

Worksheet 1; Fitting a single data curve using the function *Bayesfit*

Lydia Rickett, Contact: Lydia.Rickett@sainsbury-laboratory.ac.uk

- Open R Studio under programs and navigate to the shared folder (using “File” and “Open file...” in the toolbar at the top). In the folder “Program” open the file *example1.R*. Begin by clicking on “Session” on the toolbar and then “Set Working Directory” and “To Source File Location”. This makes sure that R is working in the same folder as our file.

USEFUL COMMANDS

* When running code, we can either run one line of code at a time by placing the cursor on the line and clicking on “Run” on the top right of the sub window (or pressing Ctrl+Enter) or we can run everything by clicking on “Source” (or Ctrl+Alt+R).

* <- is the assignment operator in R (you can use = instead if you prefer).

* `# This is a comment`

(Green) lines starting with a “ # ” symbol are commented out and do not run. To uncomment (or comment) lines so that they run (or do not run), either remove (or add) the “ # ” symbols, or place your cursor at the line or highlight the section you wish to change and click on “Code” on the toolbar at the top and then “Comment/Uncomment Lines” (or press Ctrl+Shift+C).

* The “ \$ ” symbol is used to extract data from a data frame. For instance `collectionofdata$times` picks the data `times` from the data frame `collectionofdata`.

For more help with R, visit <https://support.rstudio.com/hc/en-us/articles/200552336-Getting-Help-with-R>. A complete list of shortcuts can be found at <https://support.rstudio.com/hc/en-us/articles/200711853-Keybaord-Shortcuts>.

1 Fitting a growth curve

We begin by loading the code *Bayessingle.R* and importing and plotting the experimental data *B092_1.csv*.

- Run each of the lines of *example1.R* to perform these commands (as shown below).

```
source("Bayessingle.R")
data <- read.csv("data/B092_1.csv", header=TRUE, sep=",",
               na.strings=c("ND", "NA"))
plot(data, ylim=c(0,10))
```


1.2 Extension: fitting undetected data points

Next, we will look at fitting some data with undetected points; these are points with bacterial concentration below the level (or “threshold”) that we were able to detect using the experimental method.

- Uncomment and run the lines to import and plot curve *B119_5.csv*, which contains undetected points, choosing the threshold value 1.3.

```
data <-read.csv("data/B119_5.csv",header=TRUE,sep=",",
               na.strings=c("ND","NA"))
plot(data,ylim=c(0,10))
threshold = 1.3
for (i in 1:nrow(data)){
  if (is.na(data[i,2]) == TRUE) {
    points(data[i,1],threshold,pch=16)
  }
}
```

- Uncomment and run the line to perform the analysis including undetected points with threshold 1.3. Extract and plot the results as before.

```
results_Bar4par <- Bayesfit(data,model="Bar4par",inc.nd=TRUE,
                           threshold=1.3)
```

- Repeat the above steps with a different threshold value to see how this affects the results.

2 Performing model comparison

The available models are summarised in table 1.

- Start a new plot with the original experimental data. By uncommenting and running the relevant lines perform Bayesian analysis, fitting using the different models. An example is shown below.

```
results_logistic <- Bayesfit(data,model="logistic",inf.sigma=FALSE)
```

- By running the relevant lines, extract and plot the curves fitted by the various models.

```
t <- results_logistic$fit.t; y <- results_logistic$fit.ymean
lines(t,y,col="green",lwd=2)
```

- Extract the log evidences for the various models by running the relevant lines in the section “Compare models using Bayes’ factor”.

```
logevidence_logistic <- results_logistic$logevidence
```

- Calculate and print the results for the Bayes’ factor for the 4 parameter Baranyi versus logistic model. The results can be interpreted using table 2. Some typical output, similar to that which will appear in the command window, is shown below.

```
Bayes <- exp(logevidence_Bar4par)/exp(logevidence_logistic)
cat("Bayes' factor for first vs. second model = ", Bayes, '\n')

## Bayes' factor for first vs. second model = 0.1342

cat("Log Bayes' factor = ", log(Bayes), '\n')

## Log Bayes' factor = -2.009
```

- Try changing the models that we are comparing (by changing the model names) to see how the Bayes’ factor changes. Which is the preferred model for the data? To what degree is it preferred over the second most likely model?

Model name	Parameters
linear	y_0, μ_{max}
logistic	y_0, y_{max}, μ_{max}
Bar3par	y_0, μ_{max}, h_0
Bar4par	$y_0, y_{max}, \mu_{max}, h_0$
Bar6par	$y_0, y_{max}, \mu_{max}, \lambda, \nu, m$

Table 1: The bacterial growth models available in the function. $y_0 = \log_{10}(x_0)$, where x_0 is the initial bacterial concentration, $y_{max} = \log_{10}(x_{max})$ where x_{max} is the maximum of the bacterial concentration, μ_{max} is the maximum specific growth rate, λ is the lag time, $h_0 = \mu_{max}\lambda$, and ν and m control the curvatures from the lag to exponential phase and exponential to stationary phase respectively.

Bayes' factor	Strength of evidence for/against first model or hypothesis
10 to ∞	Strong for
3 to 10	Substantial for
1 to 3	Barely worth mentioning for
$1/3$ to 1	Barely worth mentioning against
0.1 to $1/3$	Substantial against
$-\infty$ to 0.1	Strong against

Table 2: The regions of Jeffreys' scale for interpreting the Bayes' factor.

Note: We can also use the model comparison techniques to calculate the Bayes' factor for the noise level inferred versus prescribed.

3 Extension: plotting posterior samples

The last section of *example1.R* plots the curves fitted by the posterior parameter samples. These curves are coloured by their log likelihood values (with darker curves having the highest likelihood). The curves are plotted alongside the data and the curve fitted using the mean values of the parameter samples.

- Run the lines in the section “*Plot the curves fitted by posterior samples*” to plot the posterior curves.
- Try changing the results that we plot to see how the plot changes.

