Synchronisation analysis of R-GECO1 and CDPK-FRET in guard cells

Leaves cells analysis script was adapted from Li et al. 2021 (doi.org/10.1111/nph.17202)

This script reads the excel file with the calcium and CPK-FRET traces from the guard cell and performs a synchronisation analysis of the signals

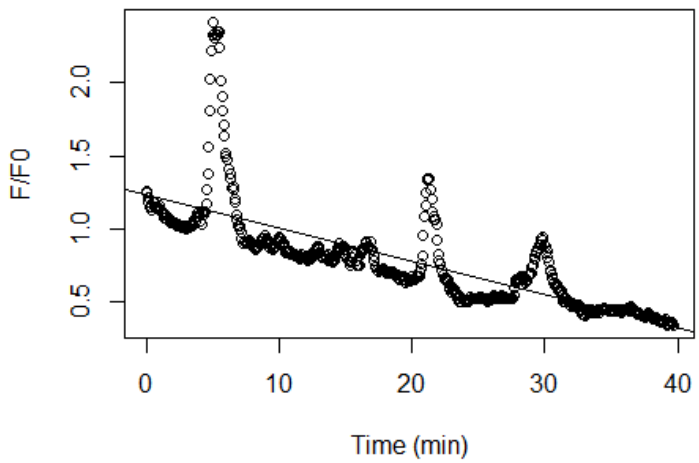
note: flg22 or ABA treatment after 3 min - data normalized 10 frames before treatment

1. Setting of the directory, installing and loading of packages, loading data from excel file and creating a results folder

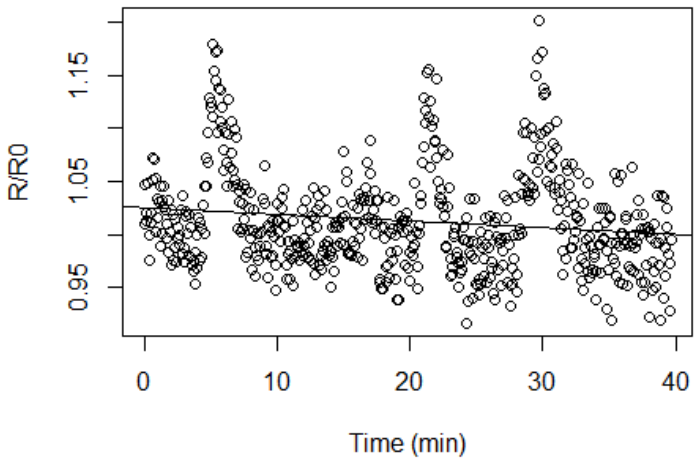
```
rm(list = ls())

setwd("C:/Users/slederer/Desktop/ABA")  # adjusting working directory (folder with data file)
requirements=c("readxl","writexl","scales","baseline","pracma","pastecs","plotly","stats",
               "ggplot2", "gridExtra", "forecast")
missing <- setdiff(requirements,
                   rownames(installed.packages()))

if (length(missing) != 0) {
  install.packages(missing, dependencies = TRUE)
}
invisible(lapply(requirements, require, character.only = TRUE))

dir.create(paste(getwd(),"/results",sep=""))
table <- read_excel("GC.xlsx")
```

2. Creating of a linear regression model with time (Time) as predictor variable and signal (RG or FRET_ratio) as response variable, assuming that the technical artefacts can be represented by a linear relationship

```
model_RG <- lm(RG ~ Time, data = table)
plot(table$RG ~ table$Time, xlab = "Time (min)", ylab = "F/F0")
abline(model_RG)
```

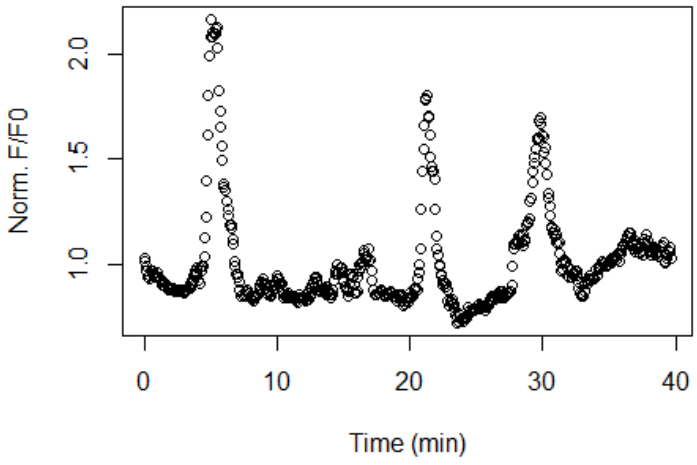


```
model_FRET <- lm(FRET_ratio ~ Time, data = table)
plot(table$FRET_ratio ~ table$Time, xlab = "Time (min)", ylab = "R/R0")
abline(model_FRET)
```

