Estimating parameters from experimental data: the fitting by Ram *et al.* [1] of rich bacterial growth data.

The aim of this practical is to learn how can quantitatively interpret data by fitting mathematical models, and to become aware of the problem of reliably identifying all parameter values, when complex models are fitted to simple data.

You will first repeat the analysis of Ram *et al.* [1] and estimate the growth parameters of a few bacterial strains under two different experimental conditions. The paper demonstrates that estimating these growth parameters suffices for estimating the relative fitness of the strains. The individual growth rate of each strain is described by the following extension of the logistic growth equation $\mathrm{d}N/\mathrm{d}t = rN(1 - N/K)$ into

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \frac{q_0}{q_0 + \mathrm{e}^{-mt}}\, rN\left[1 - \left(\frac{N}{K}\right)^v\right] ,$$

where v is a parameter defininig the curvature of the density dependent function $[1 - (N/K)^v]$, which is linear when $v = 1$ and concave when $v > 1$ (convex shapes are not considered). The leading term describes the adjustment of the bacteria to the environment of the experiment. When $t = 0$ the bacteria start with a maximum replication rate $\frac{q_0 r}{q_0+1}$ that is below r, and approach their ultimate maximum replication rate r after some time defined by the parameter m in the exponential function. This model has 6 parameters, $N(0)$, q_0, m, r, K, and v, and in Fig. 3 these parameters are independently estimated for each of the strains (called 1 (red), 2 (green), 3 (red) or 4 (green)) in three experimental conditions (called A, B or C). These estimates are given in Table S2. Note that v was not estimated for A1 and C4, and that the lag parameters, q_0 and m, were ignored in experiment B (that was done with bacteria coming from a fresh culture). You can find the paper and its appendix on tbb.bio.uu.nl/rdb/practicals/bacteria.

After estimating these six parameter from the growth curves in Fig. 3, two different strains were grown together and the total OD was measured (see Fig. 4). Not using any data on the relative frequencies of the strains, the Fig. 4 data was fitted to the summed population size, $N_1 + N_2$, of a 2-dimensional version of the same model

$$\frac{\mathrm{d}N_1}{\mathrm{d}t} = \frac{q_{0,1}}{q_{0,1} + \mathrm{e}^{-m_1 t}}\, r_1 N_1\left[1 - \left(\frac{N_1}{K_1}\right)^{v_1} - c_2\,\frac{N_2^{v_2}}{K_1^{v_1}}\right] ,$$

$$\frac{\mathrm{d}N_2}{\mathrm{d}t} = \frac{q_{0,2}}{q_{0,2} + \mathrm{e}^{-m_2 t}}\, r_2 N_2\left[1 - \left(\frac{N_2}{K_2}\right)^{v_2} - c_1\,\frac{N_1^{v_1}}{K_2^{v_2}}\right] .$$

The two c_i parameters provide the relative competitive strength of the other species (when $c_i = 1$ the intraspecific competition is equal to the interspecific competition). Fitting this model with 14 free parameters to the sigmoid curve of data points representing the total OD (see Fig. 4) would clearly be unfeasible. Instead, only c_1 and c_2 were estimated, and the 12 other parameters were copied from the fitting of the Fig. 3 data.

Finally, the model is run numerically for the best fitted parameters and the thus predicted frequencies of each strain, $\frac{N_1}{N_1+N_2}$ and $\frac{N_2}{N_1+N_2}$, are plotted together with frequencies measured by flow cytometry (see Fig. 5). The main result of the paper is that these predictions work well, and that hence for estimating the relative fitness, we need not collect frequency data. Having two individual growth curves and a total density curve from a combination experiment, i.e., three growth curves, suffices for generating simulation data from which a classic relative fitness parameter can be estimated. Additionally, using this approach one can attribute fitness differences to a variety of parameters, rather than to a single selection coefficient.

Today you will first repeat this analysis for one experiment (start with A), by fitting the data in Fig. 3 to the first model, copying the estimated parameters for the second model, to estimate the competition

parameters, c_i, from the data in Fig. 4 (write the c_i values down), and predict the frequency data in Fig. 5. This can all be done with the R-script `ram.R` using the wrapper `grind.R`. Note that the 3 data sets as we have retrieved them from their public repository are identified by a date (uncomment one of the 3 lines at the start of the `ram.R` script), and that strains are called 'R' and 'G', for red and green, respectively.

Using Grind. You will work with an R-script called `grind.R` which is a wrapper around the R-packages `deSolve`, `FME` and `rootSolve` developed by Karline Soetaert and colleagues [2–5]. These packages allow one to solve differential equations, find their steady state, and perform nonlinear parameter estimation. Today you only need two of Grind's easy-to-use functions:

- `run()` integrates a model numerically and provides a time plot,
- `fit()` fits a model to data by estimating its parameters, and depicts the result in a timeplot.

