Estimating the turnover of granulocytes by deuterium labeling

This practical describes a few approaches to estimate the life spans of granulocytes from deuterium data. Most of the granulocytes in the blood are circulating neutrophils, and a recent paper that was based upon labeling volunteers with deuterated water has stirred up the current consensus that neutrophils are short-lived [3]. Others have criticized their interpretation, and favor the previous consensus [2]. In this practical you will repeat a similar analysis and learn that one indeed needs to be careful when fitting models to data. You will learn:

- to perform non-linear parameter estimation by fitting fairly complicated data,
- that parameter estimates may depend on the initial guess of the parameters,
- that parameters can be unidentifiable even if you have a proper fit to the data,
- and hence that it is important to choose a model that is appropriate for the data at hand.

Introduction. Neutrophils are the most abundant type of granulocytes in the blood and the most abundant type of white blood cells in most mammals. They form an essential part of the immune system. They are formed by cell division of precursors cells in the bone marrow. After a few days in a "post-mitotic" pool they emigrate as non-dividing cells into the circulation as mature neutrophils. A simple mathematical model would be

$$\frac{\mathrm{d}P}{\mathrm{d}t} = \sigma - d_P P \quad \text{and} \quad \frac{\mathrm{d}G}{\mathrm{d}t} = \alpha d_P P(t-\Delta) - d_G G \ , \tag{1}$$

where P and G are the precursors in the post-mitotic pool, and the mature neutrophils in the blood, respectively, with turnover rates d_P and d_G, and where $0 < \alpha \leq 1$. The σ parameter describes the source of dividing precursors into the post-mitotic pool. Since we typically label people that are at steady state, one can set $\mathrm{d}P/\mathrm{d}t = \mathrm{d}G/\mathrm{d}t = 0$, and find that $\bar{P} = \sigma/d_P$ and $\bar{G} = \alpha\sigma/d_G$.

Volunteers and patients can drink heavy water (^2H_2O) for weeks. During the labeling period deuterium is build into the newly synthesized DNA strands of cells that divide. Labeled DNA strands will subsequently disappear by cell death. We will fit data of five volunteers whom have been drinking a glass of 4% deuterated water for nine weeks [8]. Because the fraction of heavy water in body water is similar to that in urine, one typically measures the deuterium enrichment in the urine of the volunteers. Since deuterium can be incorporated at seven positions of the adenosine moiety that is analyzed by the gas chromatography mass spectrometry (GC-MS), the enrichment of this moiety is expected to exceed that of the body water. This introduces an additional parameter known as the amplification factor, c, that we need to estimate. This is achieved by co-fitting the enrichment in the plasma (urine) with that of a fast population like granulocytes [8].

Parameter estimation with Grind. Today you will work with an R-script called Grind that is a wrapper around the R-packages `deSolve`, `FME` and `rootSolve` developed by Karline Soetaert and colleagues [4–7]. These packages allow one to solve differential equations, find their steady state, and perform nonlinear parameter estimation. Today you only need three of Grind's easy-to-use functions:

- `run()` integrates a model numerically and provides a time plot or a trajectory in the phase plane,
- `fit()` fits a model to data by estimating its parameters, and depicts the result in a timeplot.
- `timePlot()` plots a data frame having time as the first column.

The `run()` function calls `ode()` from the `deSolve` library, and the `fit()` function calls `modFit` from the `FME` library. For instance, typing `?modFit`, provides help on the `modFit()` function. The full manual of `grind.R` is available on the website `http://tbb.bio.uu.nl/rdb/practicals/grindR/`.

We will work in the RStudio environment. You will need to install the three Soetaert libraries `deSolve`, `FME` and `rootSolve`, by using `Install Packages` in the `Tools` menu of RStudio. All documents can be found on the webpage `http://tbb.bio.uu.nl/rdb/practicals/Paris2020`. Download the R-scripts `grind.R`, `granulocytes.R`, and `tcells.R`, and store them in a local directory. Later you can open them via the `File` menu. Download the data file `Vrisekoop_pnas08.csv`, and store it in the

same folder. Set the working directory of RStudio to the folder where your R-codes and data are stored (`Set working directory` in the `Session` menu of RStudio). Files will then be opened and saved in that directory.

