**The Baranyi Full models**

The analysis of the Baranyi model comes in two flavors. The full model in a Python class shows as

def BaranyiModelfunc(self, Ymax, Y0, mumax, h0, x): # x is time data

a = mumax\*x + np.log(np.exp(-mumax\*x) + np.exp(-h0) - np.exp(-mumax\*x-h0))

return Y0 + a - np.log(1.0 + (np.exp(a)-1.0)/np.exp(Ymax-Y0))

**Remember Y has to be in natural log, not log10. A conversion is needed if the users enter the log10 values of bacterial counts.**

mumax is the rate, and can be shown as max in the browser. In the Baranyi model, the lag parameter is defined as h0. The rest of the parameters is the same as Gompertz and Huang models, except in these models the lag parameter is actually the lag time of bacteria ().

The analysis of the Baranyi model has to be done in two steps. The first step is to analyze each curve to get max and h0. After all curves under different conditions are analyzed, get an average of all h0 and use it to find the max of each curve again. Therefore, the analysis of the Baranyi comes in two flavors.

**Flavor 1 – h0 is free**

For this flavor, wrapperBaranyi is used. It estimates every parameter (Y0, Ymax, max, h0).

**Flavor 2 – h0 is fixed**

For this flavor, wrapperBaranyiFixedH0 is ued. It estimates Y0, Ymax, max. But remember to get the h0 value.

**Model and Parameter UI**

For the Baranyi Full model, the parameter UI comes in two flavors. You can either use an option button to indicate a fixed h0 or use a popup alert to warn and allow the users to fix the h0:

1. Option 1: free h0 – use wrapperBaranyi.py
2. Option 2: fixed h0.- use wrapperBaranyiFixedH0.py

Once the user picks an option, call the wrapper to analyze the data. The wrappers have been tested.