

Thema08-week1-mRNA

Miguel Botter van Elburg & Jurrien de Jong

2021-05-19

```
data <- read.csv("MPL.csv", na.strings = "NA")

medians <- aggregate(data[,c("MPL_conc", "mRNA", "Free_receptor")], list(data$dose, data$time), median, na.rm = TRUE)
names(medians)[1:2] <- c("dose", "time")
medians_01 <- subset(medians, dose == 0.1 | dose == 0)
medians_03 <- subset(medians, dose == 0.3 | dose == 0)
```

1 Introduction

Zal een model altijd kloppen? Wanneer we de model data vergelijken met echte uitgevoerde onderzoeken met ratten kan er duidelijk gezegd worden of het model daadwerkelijk geldig is. Modellen zijn vooral handig om te kijken wat er precies gebeurt bij bepaalde scenario's, zo hoeven we ze niet in het echt uit te voeren, slechts met een simulatie. Het bewijzen van de geldigheid van het model en het visualiseren van verschillende scenario's komen in dit onderzoek aan bod.

1.1 Goal

Het doel is nu voornamelijk om te bekijken wat er gebeurt met de grafieken als parameters verandert worden (oftewel er moeten verschillende scenario's gerunt worden). Ook zal de geldigheid van het model op de proef gesteld worden.

1.2 Theory

- Describe biological model
- Picture of the biological model

Give an explanation of the model with citations of source [1] (replace this with actual source) and formula explanation

$$\frac{\delta R}{\delta t} = -r * R + m$$

Describe each element and the transformations

2 Methods

Over het algemeen biedt de mediaan een beter idee van de data verdeling, bij steekproefproefgemiddelden mits zij een normaal verdeling volgen. Aangezien de mediaan robuuster is dan een gemiddelde (mean), want het is minder afhankelijk van uitbijters/uitschieters, omdat het het middelste punt is van een gesorteerde reeks cijfers/getallen/gegevens en de scores niet worden gedeeld door het aantal n. Een goede illustratie zegt meer dan duizend woorden, om die reden is het beter om een grafiek van de mediaan erbij te plotten.

2.1 The software model

2.2 Model configuration

De configuratie (parameters etc) is te zien in onderstaand figuur/tabel. Let op. Bij verschillende opgaven worden andere parameters gebruikt, zoals dat $D = 39$ bij de onderstaande code block in de 'results' sectie.

```
knitr::include_graphics('Fig3_week2.png')
```

ks_Rm	2.9	fmol/g liver/h
IC50_Rm	26.2	fmol/mg protein
kon	0.00329	L/nmol/h
kT	0.63	1 / h
kre	0.57	1 / h
Rf	0.49	
kd_R	0.0572	1 / h
kd_Rm	0.612	nmol/L
ks_r	3.22	
D	53.40868	
Rm0	4.74	fmol / g liver
R0	267	fmol/mg protein
DR	0	fmol/mg protein
DR(N)	0	fmol/mg protein

3 Results

De vragen die opkomen tijdens het onderzoeken van een biologisch systeem zal je kunnen beantwoorden aan de hand van een gecreerd model. Hieronder is de code te zien dat zo'n soort model kan produceren:

```
# this function calculates the derivatives and returns it as a list.
Glucocorticoid_func <- function(t, y, parms) {
  with(as.list(c(y, parms)),{

    # Dit model bevat 4 afgeleide functies:
    # Afgeleide 1:

    dmRNAr_dt <- ks_Rm * ( 1 - (DRN / (IC50_Rm + DRN))) - kd_Rm * mRNAr

    # Afgeleide 2:

    dR_dt <- ks_r * mRNAr + Rf * kre * DRN - kon * D * R - kd_R * R

    # Afgeleide 3:

    dDR_dt <- kon * D * R - kT * DR

    # Afgeleide 4:

    dDRN_dt <- kT * DR - kre * DRN

    return(list(c(dmRNAr_dt, dR_dt, dDR_dt, dDRN_dt)))
  }
}
```

```
}
```

