Diversity of the Immune system

Handout for the Immunobiology lecture, May 3rd, 2016

Rob J. de Boer

Objectives of this exercise:
1. See that immune systems is diverse because there are so many self antigens, and not because there are so many pathogens
2. Learn to use simple probabilistic models to ask evolutionary questions about the size of the repertoire
3. See that number of MHC molecules per host is not optimal, even though several authors claim the opposite

Background

Diversity is a hallmark of the immune system. The repertoires of B cells and of CD4+ and CD8+ T cells each consist of more than 10^8 different clonotypes each characterized by a unique receptor. Each immune response is characterized by a large panel of different cytokines with –partly overlapping– functions. Each individual is characterized by a unique combination of MHC molecules that play an essential role in the selection of peptides presented to the cellular immune system. MHC loci are the most polymorphic genes known for vertebrates, i.e., for most loci several hundreds of alleles have been identified within each population. However, each individual inherits only a limited number of MHC genes from its parents, and expresses about 10 different MHC molecules. We will here address the evolutionary questions why lymphocytes are so diverse within an individual, and why MHC molecules are diverse at the population level, and not diverse within an individual.

The consensus explanation for the enormous diversity of lymphocyte repertoires is the improved recognition of many different pathogens. The consensus explanation for the limited diversity of MHC molecules within an individual is the excessive negative deletion by self tolerance processes when the number of presented self peptides is increased by increasing the diversity of MHC molecules.

Tolerance

We start with a simple toy model revealing some novel expectations for the relationships between lymphocyte specificity \( p \), the number of self epitopes \( S \), and the initial repertoire size \( R_0 \) (De Boer and Perelson, 1993). Defining the lymphocyte specificity \( p \) as the probability that a lymphocyte responds to a randomly chosen epitope, we have a definition that remains close to the conventional concept of the “precursor frequency” of an epitope. A typical viral epitope activates about one in 10^5 naive CD8+ T cells (Blattman et al., 2002; Su et al., 2013), which also says that the probability that a lymphocyte recognizes a randomly chosen epitope is about \( p = 10^{-5} \). It is difficult to estimate the number of self epitopes in general. For the peptides of nine amino acids (9-mers) that are used as epitopes by CD8+ T cells, we have made an estimate by enumerating all unique 9-mers in the human genome (Burroughs et al., 2004). Given that there are approximately 10^7 unique 9-mers in the human self, and that MHC molecules typically present about 1% of these, we would have an estimate of \( S = 10^5 \) self epitopes per T cell restricted to one particular MHC (Burroughs et al., 2004). Fortunately, for the arguments presented here the precise number of self epitopes turns out to be unimportant, we only need to know that it is large. The diversity of the repertoire before tolerization, \( R_0 \), is also a large number. Because the size of the functional CD4+ T repertoire \( R \) in man is at least 10^8 different receptors (Qi et al., 2014), the diversity of the pre-tolerance repertoire should at least be an order of magnitude higher, i.e., \( R_0 > 10^{11} \).

Having these concepts at hand we write a simple mathematical model. The diversity of the functional repertoire \( R \) is determined by the chance that a clonotype fails to recognize all self epitopes \( S \), i.e.,

\[
R = R_0(1 - p)^S.
\]  

Similarly, the chance that an individual fails to respond to a foreign epitope is the probability that none of its clonotypes in the functional repertoire \( R \) recognize the epitope. Expressing one minus the chance of failure as
Figure 1: The probability of mounting an immune response $P_i$ from Eq. (2) as a function of the specificity $p$ of the lymphocytes. Parameters $S = 10^5$ and $R_0 = 10^6$. Panel (b) depicts the effect of decreasing the initial repertoire size from $R_0 = 10^6$, $R_0 = 10^5$, to $R_0 = 10^7$. Panel (c) depicts the effect of incomplete tolerance induction, i.e., $f = 1$ and $f = 0.8$ in Eq. (6).

the probability of mounting an immune response to a foreign epitope we obtain

$$P_i = 1 - (1 - p)^R = 1 - (1 - p)^{R_0(1 - p)^S}.$$  \hspace{1cm} (2)

Plotting $P_i$ as a function of the lymphocyte specificity $p$ gives Fig. 1a which has a very wide region of specificities where the chance of mounting a successful immune response is close to one. If lymphocytes are too specific, i.e., at the left, epitopes remain unrecognized. If they are too cross-reactive, too many clonotypes are deleted by self tolerance processes, and the functional repertoire becomes too small.

Because $(1 - x)^n \approx e^{-nx}$ whenever $x \ll 1$, we can approximate this model by

$$R \approx R_0 e^{-pS} \quad \text{and} \quad P_i \approx 1 - e^{-pR} = 1 - e^{-pR_0 e^{-pS}}.$$  \hspace{1cm} (3)

When plotted for the same parameters as those of Fig. 1 the approximation is indistinguishable from the original curve (not shown). The approximation allows us to compute the “optimal” value of $P_i$ by taking the derivative $\partial_p P_i$ of Eq. (3) and solving $\partial_p P_i = 0$ to find that the maximum is at $p = 1/S$. This optimum suggests that the lymphocyte specificity is largely determined by the number of self epitopes the immune system has to be tolerant to. Thus, the specificity is not determined by the recognition of pathogens, but by the demand to remain tolerant to a large number of self epitopes. Once lymphocytes are specific, the repertoire has to be sufficiently diverse to guarantee recognition of foreign epitopes (Fig. 1b).

Incomplete tolerance

Although there is promiscuous expression of self antigens in the thymus, it remains unlikely that self tolerance is complete. Healthy individuals do harbor lymphocytes that can recognize self epitopes (Danke et al., 2004; Su et al., 2013; Malhotra et al., 2016). To study how the results change when tolerance is incomplete we define a new parameter $f$ for the fraction of self epitopes that manage to induce tolerance. For $f = 1$ the new model should be identical to the previous one. For a foreign epitope we now require that it is recognized, but that none of the clonotypes recognizing the foreign epitope also recognizes any of the $(1 - f)S$ self epitopes that fail to induce tolerance. Otherwise the clone will be held responsible for auto-immunity. Following Borghans et al. (1999) we let $\alpha$ be the fraction of clonotypes recognizing at least one ignored self epitope, i.e.,

$$\alpha = 1 - (1 - p)^{(1 - f)S}.$$  \hspace{1cm} (4)
The chance that the system remains tolerant when stimulated with a foreign epitope is the probability that none of the clones in the functional repertoire $R$ will respond (with chance $p$) and is potentially auto-reactive (with chance $\alpha$), i.e.,

$$ P_t = (1 - pa)^R \quad \text{where} \quad R = R_0(1 - p)^S. \quad (5) $$

Now the chance of a “successful immune response” is the probability that the system remains tolerant and responds to the foreign epitope, which is the chance to remain tolerant minus the chance to not respond at all:

$$ P_s = P_t - (1 - p)^R, \quad (6) $$

where the functional repertoire $R$ is given in Eq. (5). To study how incomplete tolerance affects the results we plot Eq. (6) for $f = 0.8$ and $f = 1$ in Fig. 1c.

Fig. 1c demonstrates that the effect of incomplete tolerance is enormous. The region of specificity values where the chance of a successful response $P_s$ approaches one is much narrower. Moreover the optimum has shifted leftwards, i.e., towards a specificity much smaller than $p = 1/S$. Thus the $p = 1/S$ estimate (De Boer and Perelson, 1993) is an upper bound for the lymphocyte crossreactivity: when the initial repertoire is sufficiently large the immune system operates even better when lymphocytes are more specific (Borghans et al., 1999). The conclusion remains that lymphocytes are specific to avoid auto-immunity, and not to recognize many pathogens.

