

REPORT

LOGIC BASED MODELING
METHODS FOR BIOLOGICAL
NETWORKS

NOOR PRATAP SINGH

INTRODUCTION

Usually based on data and literature and based on our understanding of system we form a network. But not always just from looking at it and from intuition alone about the network we cannot form the whole understanding of it.

Also which modeling method we chose depends on the following:

- Overall structure of the network
- Availability of Data
- Questions we want to address

There are two extreme approaches that exist when we model the system. On one extreme is the ODE(Ordinary differential equations) based approach and on the other is the PINS(Protein Interaction Networks).

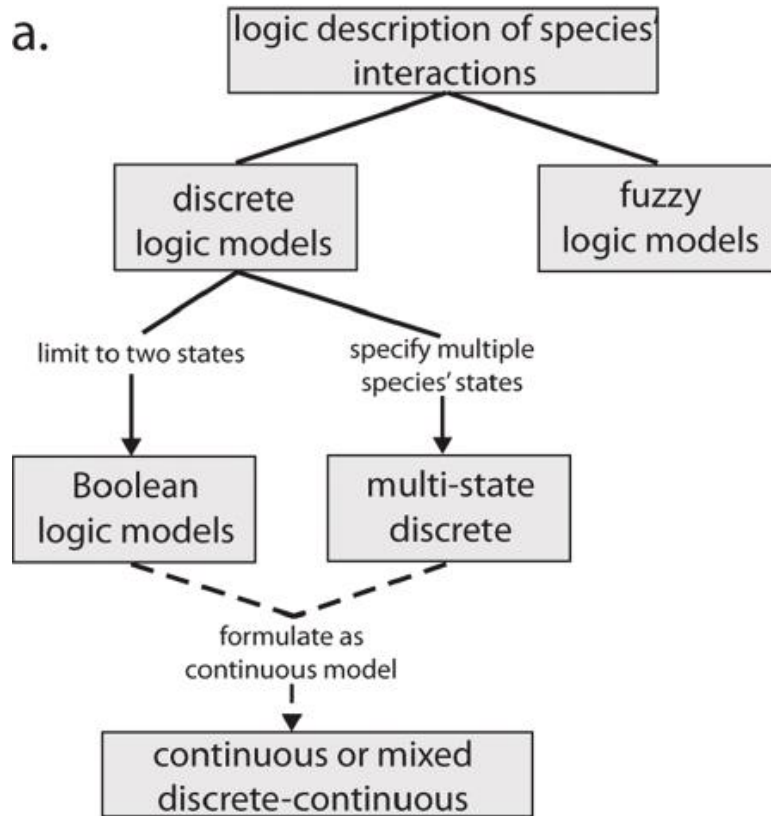
ODE based approach bear the closest relationship to a biological network. They involve forming various rate equations between the different components of a network and solving them. Though in them we get the required input-output relations but the biggest problem with them is that they involve knowing a lot parameters or (they have to be guessed and later modified based upon the fitness to actual data). Also the rate equation becomes complex as the overall length of the network increases and solving it becomes more and more difficult.

PINS on the other hand represents a graphical representation of various proteins of the network shown as nodes in which interactions between them shown as edges. It could be inhibiting or promoting. They are formed based on literature or through statistical methods. Though larger networks can be studied but their main drawbacks are that we cannot use them to compute input-output relations or dynamical changes in components with respect to other.

Logic Based Modeling is the midway approach of the methods described above. What exactly it is will be described in brief:

- First a network is formed and connections are specified.
- To map the dependency between the input and output in the network we have gates.
- These gates have a set of rules which define output states depending on the input states.

Methods



Above image shows the various modeling methods that exist in logic based modeling. I worked on **Boolean**, **multi-state** and **fuzzy logic models**. Here I shall be describing what exactly these methods are and in discussions I shall be describing the network on which I worked based upon the above models.

Boolean - As the name suggests it is self explanatory. We have only 2 states 0 and 1(on or off). The interactions between the components in a network are replaced by Boolean Gates(And, Or, Not etc) and based on those gates and inputs we calculate the state of downstream proteins (0 or 1).

Multi-State - In them apart from 0 or 1 we could have other states also such 2,3 etc depending on the system we are to model. The output again would be calculated based upon how the gate functions depending upon the inputs.

