Morphology supplementtary methods

Luciana Ant, François Le Dily, Miguel Beato, Patricia Saragüeta

Supplementary methods

1. Libraries

The libraries used for these analysis were:

```
library(ggplot2)
library(plyr)
library(gridExtra)
library(cowplot)
library("reshape2")
library(GGally)
library(nlme)
library(knitr)
library(xtable)
library(emmeans)
library(ggeffects)
library(car)
library(dplyr)
```

2. Data preparation

```
path<-"/home/luciana/Documents/LABO/morfología_DF_FINAL_5.4.csv" #path to circularity csv

df <- read.table(path, header = T, sep = ",") #open circularity csv

df<- df %>%
  mutate(placaOK = paste(treatment,time,placa)) #add a unique identifier to each pate
```

We used the mean circularity per photo as the response variable. This is due to the own nature of the variable distribution, that only varies from 0 to 1. The circularity variable did not follow a normal distribution, so we were not able to use a simple regression model. To solve this, we used the mean circularity per photo.

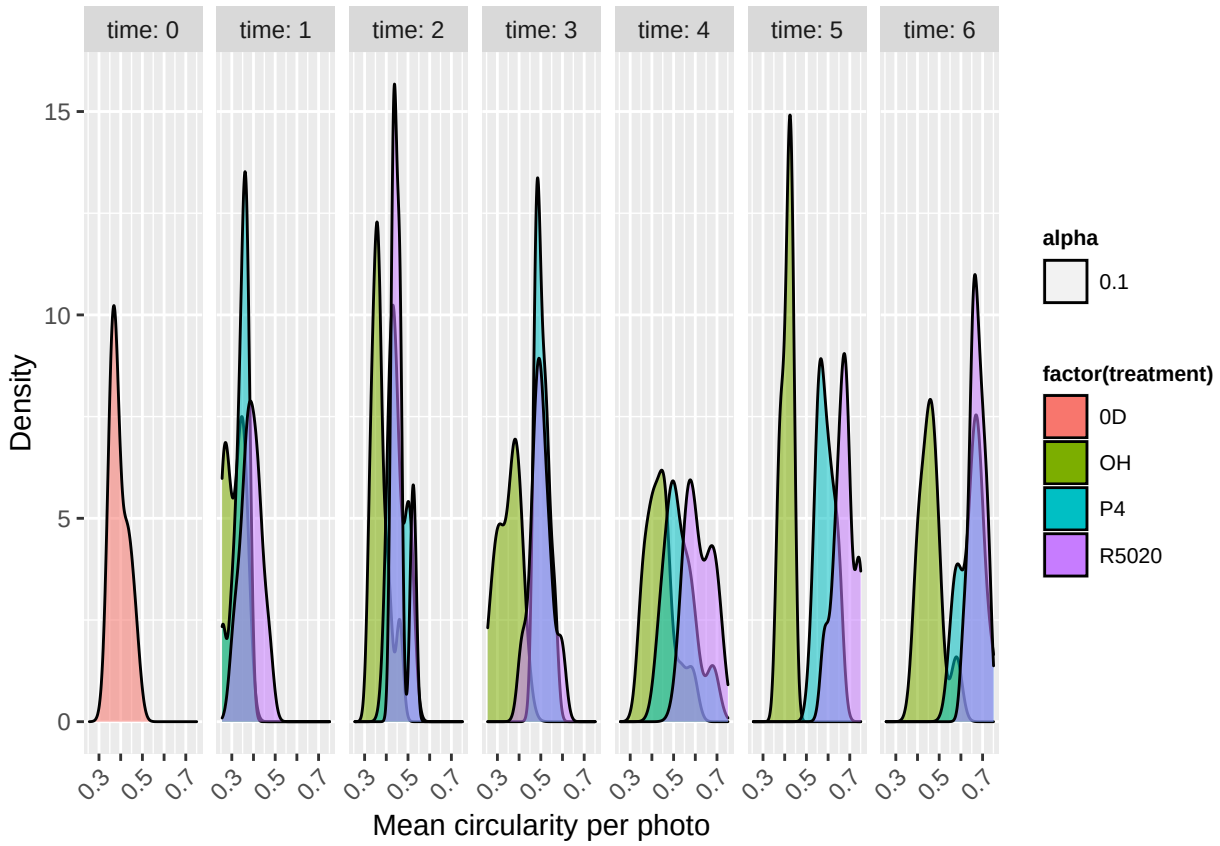
```
shapiro_test <- shapiro.test(df$Circularity) # Shapiro-test of the circularity variable, indicating a l

# Group circularity by photo
df2 <- df %>%
  group_by(file) %>%
  summarise(mean_Circularity = mean(Circularity, na.rm = TRUE),
            time= mean(time, na.rm =TRUE),
            treatment=unique(treatment), NCells=n(),
            plate=unique(placaOK), photo=unique(file))

df2$time <- as.factor(df2$time) # time as factor
```