First "Source" the `grind.R` file (button in right hand top corner) to define the Grind functions. Run the script `granulocytes.R` line-by-line to define the various models, and a few convenient functions (button in right hand top corner). The fitting starts after the line `Here the session starts`. Use "Run" or "Control Enter" to execute lines from the code panel. In the R-console one can type and call any function in R.

Urine. For the availability of deuterium in the urine we start with a model for the turnover of body water, $\mathrm{d}W/\mathrm{d}t = s - \delta W$, where W is in liters and δ per day. The steady state, $\bar{W} = s/\delta$ is the total amount of body water. When one drinks both normal, W, and deuterated, H, water this becomes

$$\frac{\mathrm{d}W}{\mathrm{d}t} = (1-f)s - \delta W \quad \text{and} \quad \frac{\mathrm{d}H}{\mathrm{d}t} = fs - \delta H \ ,$$

where f is the fraction of D_2O in the total daily consumption. Next define the *fraction* of deuterated water in the body, $u = H/(H+W) = H/\bar{W} = H\frac{\delta}{s}$ (given that the total amount remains unchanged). Since $H = \frac{s}{\delta}u$ we write

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{\delta}{s}\frac{\mathrm{d}H}{\mathrm{d}t} = \frac{\delta}{s}\Big(fs - \delta H\Big) = \frac{\delta}{s}\Big(fs - \delta\ \frac{s}{\delta}u\Big) = \delta(f-u) \ .$$

Note that s has disappeared, that this has only one parameter (the turnover rate δ). Note that $f > 0$ in the labeling period, and that $f = 0$ in the de-labeling period. This ODE is provided by the function `urine()` in the accompanying R documents.

This model is linear and the solution is $u(t) = f(1 - \mathrm{e}^{-\delta t})$, i.e., after a long labeling period this approaches the expected $\bar{u} = f$. If the experiment starts with a bolus of deuterated water on the first day, one can just add an exponential decay term to the model,

$$u(t) = f(1 - \mathrm{e}^{-\delta t}) + u(0)\mathrm{e}^{-\delta t} \ .$$

One can model the de-labeling phase by setting $f = 0$ at time τ, after which the same equation describes exponential decay starting at the of the labeling phase,

$$u(t) = u(\tau)\mathrm{e}^{-\delta(t-\tau)} \ .$$

This solution is provided by the function `urineSol()` in the accompanying R documents.

General model for a population of cells accumulating deuterium in their DNA. To get started, consider a population of cells with a source and a division rate, e.g., $\mathrm{d}N/\mathrm{d}t = \sigma + (p-d)N$, where $p < d$ and $\bar{N} = \frac{\sigma}{d-p}$ is the steady state population size. Assuming that source consists of dividing cells, the total amount of labeled, L, and unlabeled DNA, U, in the cells would be given by

$$\frac{\mathrm{d}L}{\mathrm{d}t} = cu(t)\sigma + cu(t)p(L+U) - dL \quad \text{and} \quad \frac{\mathrm{d}U}{\mathrm{d}t} = (1-cu(t))\sigma + (1-cu(t))p(L+U) - dU \ ,$$

where c is the amplification factor reflecting the "efficiency" with which dividing cells incorporate deuterium into their DNA. Note that the $(1 - cu(t))$ terms demand that $cu(t) < 1$ (which is typically no problem because $u(t)$ is small). Rewriting $\mathrm{d}L/\mathrm{d}t$ into the fraction of labeled DNA

$$\frac{\mathrm{d}l}{\mathrm{d}t} = \frac{d-p}{\sigma}\ \frac{\mathrm{d}L}{\mathrm{d}t} = \frac{d-p}{\sigma}\Big(cu(t)\sigma + cu(t)p(L+U) - dL\Big) =$$