Time Updates - This is a major consideration. How exactly do we update a system. Does the change happen instantly or there is a time delay. Usually output at a time depends on the input of a gate at some previous time. Two approaches exist here:

- Synchronous Update - In it the output value of a node at a given time step($t = i$) depends on the value of the input at previous time step($t = i - 1$).
- Asynchronous Update - In it output value of some of nodes at a given time step($t = i$) depends on the value of input node at previous node at time($t = i - 1$) and others at the current time step at ($t = i$).

The above picture will be more clear when we go under discussions.

Fuzzy Logic - Before diving into it we need to understand the concept of **fuzzy sets**. The **normal(crisp)** sets either includes or excludes an element from the universal set whereas in a **fuzzy set** partial membership can take place.

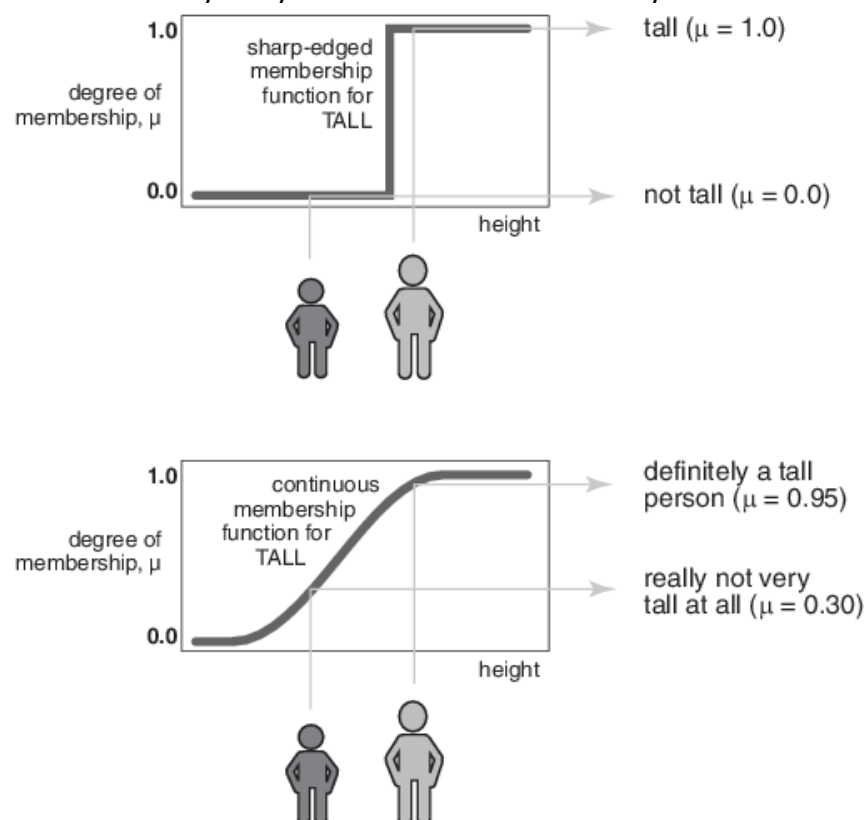
Eg - Lets consider a Biology class. It includes Btech students as well as Mtech students and the professor. The Mtech students have some background of Biology whereas Btech guys are novice. The class also includes the professor. We have defined the Universal Set.

Now lets say our set is "Persons who are proficient in Biology". Going by the normal set theory we could safely include the professor in it and exclude Btech guys but what about Mtech guys. You can't exclude them or include them. Where to put them?

Consider another set of all people and we have a set of tall people. Now we say everyone greater than 6 ft is tall. So definitely a person with a height greater than 6 ft is included and a person with height 5 ft is excluded from the normal set. But what about a person with height 5ft 9". Where does he go?

This is where fuzzy sets come. In it to each member of universal set we assign a degree of membership(b/w 0-1)(**DOM**) based on the property we are trying to study. 1 standing for complete relation and 0 for almost nil. So therefore in our examples faculty gets a DOM 1 and Btech a DOM somewhere near 0 though not exactly 0. Similarly MTech gets some DOM b/w 0.4-0.9 depending upon their knowledge. For tall we can say 5ft has DOM 0 and 5ft6" has DOM lets say 0.45 and 5ft10" has DOM 0.9. Thereby every element of Universal set is in fuzzy set.

The assigning of DOM to universal set(US) is done by **Membership Function(MF)** which specifies some property we are interested (eg *Tall* could be one MF and *short* could be another MF though both applies to same US) . Applying a MF on a US forms a **fuzzy set**. Note that every fuzzy set is a normal set but every normal set is not a fuzzy set.



