Boolean networks for modeling biological systems

From static network structure to functional behavior

The graphical models that were discussed in the previous lecture are good tools to organize knowledge about interactions between components of biological networks. They also permit insight into the structure of these networks. Unfortunately, these models are completely static.

If we want to understand in what way a network carries out a biological function, for example, regulating the concentration of cholesterol inside the body, the "nodes" in our network need to be able to change their state dynamically.

Boolean Networks

Boolean networks, named after the 19th century mathematician George Boole, are probably the simplest way to capture model dynamics. Just like directed graphical models, Boolean networks consist of nodes and two types of edges (incoming and outgoing) and the structure of the network does not change over time. But, the nodes can switch between two states - true or false.

In Boolean networks time is represented in discrete steps *t = 0, 1, 2, ...* with the network stepping through time as follows. At time *t* nodes send their state to their neighbors via the directional edges. At each node this input serves to determine the state the receiving node will adopt at time *t+1.* Then the states of all nodes are updated simultaneously, thus advancing the network to time *t+1* and the whole process is repeated.

Boolean Logic and Truth Tables

Boolean Logic only knows two states: TRUE and FALSE (often also represented as 1 and 0). It has already been stated above that whether a node will be TRUE or FALSE during the next time interval (*t+1*) depends entirely on the input it receives from its neighbors at time *t*. Because the rules that translate the input into the node's next state can be "customized", Boolean networks are very interesting objects.

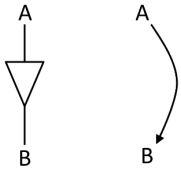
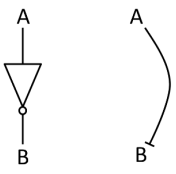
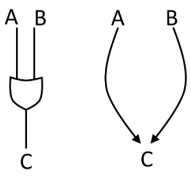
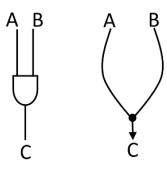
Boolean operation	unary activation (A=B)		unary inhibition (A=**NOT** B)		binary activation (A **OR** B = C)			binary activation (A **AND** B = C)		
Truth table	**input A**	**output B**	**Input A**	**output B**	**Inputs A and B**		**output C**	**Inputs A and B**		**output C**
	0	0	0	1	0	0	0	0	0	0
	1	1	1	0	1	0	1	1	0	0
					0	1	1	0	1	0
					1	1	1	1	1	1
Symbolic representations										

Figure 1 *Examples of simple Boolean logic gates showing their truth tables and the symbols used to represent them. The symbols on the left are typically used in the realm of electronics while those on the right are more typically used in a systems biology context.*

These rules can be specified in what is called a truth table. That is a table listing all possible combinations of inputs coming into a node at time *t* and indicating, for which of these input combinations the node will be TRUE and for which it will be FALSE at time *t+1.* Some simple examples for truth tables are shown in Figure 1.

If nodes receive many inputs, writing down the full truth tables becomes very laborious. In such cases it is customary to write down the rules that govern the state of node *i* in the following type of format.

$$N_i = \begin{cases} true & if \quad at\ least\ one\ \ input\ is\ true \\ false & if \quad none\ of\ the\ inputs\ is\ true \end{cases}$$

It should be noted, however, that Boolean networks only allow Boolean rules, which are of the type that **could** be formulated in a truth table. This excludes, for instance, rules that are based on the state of the node several time steps back or the state of nodes that are not connected to the node in question.

A third way to represent the rules of Boolean networks is a Boolean operator statement (Figure 2b). These are often painful to read but very useful for coding Boolean models for computer analysis.

Representing biological systems as Boolean networks

In the context of biological systems, the TRUE and FALSE states can represent a great variety of biological phenomena. If a node represents a gene, TRUE could represent that the gene is transcribed and FALSE that it is repressed. For a node representing a protein the value of the node might indicate that this protein is present (TRUE) or absent (FALSE), and further if the protein is an enzyme it might represent that this enzyme is active (TRUE) or inactive (FALSE). This simple two-state behavior also allows very different types of nodes to be connected with relative ease.

For example, to represent the fact that a transcription factor promotes the expression of a gene, one could connect the transcription factor node with a directed edge to the node of the gene and set the rule for the gene node to be:

$$N_{gene} = \begin{cases} TRUE & if \quad transcription\ factor\ is\ TRUE \\ FLASE & if \quad transcription\ factor\ is\ FALSE \end{cases}$$

Conversely, the action of a transcription factor that acts as a repressor could be represented in the following way:

$$N_{gene} = \begin{cases} TRUE & if \quad transcription\ factor\ is\ FALSE \\ FALSE & if \quad transcription\ factor\ is\ TRUE \end{cases}$$

Similarly, the relationship between two genes could be modeled such that the expression of one gene (TRUE) will turn off the expression of the other gene by setting it to FALSE.