Regulatory T cells

For additional documentation you could read the paper by Saeki et al. (2015) who extend this type of models with regulatory T cells. It is at least interesting to see how complicated these simple models become.

MHC diversity within the individual

Since individual MHC diversity increases the presentation of pathogens to the immune system, one may wonder why the number of MHC genes is not much higher than it is. The argument that is mostly invoked is that more MHC diversity within the individual would lead to T cell repertoire depletion during self tolerance induction (Nowak et al., 1992). This argument is incomplete, however, because more MHC diversity could also increase the number of clones in the T cell repertoire through positive selection. In order to be rescued in the thymus, lymphocytes need to recognize MHC–self peptide complexes with sufficient avidity. A high MHC diversity thus increases both the number of lymphocyte clones that are positively selected and the number of clones that are negatively selected. To calculate the net effect of these two opposing processes we need a mathematical model (Borghans et al., 2003).

Consider an individual with $M$ different MHC molecules and an initial T lymphocyte repertoire consisting of $R_0$ different clones. Let $p$ and $n$ denote the (unconditional) chances that a clone is positively selected by a single MHC type, because its avidity is higher than a threshold $T_1$, or negatively selected because its avidity exceeds a higher threshold $T_2$, respectively (see Fig. 2). By this definition, thymocytes can only be negatively selected by MHC molecules by which they are also positively selected, i.e., $n < p$. Since T cell clones need to be positively selected by at least one of the MHC molecules, and avoid negative selection by all of the MHC molecules, the number of clones in the functional repertoire $R$ can be expressed as

$$ R = R_0\left((1 - n)^M - (1 - p)^M\right), \quad (7) $$

(Borghans et al., 2003). The functional repertoire $R$ thus contains all T cell clones that are not negatively selected, minus the ones that fail to be positively selected by any of the $M$ different MHC molecules of the host.

Experimental estimates for the parameters of this model are difficult to obtain. In mice, around 3% of the T cells produced in the thymus end up in the mature T cell repertoire, and at least 50% of all positively selected T cells have been shown to undergo negative selection in the thymus (Van Meerwijk et al., 1997; Merkenschlager et al., 1997). Thus, in the absence of negative selection there would have been 6% survival, which means that
Figure 2: Positive and negative selection according to the avidity model (Janeway and Katz, 1984). The curve in (a) depicts the distribution of thymocyte avidities for self peptide–MHC complexes. In our model, the chance \( p \) to be positively selected by a single MHC type is the chance that the avidity between the thymocyte T cell receptor and any of the self peptide–MHC complexes exceeds threshold \( T_1 \). Thymocytes with avidities for self peptide–MHC complexes exceeding the upper threshold \( T_2 \) are negatively selected (with chance \( n \) per MHC type). Panel (b) depicts the size of the T cell repertoire as a function of MHC diversity as defined by Eq. (7). The number of clones in the functional repertoire \( R \) is plotted as a fraction of the total initial lymphocyte repertoire \( R_0 \). Parameters are: \( p = 0.01 \), and \( n = 0.005 \).

94% of the thymic T cells fail to be positively selected by any of the MHC molecules in the host, and that 3% of the thymocytes are negatively selected. These two estimates can be used to calculate the chances \( p \) and \( n \) of a T cell clone to be positively or negatively selected by a single type of MHC molecule. Taking into account that inbred mice are homozygous and therefore express 3 types of class I MHC and 3 types of class II MHC molecules, one estimates \( p \) from \( 0.94 = (1 - p)^6 \) and \( n \) from \( 0.97 = (1 - n)^6 \), which yields \( p \approx 0.01 \) and \( n \approx 0.005 \).

Using these experimental estimates, the number of clones in the functional T cell repertoire \( R \) increases with the number of different MHC molecules \( M \) in an individual until \( M = 140 \) (see Fig. 2b). In other words, the size of the functional T cell repertoire would increase if the MHC diversity \( M \) were to exceed its normal value of ten to twenty in heterozygous individuals. The intuitive reason is that only a very small part of the T cell repertoire has sufficient avidity for self peptides presented by a single MHC type to be positively selected by that MHC. As long as additional MHC types positively select hardly overlapping parts of the T cell repertoire, negative selection will only waste T cells that were not even positively selected in the absence of those MHC molecules. A net negative effect of MHC diversity on the size of the functional T cell repertoire is only attained once the individual MHC diversity is so large that thymocytes are selected by multiple MHC types, i.e. when \( M > 140 \). Summarizing, the consensus explanation that the MHC diversity per individual is limited to avoid repertoire depletion is untenable.
Computer Lab Exercises

Today we will plot various functions using “R”, which is a language with which one can easily do statistics, plot functions, and fit mathematical models to data. R is installed on the computers that we use, and the R-code for the functions that you need today is available as the file Pi.R on the webpage tbb.bio.uu.nl/rdb/practicals/diversity. Open this file in RStudio and spend some time to familiarize yourself with the R-functions in this file before making the exercises. The easiest way to work with RStudio is to highlight some text in the Pi.R window, and then hit the Run button (or use the shortcut control Enter) to have it executed in the Console window.

For instance, highlight the Pi function in the Pi.R file, hit the Run button, and then type Pi in the Console window. R then gives the definition of the function. Next click the curve(Pi(logp=x),from=-10,to=0) line and Run it. This pops up a graphics window plotting Pi for its default parameters (\(R_0 = 10^9, S = 10^5\)) as a function of \(\log_{10}[p]\). If you want to plot this function for another value of \(R_0\), just use the up-arrow to retrieve the previous command in the Console, and use the left-arrow to override the \(R_0\) parameter of the function, e.g., insert \(r0=1e8\) to obtain curve(Pi(r0=1e8,logp=x),from=-10,to=0). Actually, the next line in the Pi.R file does this for you, while using the add=TRUE option to add the curve to the same graph.

If you decide to write a report on this computer practical, you can choose to write a report about the specificity of lymphocytes (Questions 1 & 2) or about the optimal number of MHC molecules (Questions 3 & 4). Read the papers that we cite above, make sure you correctly address all the answers to the questions given below, and try to add some original results.

Exercise 1  Probability of response

Plot Eq. (2) for several values of \(S\) and \(R_0\) using R.

a. How do the results explained above depend to the precise values of \(S\) and \(R_0\)? What do you learn from this?
b. Phrase in your own words why the immune repertoire should be so diverse.
c. What diversity do you expect in other species with a different size of self?
d. Suppose the adaptive immune system evolved in a species with about \(S = 10^5\) self antigens. How diverse should this first immune system be to provide a reasonable selective advantage? So how do you think the adaptive immune system evolved?
e. What is the optimum if one has to survive a 100 pathogens? Does this narrow down the range of “good” immune systems? Note that curve(Pi(\(10^9\))^10,from=-10,to=0) depicts \(P_{\text{10}}\) as a function of \(\log_{10}[p]\).
f. Since every pathogen consists of a large number of epitopes, one could argue that the host is protected once it mounts a immune response to at least one epitope. If there are \(n\) epitopes in a typical pathogen, this probability becomes \(P'_i = 1 - (1 - P_i)^n\). What is now the optimum and does this narrow down the range of “good” immune systems?
g. Do you think these results depend much on the assumptions of the model?

Exercise 2  Incomplete tolerance

Plot Eq. (6) for several values of \(f\) using R.

a. How different is this model from that of Eq. (3)?
b. Carefully explain what you see when you change \(f\) (e.g., look at \(f = 0.999\)).
c. Phrase in your own words how diverse you expect an immune repertoire to be.
d. So how do the results of the previous model depend on its assumptions?

