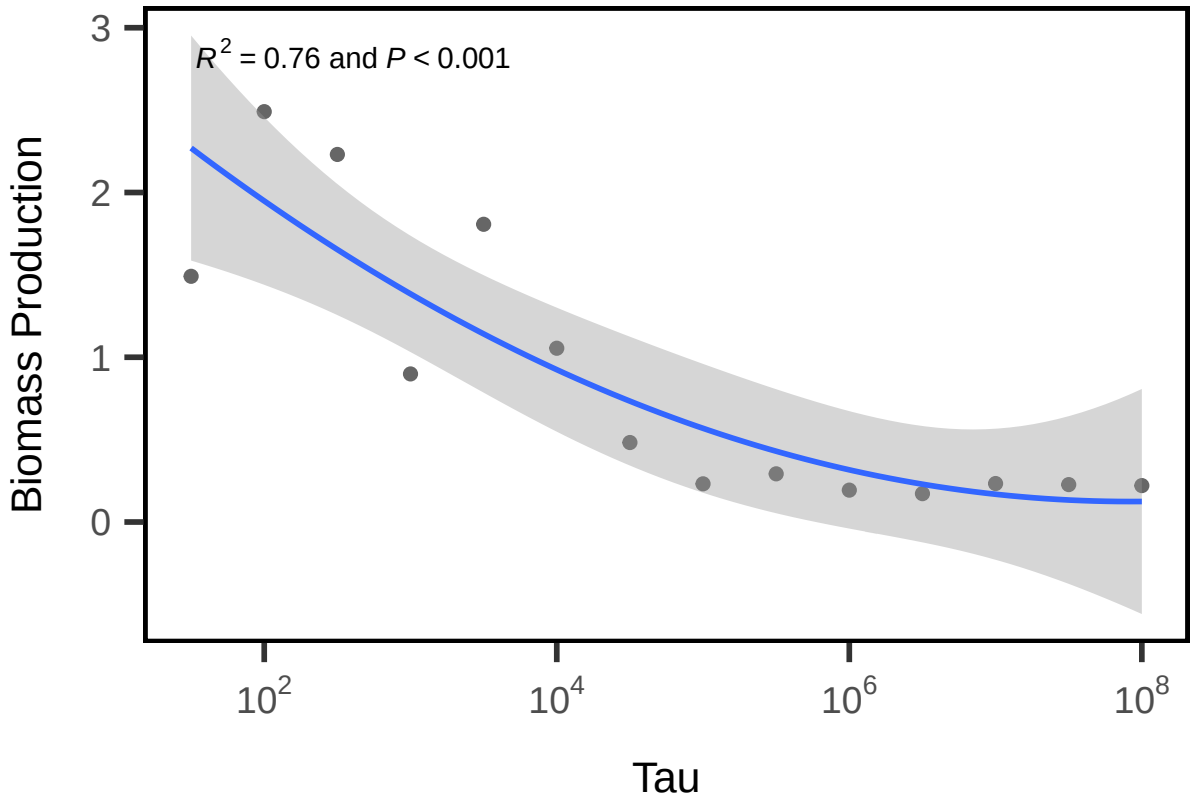
```
## [1] 22.46332
```

```
AIC(BP_lm_2)
```

```
## [1] 21.59598
```

```
BP <- ggplot(Tau, aes(x = Tau, y = uMChr))+
  geom_point(size = 2, alpha = 0.6)+
  xlab("Tau")+
  ylab("Biomass Production")+
  geom_smooth(method = "lm", formula = y~poly(x,2, raw = TRUE))+
  stat_poly_eq(aes(label =  paste(stat(rr.label), "*\" and \"*", stat(p.value.label), sep = "")),
               formula = y~poly(x,2,raw = TRUE), parse = TRUE, size = 4)+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))

BP
```




```
ggsave("./output/RTLC_BP.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_BP.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
# t_Res_rt75 <- ggplot(EcoPlate_t78,aes(x=Time,y=NumRes))+
#   geom_point(size=2,alpha=0.6)+
#   geom_smooth(method = "lm",formula =y ~ poly(x, 2, raw = TRUE))+
#   xlab("Time (hrs)")+
#   ylab("Resources Used")+
#   stat_poly_eq(formula = y ~ poly(x,2, raw = TRUE), rr.digits = 2,p.digits = 3, parse = TRUE)
#
# ggsave("./output/t_Res_rt75.pdf")
# ggsave("./output/t_Res_rt75.png")
#
# Tau.mod <- lm(EcoPlate_allrt$NumRes ~ EcoPlate_allrt$Tau)
# Tau.sq.mod <- lm(EcoPlate_allrt$NumRes ~ poly(EcoPlate_allrt$Tau, 2, raw = TRUE))
# Tau.lo <- loess(EcoPlate_allrt$NumRes ~ EcoPlate_allrt$Tau, span = 3)
# plot(Tau.mod)
# AIC(Tau.mod)
# summary(Tau.mod)
# plot(Tau.sq.mod)
# AIC(Tau.sq.mod)
# summary(Tau.sq.mod)
```

```
#
# Tau_Res <- ggplot(EcoPlate_allrt,aes(x=Tau,y=NumRes))+
#   geom_point(size=2,alpha=0.6)+
#   geom_smooth(method = "loess", span = 3)+
#   xlab(expression(paste("log(", tau,")")))+
#   ylab("# of Resources Consumed")
#
# Tau_Res
#
# ggsave("./output/Tau_Res.pdf")
# ggsave("./output/Tau_Res.png")
#
```

#Generate and save N vs. Tau pilot figure

```
# Tau.mod <- lm(log(RT$N, 10) ~ log(RT$Tau, 10))
# Tau.sq.mod <- lm(log(RT$N, 10) ~ poly(log(RT$Tau,10), 2, raw = TRUE))
# plot(Tau.mod)
# AIC(Tau.mod)
# summary(Tau.mod)
# plot(Tau.sq.mod)
# AIC(Tau.sq.mod)
# summary(Tau.sq.mod)
#
# N <- ggplot(RT,aes(x=log(Tau, 10),y=log(N, 10)))+
#   geom_point(size=2,alpha=0.6)+
#   geom_smooth(method = "lm",formula =y ~ poly(x, 2, raw = TRUE))+
#   xlab(expression(paste("log(", tau,")")))+
#   ylab("log(N)") +
#   stat_poly_eq(formula = y ~ poly(x,2, raw = TRUE), rr.digits = 2, parse = TRUE)
#
# ggsave("./output/Tau_N.pdf")
# ggsave("./output/Tau_N.png")
#
# model_OT <- lm(RT$OT ~ log(RT$Tau, 