From day 1 to 6 we used a model with two variables: time and treatment. Since the T0 plates did not share time or treatment with any of the other plates, we could not add them to this model. Because of this, we added another model, in order to compare the circularity between day 1 and T0. This last model only had one variable: "treatment", which levels were: "T0", "Day 1 EtOH", "Day 1 R5020" and "Day 1 P4".

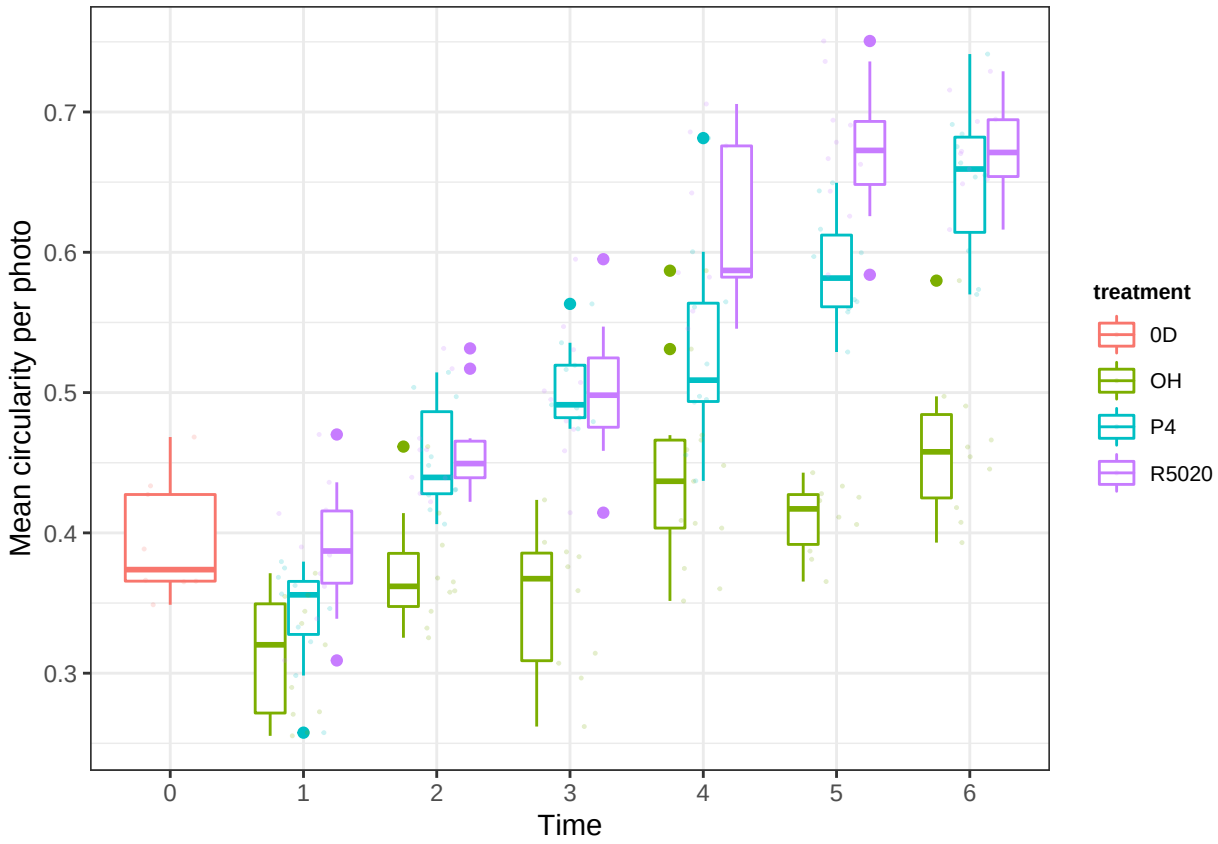
```
# Create database with only T0 and Day 1, dfT0
df_T0 <- df2[df2$time %in% c("0", "1"), ] %>% as.data.frame()

# Create database from Day 1 to Day 6
df_wo_T0 <- df2 %>% filter(time !=0)
```