Question 1.2

```
par(mfrow = c(2,2) )

# Set initial values
state <- c(mRNAr = 4.74, R = 267, DR = 0, DRN = 0)
t <- seq(0, 168, by = 1)

# -----
# Dose 0.1

parameters_01 <- c(ks_Rm = 2.90, IC50_Rm = 26.2, kon = 0.00329,
                  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
                  kd_Rm = 0.612, ks_r = 3.22, D = 39.0, Rm0 = 4.74,
                  DR = 0, DRN = 0)

out_01 <- deSolve::ode(times = t, y = state, parms = parameters_01,
                      func = Glucocorticoid_func, method = "lsoda")

out_01 <- as.data.frame(out_01)

plot(out_01$time, out_01$mRNAr,
     main="Receptor mRNA by dose 0.1",
     ylab=c("nmol/L"), xlab=c(" Time in hours"),
     type='l', lwd = 2, xlim= c(0,60), ylim= c(0,5))

lines(data$time, data$mRNA, type = "p")
lines(medians_01$time, medians_01$mRNA, type = "l", col= "red")

plot(out_01$time, out_01$R,
     main="Free receptor mRNA by dose 0.1",
     ylab=c("nmol/L"), xlab=c(" Time in hours"),
     type='l', lwd = 2, xlim= c(0,60), ylim= c(0,500))

lines(data$time, data$Free_receptor, type = "p")
lines(medians_01$time, medians_01$Free_receptor, type = "l", col= "red")

# -----
# Dose 0.3

parameters_03 <- c(ks_Rm = 2.90, IC50_Rm = 26.2, kon = 0.00329,
                  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
                  kd_Rm = 0.612, ks_r = 3.22, D = 107, Rm0 = 4.74,
                  DR = 0, DRN = 0)

out_03 <- deSolve::ode(times = t, y = state, parms = parameters_03,
                      func = Glucocorticoid_func, method = "lsoda")

out_03 <- as.data.frame(out_03)

plot(out_03$time, out_03$mRNAr,
     main="Receptor mRNA by dose 0.3",
```

```

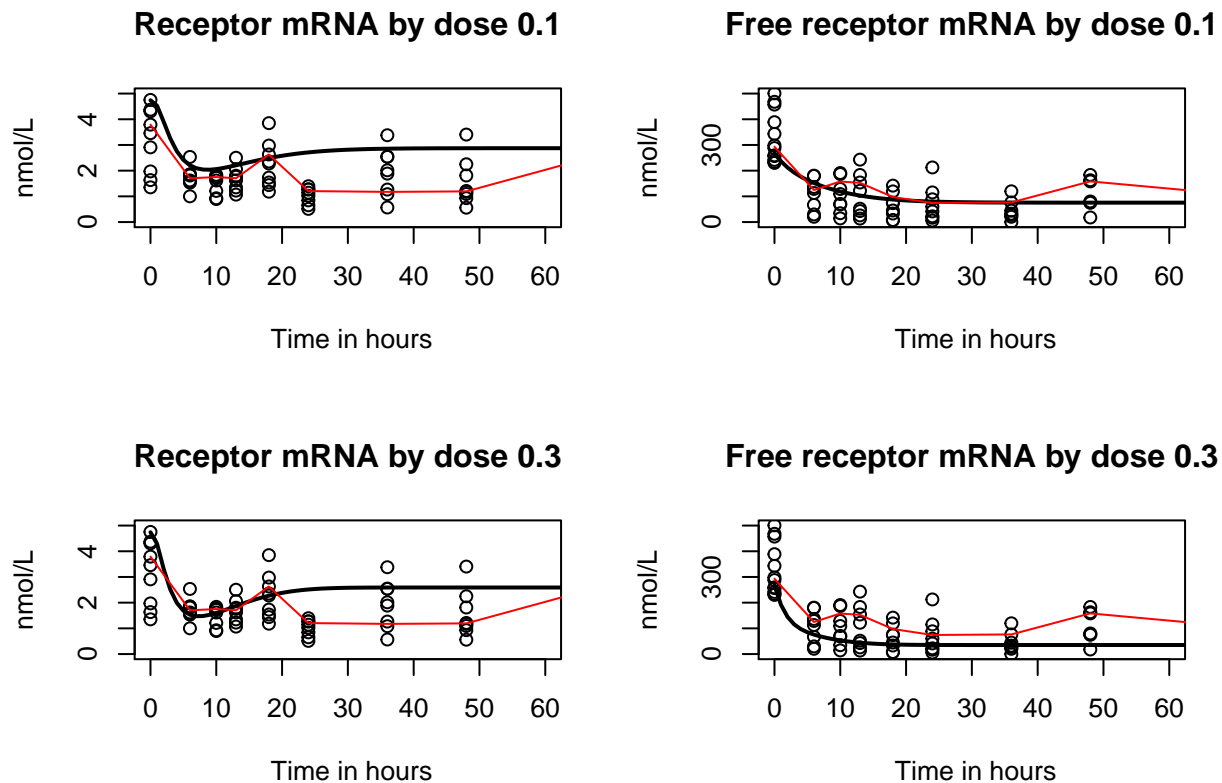
ylab=c("nmol/L"), xlab=c(" Time in hours"),
type='l', lwd = 2, xlim= c(0,60), ylim= c(0,5))

lines(data$time, data$mRNA, type = "p")
lines(medians_01$time, medians_01$mRNA, type = "l", col= "red")

plot(out_03$time, out_03$R,
      main="Free receptor mRNA by dose 0.3",
      ylab=c("nmol/L"), xlab=c(" Time in hours"),
      type='l', lwd = 2, xlim= c(0,60), ylim= c(0,500))

lines(data$time, data$Free_receptor, type = "p")
lines(medians_01$time, medians_01$Free_receptor, type = "l", col= "red")

```



Question 1.3

De resultaten van het model (zwart) komen wel overeen met elkaar, aangezien de gefitte lijnen (rode experiment data) een gevolg zijn van de data observaties (zwarte model data). Het onderlinge verschil tussen de twee model zwarte model-lijnen is mogelijk te verklaren door het onderlinge verschil in condities van model groepen (sample groups). Omdat een hogere drug concentratie leidt tot verhoogde kans van bindingen met mRNA receptoren zal de concentratie mRNA bezette receptoren bij een hogere dosis (0.3 ipv 0.1) sneller afnemen in een bepaalde tijd (zie fig Receptor mRNA by dose 0.3) dan bij een lagere dosis (zie fig Receptor mRNA by dose 0.1) Dat verschil is af te leiden tussen de grafieken!

Q2.1

De tijdsverloop van het geactiveerde drug-receptor complex is afhankelijk van de auto-regulatie van de glucocorticoide receptor. Hierom moet de formule die de auto-regulatie van de glucocorticoide receptor variabele bevat worden aangepast, dat is de formule 'Afgeleide 1' in de code chunk 'formule_modellen'. Zie de onderstaande aangepaste functie van het model en de output daarvan.

