Biofilm Traits

Jay T. Lennon

05 March, 2016

Setup Work Environment

```r
rm(list=ls())
getwd()
```

```
## [1] "/Users/lennonj/GitHub/BiofilmTrait/code/Gompertz/code"
```

```r
setwd("~/GitHub/BiofilmTrait/code/Gompertz/code")

# Load Dependencies
source("~/GitHub/BiofilmTrait/code/Gompertz/bin/modified_Gomp.R")

# Create Directory For Output
dir.create("~/GitHub/BiofilmTrait/code/Gompertz/output", showWarnings = FALSE)
```

Load data

```r
# Load complete dataset
g.data <- read.csv("../data/Biofilm_Gompertz.csv", sep = ",", header=TRUE)

# Paremeter estimates were very sensitive to 38 hr observation. Led to
# unrealistic values. So, we removed this time point from the following analysis.

g.data <- g.data[! g.data$Time. == "38",]
write.csv(g.data, "../data/Biofilm_Gompertz_Red.csv")
```

Run the modified Gompertz analysis

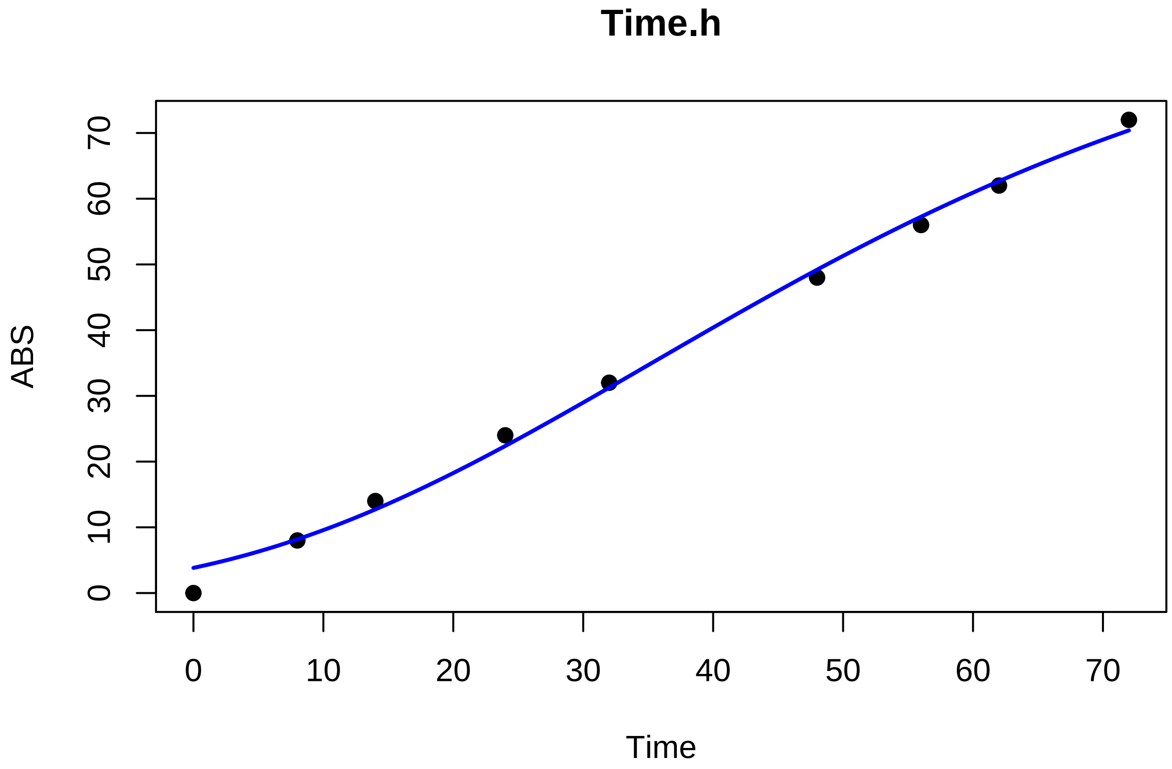
```r
growth.modGomp("../data/../data/Biofilm_Gompertz_Red.csv", "Gomp.Parms",
               intercept.guess = 0.001, synergy = F, temp = F, smooth = F)
```

```
## [1] Starting sample Time.h
```

```
## Loading required package: bbmle
## Loading required package: stats4
```

```
## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)
## [1] finished fit
```

```
## Profiling has found a better solution,so original fit had not converged:
##
## (new deviance=28.32, old deviance=34.92, diff=-6.602)
##
## Returning better fit ...
```