10))
# summary(model_OT)
#
# OT <- ggplot(RT, aes(x=log(Tau, 10), y = OT))+
#   geom_point(size=2, alpha=0.6)+
#   geom_smooth(method = "loess", span = 3)+
#   xlab(expression(paste("log(", tau,")")))+
#   xlim(c(4,7.5))+
#   ylab("Biofilm Synthesis - Abs at 550 nm")
#
# ggsave("./output/Tau_Biofilm.pdf")
# ggsave("./output/Tau_Biofilm.png")
#
# BR <- ggplot(RT, aes(x=log(Tau, 10), y = BR))+
#   geom_point(size=2, alpha=0.6)+
#   geom_smooth(method = "lm",formula =y ~ poly(x, 2, raw = TRUE))+
#   xlab(expression(paste("log(", tau,")")))+
#   ylab(expression(paste("Bacterial respirtaiton (", mu,"M O2/hr)")))+
#   stat_poly_eq(formula = y ~ poly(x,2, raw = TRUE), rr.digits = 2, parse = TRUE)
```

```
# ggsave("./output/Tau_BR.pdf")
# ggsave("./output/Tau_BR.png")
#
# BR_N <- ggplot(RT, aes(x = log(Tau,10), y = BR_N))+
#   geom_point(size=2, alpha=0.6)+
#   geom_smooth(method = "lm",formula =y ~ poly(x, 2, raw = TRUE))+
#   xlab(expression(paste("log(", tau,")")))+
#   ylab(expression(paste("Bacterial respirtaiton (", mu,"M O2/hr")))+
#   stat_poly_eq(formula = y ~ poly(x,2, raw = TRUE), rr.digits = 2, parse = TRUE)
#
# ggsave("./output/Tau_ind.R.pdf")
# ggsave("./output/Tau_ind.R.png")
#
# model_BP_N <- lm(log(RT$BP_N, 10) ~ poly(log(RT$Tau, 10), 2, raw = TRUE))
# summary(model_BP_N)
#
# BP_N <- ggplot(RT, aes(x=log(Tau, 10), y = BP_N))+
#   geom_point(size=2, alpha=0.6)+
#   geom_smooth(method = "loess", span = 3)+
#   xlab(expression(paste("log(", tau,")")))+
#   ylab(expression(paste("Individual BP (", mu,"M C/hr/cell)")))
#
# ggsave("./output/Tau_ind.P.pdf")
# ggsave("./output/Tau_ind.P.png")
#
# model_BP <- lm(log(RT$BP, 10) ~ poly(log(RT$Tau, 10), 2, raw = TRUE))
# summary(model_BP)

# BP <- ggplot(RT, aes(x=log(Tau, 10), y = BP))+
#   geom_point(size=2, alpha=0.6)+
#   geom_smooth(method = "loess", span = 3)+
#   xlab(expression(paste("log(", tau,")")))+
#   ylab(expression(paste("Total BP (",mu, "M C/hr)")))
#
# ggsave("./output/Tau_BP.pdf")
# ggsave("./output/Tau_BP.png")
#
# ggarrange(N, BP, BP_N, labels = c("A", "B", "C"), ncol = 3, nrow = 1)
# ggsave("./output/Tau_BP_N_BP_N.pdf", width = 15, height = 5)
# ggsave("./output/Tau_BP_N_BP_N.png", width = 15, height = 5)
#
# ggarrange(OT, Tau_Res, labels = c("A", "B"), ncol = 2, nrow = 1)
# ggsave("./output/Traits.pdf", width = 10, height = 5)
# ggsave("./output/Traits.png", width = 10, height = 5)
```