$(x+1)(x-1) = 0$ Bovenstaande vergelijking is alleen waar als $x = -1$ of $x = 1$, omdat dan een van beide producten dan nul is en iets keer nul is nul.

$dmRNA_r_{dt} <- ks_Rm * (1 - (DRN / (IC50_Rm + DRN))) - kd_Rm * mRNA_r$ Wanneer er geen auto-regulatie is van de glucocorticoïde receptor is, betekent het dat de uitkomst nul wordt. Dus de 'Afgeleide 1' herleiden we op nul, net als het voorbeeld van $(x+1)(x-1)$...

De parameter ks_Rm is afhankelijk van $(1 - (DRN / (IC50_Rm + DRN)))$ en wordt nul als $(1 - (DRN / (IC50_Rm + DRN)))$ nul is.

Dus moet de volgende vergelijking worden opgelost... $ks_Rm * (1 - (DRN / (IC50_Rm + DRN))) = 0$ dus... $1 - 1 * (DRN / (IC50_Rm + DRN)) = 0$ dus... $1 = +1 * (DRN / (IC50_Rm + DRN))$ dus... $1 = (DRN / (IC50_Rm + DRN))$

dus... $1 = (0 / (0 + 0))$ want $0 / 0$ is 1! oplossing: dus de parameters DRN , $IC50_Rm$, DRN zijn nul en de vergelijking kan dus verwijderd worden.

```
Glucocorticoid_func_2.1 <- function(t, y, parms) {
  with(as.list(c(y, parms)),{

    # Dit model bevat 4 afgeleide functies:
    # Afgeleide 1:

    dmRNA_r_dt <- ks_Rm - kd_Rm * mRNA_r # * (DRN / (IC50_Rm + DRN)) This has been removed to answer
                                           # the research question

    # Afgeleide 2:

    dR_dt <- ks_r * mRNA_r + Rf * kre * DRN - kon * D * R - kd_R * R

    # Afgeleide 3:

    dDR_dt <- kon * D * R - kT * DR

    # Afgeleide 4:

    dDRN_dt <- kT * DR - kre * DRN

    return(list(c(dmRNA_r_dt, dR_dt, dDR_dt, dDRN_dt)))
  }
}

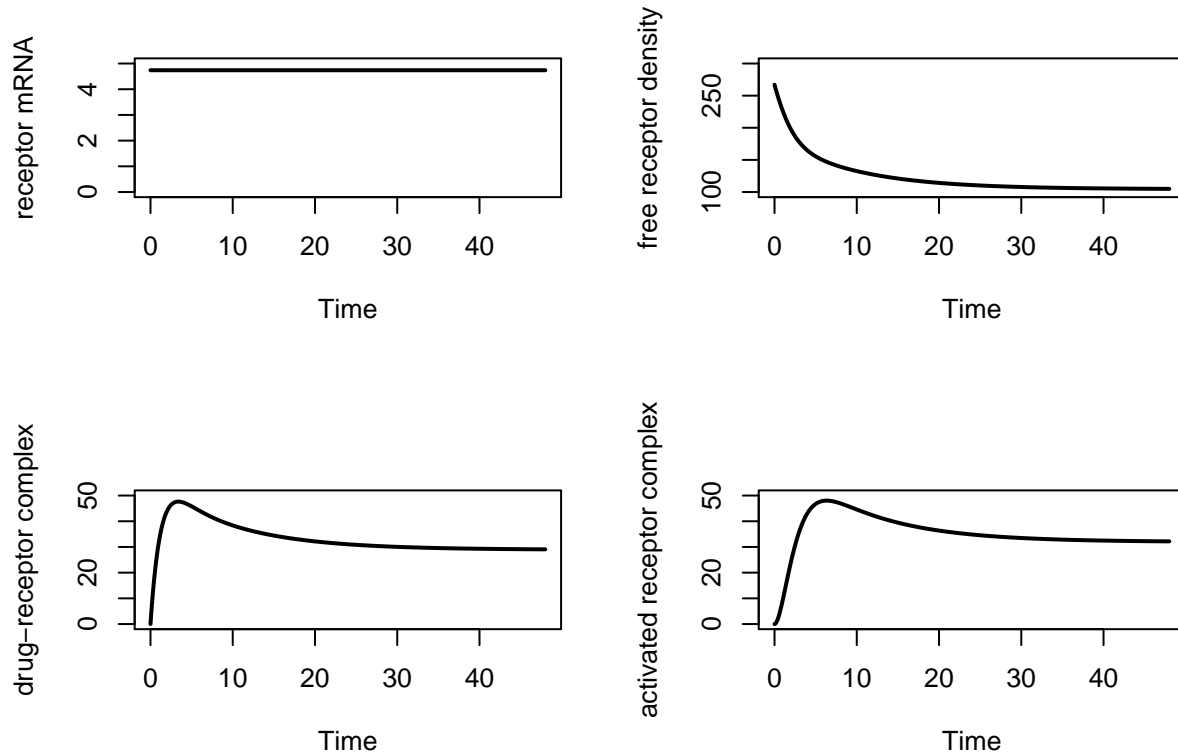
parameters <- c(ks_Rm = 2.90, IC50_Rm = 26.2, kon = 0.00329,
               kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
               kd_Rm = 0.612, ks_r = 3.22, D = 53, Rm0 = 4.74,
               DR = 0, DRN = 0)

t = seq(0, 48, by = 0.1)

out <- deSolve::ode(times = t, y = state, parms = parameters,
                   func = Glucocorticoid_func_2.1, method = "lsoda")

out <- as.data.frame(out)
```

```
par(mfrow=c(2,2))
plot(out$time,out$mRNAr,ylim = c(0,5), xlab="Time",ylab="receptor mRNA",type="l",lwd=2)
plot(out$time,out$R, ylim = c(100, 300), xlab="Time",ylab="free receptor density",type="l",lwd=2)
plot(out$time,out$DR, ylim = c(0,50), xlab="Time",ylab="drug-receptor complex",type="l",lwd=2)
plot(out$time,out$DRN, ylim = c(0,50), xlab="Time",ylab="activated receptor complex",type="l",lwd=2)
```