A full manual of Grind is available in the form of a tutorial (tbb.bio.uu.nl/rdb/grindR/tutorial.pdf). For today you need to know that the vector `p` contains the parameters, and the vector `s` the initial condition (the state). The vector `lower` in the call to `fit()` defines the lower bound of the parameters to be estimated (`lower=0` guarantees that they are all positive, use `lower[i]=1` to set the lower bound of the i^{th} parameter to one). With the vector `free` in `fit()` you can define which parameters are free and should be estimated. With the vector `differ` in `fit()` you can define which free parameters differ between the data sets. With the text `tweak` in `fit()` you can add columns to the numerical solution before this is forwarded to the fitting algorithm (e.g., add a total or a frequency). Data sets can be fitted simultaneously when they are provided as a list of data sets. Most of these options will become clear by running the example step-by-step.

We will work in the RStudio environment. To get started perform the following:

- You may need to install RStudio, R, and the Soetaert libraries. Install the three Soetaert libraries by opening RStudio, going to the Tools menu, choosing Install Packages, and then typing `deSolve`, `FME` and `rootSolve`.
- Download the practicals/bacteria/ram.zip file and unzip it somewhere in your file system. This creates a folder containing the `grind.R` and `ram.R` scripts and three subdirectories called Fig3, Fig4 and Fig5, which contain the data in the form of 'csv'-files.
- Open `ram.R` to startup RStudio (you may have to set the working directory in RStudio to the folder containing the R-script (`Set working directory` in the `Session` menu of RStudio)). Files will then be opened from that directory.
- Use "Open file" to open `grind.R` and **source** Grind to define the functions (button on the right).
- Switch back to the `ram.R` tab, and **run** that script line by line.

Next time, first "Source" the `grind.R` file (button in right hand top corner), and then "Run" the `ram.R` script line by line (select a line and hit "Run" or "Control Enter").

Question 1. Repeat the Ram *et al.* [1] analysis for at least experiment A and B.

a. Compare your parameter estimates with those in Table S2. We transform the data by taking the log. Why do we do that, did they do that, and does it make a difference?

b. What happens to the estimated parameters if the lower bound of the v parameter is zero (rather than one), and what does this imply for the form of density dependent function?

c. What happens to the estimated parameters if the lag parameters, q_0 and m, are not ignored when fitting the data from Fig. 3B?

d. What are the c_i estimates, and what do they mean? How much do you loose in the quality of the fit (i.e., in the SSR), when the c_i parameters are forced to be equal?

Question 2. When the total density of two competing strains is fitted to the OD data, each strain is given its 6 best estimated parameters and a free competition parameter, c_i, for weighting the relative competitive effect of the other strain. Although only the two competition parameters, c_i, are being estimated in Fig. 4, the two strains ultimately differ in all 7 parameters (i.e., the model to describe all

data has 14 parameters).

a. Can you describe the differences between the strains in Fig. 3 with fewer parameters? Hint: use the option `differ=c("m","v","K")` to Grind's fit-function to fit the two data sets simultaneously assuming that only these 3 parameters have to estimated separately, and that the two strains have identical values for the other free parameters.

b. Can you also fit the Fig. 3 and Fig. 4 data together, and find an even more minimal difference between the two strains in an experiment? How would you predict the Fig. 5 data given these minimal differences. Which parameter(s) play a major role in defining the relative fitness of these two strains?

Question 3. The only difference between the red bacteria A1 and B1 in Fig. 3 should be that the A1 bacteria had to wake up from their stationary phase, whereas the B1 bacteria were pre-grown in fresh media (the same should be true for the difference between the green A2 and B2 bacteria). By estimating these data separately, the A1 and B1 are different in all six parameters, however (and so are A2 and B2). For instance, the parameter v for the curvature of the density dependence could not be estimated for A1 whereas $v = 1.49$ for B1, while this strain should basically have a single curvature of its density dependence function.

a. Can you describe the A1 and A2 data while assuming that the two data sets only differ in the parameters for the lag phase (q_0 and m) and possibly the initial condition? If you can, would you then trust the other estimated differences between the two data sets? What would be your 'best' parameters for the first strain in both experiments?

b. What were your estimates for the four c_i parameters in the A and B experiment? Do you expect c_1 to differ between experiment A and B? And c_2? How would you test this? (Just propose how this *could* be done, because actually coding this in R may be too laborious for a single afternoon).

Question 4. The relative fitness of the first and second strain differs between experiment A and B because the lag phase plays hardly any role in B, and is quite long in A. Wouldn't it be best to find out which parameters really need to be different between the strains in all 6 experiments (i.e., Fig. 3 A1, B1, A2 and B2 and Fig. 4 A and B), since all experiments were done with the same two strains? How would you test this? (Again, coding this may be too tricky for a single afternoon).