The above image shows at top a normal set(excluded = DOM 0) whereas below one shows the fuzzy set on which tall MF has been applied. Similarly a **low MF** could have been applied in which a 5ft could have been assigned a DOM 1 and 6 ft DOM 0.

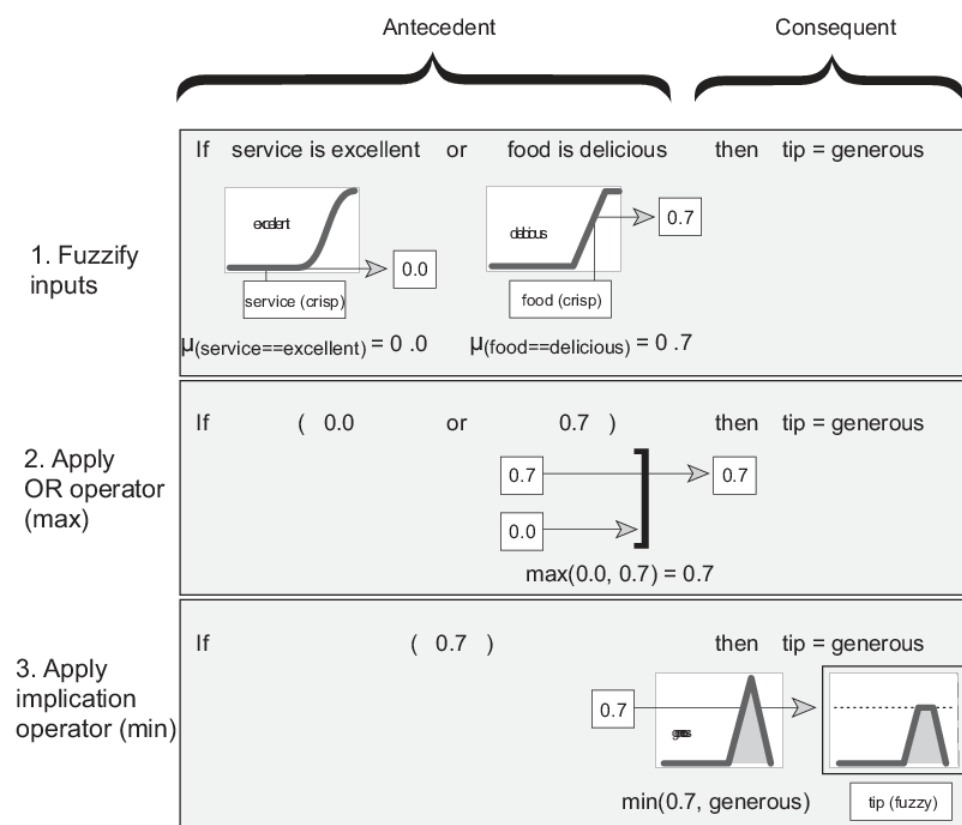
If then rules -These are the basis of computation and sort of form gates. These are of form If x is A then y is B where x,y are US and A,B are MFs. The part before then is Antecedent and part after is Consequent.

Eg *If service(x) is good(A) then tip(y) is average(B)*

If temperature(x) is cold(A) then hot water valve(y) is open(B) and cold water valve(z) is shut(C).

Fuzzy Inference Process

The following points describe the basic processes involved in fuzzy logic which calculate the final output based on inputs. These will be explained with the help of below image. Mind you below image shows only **one rule**.



1) Fuzzification - In it MFs are applied to US(excellent , delicious and generous applied to service, food and tip in above example).

2) Application of fuzzy operator in Antecedent - Usually antecedent will be plural and needs to be combined in order to apply(Or in above example could have been And also. Or means taking max of DOM)

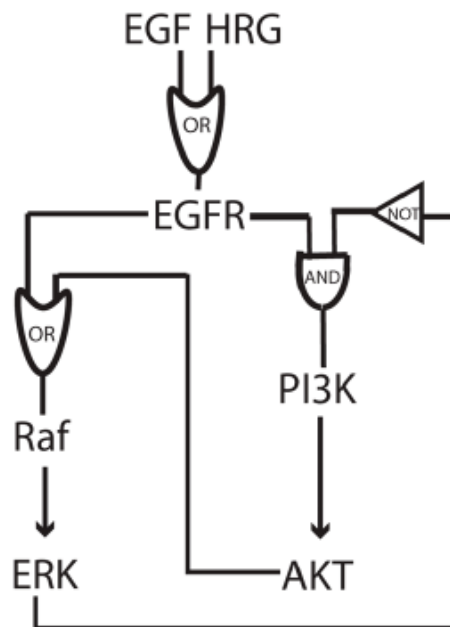
3) Implication of Antecedent to Consequent - It means after getting the antecedent value draw a line $y = \text{value}$ on the consequent. (Example from antecedent $y = 0.7$ line drawn parallel to generous MF applied on tip).