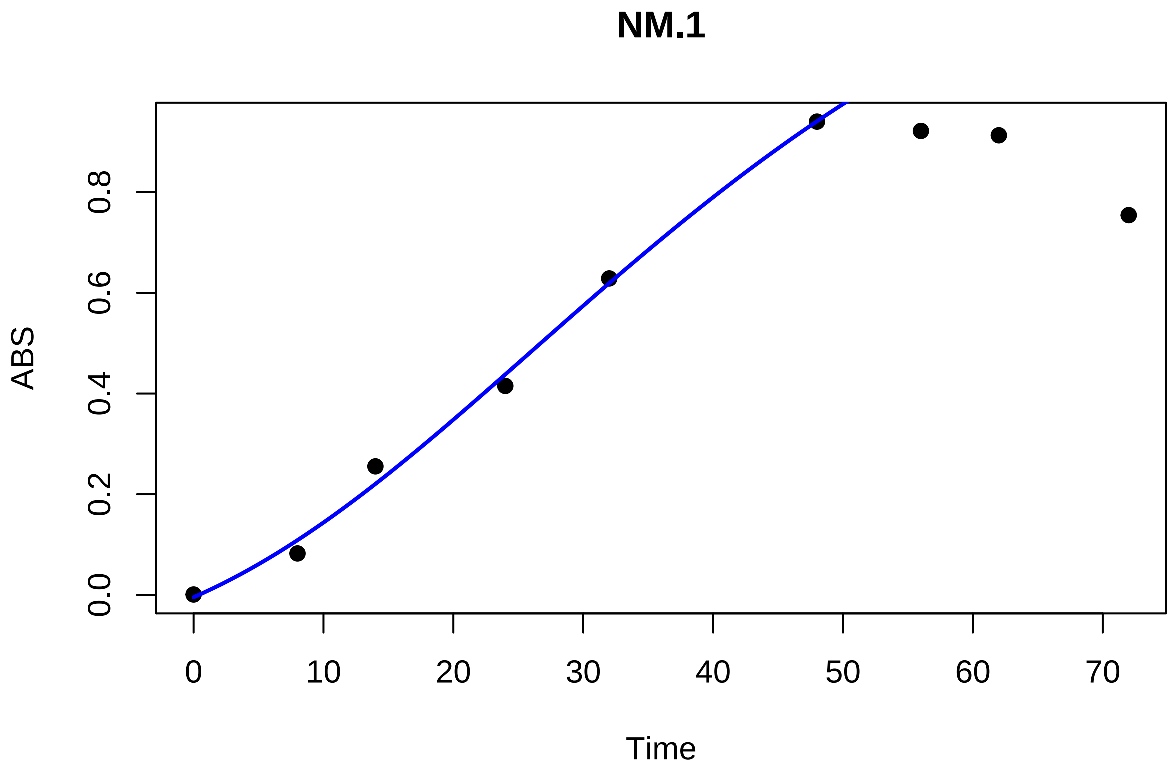


```
## [1] finished pf1
## [1] umax for Time.h = 1.146
## [1] 14% complete
## [1] Starting sample NM.1
## [1] finished fit
```

```
## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)
## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)
## Warning: stepsize effectively zero/flat profile (L)
## Warning: hit maximum number of steps (L)
```

```
## [1] finished pf1
```

```
## Warning: non-monotonic spline fit to profile (b0): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (A): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (umax): reverting from spline to linear approximation
## Warning: non-monotonic profile (L): reverting from spline to linear approximation (consider running
```