Om-
dat we de terugkoppelings reactie ($\cdot (DRN / (IC5_Rm + DRN))$) verwijderd hebben uit de functie (nieuwe functie is dus `Glucocorticoïd_func_2.1`) kan er geen extra receptor mRNA aangemaakt worden. Dit is ook te zien in de figuur linksbovenin, hij blijft staan op de value 4.74. Dit zal er dus gebeuren als de drugs geen effect hadden op de synthese van mRNA.

```
data <- read.csv("MPL.csv", na.strings = "NA")

medians <- aggregate(data[,c("MPL_conc", "mRNA", "Free_receptor")], list(data$dose, data$time), median, na.rm=T)
names(medians)[1:2] <- c("dose", "time")
medians_01 <- subset(medians, dose == 0.1 | dose == 0)
medians_03 <- subset(medians, dose == 0.3 | dose == 0)
```

Question 2.2

De D waarde heeft een groot effect op de verloop van de grafieken. Deze stelling kunnen we demonstreren aan de hand van de code hieronder. Wanneer de steady-state bereikt is (bij ongeveer $t = 48$) veranderen we de D waarde naar 0.

```
par(mfrow=c(2,2))
state <- c(mRNAr=4.74, R=267, DR=0, DRN=0)

# Creeer tijdsintervallen
t_1 <- seq(0, 48, by=1)
t_2 <- seq(0, 72, by=1)

# Model parameters
```

```

parameters <- c(D=20*1000/374.471, ks_Rm = 2.90, IC50_Rm = 26.2, kon = 0.00329,
               kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
               kd_Rm = 0.612, ks_r = 3.22, Rm0 = 4.74,
               DR = 0, DRN = 0
)

# call lsoda and store result in out
out1 <- deSolve::ode(y=state,times=t_1, func=Glucocorticoid_func, parms=parameters)

# Maak D = 0
parameters[1] <- 0

out2 <- deSolve::ode(y=out1[length(t_1),2:5], times=t_2, func=Glucocorticoid_func, parms=parameters)

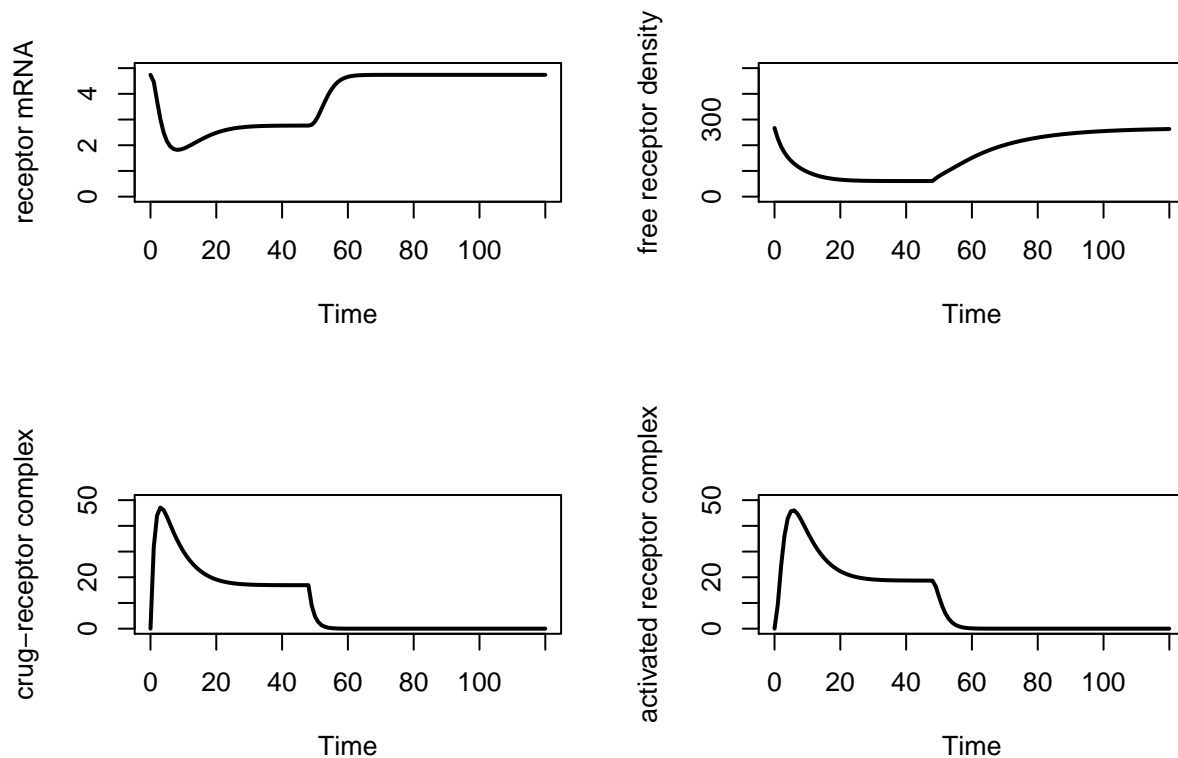
out2[,1] <- out2[,1]+t_1[length(t_1)]

out <- rbind(out1, out2)

out<-as.data.frame(out)
out$tot <- out$R + out$DR + out$DRN

plot(out$time,out$mRNAr,ylim = c(0,5), xlab="Time",ylab="receptor mRNA",type="l",lwd=2)
plot(out$time,out$R, ylim = c(0,500), xlab="Time",ylab="free receptor density",type="l",lwd=2)
plot(out$time,out$DR, ylim = c(0,50), xlab="Time",ylab="drug-receptor complex",type="l",lwd=2)
plot(out$time,out$DRN, ylim = c(0,50), xlab="Time",ylab="activated receptor complex",type="l",lwd=2)