Question 5. This is a question for students having a more technical background: The data for every strain actually consists of 30 or 32 independent experiments that we have joined into one data set. One can also fit the model to each individual culture, and hence obtain 30 or 32 estimates for each of the parameters, and report an average or median, with some range around it.

a. What did the authors do, and how do they obtain the confidence ranges in Table S2? How come that they have so narrow confidence ranges while we find that we can choose quite different values for several of the parameters?

b. What would the relative fitness be if the Fig. 5 data were fitted to the classic model from population genetics? This model is derived from

$$\frac{\mathrm{d}N_1}{\mathrm{d}t} = rN_1 \quad \text{and} \quad \frac{\mathrm{d}N_2}{\mathrm{d}t} = r(1+s)N_2 \quad \text{yielding} \quad \frac{\mathrm{d}f}{\mathrm{d}t} = rsf(1-f) \ ,$$

where $f = \frac{N_2}{N_1+N_2}$ is the fraction of the strain having a selection coefficient s (see the Appendix). What would the relative fitness be if this model were fitted to the predicted frequencies? How many identifiable free parameters does this model have if it is fitted to frequency data only? Would the estimated relative fitness depend on the growth conditions?

August 15, 2020, Rob J. de Boer, Utrecht University

References

[1] **Ram, Y., Dellus-Gur, E., Bibi, M., Karkare, K., Obolski, U., Feldman, M. W., Cooper, T. F., Berman, J., and Hadany, L.**, 2019. Predicting microbial growth in a mixed culture from growth curve data. Proc. Natl. Acad. Sci. U.S.A. **116**:14698–14707.

[2] **Soetaert, K.**, 2009. rootSolve: Nonlinear root finding, equilibrium and steady-state analysis of ordinary differential equations. R package 1.6.
[3] **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.
[4] **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.
[5] **Soetaert, K., Petzoldt, T., and Setzer, R. W.**, 2010. Solving differential equations in R: Package deSolve. Journal of Statistical Software **33**:1–25.

