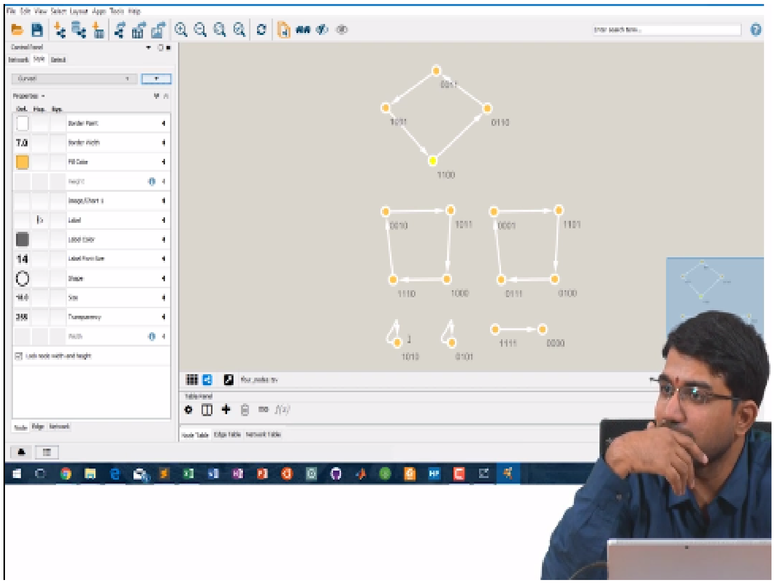
Computational Systems Biology
Karthik Raman
Department of Biotechnology
Indian Institute of Technology – Madras

Lecture – 89
Lab: Modelling Gene Regulatory Networks

In today's lab session, we will look at this tool called BooleSim which is a simple and easy to use tool based on JavaScript to simulate Boolean networks.

(Refer Slide Time: 00:20)



So, welcome to today's lab, we will work with Boolean networks and see how they can be simulated. There are 2 major ways to simulate Boolean networks, you can basically use a simple tool like BooleSim that we will start off with or you can use you know something that is based off of python which is called Boolean Net, so these are the 2 best tools in my opinion for simulating Boolean networks.

I think you can just install this using pip install Boolean net if you have python, so but this is even simpler and if we go back to the example we were seeing earlier today in the class that is actually available as the demo here, so it is called the cell cycle demo or all your rules are there, so this is the same thing, so except that in earlier in the class, we are seeing something as CycD, star = CycD and so on that is, I think that is how Boolean Net likes the rules.

For BooleSim, it is just say something like CycD = CycD or B is not of cyclin A and not of cyclin B and not of cyclin D and not of cyclin E or p 27 and not of cyclin B and not cyclin D and so on and so forth. So, these are the rules, so once you put in these rules, you can just tap simulate and so here blue is false and yellow is true, so you see a nice oscillation and which you can actually as though here as well.

So, we will see that the time step keeps changing and so this clearly a sort of cycle attractor with the period of let us say 1, 2, 3, 4, 5, 6, 7 or so, right you can clearly see the repetitive patterns here, so this is one thing maybe we should start off with the simpler example before we come here. This was just to show you what is actually possible with this tool, so let us create a new network.

Let us try to model the repressilator, let us open up the real repressilator, what were the exact genes, so there were lac I, lambda CI and tet R, so lac I is not of lambda CI not of tet R and tet R is not of lac I, so now you look at the network, save changes to network, I probably made a small typo, capital I, so you can say it is K sensitive, yeah, this is the repressilator and let us start a simulation, you see the same oscillation that we were looking at in the previous class.

And maybe I will pause, I will change the initial conditions and simulate, it is probably cycles through all those; so you found that this was the cycle attractor with 1 to 2 not; 2 state cycle and a 6 state cycle, let us see what this time series looks like, this seems to be 1,2, 3, 4, 5, 6 different states in the same node, I am just looking for where the pattern looks the same, so if you see let us pause it, so that is not distracting.

So, if you see state 65 is the same as state 71, right so you have 6 different; 65, 66, 67, 68, 69, 70; 6 different states in this attractor, so let us say we make the repressilator little larger, and starts simulating it, would this you might find a disagreement with what happens when you have a node based model right. In a node based model, I think it is understood that you cannot have 4 node repressilators, they cannot oscillate really well.

Because the; you know there is a; there are reasons for that because there is no net negative feedback onto A because of the parity here because of the fact that there are 4 nodes but you will see that there are some limitations for these kinds of models, let us just try a different initial state and let see how long are there, yeah even here you find a sort of short cycle, so there are other states that are out of the cycle.

So, one way to do this a really simple way, would be to you know even fire up Excel and put in the rules, so let us say A, B, C D and A star, B star, C star and D star and let us say this is 00001, there are better ways to do this, this is the quick and easy way to do this, now, you can set up your rules here, so this is going to be not of B, , this is going to be not of A. So, now you know what are all the states, so you can even you know somehow put this out with cytoscape and visualise it if you want and so on.