```



Hierboven is te zien dat de D waarde inderdaad een grote invloed heeft. bij $t = 48$ keert elke lijn weer terug naar een nieuwe steady-state, dit is voor nu dus steady-state-second.

Question 2.3

```

par(mfrow = c(1,2) )
t <- seq(0, 100, by = 0.1)

parameters_normal <- c(ks_Rm = 0, IC50_Rm = 26.2, kon = 0.00329,
  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
  kd_Rm = 0.612, ks_r = 3.22, D = 53, Rm0 = 4.74,
  DR = 0, DRN = 0)

parameters_1 <- c(ks_Rm = 0, IC50_Rm = 26.2, kon = 0.00329/5,
  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
  kd_Rm = 0.612, ks_r = 3.22, D = 53, Rm0 = 4.74,
  DR = 0, DRN = 0)

parameters_2 <- c(ks_Rm = 0, IC50_Rm = 26.2, kon = 0.00329/2,
  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
  kd_Rm = 0.612, ks_r = 3.22, D = 53, Rm0 = 4.74,
  DR = 0, DRN = 0)

parameters_3 <- c(ks_Rm = 0, IC50_Rm = 26.2, kon = 0.00329*2,
  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
  kd_Rm = 0.612, ks_r = 3.22, D = 53, Rm0 = 4.74,
  DR = 0, DRN = 0)

parameters_4 <- c(ks_Rm = 0, IC50_Rm = 26.2, kon = 0.00329*5,
  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
  kd_Rm = 0.612, ks_r = 3.22, D = 53, Rm0 = 4.74,
  DR = 0, DRN = 0)

parameters_5 <- c(ks_Rm = 0, IC50_Rm = 26.2, kon = 0.00329,
  kT = 0.63, kre = 0.57/5, Rf = 0.49, kd_R = 0.0572,
  kd_Rm = 0.612, ks_r = 3.22, D = 53, Rm0 = 4.74,
  DR = 0, DRN = 0)

parameters_6 <- c(ks_Rm = 0, IC50_Rm = 26.2, kon = 0.00329,
  kT = 0.63, kre = 0.57/2, Rf = 0.49, kd_R = 0.0572,
  kd_Rm = 0.612, ks_r = 3.22, D = 53, Rm0 = 4.74,
  DR = 0, DRN = 0)

parameters_7 <- c(ks_Rm = 0, IC50_Rm = 26.2, kon = 0.00329,
  kT = 0.63, kre = 0.57*2, Rf = 0.49, kd_R = 0.0572,
  kd_Rm = 0.612, ks_r = 3.22, D = 53, Rm0 = 4.74,
  DR = 0, DRN = 0)

parameters_8 <- c(ks_Rm = 0, IC50_Rm = 26.2, kon = 0.00329,
  kT = 0.63, kre = 0.57*5, Rf = 0.49, kd_R = 0.0572,
  kd_Rm = 0.612, ks_r = 3.22, D = 53, Rm0 = 4.74,
  DR = 0, DRN = 0)

out_1 <- deSolve::ode(times = t, y = state, parms = parameters_1,
  func = Glucocorticoid_func, method = "lsoda")
out_2 <- deSolve::ode(times = t, y = state, parms = parameters_2,
  func = Glucocorticoid_func, method = "lsoda")

