

```
##### Autoregulatory gene switching on and off #####
```

```
### switching on ###
```

```
simTime <- 500 # arbitrarily long time to ensure steady state
myModel <- new("odeModel",
main = function (time, init, parms, ...) {
x <- init
p <- parms
U <- approxTime1(inputs, time, rule = 2)["A.in"]
Occ2 <- function(x,y,k,KD)(k*x*k*y/(KD+k*x*k*y))
dx1 <- p["kt"]*(Occ2(x[2], x[2], p["k"], p["KD"])+ U/(7.5+U)) - p["kdr"]*x[1]
dx2 <- p["ks"]*x[1] - p["kdp"]*x[2]
list(c(dx1, dx2))
},
parms = c(kt=1, kdr=1, ks=1, kdp=1, k=1, KD=0.3),
times = c(from=0, to=simTime, by=0.5),
init = c("mRNA"=0, "protein"=0.0),
solver = "rk4"
)
```

```
nSims <- 101
```

```
results <- matrix(1:nSims, nrow=nSims, ncol=2)
for (i in 1:nSims) {
  Amplitude <- (i-1)/(nSims-1)
  tOff <- 20
  inputs(myModel) <- as.matrix(data.frame(time = c(0, tOff, (tOff+0.1), simTime),
A.in = c(Amplitude, Amplitude, Amplitude, Amplitude)))
  myModel <- sim(myModel)
  results[i, 1] <- out(myModel)[dim(out(myModel))[1],3] ### results takes SS
P value
}
```

```
windows(height=4,width=6)
```

```
plot.new()
xAxisRange <- (0:(nSims-1))/(nSims-1)
plot(xAxisRange, results[,1], xlab="activator", ylab="[protein]", type="p", col="cyan")
lines(xAxisRange, results[,1], type="l",col="blue", lwd=1)
```

```
### switching off ###
```

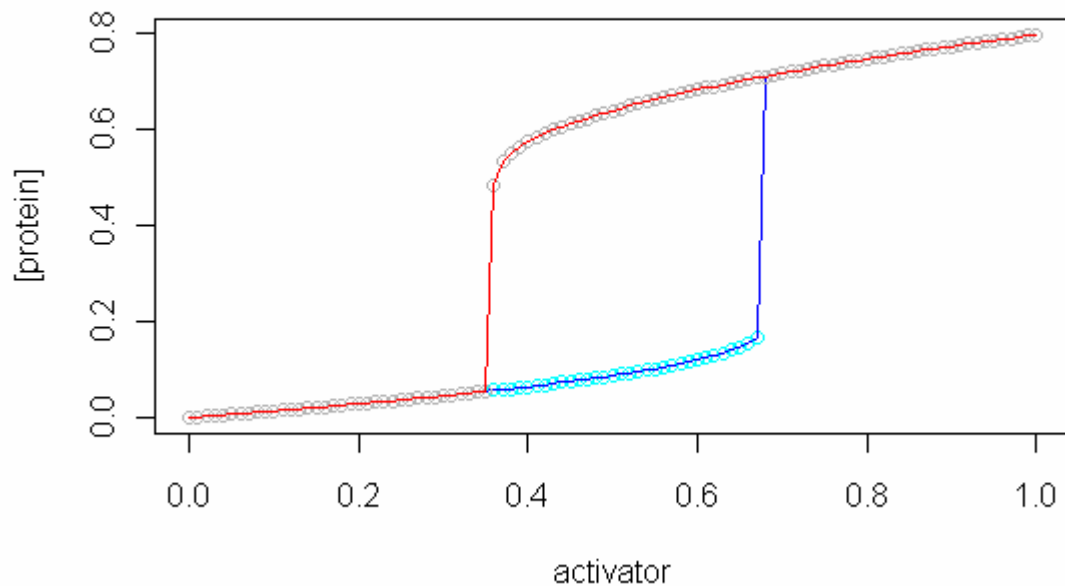
```
simTime <- 100 # switching off is faster, so shorter time is enough
times(myModel) <- c(from=0, to=simTime, by=0.5)
init(myModel) <- c("mRNA"=0, "protein"=1)
for (i in 1:nSims) {
  Amplitude <- (i-1)/(nSims-1)
  tOff <- 20
  inputs(myModel) <- as.matrix(data.frame(time = c(0, tOff, (tOff+0.1), simTime),
A.in = c(Amplitude, Amplitude, Amplitude, Amplitude)))
  myModel <- sim(myModel)
  results[i, 2] <- out(myModel)[dim(out(myModel))[1],3]
}
lines(xAxisRange, results[,2], type="p", col="gray70")
```

```
lines(xAxisRange, results[,2], type="l",col="red", lwd=1)
```