Exercise 3  Nowak’s optimal #MHC

Nowak et al. (1992) also addressed the question of the optimum number of MHC molecules within an individual with a mathematical model. According to their model the functional repertoire is given by

\[
R = R_0(1 - (1 - p)^M)(1 - n^*)^M,
\]

where \(n^*\) is the conditional probability that a positively selected clone is negatively selected by a random MHC molecule. Using the \((1 - n^*)^6 = 0.5\) cited above (Van Meerwijk et al., 1997; Merkenschlager et al., 1997), one would estimate that \(n^* = 0.11\).
a. Discuss the interpretation of each term. What is the chance of positive selection in this model?
b. What is wrong with this model?
c. Study the model with R for the same parameters as used in Fig. 2b. Note that both Eq. (7) and Eq. (8) are provided in the R-file Pi.R.

Exercise 4 General optimal MHC

The curve in Fig. 2 depends on two parameters, $p$ and $n$, that were estimated from the observation that 3% of the cells survive tolerance induction, and that 50% of the positively selected cells survive negative selection. Let us first generalize this calculation by rewriting Eq. (7) into

$$\rho = R/R_0 = (1 - n)^M - (1 - p)^M,$$

where $\rho \approx 0.03$ is the fraction of cells surviving thymic selection. Now define $\alpha$ as the fraction of cells that is positively selected, and define $\beta$ as the fraction of positively selected cells that are negatively selected (i.e., above we studied $\alpha = 0.06$ and $\beta = 0.5$). Because $\rho = \alpha(1 - \beta)$ or $\alpha = \frac{\rho}{1 - \beta}$, and we know that $\alpha = 1 - (1 - p)^M$ we solve $p$ from

$$(1 - p)^6 = 1 - \frac{\rho}{1 - \beta}, \text{ i.e., } p = 1 - \sqrt[6]{1 - \frac{\rho}{1 - \beta}}.$$

Next we solve $n$ from $\rho = (1 - n)^6 - \left[1 - \frac{\rho}{1 - \beta}\right]$ leading to

$$n = 1 - \sqrt[6]{1 - \frac{\rho\beta}{1 - \beta}}.$$

The last two expressions enable us to estimate the probabilities of positive and negative selection, $p$ and $n$, from data providing the fraction of cells surviving thymic selection, $\rho$, and the fraction of positively selected cells that are negatively selected, $\beta$. A function solving $p$ and $n$ from $\rho$ and $\beta$ is available in the R-file Pi.R, and this can be used to solve the following questions:

a. Suppose now that selection is less stringent and that 20% of the cells survive tolerance induction. How would this less stringent selection of $\rho = 0.2$ survival change Fig. 2?

b. Suppose that a much larger fraction of the positively selected cells is negatively selected, e.g., suppose that $\beta = 0.9$. How would that change Fig. 2, and how does that depend on the fraction of surviving cells?

c. The data arguing that 50% of the positively selected cells become negatively selected are based on cell counts in the thymus, and are fairly indirect. Using new methods to identify cells that undergo negative selection, Stritesky et al. (2013) recently argued that the impact of negative selection is much stronger. Enumerating positively selected cells only, they estimate that there are $7 \times 10^5$ cells dying from negative selection per hour, for every $1.23 \times 10^5$ surviving thymocytes surviving per hour. Thus the fraction of positively selected cells dying from negative selection would be $\beta = 7/8.23 = 0.85$. How does this change the results? At least study $\beta = 0.85$ keeping $\rho = 0.03$.

d. What is your favorite explanation for the fact that the diversity of MHC molecules per host is low, while their degree of polymorphism of the population is high?

```R
# Question 1

# Pi function
Pi=function(r0=1e9,s=1e5,logp=-5) {
  p = 10^logp
  r = r0*(1-p)^s
  1 - (1-p)^r
}

curve(Pi(logp=x),from=-10,to=0)
curve(Pi(r0=1e8,logp=x),add=TRUE,col="blue")

# n epitopes per pathogen:
Pin=function(r0=1e9,s=1e5,n=1,logp=-5) {
  n = 1 - \sqrt[6]{1 - \frac{\rho\beta}{1 - \beta}}
}

# Question 2

# Pi function
Pi=function(r0=1e9,s=1e5,logp=-5) {
  p = 10^logp
  r = r0*(1-p)^s
  1 - (1-p)^r
}

curve(Pi(logp=x),from=-10,to=0)
curve(Pi(r0=1e8,logp=x),add=TRUE,col="blue")
```

# Question 2

```R
# n epitopes per pathogen:
Pin=function(r0=1e9,s=1e5,n=1,logp=-5) {
  n = 1 - \sqrt[6]{1 - \frac{\rho\beta}{1 - \beta}}
}
```
\[ p = 10^{\log p} \]
\[ r = r^0(1-p)^s \]
\[ \pi = 1-(1-p)^r \]
\[ 1 - (1-\pi)^n \]

#Question 2

\begin{verbatim}
Ps=function(r0=1e9,s=1e5,f=1,logp=-5) {
  p = 10^logp
  r = r^0*(1-p)^(f*s)
  a = 1 - (1-p)^(f*s)
  pt = (1-p*a)^r
  pt - (1-p)^r
}
curve(Ps(logp=x),from=-10,to=0,add=T,col="red")
\end{verbatim}

#Question 3

# Eq. 7 for the optimal number of MHC molecules:
\begin{verbatim}
R=function(r0=1,p=0.01,n=0.005,m=6) { r^0*((1-n)^m-(1-p)^m) }
curve(R(m=x),from=0,to=1000)
\end{verbatim}

# Nowak's MHC model:
\begin{verbatim}
RN=function(r0=1,p=0.01,n=0.11,m=6) { r^0*(1-(1-p)^m)*(1-n)^m }
curve(RN(m=x),from=0,to=20)
\end{verbatim}

#Question 4

#Optimal number of MHC molecules:
\begin{verbatim}
RF=function(rho=0.03,beta=0.5,M=6,m=6) {
  p = 1 - (1-(rho/(1-beta)))^(1/M)
  n = 1 - (1-rho*beta/(1-beta))^(1/M)
  (1-n)^m-(1-p)^m
}
curve(RF(m=x),from=0,to=200)
\end{verbatim}
TRECs and Telomeres

Handout for the Immunobiology lecture, May 10th, 2016

José Borghans and Julia Drylewicz (Immunology, UMCU)

Objectives of this exercise:
1. See how a simple mathematical model can help the interpretation of experimental data.
2. Understand how a biological phenomenon can be translated into a mathematical model.
3. Learn to translate unexpected modelling results into intuitive understanding.

Exercise 1: Estimating thymic output from T-cell receptor excision circle (TREC) data

Background on TRECs

TRECs are by-products of V(D)J rearrangements that occur in the thymus when the T-cell receptor (TCR) is formed. During this genetic rearrangement process, parts of the TCR genome are excised and form stable DNA circles. These TRECs are not copied when a cell divides; they are simply passed on to one of the daughter cells. TRECs are thus uniquely formed in the thymus. When the average number of TRECs per T cell (the so-called average TREC content) was measured in healthy individuals of different ages, an exponential decline was observed (see Figure 3a). Decreased TRECs per T cell in HIV-infected patients (Figure 3b) were interpreted as evidence for impaired thymic output.

