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

1 Fitting a growth curve

Load the library *babar* and then we can import and plot some experimental data *B092_1.csv*.

```
library(babar)

B092_1.file <- system.file("extdata", "B092_1.csv", package = "babar")
B092_1.data <- read.csv(B092_1.file, header=TRUE, sep=",",
                 na.strings=c("ND","NA"))
plot(B092_1.data,ylim=c(0,10))
```

1.1 Estimating the noise level

We next use Bayesian analysis to fit a curve to the data using the *Bayesfit* function. We start by using the 4 parameter Baranyi model, Bar4par, assuming that the noise level of the data is the default value, 0.3.

Run the line to perform this command. This may take a minute or so to complete. Some typical output is shown below; this will appear in the command window.

```
set.seed(11) ## for reproducibility
results_Bar4par <- Bayesfit(B092_1.data,model="Bar4par",inf.sigma=FALSE)

## # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
## Model =  Bar4par
## mu_max prior type = Uniform
## # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
##
## log evidence =  -31.4846
## Means and standard deviations:
## Log cell count at time 0, y_0 = 3.308139 +/- 0.1642935
## Log final cell count, y_max = 8.815567 +/- 0.1326803
## Growth rate, mu_max = 0.2486727 +/- 0.01565881
## h_0 = 0.7267734 +/- 0.3265326
## Noise level, sigma = prescribed at 0.3
```

Extract a vector of mean fitted points from the results and plot them alongside our data.

```
t <- results_Bar4par$fit.t; y <- results_Bar4par$fit.ymean
lines(t,y,col="black",lwd=2)
```

Run this next line to perform the analysis with a different noise level, extracting and plotting the results as before, to see how the results change - you could try plotting using a different colour, making it easier to compare your results.

```
set.seed(11) ## for reproducibility
results_Bar4par <- Bayesfit(B092_1.data,model="Bar4par",inf.sigma=FALSE,
                            sigma=0.5)
```

Try changing the noise level manually to see how the results change again.

Repeat the analysis with the noise level inferred to see the difference.

```
set.seed(11) ## for reproducibility
results_Bar4par <- Bayesfit(B092_1.data,model="Bar4par")
```

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.

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

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

Run this line to perform the analysis including undetected points with threshold 1.3. Extract and plot the results as before.

```
set.seed(11) ## for reproducibility
results_Bar4par <- Bayesfit(B119_5.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 dataset (B092_1). An example is shown below of fitting using the different models.

```
set.seed(11) ## for reproducibility
results_logistic <- Bayesfit(B092_1.data,model="logistic",inf.sigma=FALSE)

## # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
## Model =  logistic
## mu_max prior type = Uniform
## # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
##
## log evidence =  -29.02413
## Means and standard deviations:
## Log cell count at time 0, y_0 = 2.943415 +/- 0.1267487
## Log final cell count, y_max = 8.865102 +/- 0.1403336
## Growth rate, mu_max = 0.2274081 +/- 0.01040893
## Noise level, sigma = prescribed at 0.3
```

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
plot(B092_1.data,ylim=c(0,10))
lines(t,y,col="green",lwd=2)
```

Extract the log evidences for a model by running a line like that below.

```
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)

## Bayes' factor for first vs. second model =  0.08539449
## Log Bayes' factor =  -2.460474
```

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}, μ_{max}, h_0
Bar6par	y_0, y_{max}, μ_{max}, λ, ν, 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.