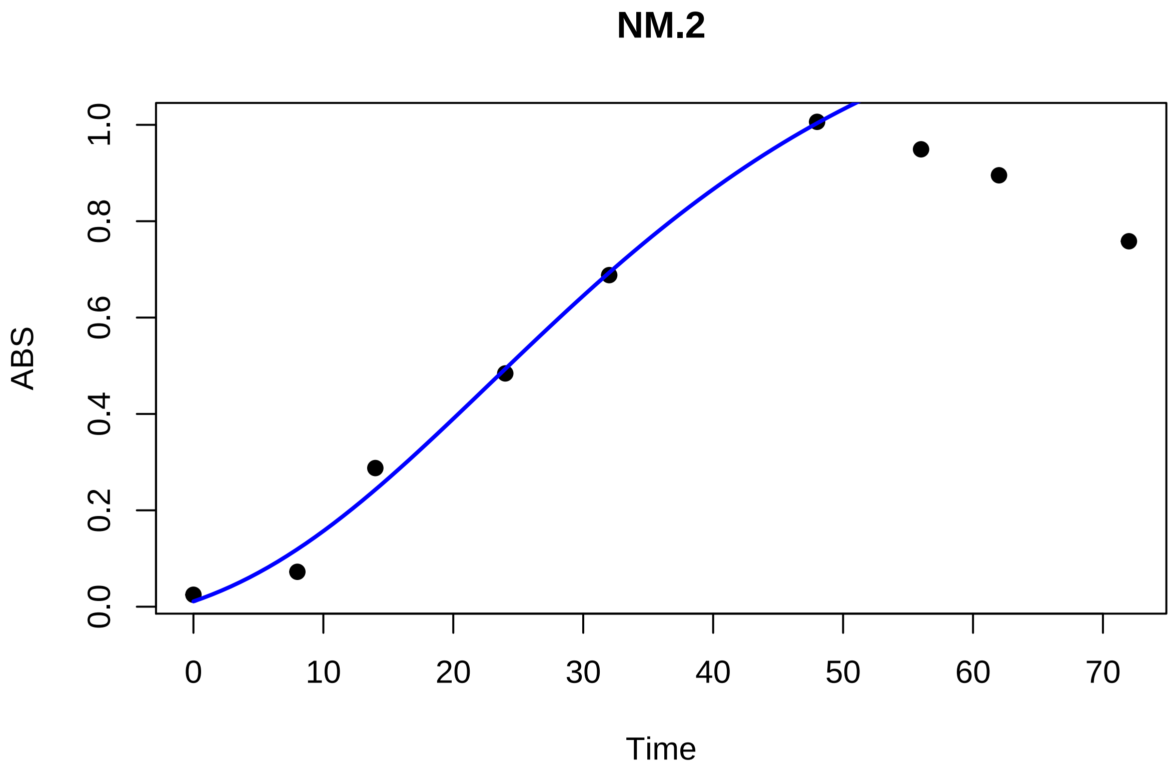


```
## [1] umax for NM.1 = 0.0228
## [1] 29% complete
## [1] Starting sample NM.2
## [1] finished fit
```

```
## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)
## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)
## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)
## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)
## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)
```

```
## [1] finished pf1
```