Figure 3: A. The average TREC content (in black) of CD4+ T cells from healthy individuals as a function of age (in years). From: Douek et al. (1998) (Discard the open circles). B. The average TREC content (in black dots) of CD4+ T cells from HIV-infected individuals of different ages in early stages of infection. The average TREC content in healthy individuals as a function of age is given by the dashed line. From: Douek et al. (1998)
A mathematical model for TREC

Hazenberg et al. (2000) developed a mathematical model to investigate how the average TREC content of a cell population depends on thymus output, cell division and cell death rates. The number of naive $T$ cells $N$ was described by:

$$\frac{dN}{dt} = \sigma(t) + pN - dN$$

In these equations, $\sigma$ represents daily thymic output, $p$ is the average proliferation rate of naive T cells, and $d$ represents the death rate of naive T cells. The dynamics of the total number of TREC$\text{s}$ $T$ was described by:

$$\frac{dT}{dt} = c\sigma(t) - dT$$

where $c = 0.125$ is the average TREC content of a T cell that has just left the thymus. The average TREC content of the naive T cell population ($A$) can subsequently be calculated from $A = T/N$.

Question

In this exercise we will use R and the file TREC.R with the following parameters to simulate what happens to an individual who gets HIV infected at the age of 30 years. Until the age of 30 years, the model is run with the following parameters: $\sigma(t) = 10^9$ cells/day (which for the simplicity assumed to be constant), $p = 0.0004$ per day, $d = 0.0005$ per day, and $c = 0.25$.

a. Simulate what would happen with the average TREC content and the number of naive T cells in this individual during the first 2 years of HIV infection, if HIV would totally block thymic output from the age of 30 onward. Compare your results to the 10-fold decline in the average TREC content that was observed in HIV patients (Fig. 1b). Can you explain intuitively what you see?

b. Simulate what would happen if HIV would increase the death rate of T cells from the age of 30 onward. Can you understand this intuitively?

c. Simulate what would happen if HIV would increase the rate of T-cell proliferation from the age of 30 onward.

d. How would you interpret the decreased TREC content of T cells in HIV infected patients? Is the average TREC content of a T cell population a good indicator of thymic output?

Exercise 2: Estimating lymphocyte turnover from telomere data

Background

Telomeres are the unique structures at the end of chromosomes that shorten with each cell division. Each cell division leads to the loss of 50-100 terminal nucleotides from the chromosomes. Weng et al. (1995) made use of this phenomenon to study the replication behaviour of naive and memory T lymphocytes. The average telomere lengths of naive and memory T cells were found to decrease with age at very similar rates. The average telomere length of naive T lymphocytes was consistently found to be 1.4 kb longer than that of memory cells (see Fig. 4).
A mathematical model for telomere shortening

In order to study how the average telomere length of a population of cells changes with age, De Boer and Noest (1998) developed a mathematical model. The model keeps track of the division history of a population of naive T cells, \( N \), and memory T cells, \( M \), each subdivided in subpopulations of cells that have undergone \( i \) cell divisions (see Fig. 3). For example, the naive cells in box \( n_0 \) have not divided yet, while the memory cells in box \( m_5 \) have undergone 5 divisions. The arrows \( p_N, p_M, d_N \) and \( d_M \) represent proliferation (= cell division) and death rates of naive and memory T cells, respectively. The arrow \( \gamma C \) represents the influx of naive cells into the memory pool through activation: Naive T cells are activated during immune responses to foreign antigens at rate \( \gamma \), and through clonal expansion form C memory T cells. During this activation into the memory population their division index increases by \( K \).
Using this model it was shown that the average division-index ($\mu$) of naive T cells ($N$) and memory T cells ($M$) are given by:

$$\frac{d\mu_N}{dt} = 2p_N$$ (11)

and

$$\frac{d\mu_M}{dt} = 2p_M - \gamma \frac{N}{M}(\mu_M - \mu_N - K)$$ (12)

Question

In this exercise we will investigate (using R and the file Telomeres.R) how we should interpret the experimental data of Figure 2. We can use the differential equations for $\mu_N$ and $\mu_M$ described above to calculate the average telomere lengths of the naive ($T_N$) and memory ($T_M$) T-cell population from:

$$T_N = T_{n0} - L\mu_N$$
$$T_M = T_{m0} - L\mu_M$$

where $L$ is the number of base pairs lost during each cell division, and $T_{n0}$ and $T_{m0}$ are the average telomere lengths of naive and memory T cells at birth, respectively.

a. The telomere data of Figure 2 were taken as evidence that naive and memory T cells divide approximately once every 5 years. Test if one division in every five years for both naive and memory T cells would indeed give rise to the data of Figure 2. You can use the following parameter values: $N/M = 1$, $\gamma = 0.001$ per day, $L = 0.1$ kb, $C = 16$ cells, $T_{N0} = 10$ kb, $T_{m0} = 8.6$ kb, $p_N = p_M = 0.0005$ divisions per day and $K = 10$. 

![Figure 5: Schematic representation of the mathematical model for telomere shortening of naive (N) and memory (M) T cells. Adapted from: De Boer and Noest (1998).]
b. The findings of Figure 2 were remarkable because many other experiments suggested that memory T cells divide a lot more frequently than naive T cells. Yet the fact that telomere lengths of naive and memory T cells declined at equal rates suggested very similar kinetics of naive and memory lymphocytes. Increase the value of \( p_M \) to investigate how the telomere data would change if memory cells divide much (e.g. 10 times) more frequently than naive T cells.

c. Using the following small steps, investigate why you found the above (perhaps unexpected) result...

c1. Investigate what happens when you decrease the value of \( \gamma \) (e.g. 100-fold) in the scenario in which memory T cells divide 10-fold more frequently than naive T cells.

c2. From the equation for \( \mu_M \) you can tell that the average division index (and hence telomere length) of the memory T-cell population does not directly reflect memory cell division. What is the effect of activation of naive T cells into the memory pool?

c3. How can we resolve the apparent contradiction between the parallel loss of telomeres in naive and memory T lymphocytes, and the fact that memory T cells are thought to divide more frequently than naive T cells?

d. Is the average telomere length a good indicator of the division history of a cell population?

TREC.R

```r
library(deSolve)
### model ###
model_TREC <- function(t, state, parms){
  with(as.list(c(state,parms)),{
    dN <- ifelse(t<30,sigma1 - d1*N + p1*N, sigma2 - d2*N + p2*N )
    dTr <- ifelse(t<30,c*sigma1 - d1*Tr, c*sigma2-d2*Tr )
    return(list(c(dN, dTr)))
  })
}
run <- function(finish=100,step=1,state=s,parms=p*year) { # run and make a table
  parms1<-c(sigma1=10**9, p1=0.0004, d1=0.0005)*365.25 # parameters for healthy
  individual
  parms2<-c(parms1,parms)
  out<-ode(times=seq(0,finish,by=step),
           func=model_TREC,y=state,parms=parms2,rtol=1e-12)
  data_run<-as.data.frame(cbind(out,out[,3]/out[,2]))
  colnames(data_run, do.NULL = FALSE)
  colnames(data_run)<-c("time","N","T","A")
  return(data_run)
}
timeplt <- function(finish=100,step=1,state=s,parms=p*year) { # run and make a time
  plot
  data<-run(finish,step,state,parms)
  par(mfrow=c(1,2))
  attach(data)
  plot(data[,1],data[,2],log="y",type='l',lwd=2,xlab="Age in years",ylab="Naive T cells and total TRECs",yaxt="n",ylim=c(10**10,10**14),xlim=c(28,32))
  lines(data[,1],data[,3],col="blue",lwd=2)
  legend("topright",legend=c("Naive T cells","Total TRECs"), col=c("black","blue"),bty="n", lty=1, lwd=2,cex=0.85)
  plot(data[,1],data[,4],log="y",col="red",lwd=2,type="l",xlab="Age in years",ylab="Average TREC content",yaxt="n",ylim=c(0.0001,1),xlim=c(28,32))
  axis(2,c(0.0001,0.001,0.01,0.1,1))
```
library(deSolve)