ram/ram.R

```
# The 3 experiments are indexed by their date:
expt <- "2015-11-18"  #Experiment A
#expt <- "2015-12-14"  #Experiment B
#expt <- "2016-01-06"  #Experiment C

# First read and plot all data:

fig3R <- read.csv(paste("Fig3/",expt,"_R.csv",sep="")) # Red
fig3G <- read.csv(paste("Fig3/",expt,"_G.csv",sep="")) # Green
plot(fig3R$Time, fig3R$OD, ylim=c(0,0.8),col="red",pch=".",xlab="Time (hr)",ylab="OD")
points(fig3G$Time, fig3G$OD, ylim=c(0,0.8),col="green",pch=".")

fig4 <- read.csv(paste("Fig4/",expt,"_RG.csv",sep=""))
plot(fig4$Time, fig4$OD, ylim=c(0,0.8), col="blue", pch=".", xlab="Time (hr)", ylab="
    Total OD")

fig5 <- read.csv(paste("Fig5/flow_df_",expt,".csv",sep=""))
fig5G <- fig5[fig5$Strain=="Green",]
fig5R <- fig5[fig5$Strain=="Red",]
plot(fig5G$time, fig5G$freq_mean, ylim=c(0,1), col="green",xlab="Time (hr)", ylab="
    Frequency")
points(fig5R$time, fig5R$freq_mean, col="red")

# Here the fitting starts

model <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    v <- max(1,v)
    a <- q0/(q0+exp(-m*t))
    dN <- r*a*N*(1-(N/K)^v)
    return(list(dN))
  })
}

s <- c(N=0.124)
p <- c(K=0.6,r=0.4,m=2,q0=0.005,v=2)
free <- c("N",names(p))
lower <- rep(0, length(free)); lower[match("v",free)] <- 1; lower  # set lower bounds
data3R <- as.data.frame(cbind(fig3R$Time,fig3R$OD)); names(data3R) <- c("time","N")
fit3R <- fit(data3R,free=free,fun=log,lower=lower,pch=".",legend=FALSE,tstep=0.1,main="
    red")
summary(fit3R)

data3G <- as.data.frame(cbind(fig3G$Time,fig3G$OD)); names(data3G) <- c("time","N")
fit3G <- fit(data3G,free=free,fun=log,lower=lower,pch=".",legend=FALSE,tstep=0.1,main="
    green")
summary(fit3G)

# The 2D model is defined:

model2 <- function(t, state, parms) {
  #state <- ifelse(state < 0, 0, state)
  with(as.list(c(state,parms)), {
    v1 <- max(1,v1); v2 <- max(1,v2)
    a1 <- q01/(q01+exp(-m1*t)); a2 <- q02/(q02+exp(-m2*t))
```

```
    dN1 <- r1*a1*N1*(1-(N1/K1)^v1-c2*(N2^v2)/(K1^v1))
    dN2 <- r2*a2*N2*(1-(N2/K2)^v2-c1*(N1^v1)/(K2^v2))
    return(list(c(dN1,dN2)))
  })
}

# Retrieve parameters from the "fit$par" list and rename them for the 2D model

pR <- fit3R$par[2:length(fit3R$par)]; names(pR) <- paste(names(pR),"1",sep="")
pG <- fit3G$par[2:length(fit3G$par)]; names(pG) <- paste(names(pG),"2",sep="")

data4 <- as.data.frame(cbind(fig4$Time,fig4$OD)); names(data4) <- c("time","OD")
p <- c(pR,pG,c1=1,c2=1); p["v1"] <- max(1,p["v1"]); p["v2"] <- max(1,p["v2"]); p
initialOD <- data4[1,2]
s <- c(N1=initialOD/2,N2=initialOD/2);s  # assume the expt was started equally
free <- c("c1","c2")
fit4 <- fit(data4,odes=model2,tweak="nsol$OD=nsol$N1+nsol$N2",free=free,fun=log,lower
    =0,upper=2,pch=".",legend=FALSE,tstep=0.1,main="blue")
summary(fit4)

p["c1"] <- fit4$par["c1"]; p["c2"] <- fit4$par["c2"]; p # Retrieve parameters from "fit
    $par"
initialF <- fig5R$freq_mean[1]
s <- c(N1=initialF*initialOD,N2=(1-initialF)*initialOD)
nsol <- run(6,0.1,odes=model2,tweak="nsol$fR=nsol$N1/(nsol$N1+nsol$N2);nsol$fG=1-nsol$
    fR",table=TRUE)
plot(nsol$time,nsol$fR,ylim=c(0,1),type="l",col="red")
points(fig5R$time, fig5R$freq_mean, col="red")
lines(nsol$time,nsol$fG,col="darkGreen")
points(fig5G$time, fig5G$freq_mean, col="darkGreen")

# Here the paper ends


# This is an example showing how to fit two data sets simultaneously

s <- c(N=0.124)
p <- c(K=0.6,r=0.4,m=2,q0=0.005,v=2)
free <- c("N",names(p))
differ <- c("N","K","v","m")
totfree <- c(free[!(free %in% differ)], differ, differ); npar <- length(totfree)
cat("Number of free parameters",npar)
lower <- rep(0,npar); lower[which(totfree == "v")] <- 1; lower  # set lower bounds
fitq1 <- fit(data=list(data3R,data3G),free=free,differ=differ,fun=log,lower=lower,pch="
    .",legend=FALSE,tstep=0.1,main="red & green",add=TRUE)
summary(fitq1)

# This is an example of fitting the population genetics model to frequency data

model <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    df <- r*s*f*(1-f)
    return(list(df))
  })
}

s <- c(f=0.5); p <- c(r=1,s=0.1); free=c("f","s")

data <- as.data.frame(cbind(fig5R$time,fig5R$freq_mean)); names(data) <- c("time","f")
fitq4d1 <- fit(data,free=free)
data <- as.data.frame(cbind(nsol$time,nsol$fR)); names(data) <- c("time","f")
fitq4d2 <- fit(data,free=free)
```

Appendix

Consider two exponentially expanding populations, e.g., a wild type N_1 and a mutant N_2,

$$\frac{\mathrm{d}N_1}{\mathrm{d}t} = rN_1 \quad \text{and} \quad \frac{\mathrm{d}N_2}{\mathrm{d}t} = r(1+s)N_2 \ ,$$

where s is the selection coefficient of the mutant (s can be positive or negative). In a competition experiment one would plot how the fraction mutant $f \equiv N_2/(N_1+N_2)$ evolves over time. To compute how the fraction $f(t)$ changes one needs to employ the quotient rule of differentiation: $[f(x)/g(x)]' = (f(x)'g(x) - f(x)g(x)')/g(x)^2$. Thus, using $'$ to denote the time derivative, one obtains for $\mathrm{d}f/\mathrm{d}t$:

$$\begin{aligned}
\frac{\mathrm{d}f}{\mathrm{d}t} &= \frac{N_2'(N_1+N_2) - (N_1'+N_2')N_2}{(N_1+N_2)^2} \ , \\
&= \frac{N_2'N_1 - N_1'N_2}{(N_1+N_2)^2} \ , \\
&= \frac{r(1+s)N_2N_1 - rN_1N_2}{(N_1+N_2)^2} \ , \\
&= r(1+s)(1-f)f - rf(1-f) \ , \\
&= rsf(1-f) \ ,
\end{aligned}$$

with the solution

$$f(t) = \frac{1}{1+\mathrm{e}^{-rst}\ \frac{1-f(0)}{f(0)}} \ ,$$

that is also written in the legend of Fig. 5 of the Ram *et al.* [1] paper.