Estimating the turnover of T cells by deuterium labeling

This practical describes a few approaches to estimate the life spans of lymphocytes from deuterium labeling data. To perform this practical, first make the practical on estimating the amplification factor (see the accompanying document `granulocytes.pdf`). This text is based upon the Appendix of the paper of Vrisekoop *et al.* [6] and the review of De Boer and Perelson [4]. You will learn

- how to perform non-linear parameter estimation by fitting fairly complicated data
- that T cell memory is maintained by short lived cells, and that naive T cells are extremely long-lived,
- that the selection of the most appropriate model is of utmost importance because different models can give different estimates,
- and hence that quantification of cellular population dynamics is far from trivial.

Introduction. In the accompanying practical we have derived a general model for the deuterium labeling of a population of cells that is maintained by a source, σ, by self-renewal at a rate, p, and has a loss rate, d. The equations for the fraction of labeled DNA in the population, $l(t)$, and the fraction of deuterium in the body water, $u(t)$, are the elegantly simple

$$\frac{\mathrm{d}l}{\mathrm{d}t} = d(cu(t) - l) \quad \text{and} \quad \frac{\mathrm{d}u}{\mathrm{d}t} = \delta(f - u) \ ,$$

respectively. Here c is the amplification factor, δ the turnover rate of the body water, and f the fraction of deuterium in the daily consumption. For the cells, this is a model with a single exponential which should hence be sufficient for kinetically homogenous populations. It is available as the model `model()` in the R-script `tcells.R`.

Kinetic heterogeneity. A popular model to describe deuterium labeling data of kinetically heterogeneous populations was proposed almost 20 years ago by Becca Asquith and her co-workers [1]. They argue that in kinetically heterogeneous populations the label should accumulate faster in subpopulations that are turning over more rapidly. The initial up-slope during the labeling phase would then reflect the average turnover rate, p, but the labeled cells should be enriched in fast cells, and die at a rate d^* that is faster than p. The modeling started with a labeling equation that was defined in terms of the total amount of deoxyadenosine, A, in the population of interest (which remains constant). Since labeled deoxyadenosine, A^*, increases in the population when cells divide, they wrote something like $\mathrm{d}A^*/\mathrm{d}t = cu(t)pA - d^*A^*$, where p is the division rate, and $d^* \geq p$ is the loss rate of labeled cells. After dividing by A this delivered, $l(t)$, for fraction labeled DNA, A^*/A, in the labeling phase

$$\frac{\mathrm{d}l}{\mathrm{d}t} = pcu(t) - d^*l \ ,$$

which has an initial up-slope, p, a down-slope, d^*, and approaches the asymptote pcf/d^* in an infinite labeling experiment. The latter is lower than the maximum labeling, cf, approached by the granulocytes, because $p < d^*$. However, the degree of labeling should become more and more uniform over all subpopulations over time, i.e., $p \to d^*$ in an infinite labeling experiment. This nicely accounts for the fact that the down-slope of labeling experiments decreases with the length of the labeling phase [1, 2]. This model is available as the function `pdstar()` in the R-script `tcells.R`.

Multi-compartment model. A more mechanistic model to describe kinetically heterogeneous populations was proposed by Ganusov *et al.* [5], and basically generalizes the general model into the sum of several sub-populations with different turnover rates d_i, and where $L = \sum_i^n \alpha_i L_i$. Here α_i is the fractional size of sub-population i, and the equation for the fraction labeled DNA, l_i, obeys the very similar

$$\frac{\mathrm{d}l_i}{\mathrm{d}t} = d_i(cu(t) - l_i) \ .$$

The basic procedure to fit this model to deuterium data is to start with one sub-population, $n = 1$, and increase the number of compartments, n, until the best fit is obtained or some average turnover rate is

approached [5, 7]. (What is best should be defined statistically by the F-test, as the quality of the fit will always improve when the number of parameters is increased). A two compartment version of this model is provided by the function `model2()` in the `tcells.R` script. The average turnover rate of this model is defined as $\bar{d} = \sum_i^n \alpha_i d_i$, and $\bar{d}$ is fortunately much more identifiable than the estimates of the individual α_i and d_i parameters [3, 5, 7]. The average life span of the cells in the entire population is defined as $1/\bar{d}$. Finally, note these models are "nested", as one can define the general one-parameter model by setting $n = 1$ and $\alpha_1 = 1$, the two-parameter Asquith model by setting $n = 2$, estimating d_1 and α_1 while restricting $d_2 = 0$, and allow for a three-parameter model by also estimating d_2, and so on.