### model ###
model_telomere <- function(t, state, parms){
  with(as.list(c(state,parms)),{
    dmuN <- 2*pn
    dmuM <- 2*pm-gamma*C*(muM-muN-K)
    return(list(c(dmuN, dmuM)))
  })
}

run <- function(finish=100, step=1, state=s, parms=p*year) {
  # run and make a table
  out<-ode(times=seq(0,finish,by=step),
    func=model_telomere,y=state,parms=parms,rtol=1e-12)
  TN<-Tn0-L*out[,2]
  TM<-Tm0-L*out[,3]
  data_run<-as.data.frame(cbind(out[,1],TN,TM))
  colnames(data_run, do.NULL = FALSE)
  colnames(data_run)<-c("time","TN","TM")
  return(data_run)
}
```r
timeplt <- function(finish=100, step=0.1, state=s, parms=p*year) {
  # run and make a time plot
  data<-run(finish, step, state, parms)
  par(mfrow=c(1,1))
  attach(data)
  plot(data[,1], data[,2]/1000, type='l', lwd=2, xlab="Age in years", ylab="Average telomere length", ylim=c(0,11), xlim=c(20,80))
  lines(data[,1], data[,3]/1000, col="blue", lwd=2)
  legend("topright", legend=c("Naive T cells", "Memory T cells"), col=c("black", "blue"), bty="n", lty=1, lwd=2, cex=0.85)
  detach(data)
}

year=365.25
Tn0=10000
Tm0=8600
L=100
C=16
K=0
s <- c(muN=0, muM=0) # initialize the model

# Here the session starts:

## Question a
p <- c(pn=0.0005, pm=0.0005, gamma=0.001)
run()
timeplt()

## Question b: increase pm
p <- c(pn=0.0005, pm=, gamma=0.001)
run()
timeplt()

## Question c: decrease gamma
p <- c(pn=0.0005, pm=, gamma=)
run()
timeplt()
```

Objectives of this exercise:

1. Learn to quantify cell migration based on cell tracks from intravital movies.
2. Learn to associate different patterns of migration with different biological functions.
3. Learn to use the simple Beauchemin model to simulate cell migration.
4. Apply this model to study the impact of imaging setups on conclusions drawn from cell tracking.

All material for this exercise is online at theory.bio.uu.nl/immbio/migration, where you can also download the PDF version of this sheet with clickable links.

Exercise 1 Analyzing cell tracks

For this exercise, we prepared three different cell track datasets from intravital imaging experiments. Three different cell populations have been imaged:

1. a population of neutrophils homing towards an infection in the lung;
2. a population of T cells searching for antigen in a lymph node; and
3. a population of B cells searching for antigen in a lymph node.

Sadly, something went wrong and we don’t know which file represents which cell population. Can you help?

a. Download the files dataset1.xls, dataset2.xls, and dataset3.xls from the course website (see above link), and open them in Excel. Take a minute to look at the raw data. How is the data organized? What do you think each individual column represents? Can you infer the duration of each experiment from the raw data?

b. Open the MotilityLab website, an online tool for analyzing cell track datasets, at 2ptrack.net. We recommend using Mozilla Firefox for this, as Internet Explorer lacks support for 3D graphics. Click on “Launch MotilityLab”. Import your three datafiles into the MotilityLab session: Click on “Import tracks” underneath “File” at the top left corner of the page. Switch to your Excel file, and copy all the data by pressing “Ctrl+A, Ctrl+C”. Click on the top left cell of the grid under “Option 1”on the MotilityLab import page (highlighted by the red arrow in the image on the right hand side) and paste by pressing “Ctrl+V”. Type in a short file name for the data (e.g. “data1”) and click “save data”. Do this for all three datasets.

c. Once all data is uploaded, click on “Quantitative Analysis” in the MotilityLab menu. First, display a 2D projection of all of your tracks together by ticking the corresponding textboxes at “File list”. What differences do you observe by eye? Which population stands out most?

d. Make a plot of mean track speed by selecting “mean track speed” from the drop-down in the “Plots” subwindow. How do the three populations compare speed-wise? If you compare the speed to the typical diameter of a lymphocyte (5-10 \( \mu \)m), would you describe the cell motion as fast or slow?

e. Make a mean square displacement plot. Do you reach similar conclusions as with mean track speed when you compare the populations? Does any of the populations seem to exhibit linear motion? Do your conclusions change if you only look at the first 5 minutes of your data (using the “Crop tracks” filter)?

f. Make an autocorrelation plot. For each cell population, at what time does the autocorrelation function touch the zero axis? Based on this value, what are the timescales of linear and Brownian motion for each cell population – e.g., does one of them look linear on a short timescale, but Brownian on a long timescale?
By now, you probably know which datafile is which cell population – if you’re not sure about the T and B cells, check figures 2 and 3 of the Miller et al paper (Miller et al., 2002) or ask the tutor for advice. Can you link your quantitative results to the biological functions T cells, B cells and neutrophils in their respective experimental settings? For instance, why does the autocorrelation of the neutrophil tracks make sense given that they are fighting an infection in the lung? What about speed?

(h. (Optional) The lymph nodes in which our T and B cells were recorded were not infected. How do you expect an infection to influence the migration of T and B cells in lymph nodes? Which motility parameters would change, and would they decrease or increase?

Exercise 2 Simulating lymphocyte migration with the Beauchemin model

As discussed in the lecture, many different models exist to simulate lymphocyte migration. Here we will use the probably simplest one, called the Beauchemin model (Beauchemin et al., 2007).

Cells in the Beauchemin model are represented as points without a shape and without a mass (this means that, like ghosts, they cannot “bump into each other” or block each other out). Each cell-point moves according to a deterministic program, which has three parameters: a speed $v_{\text{free}}$, a time interval $t_{\text{free}}$, and another time interval $t_{\text{pause}}$. The cell-point starts at a fixed position, chooses a random direction, and moves towards the chosen direction with speed $v_{\text{free}}$. After a time interval $t_{\text{free}}$, the movement stops, and the cell-point pauses for time $t_{\text{pause}}$. These steps are repeated as often as desired.

a. To visualize the Beauchemin model, we have created an online implementation at beauchemin.html. To view this implementation on the lab computers, you will have to use Firefox with 3D graphics (WebGL) enabled – ask your tutor how to do this. You will save time later on if you open the simulation in a new tab or window and keep your MotilityLab window open. Click “start simulation” to see the model in action. Play a little with the parameters to get a feeling for the model. For example, what happens if you increase $t_{\text{free}}$ and/or $t_{\text{pause}}$ by a factor of 10?

b. Simulate 200 cells using the parameters $v_{\text{free}}=12 \mu m/min$, $t_{\text{free}}=2 \text{ min}$, $t_{\text{pause}}=0.5 \text{ min}$, $\Delta t=0.5 \text{ min}$. Set the simulation time (which runs much faster than real time) to 1 hour and the depth of the imaging volume to a realistic 40 $\mu m$. Like in real experiments, simulated cells are only tracked inside this volume, which is displayed in red. Import the simulated tracks into MotilityLab, and compare them to the real data. Are they most similar to B cells, T cells or neutrophils? Do mean speed, autocorrelation, and mean square displacement look realistic? Which differences to the real data do you observe?

c. (Optional) Quantitative data on lymphocyte migration varies a lot between studies. For example, the mean speed of T cells in uninfected murine lymph nodes was found to be 18 $\mu m/min$ in one study (Textor et al., 2011) and 10.8 $\mu m/min$ in another (Miller et al., 2002), an almost 2-fold difference. Could such inconsistencies be due to different imaging setups? Answer this question using the Beauchemin model. Generate 2 datasets where cells are “imaged” in 2 different setups:

1. Imaging volume depth: 100 $\mu m$, imaging interval: 0.75 min.
2. Imaging volume depth: 40 $\mu m$, imaging interval: 0.25 min.