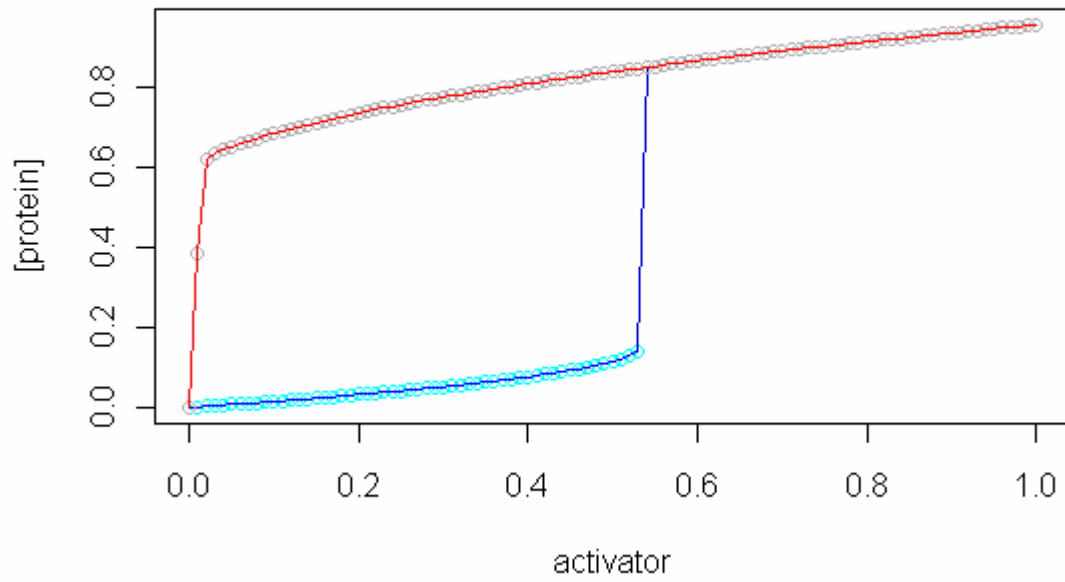
Below is the output. The red curve is for the gene starting in off state. The blue curve is for the gene starting in the on state (init protein =0.6). Note that as the activator is reduced, the gene stays on until a threshold point at ~ 0.068 (cf the turning on threshold of ~ 0.083).

Referring to my book, p249, let $\beta = 2\alpha = kt*ks/kdm*kdp$, then this model basically has only 2 independent parameters: β and KD. The only other user-specified input is the initial value of [p]. We can vary each of these by +/- N% and see how the behavior of the switch is affected.