```



```

out_3 <- deSolve::ode(times = t, y = state, parms = parameters_3,
  func = Glucocorticoid_func, method = "lsoda")
out_4 <- deSolve::ode(times = t, y = state, parms = parameters_4,
  func = Glucocorticoid_func, method = "lsoda")

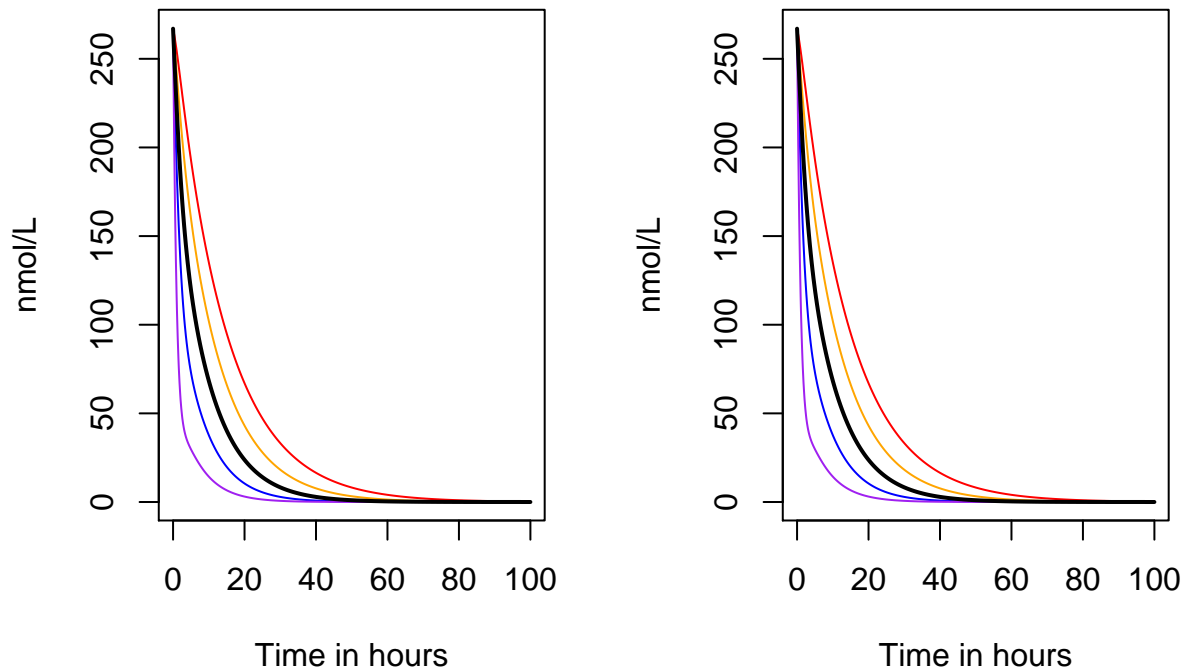
out_5 <- deSolve::ode(times = t, y = state, parms = parameters_5,
  func = Glucocorticoid_func, method = "lsoda")
out_6 <- deSolve::ode(times = t, y = state, parms = parameters_6,
  func = Glucocorticoid_func, method = "lsoda")
out_7 <- deSolve::ode(times = t, y = state, parms = parameters_7,
  func = Glucocorticoid_func, method = "lsoda")
out_8 <- deSolve::ode(times = t, y = state, parms = parameters_8,
  func = Glucocorticoid_func, method = "lsoda")
out <- deSolve::ode(times = t, y = state, parms = parameters_normal,
  func = Glucocorticoid_func, method = "lsoda")

out_1 <- as.data.frame(out_1)
out_2 <- as.data.frame(out_2)
out_3 <- as.data.frame(out_3)
out_4 <- as.data.frame(out_4)
out_5 <- as.data.frame(out_1)
out_6 <- as.data.frame(out_2)
out_7 <- as.data.frame(out_3)
out_8 <- as.data.frame(out_4)
out <- as.data.frame(out)

plot(out_1$time, out_1$R, type = "l", lwd = 1, xlab = "Time in hours", ylab = "nmol/L", col = "red")
lines(out_2$time, out_2$R, col = "orange")
lines(out_3$time, out_3$R, col = "blue")
lines(out_4$time, out_4$R, col = "purple")
lines(out$time, out$R, col = "black", lwd = 2)

plot(out_5$time, out_5$R, type = "l", lwd = 1, xlab = "Time in hours", ylab = "nmol/L", col = "red")
lines(out_6$time, out_6$R, col = "orange")
lines(out_7$time, out_7$R, col = "blue")
lines(out_8$time, out_8$R, col = "purple")
lines(out$time, out$R, col = "black", lwd = 2)