You want to compute the attractors and so on, this is like very quick and dirty way to do it, the right way to do it would be using python, Boolean Net and may be network Xn so on but for those of you are not very comfortable with coding yet, this would be a quick and easy way to understand if your network is working. Now, I can just input that into a cytoscape, if you want to really just visualise how things look.

So, it should be directed, let me just, yeah, so every state is a node, so no surprises here, every state is a node and every edge is from a source to a target meaning from start node from 111, you go to 000, oh, wait, I think 000 to 111 was lost as the header, right so this would have been a cycle that was lost as a header, so you will see only 15 edges here instead of 16, we did take 16 edges as input, right.

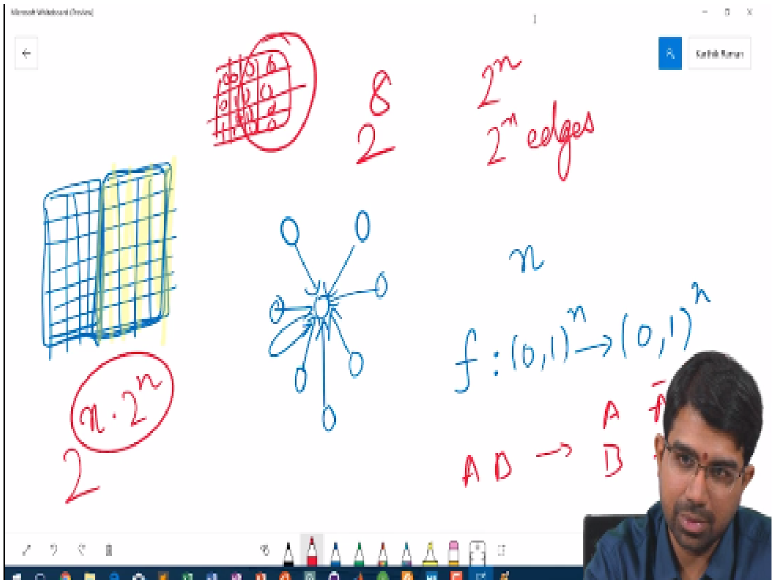
So, the first line was lost as a header, when I imported to cytoscape, I was in careful enough but essentially you see so, 1010 again comes back to 1010, 0101 comes back to 0101, and we can try that here as well, network this is 1010, so it is basically frozen, so it is stopped in the second hydration cells, if you look at the time series, wait maybe I should like reset something, wait, let us just copy this. Yes, discard, rules, save changes.

Now, A on, B off, D off and simulate, time series that it is it, right, so if I do the reverse of this, when I mouse were, it shows me the rule, yeah, we have the (()) (12:35), so this is just one simple way to visualise, so you have 3 cycle attractors of size 4 and one cycle attractor of size 2 and this is actually a cycle attractor even if it seems like a; I am missing the other edge and these 2 are point attractors, this sort of called attractors.

Because you know as the simulation happens, points get suck towards this point right, the moment you fall; if you have a very large system, the moment you come to 1011, you start, you might be anywhere, so this may have other inputs as well, right, the moment you transit to 1011, you are caught in this loop or the moment you transit to 1010, you are caught in the cycle, it turns out that there is no other way to come here in this model.

But there are obviously models where you will have that kind of behaviour, you could theoretically think of a model where in you know, the transitions happen only in; see, you need 15 edges, you need 16 edges, you can lay them out whichever way you want right, you can have certainly not having a cycle.

(Refer Slide Time: 13:47)



Let us consider a very simple example, right where you need 8 states, I could just have them like this, how many functions are possible; **“Professor – student conversation starts”** yeah, how many functions are possible? No, I am saying for a given system, system with n states, how many functions are possible, what is the function? A function maps, is that right or in other words, how many truth tables are possible; n states, think about it, you have 2 to the n values here, so what values, how many different truth table can you have?

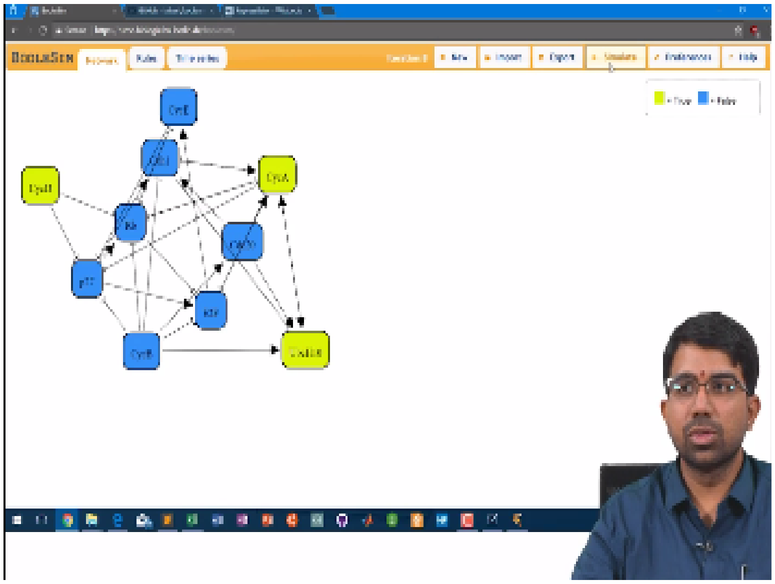
This is fixed, how many different truth tables can you have, so this is going to be n * 2 to the n; n is the number of nodes, so just think of it this way, so this is; let us ignore this left half right, let us look at this part, this is the truth table, how many entries in the truth table, how many entries? How many squares are there here man, n * 2 to the n, each of them can take 0 or 1 that is 2 to they know, so if each of this values can take get; each of so many values can take 0 or 1.