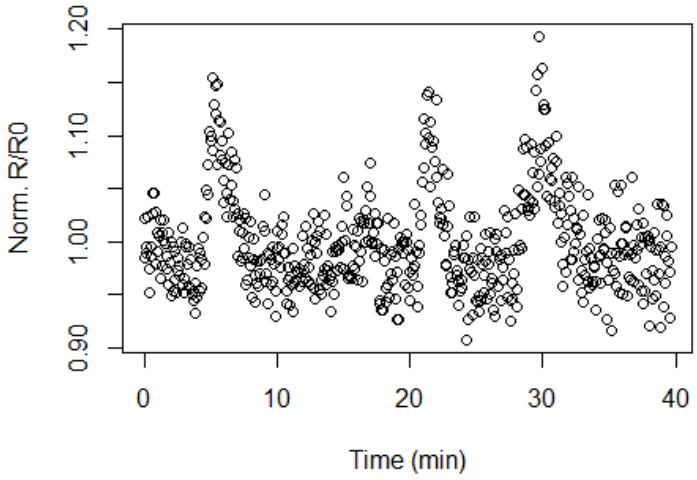


3. Normalization of the data

```
lm_RG <- (table$Time * model_RG$coefficients[-c(1)]) + model_RG$coefficients[c(1)]
lm_FRET <- (table$Time * model_FRET$coefficients[-c(1)]) +
            model_FRET$coefficients[c(1)]

table$normalized_RG <- table$RG/lm_RG
table$normalized_FRET <- table$FRET_ratio/lm_FRET

plot(table$normalized_RG ~ table$Time, xlab = "Time (min)", ylab = "Norm. F/F0")
```

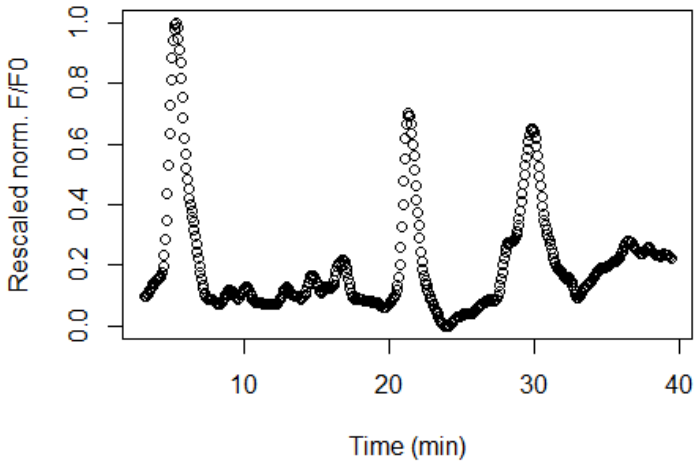



```
plot(table$normalized_FRET ~ table$Time, xlab = "Time (min)", ylab = "Norm. R/R0")
```

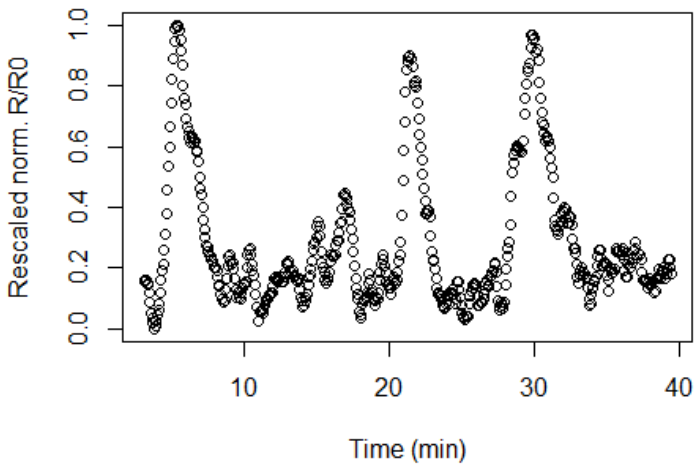


4. Re-scaling between 0 and 1 after treatment application (frame 45) and smoothing of data

```
table$RG_rescaled <- NA
table$FRET_rescaled <- NA
smoothed_normalized_RG <- as.numeric(ma(table$normalized_RG, order = 8,
                                        centre = TRUE))
smoothed_normalized_FRET <- as.numeric(ma(table$normalized_FRET, order = 8,
                                          centre = TRUE))

table$RG_rescaled[45:(nrow(table) - 4)] <- rescale(smoothed_normalized_RG
                                                   [45:(nrow(table) - 4)], c(0,1))
table$FRET_rescaled[45:(nrow(table) - 4)] <- rescale(smoothed_normalized_FRET
                                                     [45:(nrow(table) - 4)], c(0,1))

plot(table$RG_rescaled[45:(nrow(table) - 4)] ~ table$Time[45:(nrow(table) - 4)],
     xlab = "Time (min)", ylab = "Rescaled norm. F/F0")
```