```



Question 2.4

De code hieronder genereert grafieken die een scenario uitbeeld waarbij de synthese van de receptor geblokkeerd wordt. De parameter die hiervoor veranderd moet worden is ks_r . Dit is omdat deze waarde in de 2e afgeleide de hoeveelheid receptor regelt. Als we ks_r dus op 0 zetten maken we een situatie waarbij er geen synthese meer is van de receptor.

```
par(mfrow = c(2,2) )

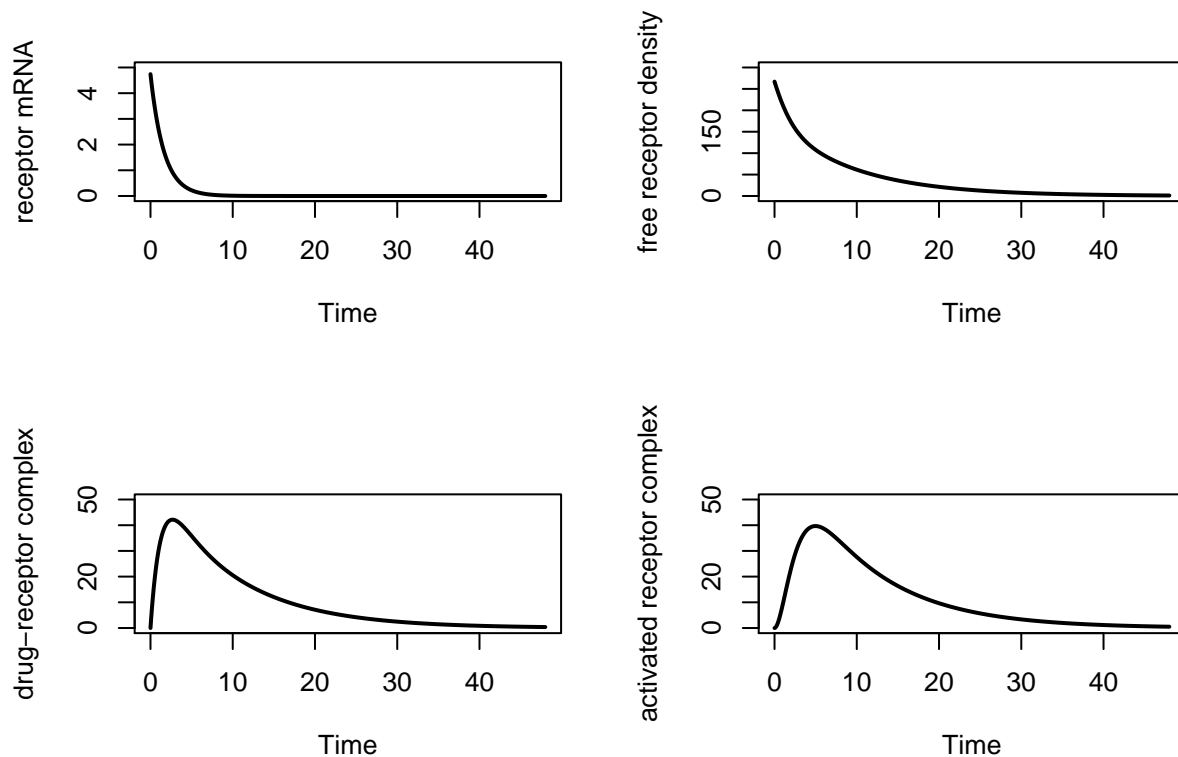
parameters <- c(ks_Rm = 0, IC50_Rm = 26.2, kon = 0.00329,
               kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
               kd_Rm = 0.612, ks_r = 0, D = 53, Rm0 = 4.74,
               DR = 0, DRN = 0)

t = seq(0, 48, by = 0.1)

out <- deSolve::ode(times = t, y = state, parms = parameters,
                   func = Glucocorticoid_func, method = "lsoda")

out <- as.data.frame(out)

plot(out$time, out$mRNAr, ylim = c(0,5), xlab="Time", ylab="receptor mRNA", type="l", lwd=2)
plot(out$time, out$R, ylim = c(0,300), xlab="Time", ylab="free receptor density", type="l", lwd=2)
plot(out$time, out$DR, ylim = c(0,50), xlab="Time", ylab="drug-receptor complex", type="l", lwd=2)
plot(out$time, out$DRN, ylim = c(0,50), xlab="Time", ylab="activated receptor complex", type="l", lwd=2)
```



Voor-
namelijk in de figuren onderaan is te zien dat de drug- en activated receptor dichtheid vrij snel daalt omdat de synthese van de receptor geblokkeerd is. Als je dit vergelijkt met de normale situatie is er zeker een verschil.