3. Exploratory analysis

```
#### Gráfico de densidad
dens<-ggplot(data = df2, aes(mean_Circularity)) +
  geom_density(aes(fill=factor(treatment), alpha=0.1)) +
  xlab("Mean circularity per photo")+ylab("Density")+
scale_x_continuous(guide = guide_axis(check.overlap = TRUE,
                                      angle = 45))+
   theme(legend.position="right",
         legend.text=element_text(size = 8),
         legend.title = element_text(size=8, face="bold")) +
  facet_grid(. ~time,labeller=label_both )

dens
```




```
#### Boxplot
box <- ggplot(df2, aes(x=time, y=mean_Circularity,
                       colour=treatment)) +
  geom_boxplot(aes(color=treatment), coef=TRUE)+
  theme_bw()+
  geom_jitter(alpha=0.2, size=0.2,aes(color=treatment),
              position = position_jitter(width = .2))+
  theme(legend.position="right",
        legend.text=element_text(size = 8),
        legend.title = element_text(size=8, face="bold")) +
  ylab("Mean circularity per photo")+xlab("Time")

box
```



4. Generalized linear mixed models

```
# Model 1: conditional model
m1 <- lme(mean_Circularity ~ treatment*time, random= ~1|plate,
          weights = ~(1/NCells),data = df_wo_T0)


# Model T0: Model to analyze T0 and Day1
mT0 <- lme(mean_Circularity ~ treatment,
           random= ~1|plate, weights = ~(1/NCells),
           data = df_T0)
```

5. Check generalized linear mixed model assumptions

The assumptions of these models are: homoscedasticity, normality of errors and normality of random coefficients

```
#### M1 ####
par(mfrow=c(3,3))
```

```r
e_m1 <- resid(m1, type="pearson") # Pearson Residues
pre_m1 <- predict(m1) # Predicted circularity
Bi_m1 <- m1[["coefficients"]][["random"]][["plate"]] # estimated random coefficients

{plot(pre_m1, e_m1, xlab="Pred.", ylab =
                          "Pearson Res.",main="RE vs PRED")
  abline(0,0)} # Residues vs Predicted values plot, to prove homocedasticity

# Residues QQplot, to prove normality of errors
qqPlot(e_m1, main = "QQPlot standarized res.", ylab="standarized residues")
```

```
## P4 4 2 OH 4 2
##    107     98
```

```r
# Random coefficients QQplot, , to prove normality of random coefficients
qqPlot(Bi_m1, main="QQPlot Random effects", ylab="Random effects")
```

```
## [1] 17 13
```

```r
# Shapiro test for the errors and random coefficients, to also prove normality
shapiro_random_m1 <- shapiro.test(Bi_m1)
shapiro_random_m1
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Bi_m1
## W = 0.97783, p-value = 0.1316
```

```r
shapiro_errors_m1 <- shapiro.test(e_m1)
shapiro_errors_m1
```

```
##
##  Shapiro-Wilk normality test
##
## data:  e_m1
## W = 0.98832, p-value = 0.1368
```

```r
#### MT0 ####
e_mT0 <- resid(mT0, type="pearson") # Pearson Residues
pre_mT0 <- predict(mT0) # Predicted circularity
# Coeficientes aleatorios estimados
Bi_mT0 <- mT0[["coefficients"]][["random"]][["plate"]]# estimated random coefficients


{plot(pre_mT0, e_mT0, xlab="Pred.",
                            ylab = "Pearson Res.",main="RE vs PRED")
  abline(0,0)} # Residues vs Predicted values plot, to prove homocedasticity

# Residues QQplot, to prove normality of errors
qqPlot(e_mT0, main = "QQPlot standarized res.", ylab="standarized residues")
```

```
##    0D 0 4 R5020 1 4
##         7        39
```

```
# Random coefficients QQplot, , to prove normality of random coefficients
qqPlot(Bi_mT0, main="QQPlot Random effects", ylab="Random effects")
```