4) Aggregation of Consequents across the rules- Above shows only one rule. One output could be defined by various rules. (For tip could be poor if food is poor this defines another rule and final consequent is combination of both rules.)

5) Defuzzification - Once when all the rules have been applied then we get a combined consequent. On it defuzzification is applied and from it we get one value. It could be applied in various ways(centroid, bisector etc).

DISCUSSION

After describing the methods now I shall be describing the system I worked on. It is a hypothetical network on which I tried to apply Boolean, multi-state and fuzzy logic methods. It is a network of EGFR and downstream proteins. The network is shown below:



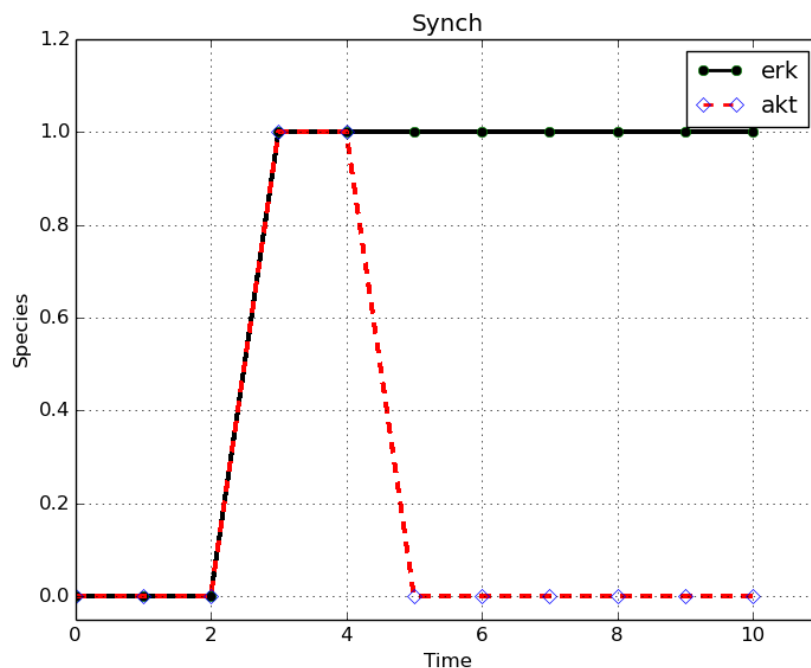
In general for implementation purpose Python has been used (libraries NumPy, matplotlib.pyplot, skfuzzy) have been used. In addition MatLab especially fuzzy toolbox has been to model the systems and get some visual idea before implementing fuzzy logic in Python.

Boolean- Below image shows the rules of the logic gate used in modeling of system.

Logic Gate	Truth Table	Examples in network															
Activation Input \longrightarrow Output	<table> <tr><th>Input</th><th>Output</th></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table>	Input	Output	0	0	1	1	Raf \longrightarrow ERK PI3K \longrightarrow AKT									
Input	Output																
0	0																
1	1																
NOT (Inhibitor) Input \longrightarrow NOT \longrightarrow Output	<table> <tr><th>Input</th><th>Output</th></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	Input	Output	0	1	1	0										
Input	Output																
0	1																
1	0																
OR Input 1 \longrightarrow OR \longrightarrow Output Input 2	<table> <tr><th>Input 1</th><th>Input 2</th><th>Output</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	Input 1	Input 2	Output	0	0	0	0	1	1	1	0	1	1	1	1	EGF \longrightarrow OR \longrightarrow EGFR HRG EGFR \longrightarrow OR \longrightarrow Raf AKT
Input 1	Input 2	Output															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
AND Input 1 \longrightarrow AND \longrightarrow Output Input 2	<table> <tr><th>Input 1</th><th>Input 2</th><th>Output</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	Input 1	Input 2	Output	0	0	0	0	1	0	1	0	0	1	1	1	
Input 1	Input 2	Output															
0	0	0															
0	1	0															
1	0	0															
1	1	1															

Combination of Logic Gates	Possible Truth Table	Example in network															
AND NOT	<table> <tr><th>Input 1</th><th>Input 2</th><th>Output</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	Input 1	Input 2	Output	0	0	0	0	1	0	1	0	1	1	1	0	ERK \longrightarrow NOT \longrightarrow AND \longrightarrow PI3K EGFR
Input 1	Input 2	Output															
0	0	0															
0	1	0															
1	0	1															
1	1	0															

The previous image shows the truth values which have been implemented in our code. Now when we plot the graph of "**AKT,ERK vs time**" we get the following graph. Note that we use synchronous update.