Leave all other parameters at their default values (see above), and simulate 500 cells each for 1 hour. Import the tracks into MotilityLab and give them meaningful names to make the comparison easier (e.g. “tracks_large_volume”, “tracks_small_volume”). Quantify the tracks. What differences to you find? Which of these do you think are due to the imaging interval, and which are due to the imaging volume? Imagine you were comparing these data without knowing the technical parameters of the imaging experiment (e.g., in journal articles such details are not always reported). Could these quantitative differences be misinterpreted as biological effects?

d. (Optional) Try to mimic neutrophil migration. Which parameter(s) do you need to modify for that?

e. (Optional) What do you think of the Beauchemin model in general? How could you make it more realistic?

---

For the tech-affine students: You can enable WebGL yourself by typing “about:config” into the Firefox address bar. Click “I’ll be careful” and then type “webgl” into the search box. Set the value “webgl.force-enabled” to “true”.

16
T cell turnover

Handout for the Immunobiology lecture, May 17th, 2016

José Borghans and Julia Drylewicz (Immunology, UMCU)

Objectives of this exercise:
1. See how a simple mathematical model can help the interpretation of experimental data.
2. Understand how to interpret different kinetic and labeling data.

Exercise 1: T-cell dynamics after stem-cell transplantation

During HIV infection, CD4+ T cells show increased levels of T-cell proliferation (as measured by Ki67 expression). Although some have argued that increased T-cell proliferation in HIV infection is a consequence of the low numbers of T cells (the immune system is trying to compensate for the loss of cells), others have argued that increased T-cell proliferation is induced directly by the virus, and is the cause of low CD4+ T-cell numbers. In the article by Hazenberg et al. (Blood 2002) it was found that shortly after stem-cell transplantation, when cell numbers are very low, T cells have increased levels of Ki67 expression.

a. Although these data seem to support the idea that low T-cell numbers (even in the absence of HIV infection) lead to increased levels of T-cell proliferation through a homeostatic mechanism, Hazenberg et al. argue that even after stem-cell transplantation there is no homeostatic response of the immune system. Give two arguments for their interpretation, and point out which figures support their interpretation.

b. Calculate which average TREC content you expect in a healthy individual with $s = 0.2 \text{cells/µl}$ blood per day, $p = 0.0005$ per day, $d = 0.001$ per day, and $c = 0.25$. Check with the file TREC2.R whether this is indeed the equilibrium.

c. Before patients can receive a stem-cell transplantation their lymphocyte numbers are dramatically decreased because of ‘conditioning’ for the transplantation. Investigate what happens with the average TREC content if the T-cell pool (and the total number of TRECs) in an individual is reduced 10-fold due to conditioning for stem-cell transplantation?

d. How can you explain intuitively what happens to the TREC content after stem cell transplantation?

e. There is some dispute in the field as to whether the thymus in healthy adults is still contributing new T cells. What do the findings of Hazenberg’s Figure 2A tell you about the thymus in a healthy adult?

Exercise 2: Quantification of T-cell turnover using BrdU or stable isotope labelling

Background

Hellerstein et al. (2003) studied the increase in deuterium label during 9 weeks of heavy water (D2O) administration in the DNA of T cells from healthy and HIV-infected subjects. See the abstract and figure in the following.
Subpopulations of long-lived and short-lived T cells in advanced HIV-1 infection

Marc K. Hellerstein,1,2 Rebecca A. Hoh,2 Mary Beth Hanley,3 Denise Cesar,1 Daniel Lee,1 Richard A. Neese,1,2 and Joseph M. McCune2,3,4

1Department of Nutritional Sciences and Toxicology, University of California, Berkeley, Berkeley, California, USA
2Department of Medicine, University of California, San Francisco, and Medical Service, San Francisco General Hospital, University of California, San Francisco, San Francisco, California, USA
3Gladstone Institute of Virology and Immunology, San Francisco General Hospital, University of California, San Francisco, San Francisco, California, USA
4Department of Microbiology and Immunology, University of California, San Francisco, San Francisco, California, USA

Antigenic stimulation of T cells gives rise to short-lived effector cells and long-lived memory cells. We used two stable isotope-labeling techniques to identify kinetically distinct subpopulations of T cells and to determine the effect of advanced infection with HIV-1. Long-term deuterated water (2H2O) incorporation into DNA demonstrated biphasic accrual of total and of memory/effector (m/e)–phenotype but not naive-phenotype T cells, consistent with the presence of short-lived and longer-lived subpopulations within the m/e-phenotype T cell pool. These results were mirrored by biphasic die-away kinetics in m/e- but not naive-phenotype T cells after short-term 2H-glucose labeling. Persistent label retention was observed in a subset of m/e-phenotype T cells (presumably memory T cells), confirming the presence of T cells with very different life spans in humans. In advanced HIV-1 infection, much higher proportions of T cells were short-lived, compared to healthy controls. Effective long-term anti-retroviral therapy restored values to normal. These results provide the first quantitative evidence that long-lived and quiescent T cells do indeed predominate in the T cell pool in humans and determine T cell pool size, as in rodents. The greatest impact of advanced HIV-1 infection is to reduce the generation of long-lived, potential progenitor T cells.


Figure 6: Note that LT ARV stands for patients on long term anti-retroviral therapy, m/e stands for memory or effector T cells and f (on y-axis) is the percentage of cells that have proliferated (and hence have taken up deuterium)
Model for stable isotopes

As shown in today’s lecture, the fraction of deuterium labelled DNA $L$ during deuterium administration changes according to:

$$\frac{dL}{dt} = p - dL$$  \hspace{1cm} (13)

where $p$ is the production rate of the cells (in the thymus or through peripheral T-cell proliferation) and $d$ their death rate. After stop of deuterium administration the fraction of deuterium labelled DNA $L$ changes according to:

$$\frac{dL}{dt} = -dL$$  \hspace{1cm} (14)

In this exercise we will use the file labeling.R to study deuterium and BrdU labelling.

a. Hellerstein et al. (2003) reported that the accrual of deuterium in the DNA of total and memory/effectector CD4 T cells occurs “bi-phatically”. Based on this it was concluded that the memory/effectector CD4 T-cell pool consists of two different subpopulations: one with short-lived and one with long-lived cells.

Simulate a deuterium labelling experiment for a cell population with $p = 0.015$ per week and $d = 0.05$ per week when the individual is healthy. Increase the proliferation and death rate of cells 10-fold when the individual is HIV-infected.

What is causing the non-linear increase? Was Hellerstein’s interpretation correct?

b. In many deuterium labelling studies the average turnover rate of cells is determined by finding the maximum level of deuterium incorporation that is ever attained during label administration, and dividing by the number of days it took to reach that maximum.

Based on your simulations of previous question, describe what can go wrong using this method.

How would you design a labelling study to avoid such problems?

How would you design it if you were only allowed to take one blood sample during labeling?

c. Consider a mouse T-cell population for which $p = 0.5$ per day and $d = 0.5$ per day, which is labelled for 10 days with both BrdU and deuterium. Simulate how the percentage of BrdU labeled cells and deuterium labeled DNA increases during the labeling period.