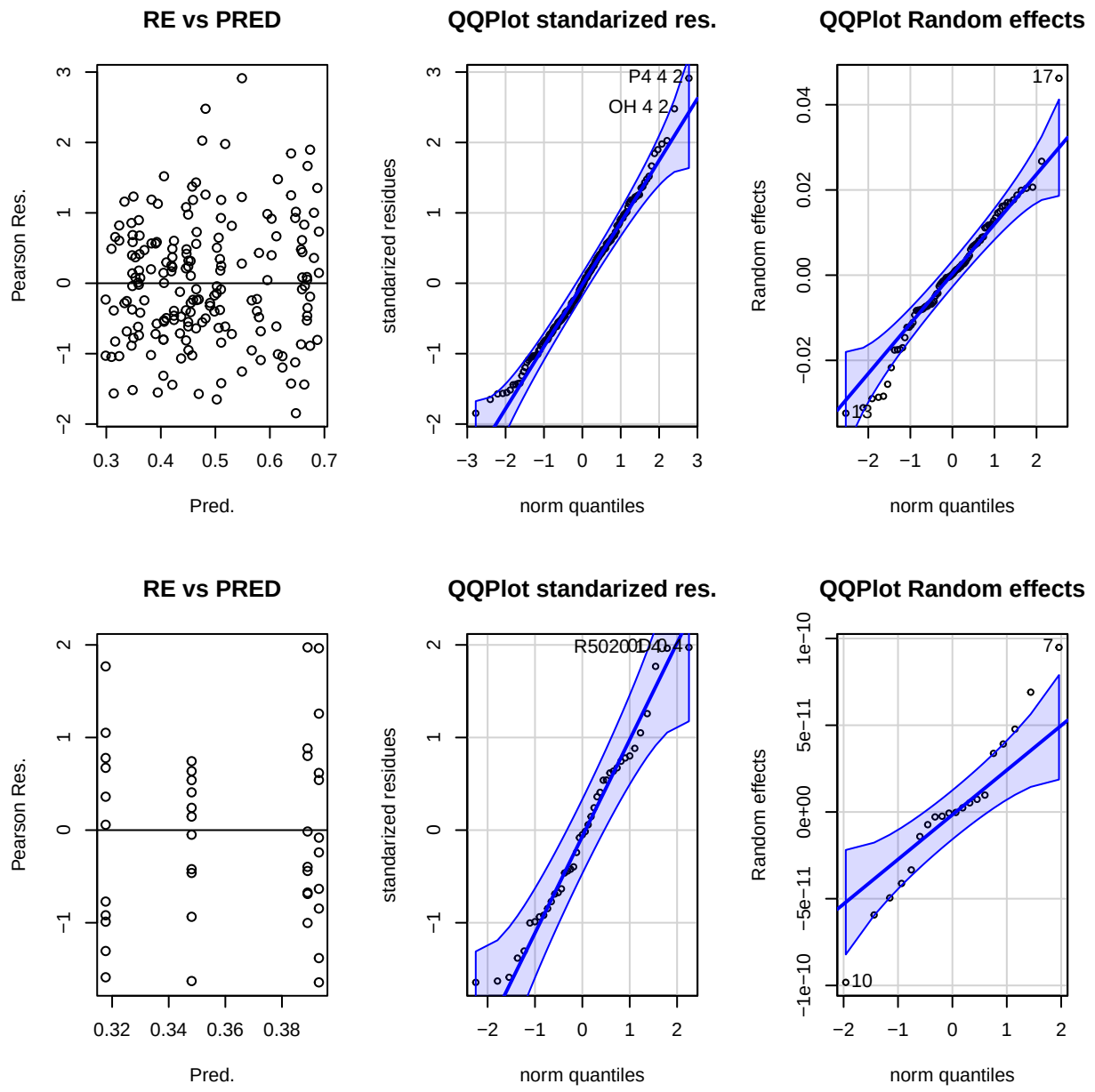
```
## [1] 10  7
```

```
# Shapiro test for the errors and random coefficients, to also prove normality
shapiro_random_mt0  <- shapiro.test(Bi_mT0)
shapiro_random_mt0
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Bi_mT0
## W = 0.97121, p-value = 0.7803
```

```
shapiro_errors_mt0 <- shapiro.test(e_mT0)
shapiro_errors_mt0
```

```
##
##  Shapiro-Wilk normality test
##
## data:  e_mT0
## W = 0.97118, p-value = 0.3764
```