Multi State - In multi state now for egfr we add an additional state 2 for egfr. It has been experimentally found egfr being more sensitive to egf than hrg and different activation of egfr has different impacts on downstream proteins. The following truth table makes the picture more clear.

g. Multi-level description of example network

Truth Tables

Input	Input	Output
EGF	HRG	EGFR
0	0	0
0	1	1
1	0	2
1	1	2

Input	Input	Output
EGFR	AKT	Raf
0	0	0
0	1	1
1	0	0
1	1	1
2	0	1
2	1	1

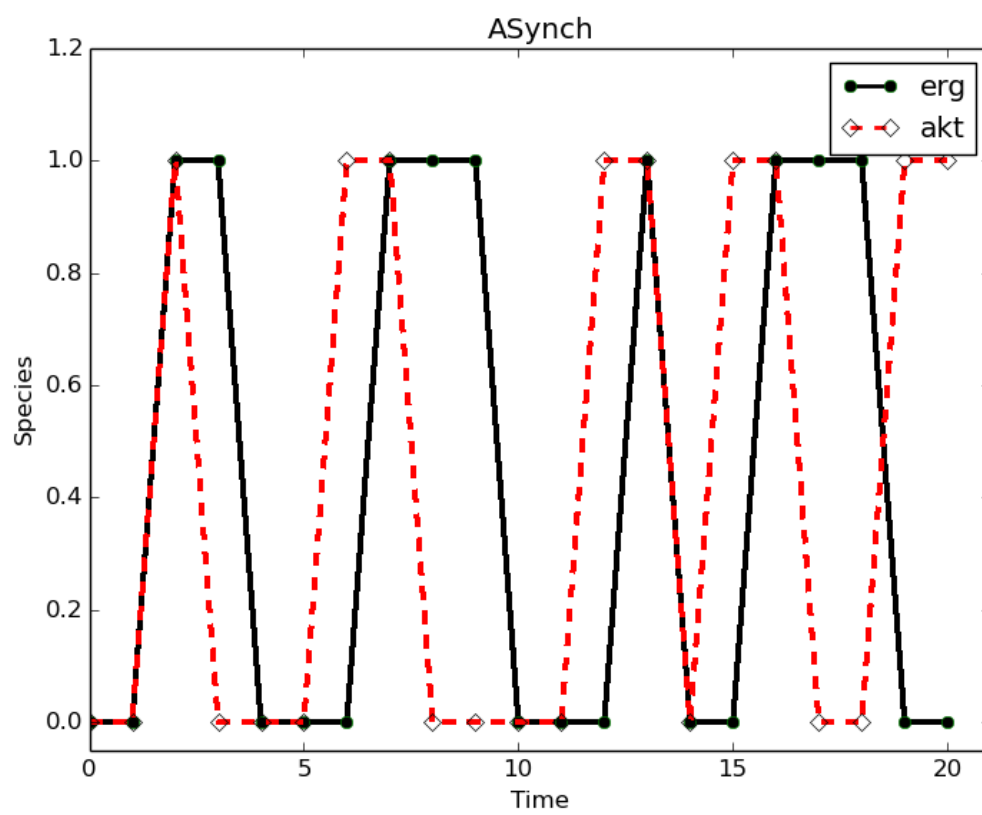
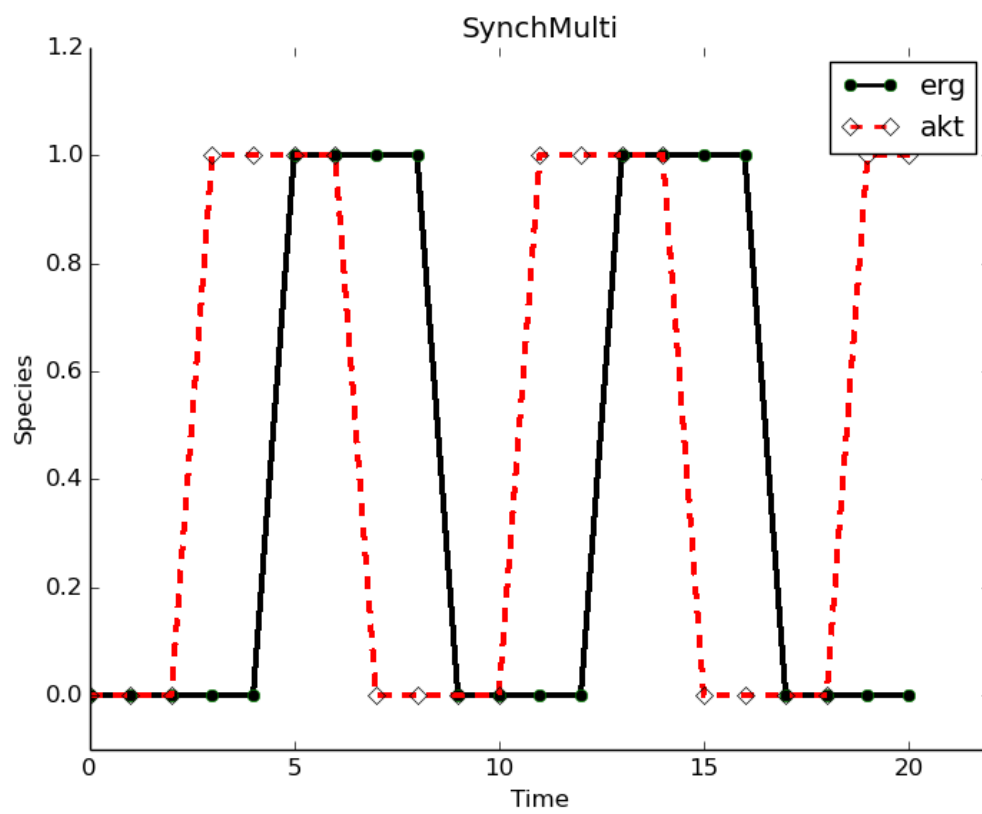
Input	Input	Output
EGFR	ERK	PI3K
0	0	0
0	1	0
1	0	1
1	1	0
2	0	1
2	1	1