Why is the increase in the fraction of BrdU labelled cells faster than the increase in the fraction of deuterium labelled DNA?

d. Simulate what happens with this cell population when BrdU and deuterium labelling are stopped. What is going on?

e. What could explain the BrdU loss that is typically observed after label withdrawal?
library(deSolve)

### model ###

model_TREC <- function(t, state, parms){
  with(as.list(c(state,parms)),{
    dN <- sigma - d*N + p*N
    dTr <- c*sigma - d*Tr
    return(list(c(dN, dTr)))
  })
}

run <- function(finish=50, step=0.1, state=s, parms=p*year) {  # run and make a table
  out<-ode(times=seq(0,finish,by=step),
    func=model_TREC, y=state, parms=parms, rtol=1e-12)
  data_run<-as.data.frame(cbind(out,out[,3]/out[,2]))
  colnames(data_run, do.NULL = FALSE)
  colnames(data_run)<-c("time","N","T","A")
  return(data_run)
}

timeplt <- function(finish=50, step=0.1, state=s, parms=p*year) {  # run and make a time plot
  data<-run(finish, step, state, parms)
  par(mfrow=c(1,2))
  attach(data)
  plot(data[,1],data[,2],log="y",type='l',lwd=2,xlab="Time in years",ylab="Naive T cells and total TREC")
  ylim=c(1,100000),xlim=c(0,50))
  lines(data[,1],data[,3],col="blue",lwd=2)
  legend("topright",legend=c("Naive T cells","Total TREC"), col=c("black","blue"),bty="n", lty=1, lwd=2,cex=0.85)
  axis(2,c(1,10,10**2,10**3))
  plot(data[,1],data[,4],log="y",col="red",lwd=2,type="l",xlab="Age in years",ylab="Average TREC content",yaxt="n",ylim=c(0.01,1),xlim=c(0,10))
  axis(2,c(0,0.01,0.1,1))
  legend("topright",legend="Average TREC content", col="red", lty=1, lwd=2,cex=0.85,bty="n")
  detach(data)
}

########################### Here the session starts:

c=0.25
year=365.25

## Question b:

s <- c(N=, Tr=) # values at time = 0 for an healthy individual
p <- c(sigma=, p=, d=)

run()
timeplt()

## Question c:
\texttt{s <- c(N=, Tr=)} \# values at time = 0 for an individual after SCT
\texttt{p <- c(sigma=, p=, d=)}

\texttt{run()}
\texttt{timeplt()}

library(deSolve)

### model deuterium ###
model_deuterium <- function(t, state, parms){
  with(as.list(c(state,parms)),{
    dLd <- p - d*Ld
    return(list(c(dLd)))
  })
}

model_deuterium2 <- function(t, state, parms){
  with(as.list(c(state,parms)),{
    dLd <- ifelse(t<10, p - d*Ld, -d*Ld)
    return(list(c(dLd)))
  })
}

### model BrdU ###
model_brdu <- function(t, state, parms){
  with(as.list(c(state,parms)),{
    dUb <- -p*Ub - d*Ub
    return(list(c(dUb)))
  })
}

model_brdu2 <- function(t, state, parms){
  with(as.list(c(state,parms)),{
    dLb <- p*Lb - d*Lb
    return(list(c(dLb)))
  })
}

#### For deuterium up-labeling only
run1 <- function(finish=10, step=0.1, state=s, parms=p) {
  out<-ode(times=seq(0,finish,by=step),
    func=model_deuterium, y=state, parms=parms, rtol=1e-12)
  data_run<-as.data.frame(cbind(out[,1],out[,2]))
  colnames(data_run, do.NULL = FALSE)
  colnames(data_run)<-c("time","Ld")
  return(data_run)
}
timeplt1 <- function(finish=10, step=0.1, state=s, parms=p) {
  # run and make a time plot
  data <- run1(finish, step, state, parms)
  par(mfrow=c(1,1))
  attach(data)
  plot(data[,1], data[,2], type='l', lwd=2, xlab="Times in weeks", ylab="Fraction labeled DNA", ylim=c(0,1), xlim=c(0,20))
  legend("topright", legend="Deuterium Labeling", col="black", bty="n", lty=1, lwd=2, cex=0.85)
  detach(data)
}

#### For deuterium and BrdU up-labeling only
run2 <- function(finish=10, step=0.1, state=s, parms=p) {
  # run and make a table
  out1 <- ode(times=seq(0, finish, by=step),
              func=model_deuterium, y=state[1], parms=parms, rtol=1e-12)
  out2 <- ode(times=seq(0, finish, by=step),
              func=model_brdu, y=state[2], parms=parms, rtol=1e-12)
  Lb <- 1-out2[,2]
  data_run <- as.data.frame(cbind(out1[,1], out1[,2], Lb))
  colnames(data_run, do.NULL = FALSE)
  colnames(data_run)<-c("time","Ld","Lb")
  return(data_run)
}

timeplt2 <- function(finish=10, step=0.1, state=s, parms=p) {
  # run and make a time plot
  data <- run2(finish, step, state, parms)
  par(mfrow=c(1,1))
  attach(data)
  plot(data[,1], data[,2], type='l', lwd=2, xlab="Time in weeks", ylab="Fraction labeled", ylim=c(0,1), xlim=c(0,20))
  lines(data[,1], data[,3], col="blue", lwd=2)
  legend("topright", legend=c("Deuterium labeling","BrdU labeling"), col=c("black", "blue"), bty="n", lty=1, lwd=2, cex=0.85)
  detach(data)
}

#### For deuterium and BrdU up- and de-labeling
run3 <- function(finish=20, step=0.1, state=s, parms=p) {
  # run and make a table
  out1 <- ode(times=seq(0, finish, by=step),
              func=model_deuterium2, y=state[1], parms=parms, rtol=1e-12)
  out2 <- ode(times=seq(0, finish, by=step),
              func=model_brdu, y=state[2], parms=parms, rtol=1e-12)
  out3 <- ode(times=seq(0, finish, by=step),
              func=model_brdu2, y=state[3], parms=parms, rtol=1e-12)
  data_run <- as.data.frame(cbind(out1[,1], out1[,2], 1-out2[,2], out3[,2]))
  colnames(data_run, do.NULL = FALSE)
  colnames(data_run)<-c("time","Ld","Lb","Lb2")
}
return(data_run)

}  

timeplt3 <- function(finish=20, step=0.1, state=s, parms=p) {  # run and make a time plot
data<-run3(finish, step, state, parms)
par(mfrow=c(1,1))
attach(data)
plot(data[,1],data[,2],type='l',lwd=2,xlab="Time in weeks",ylab="Fraction labeled",ylim=c(0,1),xlim=c(0,20))
lines(data[,1],data[,3],col="blue",lwd=2)
lines(data[,1]+20,data[,4],col="blue",lwd=2)
legend("topright",legend=c("Deuterium labeling","BrdU labeling"), col=c("black","blue"),bty="n", lty=1, lwd=2,cex=0.85)
detach(data)

}  

##################### Here the session starts:  

## Question a: deuterium up-labeling
s<-c(Ld=0)
p <- c(p=0.5,d=0.5)
run1()
timeplt1()

## Question c: deuterium and BrdU up-labeling
s<-c(Ld=0,Ub=1)
p <- c(p=0.5,d=0.5)
run2()
timeplot2()

## Question d: deuterium and BrdU
s<-c(Ld=0,Ub=1,Lb=1)
p <- c(p=0.5,d=0.5)
run3()
timeplot3()}
Acute Immune Responses to viruses

Handout for the Immunobiology lecture, June 9th, 2016

Rob J. de Boer

Objectives of this exercise:
1. Learn to think in a quantitative manner about the immune system
2. Learn to parametrize simple mathematical models using quantitative information about steady states
3. Learn to fit mathematical models to experimental data to obtain quantitative parameter estimates.
4. Identify the mechanisms underlying the immunodominance ranking of primary immune reactions.

