Assignment week 3

Detailed analysis of the glucocorticoid receptor dynamica model

Student: Vincent Talen

Student number: 389015

Class: BFV2

Study: Bio-Informatics

Institute: Institute for Life Science & Technology

Teacher: Tsjerk Wassenaar

Date: 2022-05-23

Contents

1 Assignment 1: Assessing Model Validity

To assess the validity of the model, the experimental data will be compared to the simulation data from previous week. The comparison will be done by plotting the experimental data and the simulation data in the same graph.

1.1 Loading in the data

```
# Load in data from file
data <- read.csv("MPL.csv", na.strings = "NA")
# Rename 'mRNA' and 'Free_receptor' columns to use the same name scheme
names(data)[4:5] <- c("Rm", "R")
```

The actual experiment data contains multiple values per time, if a plot is made with those raw values the plot below will result.

```
# Create plot with raw values to demonstrate the need of using medians
ggplot(data, mapping = aes(x = time, y = R, width = 2)) + geom_line() + geom_point()
```

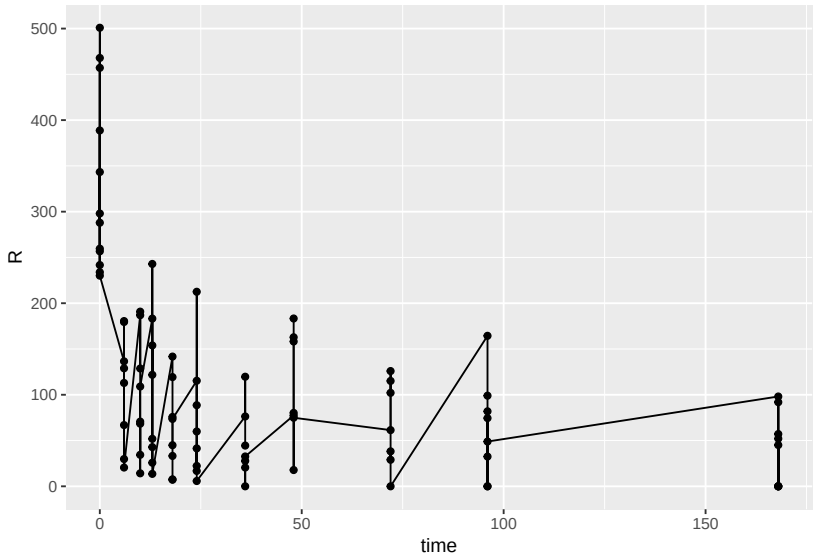


Figure 1: Demonstration of plotting raw data

As can be seen the line goes through each single point and thus is the line not that useful. So to be able to get a meaningful plot the medians are used to create a useful line of the medians that can then be used to compare the values.

```
# Create new data frame containing the medians per time
data.medians <- aggregate(data[,c("MPL_conc","Rm","R")],
                          list(data$dose,data$time), median, na.rm=T)
names(data.medians)[1:2] <- c("dose","time")
```

1.2 Implementing GRD model with given values

Table 1: Initial Values for MPL

Parameter	Value	Unit	Explanation
Rm	4.74	$fmol/g\ liver$	concentration of receptor mRNA
R	267	$fmol/mg\ protein$	concentration of free receptor in cytosol
DR	0	$fmol/mg\ protein$	concentration of receptor complex in cytosol
DR_N	0	$fmol/mg\ protein$	concentration of receptor complex in nucleus

Table 2: Parameter Values for MPL

Parameter	Value	Unit	Explanation
k.s_Rm	2.9	$fmol/g\ liver/h$	zero-order rate constant of receptor mRNA synthesis
k.d_Rm	0.612	-	first-order rate constant receptor mRNA degradation
IC.50_Rm	26.2	$fmol/mg\ protein$	concentration of DR_N where receptor mRNA synthesis drops to 50% of base value
k.on	0.00329	$L/nmol/h$	second-order rate constant of receptor complex formation
k.T	0.63	$1/h$	first-order rate constant of translocation of receptor complex to nucleus
k.re	0.57	$1/h$	first-order rate constant of receptor 'recovery' from nucleus to cytosol
R.f	0.49	-	fraction of receptor being recycled from complexes
k.s_R	3.22	-	first-order rate constant of receptor synthesis
k.d_R	0.0572	$1/h$	first-order rate constant of receptor degradation
D	-	$nmol/L$	plasma concentration of corticosteroid

```
# Create function of GRD model
model <- function(time, cur.state, parameters) {
  ## Unpack the current state and the parameters for instant access
  with(as.list(c(cur.state, parameters)), {
    ## Calculate delta for each equation and return them in a list
    delta.Rm <- k.s_Rm * ( 1 - DR_N / ( IC.50_Rm + DR_N )) - k.d_Rm * Rm
    delta.R <- k.s_R * Rm + R.f * k.re * DR_N - k.on * D * R - k.d_R * R
    delta.DR <- k.on * D * R - k.T * DR
    delta.DR_N <- k.T * DR - k.re * DR_N
    return(list(c(delta.Rm, delta.R, delta.DR, delta.DR_N)))
    })
}

# Initial values and time frame
initial.values <- c(Rm = 4.74, R = 267, DR = 0, DR_N = 0)
times <- seq(0, 168, by = 1)

# Create function that converts concentration of MPL from ng/mL to nmol/L
calc.D <- function(ng.ml.concentration) { return(ng.ml.concentration * 1000 / 374.471) }
# Define parameters determined for methylprednisolone (MPL)
parameters <- c(k.s_Rm = 2.9, k.d_Rm = 0.612, IC.50_Rm = 26.2, k.on = 0.00329, k.T = 0.63,
                k.re = 0.57, R.f = 0.49, k.s_R = 3.22, k.d_R = 0.0572, D = calc.D(20))
```