6. Anova

```
anovam1 <- anova(m1)
anovam1
```

```
##                numDF denDF   F-value p-value
## (Intercept)        1    94 14717.905  <.0001
## treatment          2    71   149.506  <.0001
## time               5    71    86.893  <.0001
## treatment:time    10    71     5.666  <.0001
```

```
anovaT0 <- anova(mT0)
anovaT0
```

```
##             numDF denDF  F-value p-value
## (Intercept)     1    21 3323.645  <.0001
## treatment       3    16    8.343  0.0014
```

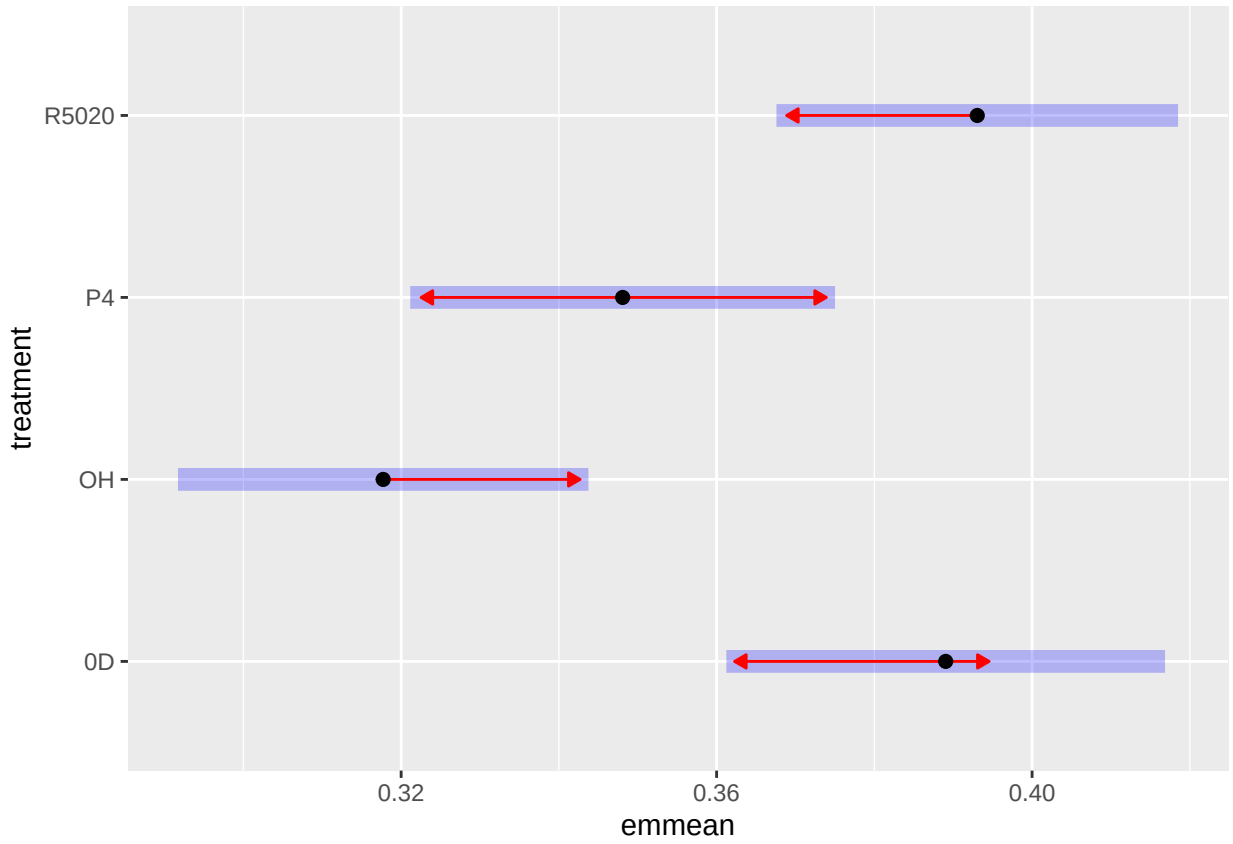
7. Simple effect comparison

For each day, we compared the circularity of the cells expected for each treatment using the emmeans package.