Input	Output
PI3K	AKT
0	0
1	1

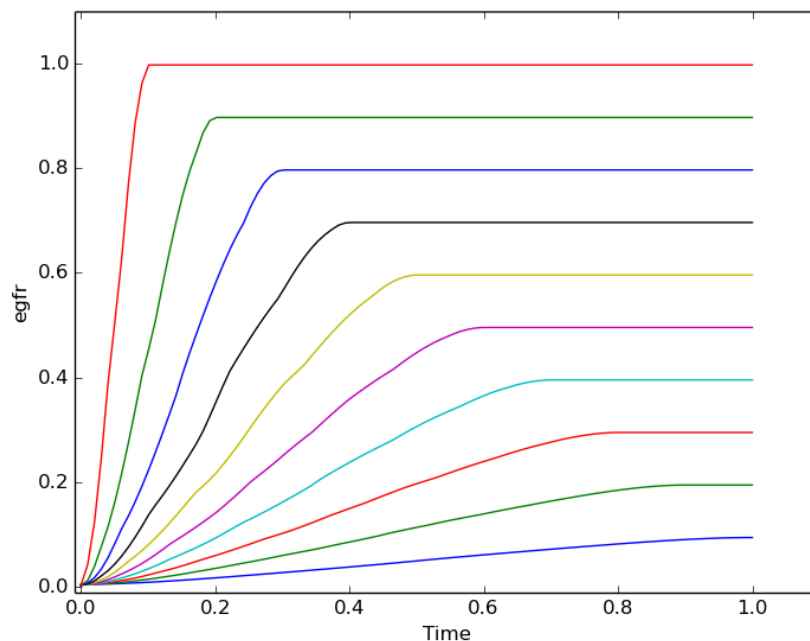
Input	Output
Raf	ERK
0	0
1	1

Now we see that egf is present than egfr has state 2 which in turn always activates the downstream proteins. But when only hrg is present egfr has state 1 which then behaves differently as shown above. Following we plot 2 graphs which are plotted when hrg is only present. First one is with *Synchronous Updates* whereas second one is with *Asynchronous updates*.