Estimating the half lives of T cell subsets in the blood of young healthy volunteers. The data set that you worked on in the previous practical has 4 additional columns, `N4`, `M4`, `N8`, and `M8`, for naive and memory $CD4^+$ and $CD8^+$ T cells, respectively. Leave your previous R-session open such that the estimated c-values are kept, and open the R-script `tcells.R` in another tab. This script defines the three functions explained above. The functions define ODEs computing U and L, where L is just a generic name for the fraction labeled DNA of any T cell subset. Run the first part of the script to define the models. After the line `Here the session starts` you select a T cell subset, and pick an identifier for a volunteer, and fit that data to the three models. Compare your fits and estimates to those in Figure 2 and Tables 1 and 2 of Vrisekoop *et al.* [6] (note that half lives of for all volunteers are stored in a list `Thalf`):

1. Fit the `N4` data of every volunteer to each of the three models. Which model describes the data best (use the `ftest()` function)? Do the estimates depend on the assumptions of the model? What is your best estimate for the half life of naive $CD4^+$ T cells?
2. In some of the fits of the Kinetic heterogeneity model the estimated p is larger than the estimated d^*. The authors therefore concluded that newly produced cells have better survival. What do you think of this?
3. Fit the `M4` data of every volunteer to each of the three models. Which model describes the data best? Do the estimates depend on the assumptions of the model? What is your best estimate for the half life of memory $CD4^+$ T cells?
4. If you have time repeat all of this for the $CD8^+$ T cells.
5. How confident are you of these estimates: check the standard errors reported by `fit` (and/or perform bootstrapping)?
6. How is life-long T-cell memory maintained?
7. The average life span of the multi-compartment model can be defined in two ways: as the inverse of the average turnover rate, $1/\bar{d} = 1/\sum_i^n \alpha_i d_i$, or as the average of the various life spans $\sum_i^n \alpha_i/d_i$. Can this be very different, and –if so– what do you think is best? What happens if you compare the estimated life spans, rather than the half lives $1/\bar{d}$, for the naive T cells between the models?

March 1, 2020, Rob J. de Boer

References

[1] **Asquith, B., Debacq, C., Macallan, D. C., Willems, L., and Bangham, C. R.**, 2002. Lymphocyte kinetics: the interpretation of labelling data. Trends Immunol. **23**:596–601.

[2] **Borghans, J. A. M. and De Boer, R. J.**, 2007. Quantification of T-cell dynamics: From telomeres to DNA labelling. Immunol. Reviews **216**:35–47.

[3] **De Boer, R. J., Mohri, H., Ho, D. D., and Perelson, A. S.**, 2003. Turnover rates of B cells, T cells, and NK cells in simian immunodeficiency virus-infected and uninfected rhesus macaques. J. Immunol. **170**:2479–2487.

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

[5] **Ganusov, V. V., Borghans, J. A., and De Boer, R. J.**, 2010. Explicit kinetic heterogeneity: mathematical models for interpretation of deuterium labeling of heterogeneous cell populations. PLoS Comput. Biol. **6**:e1000666.

[6] **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.

[7] **Westera, L., Drylewicz, J., Den Braber, I., Mugwagwa, T., Van der Maas, I., Kwast, L., Volman, T., Van de Weg-Schrijver, E. H., Bartha, I., Spierenburg, G., Gaiser, K., Ackermans, M. T., Asquith, B., De Boer, R. J., Tesselaar, K., and Borghans, J. A.**, 2013. Closing the gap between T-cell life span estimates from stable isotope-labeling studies in mice and humans. Blood **122**:2205–2212.

tcells.R

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

pdstar <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    dtU <- ifelse(t < tau, delta*(f - U), -delta*U)
    dtL <- p*c*U - d*L
    return(list(c(dtU,dtL)))
  })
}

model2 <- function (t, state, parms) {
  with(as.list(c(state,parms)), {
    d1 <- max(0,(d + (alpha-1)*d2)/alpha)
    dtU <- ifelse(t < tau, delta*(f - U), -delta*U)
    dtL1 <- d1*(c*U - L1)
    dtL2 <- d2*(c*U - L2)
    return(list(c(dtU,dtL1,dtL2)))
  })
}

selectType <- function(i,celltype) {
  person <- subset(rawdata,rawdata$patient == i)
  person <- person[,c("time","U",celltype)]
  person <- person[(person$time < 200),] # Use early time points only
  names(person)[3] <- "L" # Rename last column to a generic name
  return(person)
}

# Here the session starts:

Thalf <- list()
data <- lapply(seq(5),selectType,"N4")
id <- 5

p <- c(f=0.012,delta=0.05,tau=63,c=5,p=0.001,d=0.01)
p["c"] <- cValues[id]
free <- c("f","delta")
fit0 <- myfit(data[[id]][,1:2],urine,c(U=data[[id]][1,2]),free,main="urine")
p[free] <- fit0$par; P <- p

free <- c("d")
fit1 <- myfit(data[[id]],model,c(U=data[[id]][1,2],L=0),free,main="one")
summary(fit1)

free <- c("p","d")
fit2 <- myfit(data[[id]],pdstar,c(U=data[[id]][1,2],L=0),free,main="pdstar")
summary(fit2)

p <- c(P,alpha=0.5,d2=0.001)
free <- c("alpha","d","d2")
```

```
tweak <- "nsol$L<-nsol$L1*parms[\"alpha\"]+(1-parms[\"alpha\"])*nsol$L2"
fit3 <- myfit(data[[id]],model2,c(U=data[[id]][1,2],L1=0,L2=0),free,tweak=tweak,main="two")
summary(fit3)

d <- round(c(fit1$par["d"],fit2$par["p"],fit3$par["d"]),5)
h <- round(c(log(2)/fit1$par["d"],log(2)/fit2$par["p"],log(2)/fit3$par["d"]),1)
print(c("Average_turnover:",d));print(c("Half_life:",h))
Thalf[[id]] <- h

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