$$\frac{d-p}{\sigma}\Big(cu(t)\sigma + cu(t)p\ \frac{\sigma}{d-p} - d\ \frac{\sigma}{d-p}\ l\Big) = cu(t)(d-p) + cu(t)p - dl = d(cu(t) - l) \ ,$$

we end up with a single parameter model. Both σ and p have disappeared, and the only thing we can get from the data is the turnover rate d. A sanity check of a long labeling experiment readily confirms that $l(\infty) = cf$. This equation is provided by the function `cells()` in the accompanying R documents. Fortuitously, this model can be used both for populations maintained entirely by the source, i.e., $\mathrm{d}T/\mathrm{d}t = \sigma - dT$, and for populations maintained entirely by self renewal, i.e., $\mathrm{d}T/\mathrm{d}t = (p-d)T$. In both cases the turnover rate, d, is the only identifiable parameter [1].

A simple chain model for neutrophils. Since Eq. (1) is a chain of two equations with a time delay, we next derive the labeling equations for such a chain. For the amounts of labeled DNA of bone marrow precursors, L_P, and mature neutrophils, L_N, we write

$$\frac{\mathrm{d}L_P}{\mathrm{d}t} = \sigma c u(t) - d_P L_P \quad \text{and} \quad \frac{\mathrm{d}L_G}{\mathrm{d}t} = \alpha d_P L_P(t-\Delta) - d_G L_G \ ,$$

and for the fraction of labeled DNA

$$\frac{\mathrm{d}l_P}{\mathrm{d}t} = \frac{d_P}{\sigma}\frac{\mathrm{d}L_P}{\mathrm{d}t} = \frac{d_P}{\sigma}\Big(\sigma c u(t) - d_P L_P\Big) = \frac{d_P}{\sigma}\Big(\sigma c u(t) - d_P \frac{\sigma}{d_P} l_P\Big) = d_P(cu(t) - l_P)$$

and

$$\frac{\mathrm{d}l_G}{\mathrm{d}t} = \frac{d_G}{\alpha\sigma}\frac{\mathrm{d}L_G}{\mathrm{d}t} = \frac{d_G}{\alpha\sigma}\Big(\alpha d_P \frac{\sigma}{d_P} l_P(t-\Delta) - d_G \frac{\alpha\sigma}{d_G} l_G\Big) = d_G(l_P(t-\Delta) - l_G) \ ,$$

where α disappeared. One can perform a sanity check by considering an infinite labeling experiment: $l_P(\infty) \to cf$ and $l_G(\infty) \to l_P$. Again we lose σ, and each equation only contains its own turnover rate. The last equation also shows that the fraction of label in the circulating cells will start to go up when $t = \Delta$, after which $l_P(t-\Delta) > l_G(t)$, and will start to decline when $l_P(t-\Delta) = l_G(t)$. Finally, note that the maximum upslope of the circulating cells is d_G, which is only achieved when the precursors get labeled infinitely fast.

Estimating the turnover rate of neutrophils or granulocytes in the blood. The Vrisekoop *et al.* [8] paper provides data fraction labeled DNA of granulocytes in five healthy volunteers. For obvious ethical reasons they have no data on the degree of labeling of their precursors in the bone marrow. Since the model for the fraction of labeled DNA in neutrophils has four parameters, c, d_P, d_G and Δ, we could start by just estimating all parameters from the data. For fitting we need an initial guess for these parameters. For the simple $cu(t)$ term we could start with $c = 5$ (or a less because not all hydrogens are equally replaceable) The time delay typically falls in a range of several days to a week, making $\Delta = 6$ days a reasonable initial guess. The turnover of precursors in the bone marrow is not known. A reasonable fraction (20%) of precursor cells expresses markers of cell division (like Ki67), but it is difficult to translate the expression of a marker into a rate. Pillay *et al.* [3] conclude that neutrophils have an average life span (or residence time) of five days, making $d_G = 0.25\mathrm{d}^{-1}$ a reasonable guess. Lahoz-Beneytez *et al.* [2] argue that mature neutrophils are short lived, $d_G \geq 1\mathrm{d}^{-1}$, and that the slow time scale in the data is due to slow turnover in the bone marrow, making $d_P = 0.25\mathrm{d}^{-1}$ a reasonable guess. Since we do not know we need to consider both, and take $c = 5, d_P = 5, d_G = 0.25, \Delta = 6$ and $c = 5, d_P = 0.25, d_G = 5, \Delta = 6$ as alternative initial guesses.