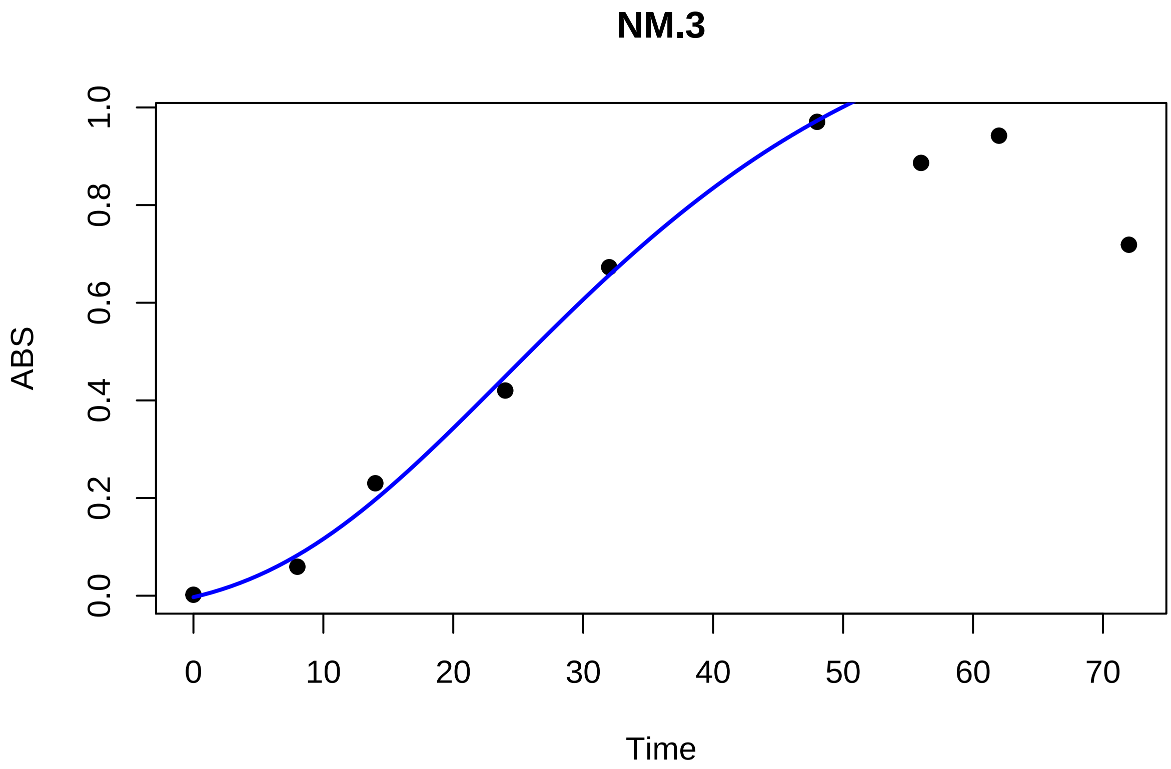
```
## Warning: non-monotonic spline fit to profile (b0): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (A): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (L): reverting from spline to linear approximation
```



```
## [1] umax for NM.2 = 0.0259
## [1] 43% complete
## [1] Starting sample NM.3
## [1] finished fit
```

```
## [1] finished pf1
```

```
## Warning: non-monotonic spline fit to profile (b0): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (A): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (L): reverting from spline to linear approximation
```

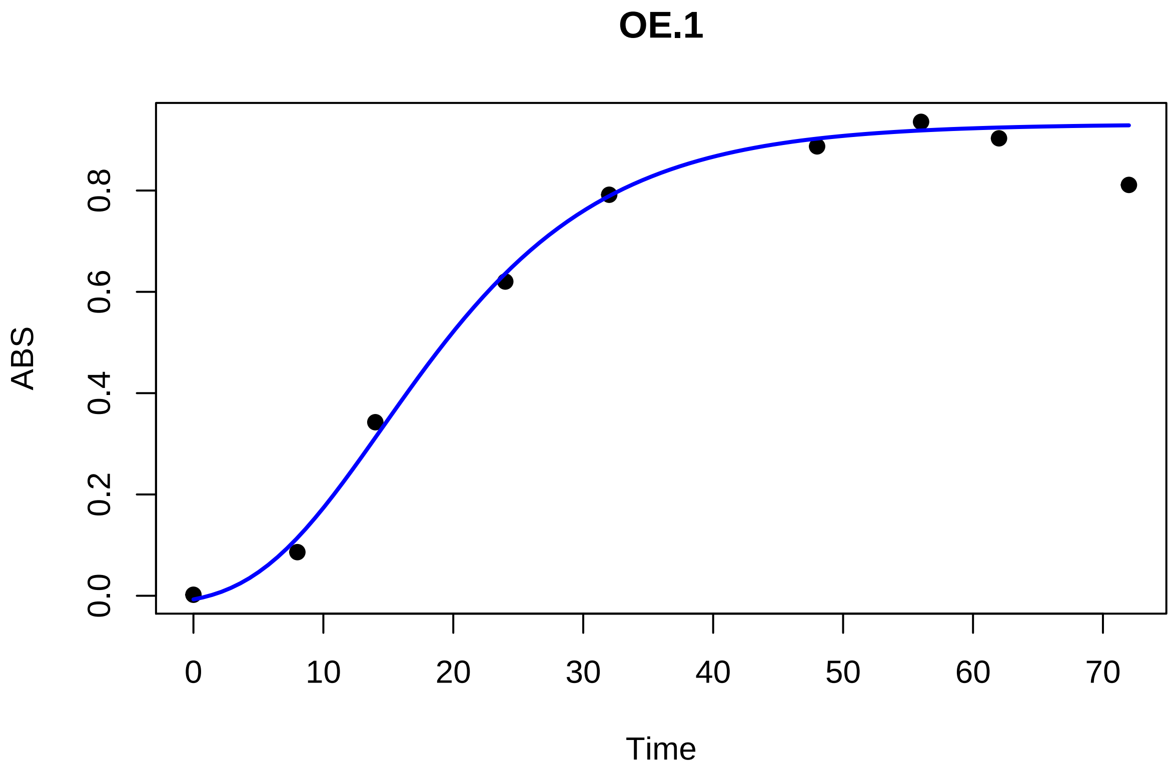


```
## [1] umax for NM.3 = 0.0267
## [1] 57% complete
## [1] Starting sample OE.1
## [1] finished fit

## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)

## [1] finished pf1

## Warning: non-monotonic spline fit to profile (b0): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (A): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (umax): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (L): reverting from spline to linear approximation
```

