# Systems Biology Final Project

Mathias D. Kallick

December 10, 2017

## 1 Kinetics and Motifs

For this project, we are looking at the feedback loop model from Becker-Weimann's 2004 paper "Modeling Feedback Loops of the Mammalian Circadian Oscillator". In order to do this, we must understand the kinetics that the authors use in this model. The primary type of kinetics used in this model is simply first-order mass action. Most of the equations for the variable are various forms of activation by other variables, combined with various forms of degradation of the variable itself. The two notable exceptions are $y_1$ and $y_4$, the concentration of *Per2* or *Cry* mRNA, and the concentration of *Bmal1* mRNA respectively. For $y_1$, we see what can be described as a AND NOT gate. In computational biology, we often describe certain mathematical constructs as gates, making an analogy to binary logic (although these gates are very much non-binary). An OR gate represents a situation where a variable might be activated by either one of two other variables - if either $X$ OR $Y$ are present, then the variable will be activated - the truth table for this can be see in Table 1. OR gates are often thought of as sum gates - you add the two variables together. The second common gate used is the AND gate - this represents the situation where we need both $X$ AND $Y$ to be present for the activation to occur. AND gates are often thought of as product gates - you multiply the two variables together. The truth table for AND gates can be seen in Table 2. In this case, the authors are using a gate very similar to an AND gate, but instead of having it necessary for both variables to be there, the first variable needs to be there and the second variable needs to not be there - this is what I describe as an AND NOT gate - the truth table for this can be seen in Table 3. The equation they use for this is a Hill function with a Hill coefficient of 1 - the equation is shown in Equation 1. The other interesting equation here is the one used for $y_4$ - This is a Hill function with a Hill coefficient $r$, which they use a relatively high value of 3 for. This can be seen in Equation 2

$$f(trans_{Per2/Cry}) = \frac{v_{1b} \cdot (y_7 + c)}{k_{1b} \cdot (1 + (\frac{y3}{k_{1i}})^p) + (y7 + c)} \tag{1}$$

$$f(trans_{Bmal1}) = \frac{v_{4b} \cdot y3^r}{k_{4b}^r + y3^r} \tag{2}$$

| $X$ | | $Y$ | | result |
|---|---|---|---|---|
| T | $+$ | F | $=$ | T |
| F | $+$ | T | $=$ | T |
| F | $+$ | F | $=$ | F |
| T | $+$ | T | $=$ | T |

Table 1: Truth table for an OR gate.

| $X$ | | $Y$ | | result |
|---|---|---|---|---|
| T | $\cdot$ | F | $=$ | F |
| F | $\cdot$ | T | $=$ | F |
| F | $\cdot$ | F | $=$ | F |
| T | $\cdot$ | T | $=$ | T |

Table 2: Truth table for an AND gate.

## 2 Numerical Solvers

In order to simulate this model, we coded up the ODE in both MatLab and Python - our main codebase is in Python, so it was important to have that code. However, we wanted to analyze the efficiency of various solvers for this model, and MatLab has a higher variety of available solvers than Python, so we also needed MatLab code for it. We were able to recreate Figure 3A from the paper (using Python), as can be seen in Figure 1.

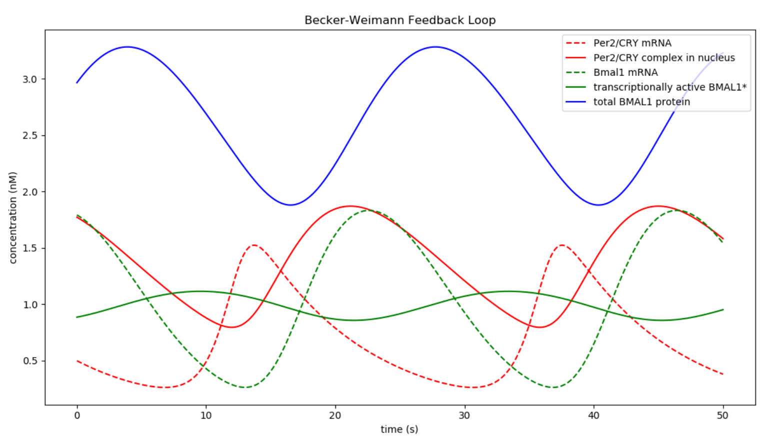


Figure 1: Recreation of Figure 3A from Becker-Weimann Paper. Recreated using ode15s in Python.

| $X$ | | $Y$ | | result |
|---|---|---|---|---|
| T | $\cdot$ | F | $=$ | T |
| F | $\cdot$ | T | $=$ | F |
| F | $\cdot$ | F | $=$ | F |
| T | $\cdot$ | T | $=$ | F |

Table 3: Truth table for an AND gate.