Questions:

1. Fit the `urine()` function to the data from volunteer one. What is the turnover rate of body water per day? How confident are you of this estimate? Check the standard errors provided when calling `summary(fit)`.
2. Which parameter estimates for the neutrophils and their precursors do you obtain for the two initial guesses?
3. Is one fit better than the other?
4. Is that any different for the four other volunteers?
5. What can you conclude expected life span of neutrophils?

Estimating the amplification factor. Although we have learned in the previous section that labeling data of mature neutrophils remains insufficient for estimating their turnover rate, the same

data do allow us to estimate the amplification factor c. This is important because we need to know c for estimating the life spans of other cell types in the blood [8].

Above we saw that the two turnover rates of the complete model,

$$\frac{\mathrm{d}l_P}{\mathrm{d}t} = d_p(cu(t) - l_P) \quad \text{and} \quad \frac{\mathrm{d}l_G}{\mathrm{d}t} = d_G(l_P(t-\Delta) - l_G) \ ,$$

cannot be identified from labeling data on peripheral neutrophils, l_G, only. The weekly time scale we observe in the data can be explained when the bone marrow is slow, i.e., $d_P \simeq 0.2\mathrm{d}^{-1}$, and the periphery is fast, i.e., $d_G \gg d_P$, and when the bone marrow is fast, i.e., $d_P \gg d_G$, and the blood is slow, i.e., $d_G \simeq 0.2\mathrm{d}^{-1}$. Since we still would like to estimate the amplification factor, c, from this data, and because it is cumbersome to work with a DDE model with unidentifiable parameters, we simplify the model using this separation of the two time scales, i.e., by making a quasi steady state assumption for the fast population.

For the case $d_G \gg d_P$ we assume $\mathrm{d}l_G/\mathrm{d}t = 0$ to obtain $l_G = l_P(t-\Delta)$, which make intuitive sense because the labeling of the neutrophils in the circulation would just reflect that of their precursors in the bone marrow Δ days earlier. When $l_G = l_p(t-\Delta)$ we can also write $\mathrm{d}l_G/\mathrm{d}t = \mathrm{d}l_P(t-\Delta)/\mathrm{d}t$, and because

$$\frac{\mathrm{d}l_P(t-\Delta)}{\mathrm{d}t} = d_p cu(t-\Delta) - d_p l_P(t-\Delta) \ ,$$

we obtain

$$\frac{\mathrm{d}l_G}{\mathrm{d}t} = d_p cu(t-\Delta) - d_p l_P(t-\Delta) = d_p cu(t-\Delta) - d_p l_G \ ,$$

which is a single ODE with just three parameters, d_p, Δ and c. Similarly for the case $d_P \gg d_G$ we assume $\mathrm{d}l_P/\mathrm{d}t = 0$ to obtain $l_P = cu(t)$, which by substituting the equivalent $l_P(t-\Delta) = cu(t-\Delta)$ into $\mathrm{d}l_G/\mathrm{d}t$ gives

$$\frac{\mathrm{d}l_G}{\mathrm{d}t} = d_G cu(t-\Delta) - d_G l_G \ ,$$

which is the same ODE with just three parameters. Thus, when the time scales in the bone marrow and blood differ sufficiently, this would be an appropriate model for describing the labeling of mature neutrophils in the circulation, that would allow us to estimate c in a reliable and robust manner. Finally, because we have an analytic solution for $u(t)$ we can compute $u(t-\Delta)$ without having to solve a DDE (see the function `simple()` in the R-script). The standard sanity check by considering an infinite labeling experiment readily confirms that $l_G \to fc$.