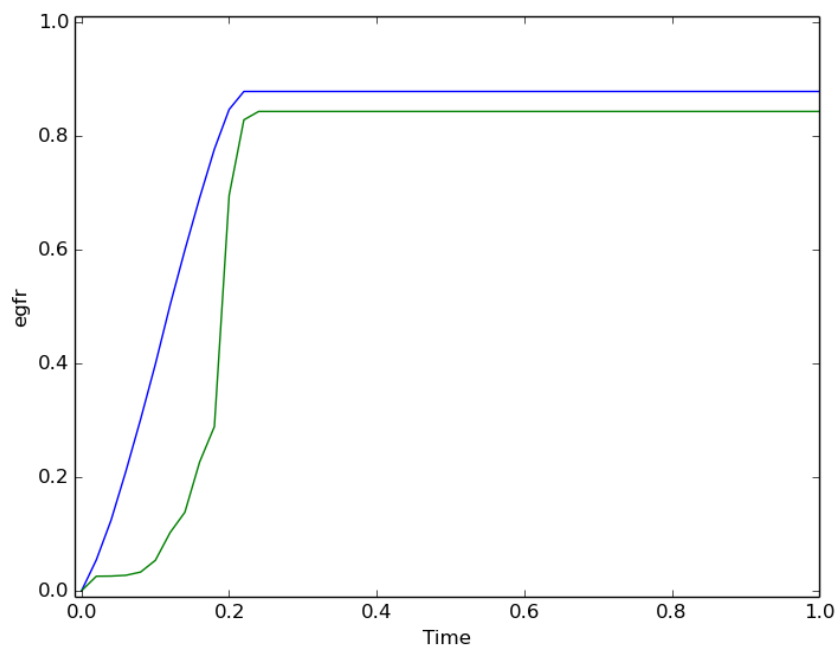
Note that in *Synchronous updates* we get same graph every time while in *Asynchronous updates* things are randomised as you just can't predict which node is being updated based on input value at $(t = i - 1)$ and which at input value at $(t = i)$. The rise and fall in akt and erk is because first since **ERK** is 0 it activates **PI3K** which in turn activates **AKT** but now it forms a feedback to **RAF** activating it and hence in turn activating **ERK** but now it in turn deactivates **PI3K** which in turn deactivates **AKT** and this entire cycle goes on. In synchronous that activation/deactivation gets fixed at a particular time step while same cannot be said for asynchronous because that update is randomised though the basic rules on which update takes place if fixed for both Synchronous and Asynchronous.



Fuzzy Logic - So apparently the biggest problem was how to take time into account while modeling above network. The oscillations seen above were because of time delay added to almost instantaneous changes. So after initially going on the wrong the approach was started again with going again starting from basic steps. First task was to compute egfr at different egf concentrations and each will take different times depending on the input greater the input lesser the time required. So the graph below shows time required for egfr to reach its steady state value based on different egf values. Note fuzzy logic has been used.

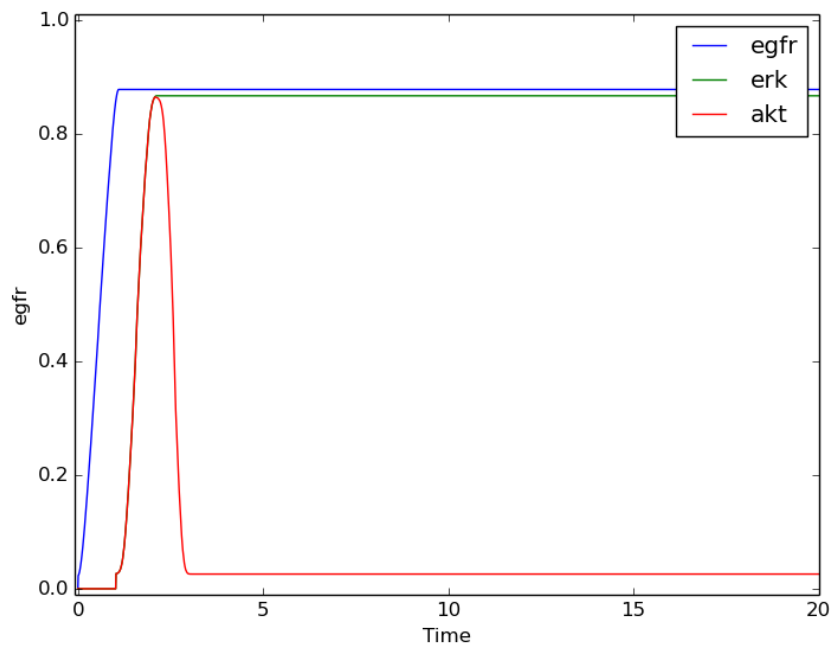


Next we started with **RAF**. We were trying to get some delayed picture without actually adding time delay. (Main idea being compute how egfr varies with time and how raf varies with egfr and so automatically get RAF with time). This was one of the pictures we get.