Question 2.5

In dit scenario kijken we naar wat er gebeurt als we de parameters, ks_rm en kd_rm vergroten en verkleinen. We testen dit met de volgende 'fold changes': $/2$, $/5$, $\times 2$ en $\times 5$. Hier onder is de code te zien die deze situatie tot leven brengt:

```
par(mfrow = c(2,2) )
t <- seq(0, 50, by = 0.1)

parameters_1 <- c(ks_Rm = 2.90/5, IC50_Rm = 26.2, kon = 0.00329,
                  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
                  kd_Rm = 2.9/5/4.74, ks_r = 3.22, D = 53, Rm0 = 4.74,
                  DR = 0, DRN = 0)

parameters_2 <- c(ks_Rm = 2.90/2, IC50_Rm = 26.2, kon = 0.00329,
                  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
                  kd_Rm = 2.9/2/4.74, ks_r = 3.22, D = 53, Rm0 = 4.74,
                  DR = 0, DRN = 0)

parameters_3 <- c(ks_Rm = 2.90*2, IC50_Rm = 26.2, kon = 0.00329,
                  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
                  kd_Rm = 2.9*2/4.74, ks_r = 3.22, D = 53, Rm0 = 4.74,
                  DR = 0, DRN = 0)

parameters_4 <- c(ks_Rm = 2.90*5, IC50_Rm = 26.2, kon = 0.00329,
                  kT = 0.63, kre = 0.57, Rf = 0.49, kd_R = 0.0572,
                  kd_Rm = 2.9*5/4.74, ks_r = 3.22, D = 53, Rm0 = 4.74,
                  DR = 0, DRN = 0)
```

```

out_1 <- deSolve::ode(times = t, y = state, parms = parameters_1,
  func = Glucocorticoid_func, method = "lsoda")

out_2 <- deSolve::ode(times = t, y = state, parms = parameters_2,
  func = Glucocorticoid_func, method = "lsoda")

out_3 <- deSolve::ode(times = t, y = state, parms = parameters_3,
  func = Glucocorticoid_func, method = "lsoda")

out_4 <- deSolve::ode(times = t, y = state, parms = parameters_4,
  func = Glucocorticoid_func, method = "lsoda")

out_1 <- as.data.frame(out_1)
out_2 <- as.data.frame(out_2)
out_3 <- as.data.frame(out_3)
out_4 <- as.data.frame(out_4)

plot(out_1$time, out_1$mRNAr,
  ylab=c("Receptor mRNA"), xlab=c(" Time in hours"),
  type='l', lwd = 2, xlim= c(0,50), ylim= c(5, 0))

lines(out_2$time, out_2$mRNAr, lwd = 2, col = "red")
lines(out_3$time, out_3$mRNAr, lwd = 2, col = "blue")
lines(out_4$time, out_4$mRNAr, lwd = 2, col = "purple")

plot(out_1$time, out_1$R,
  ylab=c("Free receptor density"), xlab=c(" Time in hours"),
  type='l', lwd = 2, xlim= c(0,50), ylim= c(0, 400))

lines(out_2$time, out_2$R, lwd = 2, col = "red")
lines(out_3$time, out_3$R, lwd = 2, col = "blue")
lines(out_4$time, out_4$R, lwd = 2, col = "purple")

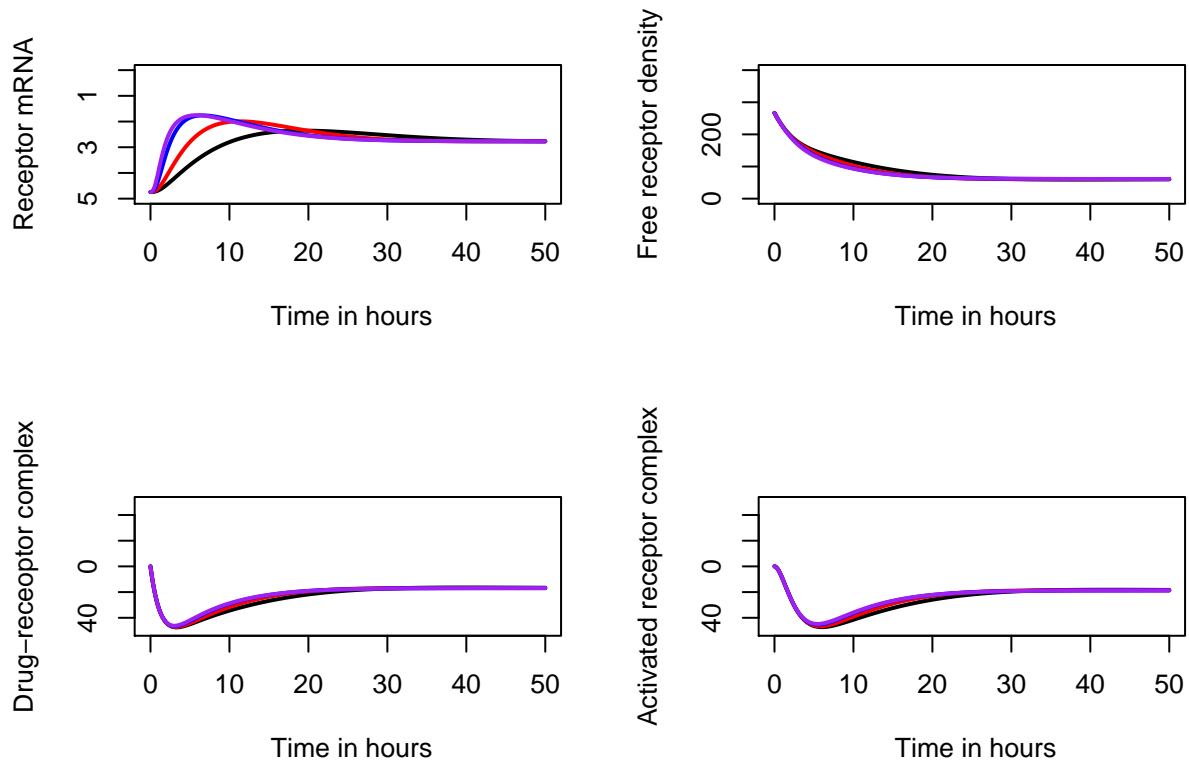
plot(out_1$time, out_1$DR,
  ylab=c("Drug-receptor complex"), xlab=c(" Time in hours"),
  type='l', lwd = 2, xlim= c(0,50), ylim= c(50, -50))

lines(out_2$time, out_2$DR, lwd = 2, col = "red")
lines(out_3$time, out_3$DR, lwd = 2, col = "blue")
lines(out_4$time, out_4$DR, lwd = 2, col = "purple")

plot(out_1$time, out_1$DRN,
  ylab=c("Activated receptor complex"), xlab=c(" Time in hours"),
  type='l', lwd = 2, xlim= c(0,50), ylim= c(50, -50))

lines(out_2$time, out_2$DRN, lwd = 2, col = "red")
lines(out_3$time, out_3$DRN, lwd = 2, col = "blue")
lines(out_4$time, out_4$DRN, lwd = 2, col = "purple")

```



In de plotjes hierboven is vooral bij de figuur linksboven een duidelijk verschil te zien tussen de fold changes. Dus de receptor mRNA hangt het meest af van de ks_rm & kd_rm parameters af in vergelijking met de andere figuren. De kleuren duiden de hoeveelheid change aan: zwart = $\times 5$, rood = $\times 2$, blauw = $/ 2$ en paars is $/ 5$.

4 Discussion and Conclusion

4.1 Discussion

- Compare your results with what is expecting from the literature and discuss differences with them.
- Discuss striking and surprising results.
- Discuss weaknesses in your research and how they could be addressed.

4.2 General conclusion and perspective

Discuss what your goal was, what the end result is and how you could continue working from here.

References

- [1] Soetaert, K., Petzoldt, T., and Woodrow Setzer, R.: *Solving differential equations in R: package deSolve*, J. Stat. Softw., 33, 1-25, 2010.

Soetaert, K., Petzoldt, T., and Woodrow Setzer, R.: *Solving differential equations in R: package deSolve*, J. Stat. Softw., 33, 1-25, 2010.