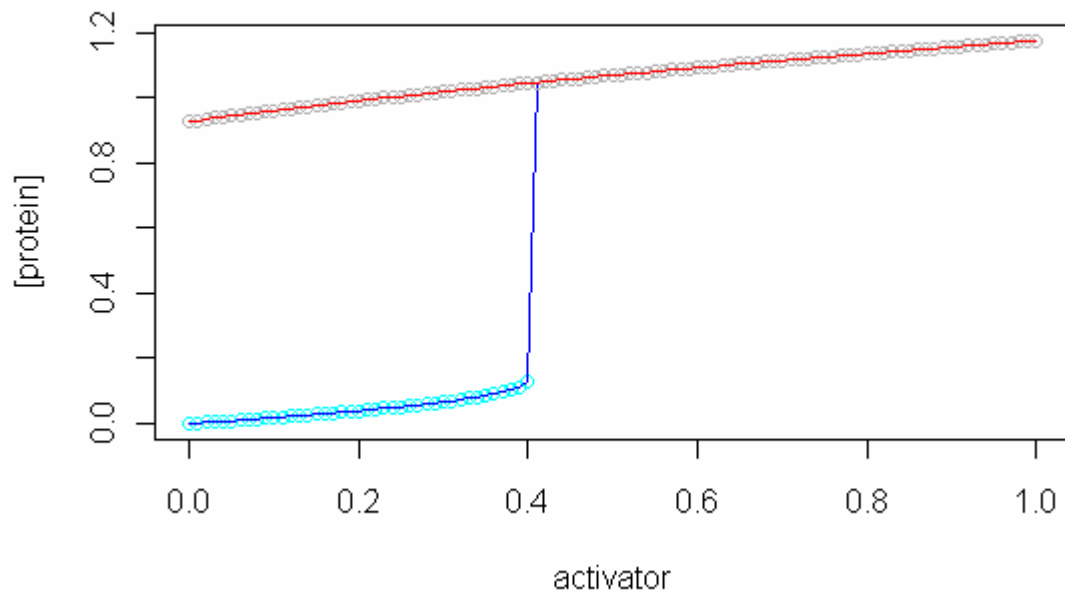
WT



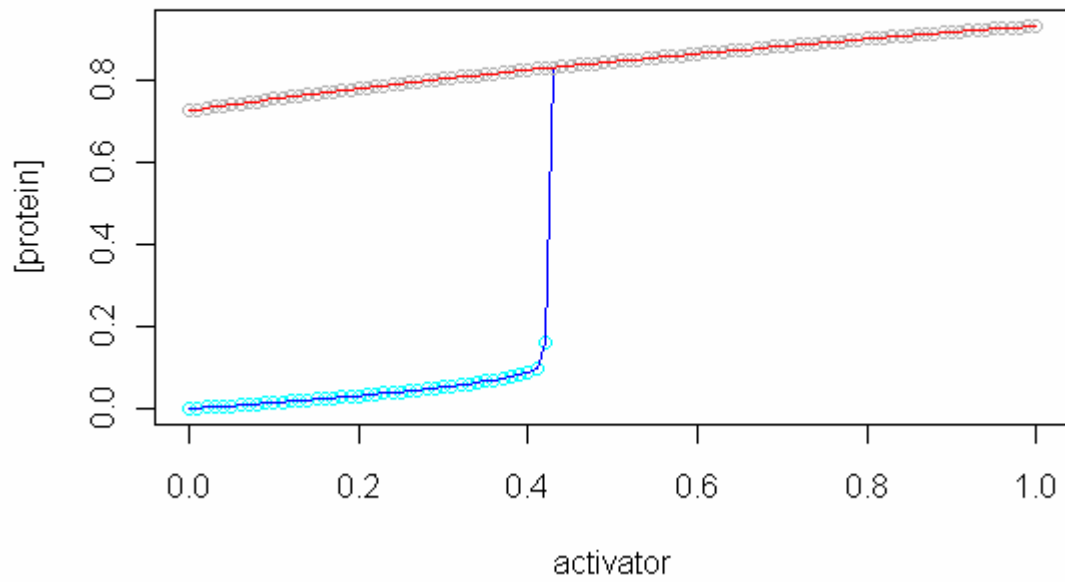
kt \rightarrow 1.1



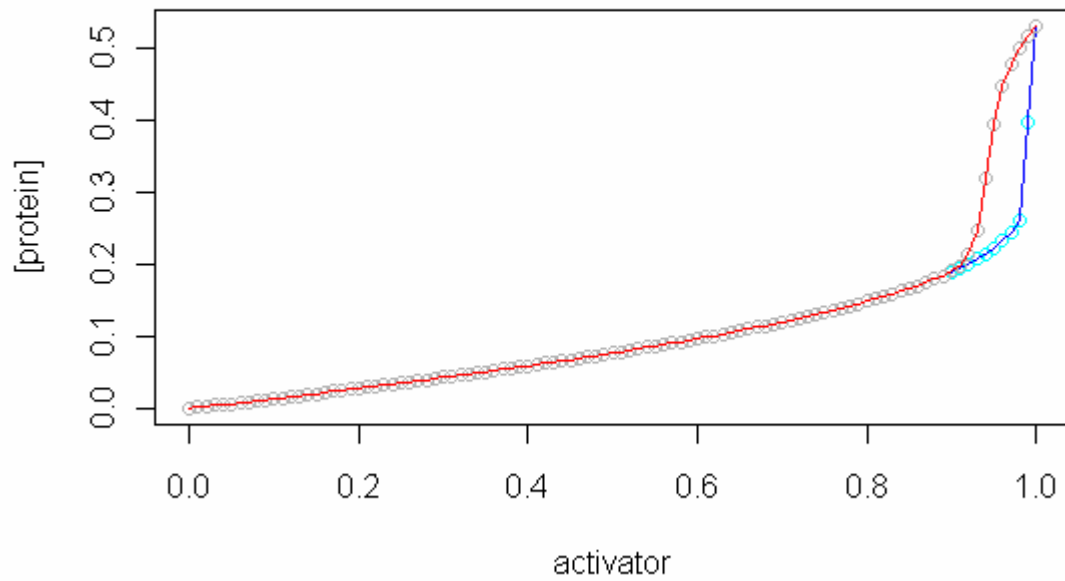
kt \rightarrow 1.25
Gene can be switched on, but not off



KD \rightarrow 0.2



KD \rightarrow 0.4



kt \rightarrow 0.9