1.3 Creating plots to compare experiment data with simulation data

Everything is coded into functions so they can be used dynamically.
The main function `createPlotsForDoses` collects the data for a dose and creates the plots with it, it also uses another function called `addPlotDataAndStyle` which places the data into the plots and styles it.

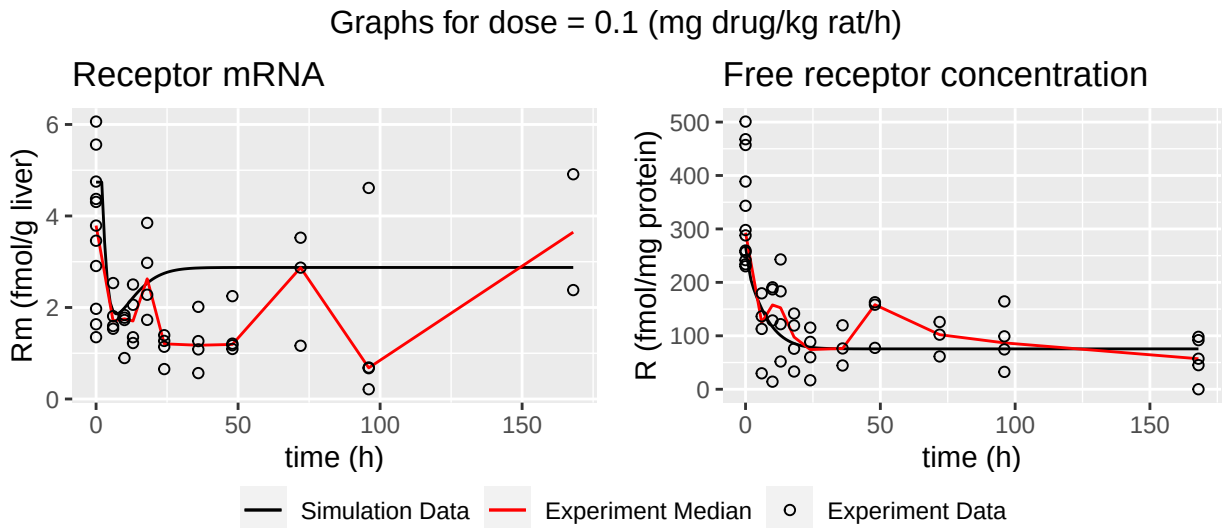
```
# Function to create plots of R and Rm for a dose
createPlotsForDoses <- function(current.dose) {
  ## Change to correct D parameter for current dose
  parameters$D <- calc.D(median(data$MPL_conc[data$dose == current.dose], na.rm = T))

  ## Get subset of experiment data for current dose
  cur.data.points <- subset(data, dose == 0.0 | dose == current.dose)
  ## Get subset of experiment median data for current dose
  cur.data.medians <- subset(data.medians, dose == 0.0 | dose == current.dose)
  ## Perform ODE function with the model to get simulation data
  cur.data.simulated <- as.data.frame(ode(times = times, y = initial.values,
                                          parms = parameters, func = model,
                                          method = "euler"))

  ## Create plot for receptor mRNA
  plot.Rm <- ggplot(cur.data.simulated, aes(x = time, y = Rm)) +
    labs(title = "Receptor mRNA", y = "Rm (fmol/g liver)", x = "time (h)")
  plot.Rm <- addPlotDataAndStyle(plot.Rm, cur.data.medians, cur.data.points)

  ## Create plot for free receptor concentration
  plot.R <- ggplot(cur.data.simulated, aes(x = time, y = R)) +
    labs(title = "Free receptor concentration", y = "R (fmol/mg protein)", x = "time (h)")
  plot.R <- addPlotDataAndStyle(plot.R, cur.data.medians, cur.data.points)

  ## Print the plots in an arranged grid with the legend at the bottom
  grid.arrange(plot.Rm, plot.R, my.legend, ncol = 2, nrow = 2,
               layout_matrix = rbind(c(1,2), c(3,3)),
               widths = c(2.7, 2.7), heights = c(2.5, 0.2),
               top = sprintf("Graphs for dose = %s (mg drug/kg rat/h)", current.dose))
  remove(my.legend)
}
```