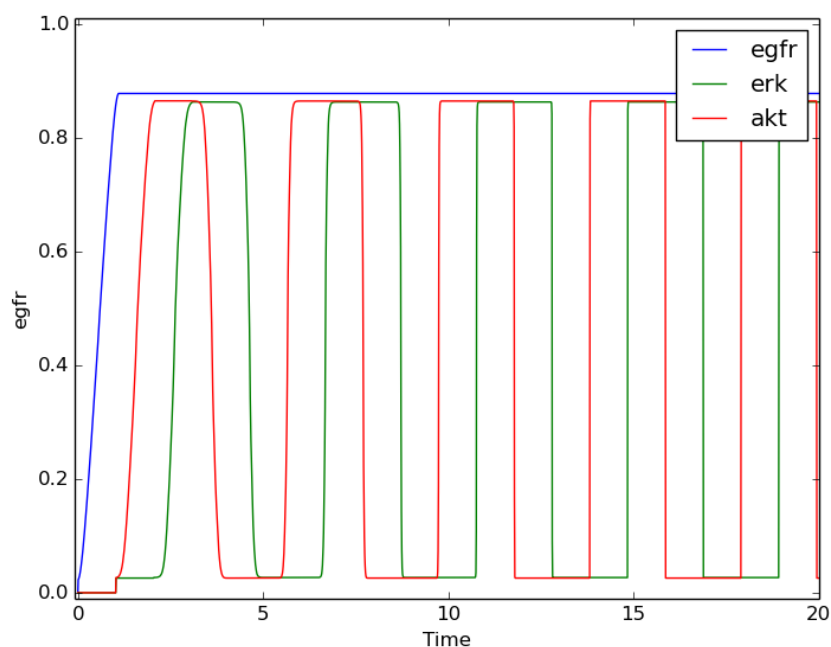
However ultimately we moved to time delay. We calculate the instantaneous change but plot it after some fixed delay.

So far apart topmost OR gate the entire network has been implemented. We plot the graph as again but this time using fuzzy logic and this is what we get. Note that it is quite similar to our initial plot.



Note above width could be increased by increasing time delay.

Next to check the oscillatory behavior we remove the egfr connection from RAF. We then try to plot the graph. This is as follows:



Note though the first oscillations are expected further from there we begin to see very steep oscillations. This is something which is not correct. A solution has been found but I am still working on it.

FUTURE WORK

As mentioned I am still working on fuzzy logic so perfecting it is still my top priority. Further most of this was based on assumptions and I did not have any real data with me to actually chose the correct MFs or build the system properly. The more the knowledge you have of the system the better you can apply fuzzy logic.

So at the moment my next task is to learn some statistical methods and do some data analysis. I primarily aim to learn data analysis techniques and work with microarray.

Also that would include learning R and bioconductor package along with some statistics.

If everything goes fine I might be working on an actual system and doing the boolean modeling of that system.

BIBLIOGRAPHY

1. Logic-Based Models for the Analysis of Cell Signaling Networks †
Melody K. Morris, ‡ Julio Saez-Rodriguez, ‡,§ Peter K. Sorger, ‡,§ and Douglas A. Lauffenburger*,‡
2. Fuzzy Logic Analysis of Kinase Pathway Crosstalk in TNF/EGF/Insulin-Induced Signaling
Bree B. Aldridge 1,2,3‡, Julio Saez-Rodriguez 1,2,3, Jeremy L. Muhlich 1,3, Peter K. Sorger 1,2,3, Douglas A. Lauffenburger 1,2,3 *
3. Fuzzy Logic and Computation with words
David Pruitt
4. <http://in.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html>
5. <http://in.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html>
6. <http://in.mathworks.com/help/fuzzy/fuzzy-inference-process.html>
7. <https://www.youtube.com/watch?v=6D3G1ywOOzo&index=27&list=PLD52D4FEA6E477AFC>
8. <https://www.youtube.com/watch?v=-42QUdFdb9I&list=PLk7mMVIKP4DGQp81pBMVoEMRjoZBpErUr>
9. <https://www.youtube.com/watch?v=qUQf1JxnTnY&index=6&list=PLk7mMVIKP4DGQp81pBMVoEMRjoZBpErUr>
10. <https://www.youtube.com/watch?v=YM4PVm5HDDY&list=PL641C69625C7C9000>