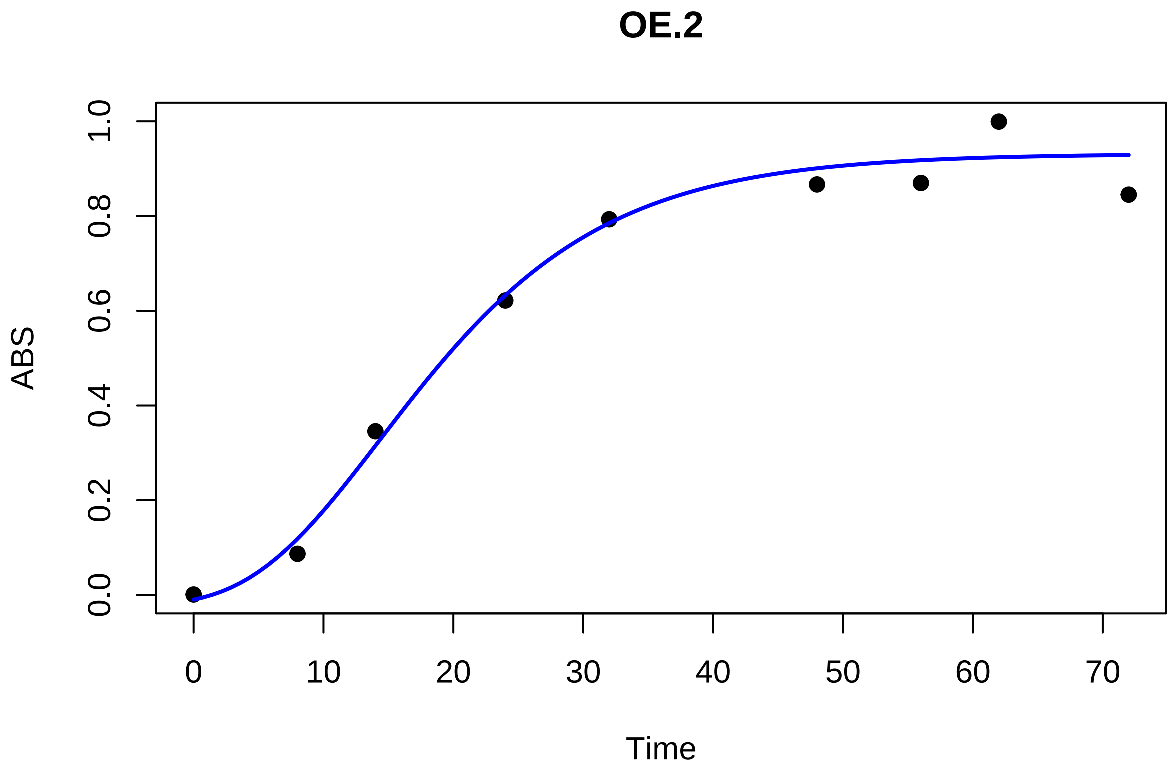


```
## [1] umax for OE.1 = 0.0364
## [1] 71% complete
## [1] Starting sample OE.2
## [1] finished fit

## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)
## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)

## [1] finished pf1

## Warning: non-monotonic spline fit to profile (b0): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (A): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (umax): reverting from spline to linear approximation
## Warning: non-monotonic profile (L): reverting from spline to linear approximation (consider running
```