Background

If you choose to do a project about this you could start to see how in Fig. 2 of De Boer and Perelson (2013) the data from Kotturi et al. (2008) on the precursor frequencies are included in an extended LCMV model, and how the same data on the GP33 and the NP396 response to LCMV are explained by this model. Other questions to address are: How would the model change when memory cells are already formed during the expansion phase (see Kohler (2007))? Can this explain the data equally well?

Exercise 1 The LCMV primary immune response

Vigorous infections with rapidly replicating pathogenic bacteria or viruses trigger strong adaptive immune responses. A few days after the initial exposure, antigen specific naïve T cells will become activated to undergo rapid clonal expansion, until the peak of the response, after which most of the activated cells die by apoptosis. The clonal expansion in mice infected with bacteria typically continues if the antigenic stimulus is removed by treating the mice with antibiotics, but the peak response is somewhat lower (Badovinac et al., 2002). Similarly, the contraction after the peak will take place even if the antigen persists (Badovinac et al., 2002). It has therefore been suggested that after proper antigenic stimulation the acute immune response of T cells is “programmed”, and hardly regulated by the current concentration of antigen. Such a program can conveniently be modeled with piece-wise models (De Boer et al., 2001, 2003; De Boer and Perelson, 2013), that can be fitted to the experimental data from acute immune responses to viruses and bacteria. Consider antigen specific activated T cells, $A$, and memory cells, $M$, e.g.,

\[
\begin{align*}
\frac{dA}{dt} &= 0 \quad \text{and} \quad \frac{dM}{dt} = 0, \\
\frac{dA}{dt} &= pA \quad \text{and} \quad \frac{dM}{dt} = 0, \\
\frac{dA}{dt} &= -(d_A + r)A \quad \text{and} \quad \frac{dM}{dt} = rA - d_MM, \\
\end{align*}
\]

(19)

where $A(0)$ is the initial number of cells specific for the epitope of interest (and $M(0) = 0$). An example for coding these equations in R is available as the script lcmv.R.

On the webpage tbb.bio.uu.nl/rdb/practicals/lcmv you will also find the data of Homann et al. (2001) for one particular epitope (gp33) in the file gp33.txt. Open that file to see what is in there (you will plot the data later using the R-script lcmv.R). Read the R-script, and check how the model of Eq. (19) is defined as a solution. Check how to read the data, plot them, and fit them. Note the model returns the sum of $A$ and $M$ to be fitted to the data.

a. Fit the model to the data using ModFit function of the FME package.
b. What are the parameter values and how good do you find this fit?
c. How robust are these parameter estimates? How do the fits change if you fix the values of one parameter?
d. How would the parameter estimates change if you don’t take the logarithm of the data?
e. We only fit $A0$ and not $Ton$ (i.e., $T_{on} = 0$). Why can they not both be estimated?

On the same webpage you can also find the data of the NP396 epitope from Homann et al. (2001).
f. Download this file, depict the data to familiarize yourself with it, and fit the model of Eq. (19) to the data of this NP396 epitope.

g. What is the best way to explain the differences between these responses?

The second part of the lcmv.R R-script defines a new cost function with which both responses can be fitted simultaneously to the data. The parameters that are allowed to differ are defined in the vector `differ`.

h. Play with the shared and unique parameters per immune response.

i. Which parameter(s) best explains the immunodominance?

j. We used the fitted parameters for the GP33 response as an initial guess for the NP396 response. Does that influence the results? How would you test that?

Kotturi et al. (2008) determined the affinity and the precursor frequencies of several CD8+ T cell responses to LCMV and report that naive B6 mice contain on average 449 and 117 precursors for the GP33 and the NP396 epitopes, respectively.

k. Would this be consistent with your explanation for the immunodominance relation of these two responses?

l. Students choosing this for their project should now continue with the R script that we provide in the file kotturi.R.

```r
library(FME)

# This function provides the solution of the model:

sol=function(t,parms=pars) {
  with(as.list(parms), {
    tot <- ifelse(t < Ton, A0,
    ifelse(t < Toff, A0*exp(p*(t-Ton)),
    ifelse(t < Toff, A0*exp(p*(Toff-Ton))
    A <- AToff*exp(-(da+r)*(t-Toff))
    M <- (r*AToff/(da+r-dm))*(exp(-dm*(t-Toff)) - exp(-(da+r)*(t-Toff)))
    A + M
  )
  )
  return(tot)
})

cost <- function(p, data, cost=NULL) {
  pars[index] <- p
  tot <- log10(sol(data[,1],parm=pars))
  tot <- as.data.frame(cbind(data[,1],tot))
  names(tot) <- c("t", "v")
  return (modCost(x="t", model=tot, obs=data, cost=cost))
}

plotfit <- function(p, data, finish=1000) { # Plot fit as a continous curve
  n <- length(data[,1])
  plot(range(0:finish),range(0:8),type="n",xlab="Time in days",ylab="Log10 population size")
  curve(log10(sol(t=x,parm=p)),from=0,to=finish,n=2*finish,col="red",add=TRUE)
  points(data)
}

25
```
# Here the analysis starts:

pars <- c(A0=10, Ton=0, Toff=6, p=2, da=0.5, r=0.05, dm=0)  # Define parameter values

# Read gp33 data and plot them with initial guess of parameters

# take the log of the data column

plot(gp33)  # Plot the data and add the guessed solution

curve(log10(sol(t=x, parm=pars)), from=0, to=1000, add=TRUE)

# Select subset of parameters to fit:

index <- c("A0","Toff","p","da","r")

guess <- pars[ index ]  # Take values from pars as first guess

# Fit gp33 data:

fit33 <- modFit(f=cost, p=guess, data=gp33, lower=0, method="Marq")

summary(fit33)  # Note that some parameters are highly correlated and difficult to estimate

par33 <- pars
par33[index] <- fit33$par  # Save fit as a new parameter setting

plotfit(par33, gp33)

# Fit np396 data:

np396 <- read.table("np396.txt", header=TRUE)

cost1 <- cost(p[1:n1], gp33)

n2 <- length(differ)

p[differ] = p[(n1+1):(n1+n2)]

return (cost(p[1:n1], np396, cost=cost1))

getpar <- function(pars, fit, dataset=0) {

n1 <- length(index)

cost1 <- cost(p[1:n1], gp33)

n2 <- length(differ)

p <- pars

p[index] <- fit$par[1:n1]

if (dataset > 1)

differ <- c("A0","p")

guess <- c(pars[index], pars[differ])

fit2 <- modFit(f=cost2, p=guess, lower=0, method="Marq")

summary(fit2)

both33 <- getpar(pars, fit2, 1)

both396 <- getpar(pars, fit2, 2)

par(mfrow=c(1,2))
run <- plotfit(both33,gp33,100)
points(gp33)
run <- plotfit(both396,np396,100)
points(np396)
par(mfrow=c(1,1))

# Questions:
# 1. Which parameters explain the difference between the two immune responses best?
# 2. How would the parameter estimates change if you don't take the logarithm of the data?
# 3. We used the fitted parameters for the GP33 response as an initial guess for the NP396 response. Does that influence the results? How would you test that?
# 4. We only fit A0 and not Ton. Why can they not both be estimated?

References