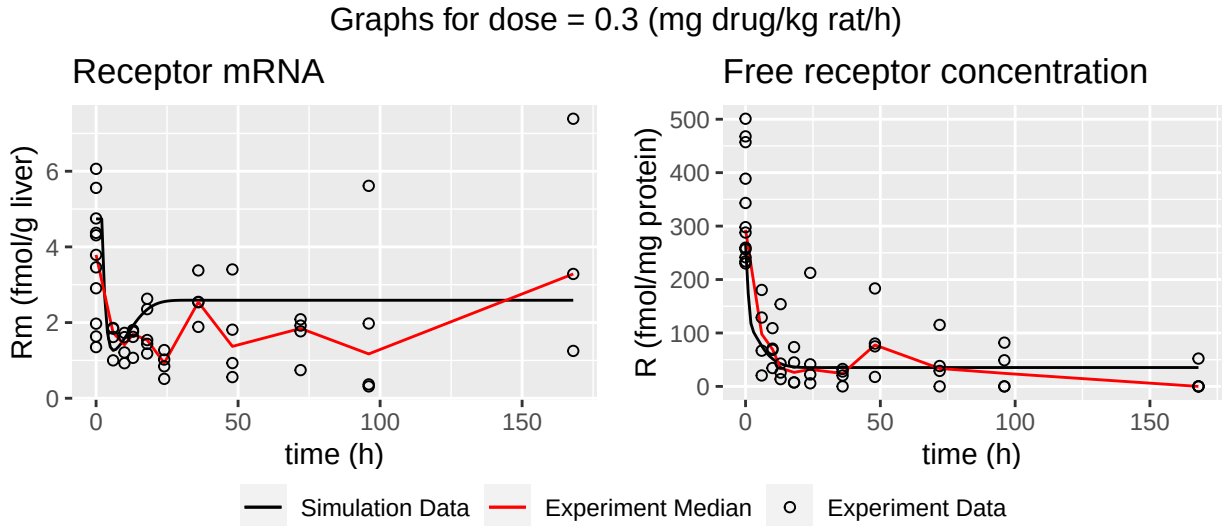
In the `addPlotDataAndStyle` function the lines and points are added and after which a legend is made. The legend then gets extracted using the `getLegend` function, put into it's own variable that can be used outside this function scope and then the legend is removed from the plot so it can be shown as a collective legend below both plots.

```
# Function to add the data and styles to the plots
addPlotDataAndStyle <- function(my.plot, cur.data.medians, cur.data.points) {
  my.plot <- my.plot +
    ## Add the lines and data points
    geom_line(aes(color = "Simulation Data")) +
    geom_line(aes(color = "Experiment Median"), data = cur.data.medians) +
    geom_point(aes(color = "Experiment Data"), data = cur.data.points, shape = 1) +
    theme(
      legend.position = "bottom",
      plot.margin = margin(0.25, 0.25, 0.25, 0.25, "cm")) +
    ## Use scale_color_manual to color the data and also set their names in the legend
    scale_color_manual("",
      values = c("black","red","black"),
      limits = c("Simulation Data", "Experiment Median", "Experiment Data")) +
    ## Correct the displayed line types in the legend
    guides(color = guide_legend(
      override.aes = list(linetype = c(1, 1, NA), shape = c(NA, NA, 1))))

  ## Gather legend into it's own variable and remove from plot
  my.legend <<- getLegend(my.plot)
  my.plot <- my.plot + theme(legend.position="none")
  return(my.plot)
}

# Function to extract the legend from a plot
getLegend <- function(myggplot) {
  tmp <- ggplot_gtable(ggplot_build(myggplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  return(legend)
}
```

Finally, after making all the functions gather the unique doses (except 0.0) and use lapply to creates the plots for each dose.

```
# Get all unique doses present in the data except 0.0
all.doses <- unique(data$dose)
all.doses <- all.doses[!all.doses %in% (0.0)]

# Create plots for each dose by using lapply() on createPlotsForDoses()
invisible(lapply(all.doses, createPlotsForDoses))
```





1.4 Results of plot comparison

[2] How do the results of the simulations depend on the dose and concentration of the drug?
Compare the model variables mRNA, R with the experimental data

[3] Are the results of the model in line with experimental data? If not, what could be the reason? Think of at least one explanation.
Try to test it with simulations (you will get bonus points for that, your explanation does not need to be correct, but should be logical).

References

[1] Barnes, P.J. (2011), *Glucocorticosteroids: current and future directions*, British Journal of Pharmacology, 163: 29-43, https://doi.org/10.1111/j.1476-5381.2010.01199.x (accessed May 11, 2022).