You have 2 choices for each, so multiply 2 so also many times, how many different truth tables are possible? Correct, how many different truth tables are possible? 2 nodes means we are talking about A and B; A and B how many ways can you combine; so you can say A is one function, B is one function, A bar is a function, B bar is a function, A bar and B bar is a function, there are so many different functions you can make with A and B.

How many such functions are possible, which means how many truth tables can you build? So, in 2 states, you will have these; this is the size of your truth table, this is always going to be fixed, 00, 01, 10, 11 right, on the right hand side, we have 00 for everything, you can have just 11 changed, you can just have a one more change like that you can have one’s for everything. How many were entries are here, which happens to be 8, you can have 256 different truth tables with just 2 variables.

There are 256 ways in which you can combine 2 variables basically. If you talk about the graph thing, the graph will have 2 to the n nodes and it can have; how many edges, the graph will have exactly 2 to the n edges, this number is absolutely correct, I have no doubts about this, so here the number of nodes are fixed, so each edge is a different truth table, each set of edges; yeah, okay, I had not intended to do any of these, right. **“Professor – student conversation ends.”**

(Refer Slide Time: 18:28)



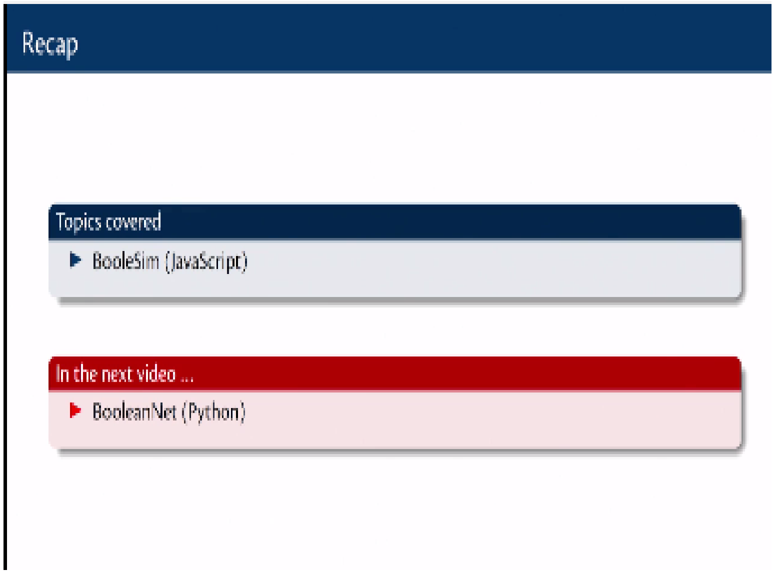
So, what I intended to do was this, so you have a network and you have rules, so let us go back to that network we saw in the previous class which is the cell cycle demo, so this is the network and these are the rules, you have only 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 states right as we saw earlier, so we have total of 2 to the 10 possible global states of the model, you have 10 nodes, so you will have 2 to the 10 different global states.

Now, you can; so now you can just knock out cdh1 and see what happens, right, so let us say it changes the network and simulate, so this system can still oscillate, so cdh1 is not critical to research cycle as far as this model goes, right, so now we go here, so now let us knock out seemingly important cyclin protein, Cyc B = Cyc B and not Cyc E, save changes and now let us simulate, no, it is all dead, right.

In 4 hydrations it; **"Professor – student conversation starts"** it is a cell cycle, it is supposed to oscillate, the cell cycle is an oscillation of circles know, so it goes with g, s and you know, yeah cannot sustain oscillations, so these are the kind of conclusions one can draw out of simulating these systems obviously, our conclusions are going to be little limited because as we discussed earlier, you are just restricting yourself to 2 states as on and there is off. **"Professor – student conversation ends."**

There is no intermediate concentration, you have either zero moles or something or you have like 100, right, there is no; you cannot say that there are other ways actually, there are polynomial you know systems and so on, wherein your state do not are not in 01 but say 0, 1, 2, 3, 4, so you have 5 different states, I have like no or off, low levels, medium levels, medium high levels, very high levels or on.

(Refer Slide Time: 21:23)



Recap

Topics covered

- BooleSim (JavaScript)

In the next video ...

- BooleanNet (Python)

So, you can have a gradation which may be useful in certain scenarios as well. I hope you got a good overview of how to simulate Boolean networks using a simple tool called BooleSim today; this tool was based on JavaScript. In the next video, we will look at a tool called Boolean net, which is based on python and can be used to simulate even larger Boolean networks.