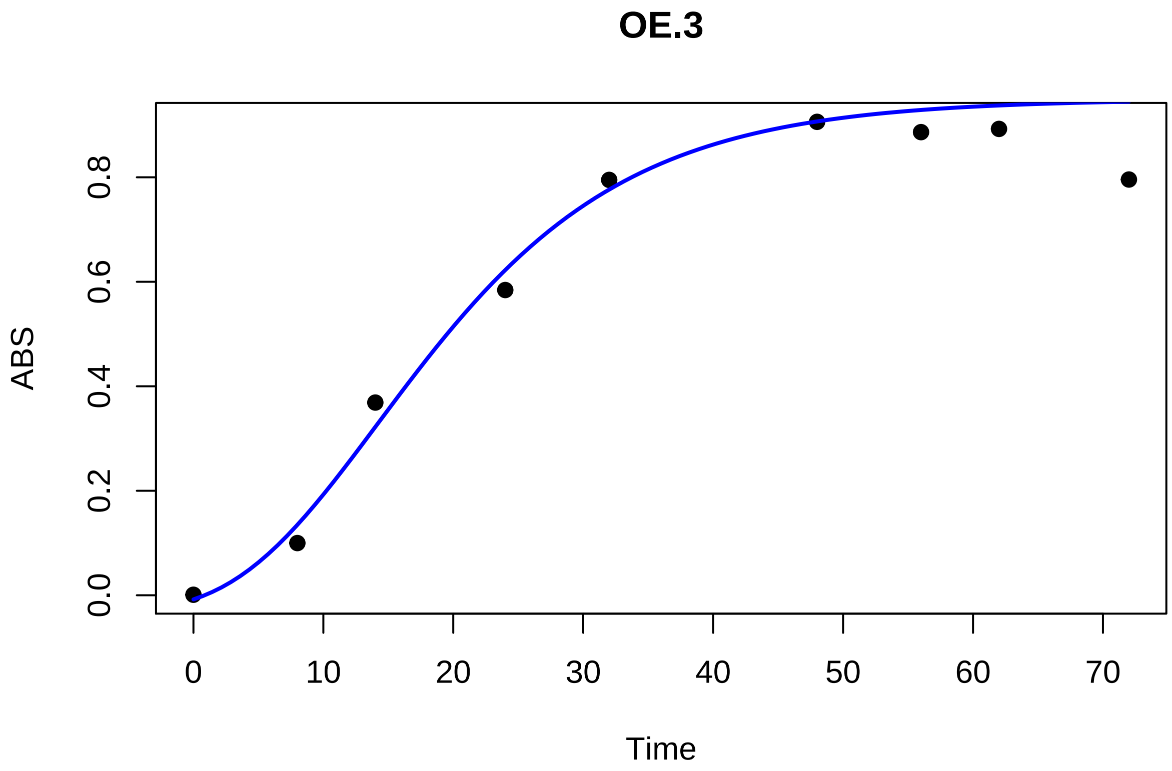


```
## [1] umax for OE.2 = 0.0357
## [1] 86% complete
## [1] Starting sample OE.3
## [1] finished fit
```

```
## Warning: convergence failure: code=1 (iteration limit 'maxit' reached)
```

```
## [1] finished pf1
```

```
## Warning: non-monotonic spline fit to profile (b0): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (A): reverting from spline to linear approximation
## Warning: non-monotonic spline fit to profile (umax): reverting from spline to linear approximation
## Warning: non-monotonic profile (L): reverting from spline to linear approximation (consider running
```



```
## [1] umax for OE.3 = 0.0333
## [1] 100% complete
```

Evaluate growth parameters for OE and NM

```
g.traits <- read.table("../output/Gomp.Parms.txt", sep=",", header=TRUE)
strain <- ifelse(grepl("NM", g.traits[,1]), "NM", "OE")
sem <- function(x) sqrt(var(x)/length(x))

# Final productivity (A)
A.means <- tapply(g.traits$A, strain, mean)
A.sem <- tapply(g.traits$A, strain, sem)
A.t.test <- t.test(g.traits$A ~ strain)

# Maximum growth rate (umax)
umax.means <- tapply(g.traits$umax, strain, mean)
umax.sem <- tapply(g.traits$umax, strain, sem)
umax.t.test <- t.test(g.traits$umax ~ strain)

# Lag time (L)
lag.means <- tapply(g.traits$L, strain, mean)
lag.sem <- tapply(g.traits$L, strain, sem)
lag.t.test <- t.test(g.traits$L ~ strain)
```