Questions:

1. Fit the neutrophil data with the simplified model and the full model to test whether or not the simplified model suffices to explain the data. Do you observe a visual difference in the quality of the fit? Is the SSR different? What does the F-test suggest about the most appropriate model?
2. One can fit all data together by assuming that people only differ in the amount of fluids that they drink. What do you estimate for the amplification factor of the five volunteers? Keep these five values because you will need them in the next practical where you will estimate the turnover rates of naive and memory T cells.
3. Since the predicted asymptote of the neutrophils in an infinite labeling experiment is fc, it may be difficult to estimate c and f independently. Plot c as a function of f for the five cs and fs you estimate for the five volunteers.

Estimating T cell turnover. Now that you have a reasonable value for c for every volunteer you can proceed with the next practical and estimate the turnover rates of their naive and memory T cells. If you have time open the document `tcells.pdf` and proceed.

February 27, 2020, Rob J. de Boer

References

[1] **De Boer, R. J. and Perelson, A. S.**, 2013. Quantifying T lymphocyte turnover. J. theor. Biol. **327**:45–87.

[2] **Lahoz-Beneytez, J., Elemans, M., Zhang, Y., Ahmed, R., Salam, A., Block, M., Niederalt, C., Asquith, B., and Macallan, D.**, 2016. Human neutrophil kinetics: modeling of stable isotope labeling data supports short blood neutrophil half-lives. Blood **127**:3431–3438.

[3] **Pillay, J., Den Braber, I., Vrisekoop, N., Kwast, L. M., De Boer, R. J., Borghans, J. A., Tesselaar, K., and Koenderman, L.**, 2010. In vivo labeling with ^2H_2O reveals a human neutrophil lifespan of 5.4 days. Blood **116**:625–627.

[4] **Soetaert, K.**, 2009. rootSolve: Nonlinear root finding, equilibrium and steady-state analysis of ordinary differential equations. R package 1.6.

[5] **Soetaert, K. and Herman, P. M.**, 2009. A Practical Guide to Ecological Modelling. Using R as a Simulation Platform. Springer. ISBN 978-1-4020-8623-6.

[6] **Soetaert, K. and Petzoldt, T.**, 2010. Inverse modelling, sensitivity and Monte Carlo analysis in R using package FME. Journal of Statistical Software **33**:1–28.

[7] **Soetaert, K., Petzoldt, T., and Setzer, R. W.**, 2010. Solving differential equations in R: Package deSolve. Journal of Statistical Software **33**:1–25.

[8] **Vrisekoop, N., Den Braber, I., De Boer, A. B., Ruiter, A. F., Ackermans, M. T., Van der Crabben, S. N., Schrijver, E. H., Spierenburg, G., Sauerwein, H. P., Hazenberg, M. D., De Boer, R. J., Miedema, F., Borghans, J. A., and Tesselaar, K.**, 2008. Sparse production but preferential incorporation of recently produced naive T cells in the human peripheral pool. Proc. Natl. Acad. Sci. U.S.A. **105**:6115–6120.