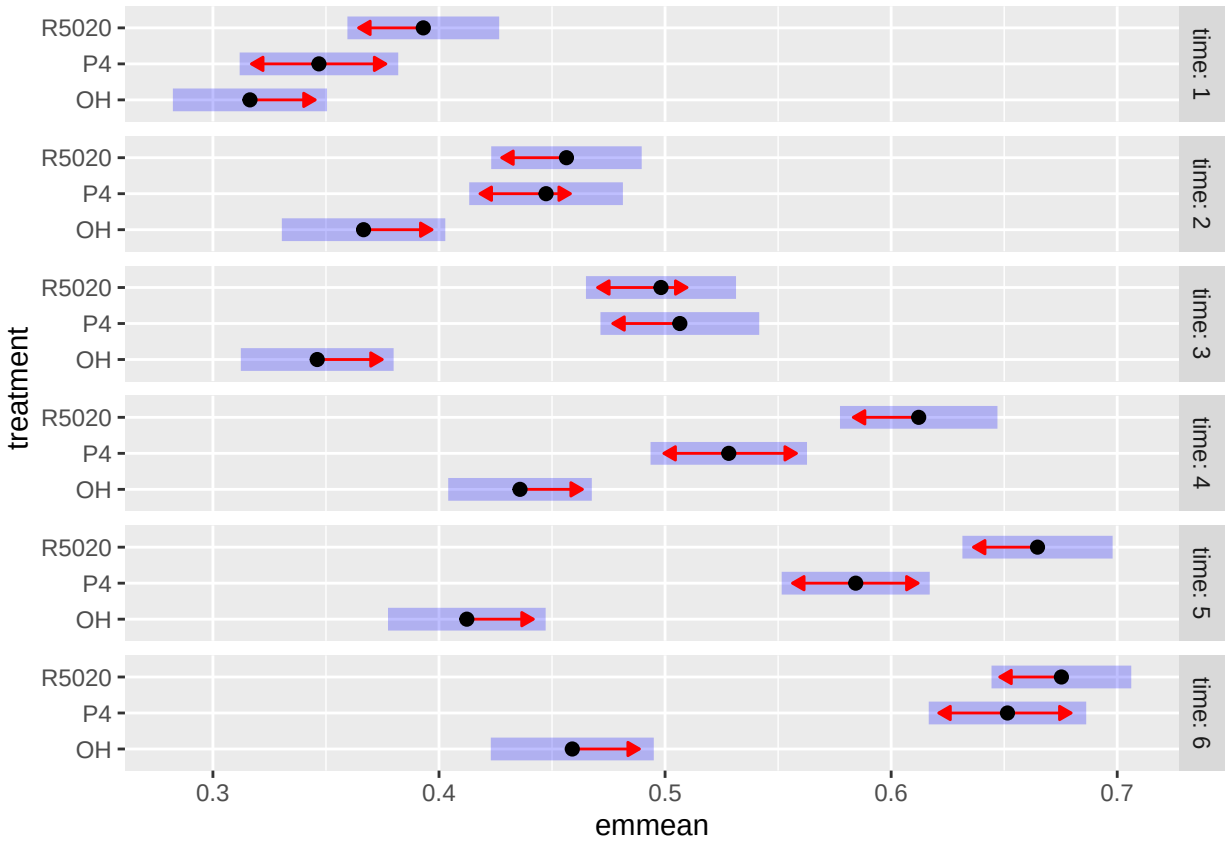
```
# Setting emmeans
options(emmeans= list(emmeans = list(infer = c(TRUE, TRUE)),
                      contrast = list(infer = c(TRUE, TRUE))))

# m1 comparissons
comp1<-emmeans(m1, pairwise ~ treatment|time )

# mT0 comparissons
compT0<-emmeans(mT0, pairwise ~ treatment )
```

Graphics depicting the significant differences between treatments for each day. The black dots indicate the expected circularity, the blue lines the 95% confidence interval and the red arrows the differences between treatments with a 95% confidence, were the overlapping of the arrows indicates the difference in circularity is not significant.

```
plot(compT0, comparisons = TRUE)
```




```
plot(comp1, comparisons = TRUE)
```


treatment
R5020
P4
OH
time: 1
time: 2
time: 3
time: 4
time: 5
time: 6
0.3
0.4
0.5
0.6
0.7
emmean