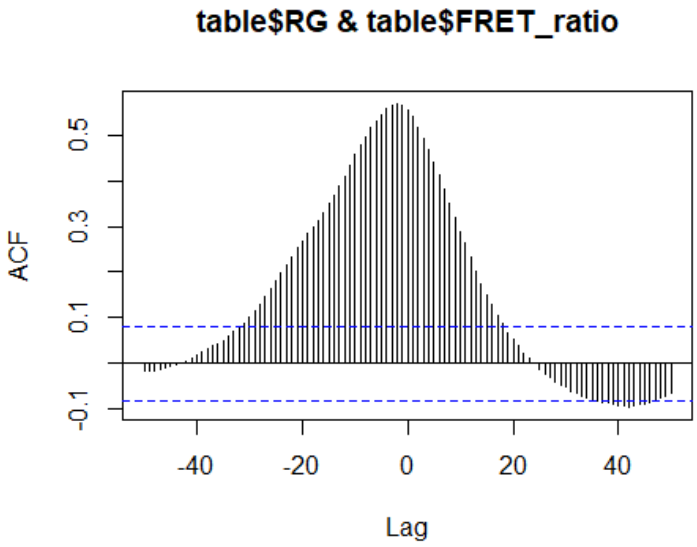



```
plot(table$FRET_rescaled[45:(nrow(table) - 4)] ~ table$Time[45:(nrow(table) - 4)],
     xlab = "Time (min)", ylab = "Rescaled norm. R/R0")
```

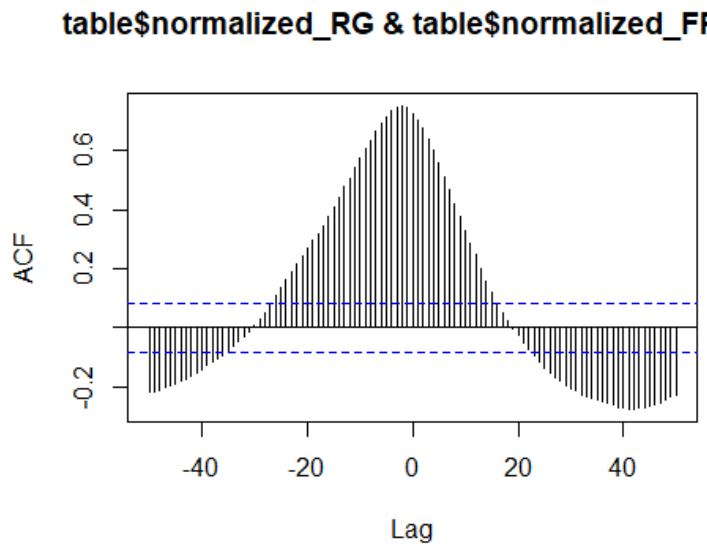


5. Cross correlation analysis between R-GECO1 signal and CDPK-FRET with (ccf_normalized) and without normalization by linear model (ccf)

```
max_lag = 50
cross_corr_matrix = matrix(0,nrow=105,ncol = 4)
colnames(cross_corr_matrix) <- c("lag", "Time", "Correlation coefficients",
                                 "Correlation coefficients normalized")

ccf <- ccf(table$RG, table$FRET_ratio, lag.max = max_lag)
```




```
ccf_normalized <- ccf(table$normalized_RG, table$normalized_FRET, lag.max = max_lag)
```




```
cross_corr_matrix[1:101,1] = ccf$lag
cross_corr_matrix[1:101,2] = ccf$lag * table$Time[2]
cross_corr_matrix[1:101,3] = ccf$acf
cross_corr_matrix[1:101,4] = ccf_normalized$acf
cross_corr_matrix[102,1] = "lag max Time ccf"
cross_corr_matrix[103,1] = "max correlation coefficient cff"
cross_corr_matrix[102,2] = ccf$lag[which.max(cross_corr_matrix[1:101,3])] *
                           table$Time[2]
cross_corr_matrix[103,2] = ccf$acf[which.max(cross_corr_matrix[1:101,3])]
cross_corr_matrix[104,1] = "lag max Time ccf normalized"
cross_corr_matrix[105,1] = "max correlation coefficient cff normalized"
cross_corr_matrix[104,2] = ccf_normalized$lag[which.max(cross_corr_matrix[1:101,4])] *
                           table$Time[2]
cross_corr_matrix[105,2] = ccf_normalized$acf[which.max(cross_corr_matrix[1:101,4])]
```

6. Output as excel files in result folder

```
write_xlsx(data.frame(cross_corr_matrix),
           path=paste( getwd(),"/results/cross correlation.xlsx",sep=""), col_names = TRUE)
```

```
write_xlsx(data.frame(table),path=paste(getwd(),"/results/data.xlsx",sep=""), col_names = TRUE)
```