granulocytes.R

```
urine <- function (t, state, parms) {
  with(as.list(c(state,parms)), {
    dtU <- ifelse(t < tau, delta*(f - U), -delta*U)
    return(list(dtU))
  })
}

urineSol <- function(t, state, parms) { # state is a dummy parameter
  with(as.list(parms), {
    U <- ifelse(t<0, 0, ifelse(t<tau, f*(1-exp(-delta*t))+u0*exp(-delta*t),
      (f*(1-exp(-delta*tau))+u0*exp(-delta*tau))*exp(-delta*(t-tau))))
    return(U)
  })
}

granulos <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    lagP <- ifelse(t-Delta<0, 0, lagvalue(t-Delta,2))
    dtU <- ifelse(t < tau, delta*(f - U), -delta*U)
    dtP <- dP*(c*U - P)
    dtG <- dG*(lagP - G)
    return(list(c(dtU,dtP,dtG)))
  })
}

cells <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    dtU <- ifelse(t < tau, delta*(f - U), -delta*U)
    dtG <- dG*(c*U - G)
    return(list(c(dtU,dtG)))
  })
}

simple <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    lagU <- urineSol(t-Delta,NULL,parms=parms) # Use the solution for the delay
    dtG <- dG*(c*lagU - G)
    return(list(dtG))
  })
}
```

```
asinsqrt <- function(x) return(asin(sqrt(x))) # Transformation function

ftest=function(ssr1, p1, ssr2, p2, n) {
  if (p2 > p1) {
    df1 <- p2-p1
    df2 <- n-p2
    f <- ((ssr1-ssr2)/df1)/(ssr2/df2)
    cat("F[",df1,",",df2,"] = ", f, ": P = ",1-pf(f,df1,df2),"\n")
  } else
    ftest(ssr2,p2,ssr1,p1,n)
}

myfit <- function(data, model, state, free, ...) {
  return(fit(datas=data,odes=model,state=state,free=free,fun=asinsqrt,lower=0,arrest="tau",atol=1e-12,rtol=1e
      -12,...))
}

select <- function(i) {
  person <- subset(rawdata,rawdata$patient == i)
  person <- person[(person$time < 200),] # Use early time points only
  return(person[2:4])
}

opar <- par();par(mar=c(2.6,2.6,1.6,0.2),mgp=c(1.5,0.5,0)) # for better margins

# Here the session starts:
rawdata <- read.csv("Vrisekoop_pnas08.csv")
for (i in seq(4,8)) # Set negative values to zero
  rawdata[,i] <- sapply(rawdata[,i],max,0)
data <- lapply(seq(5),select) # Urine & granulos from 5 persons

# Example of a model generating a urine curve
p <- c(f=1,delta=0.05,tau=63)
run(100,odes=urine,state=c(U=0.1))

id <- 1 # Select a person
# Good guesses for the urine parameters
p <- c(u0=0.1,f=0.018,delta=0.05,tau=63)
free <- c("u0","f","delta")
fit0 <- myfit(data[[id]][,1:2],urineSol,c(U=0),free,solution=TRUE)
summary(fit0)
p[free] <- fit0$par; P <- p; U0 <- as.numeric(fit0$par["u0"])

# Add first guess for neutrophil parameters and fit the Granulocyte model
p <- c(P,c=5,dP=0.25,dG=5,Delta=6);p
free <- c("c","dP","dG","Delta")
fit1 <- myfit(data[[id]],granulos,state=c(U=U0,P=0,G=0),free,delay=TRUE)
summary(fit1)
# Second guess
p <- c(P,c=5,dP=5,dG=0.25,Delta=6);p
fit2 <- myfit(data[[id]],granulos,state=c(U=U0,P=0,G=0),free,delay=TRUE)
summary(fit2)

print(fit1$par);print(fit2$par)
print(c(fit1$ssr,fit2$ssr))

# Next fit the Quasi steady state model
p <- c(P,c=5,dG=0.25,Delta=6);p
free <- c("c","dG","Delta")
fit3 <- myfit(data[[id]],simple,state=c(G=0),free,tweak="nsol$U<-urineSol(times,state,parms)")
summary(fit3)
print(c(fit1$ssr,fit2$ssr,fit3$ssr))

ftest(fit1$ssr,length(fit1$par),fit3$ssr,length(fit3$par),nrow(data[[id]]))
ftest(fit2$ssr,length(fit2$par),fit3$ssr,length(fit3$par),nrow(data[[id]]))
```

```
# Fit all data together allowing only c and the urine parameters to differ between persons
differ <- c("c","u0","f","delta")
par(mfrow=c(3,2))
fit4 <- myfit(data,simple,state=c(G=0),free,differ=differ,tweak="nsol$U<-urineSol(times,state,parms)",main=seq(5)
    )
cValues <- fit4$par[names(fit4$par)=="c"]
fValues <- fit4$par[names(fit4$par)=="f"]
print(c(CV=sd(cValues)/mean(cValues)))
plot(fValues,cValues,main=round(fit4$ssr,digits=3))
par(mfrow=c(1,1))
print(fit4$par)
```