Or the relationship between an enzyme and its substrate could be modeled by a rule that when the enzyme is active (TRUE) the metabolite is degraded (i.e. the metabolite node is set to FALSE).

So even for relatively complex interactions (figure 2) and systems not yet entirely understood, constructing a Boolean network is fast and easy. For to build a Boolean network model is not necessarily to grasp the mechanistic details of the regulatory interactions that link two system components. The only requirement is to know if two components regulate one another and to be aware of which component regulates which.

Boolean networks can be simulated very efficiently

Another major advantage of Boolean networks is the ease and speed with which they can be analyzed computationally. Just how efficient these simulations have become is demonstrated by the browser-based app called BooleSim that can simulate even complicated Boolean networks interactively - right in the browser of your laptop. A short tutorial on BooleSim can be found on the course's Moodle page. In Figure 2 you can see an image generated with BooleSim.

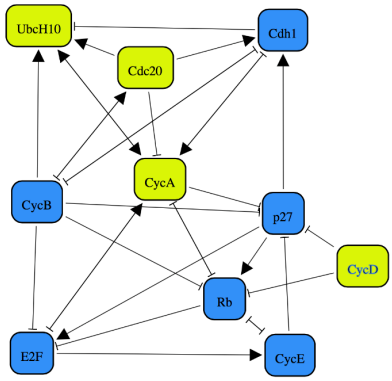
Simplicity is the main strength - and weakness - of Boolean Networks

The ease, with which Boolean networks can incorporate and model relationships between so many different types of components of biological systems, is due to the extreme simplicity of the Boolean model. The binary all-or-nothing approach to modeling interactions circumvents all questions of stoichiometry. Similarly, modeling time as a series of discrete steps of undetermined length sidesteps questions related to kinetics.

But, the simplicity of Boolean networks is also their major weakness. For example, every process is forced to take place on the same "time-scale", regardless of whether in reality they are fast (e.g. an enzymatic reaction happening in seconds) or slow (e.g. a gene-regulatory process taking hours to days). Clearly this is unrealistic and can easily result in nonsensical behavior of the modeled system, in particular if the real system contains multiple feedback loops that operate on different time scales.

Boolean networks also are poorly suited to facilitate understanding of the regulatory processes that fine-tune the concentration or activity of some system components.

a



b

```
CycD = CycD
Rb = (! CycA && ! CycB && ! CycD && ! CycE) || (p27 && ! CycB && ! CycD)
E2F = (! Rb && ! CycA && ! CycB) || (p27 && ! Rb && ! CycB)
CycE = (E2F && ! Rb)
CycA =(E2F && !Rb && !Cdc20 && !(Cdh1 && UbcH10))||(CycA && !Rb && !Cdc20 && !(Cdh1&&UbcH10))
p27 = (! CycD && ! CycE && ! CycA && ! CycB) || (p27 && ! (CycE && CycA) && ! CycB && ! CycD)
Cdc20 = CycB
Cdh1 = (! CycA && ! CycB) || (Cdc20) || (p27 && ! CycB)
UbcH10 = ! Cdh1 || (Cdh1 && UbcH10 && (Cdc20 || CycA || CycB))
CycB = ! Cdc20 && ! Cdh1
```

Figure 2 *A Graphical (**a**) representation of a Boolean network model for the protein signaling network that controls the cell cycle. Each node represents a protein and this protein can be either active (blue = TRUE) or inactive (green = FALSE). The Boolean interactions between the nodes are represented by directed edges. An arrow tip represents activation and a cross bar inhibition. The precise Boolean rules for each node in code form are shown in (**b**)."!" corresponds to the boolean NOT, "&&" to AND and "||" to OR (image generated with BooleSim)*

Boolean Networks have been successful in modeling complicated protein signaling networks.

Despite these limitations and the resulting inability of Boolean networks to simulate certain types of biological processes, they have been employed successfully in the simulation of cellular protein signaling networks. Figure 3 shows an example thereof. This network represents a human cell's immediate response to the combined input of seven cytokine signals, which is of particular interest, because it is one of the processes frequently found to be deregulated in cancer.

Despite the obvious complexity of this protein-protein signaling network, a Boolean network model was able to correctly predict the response of this system to a wide variety of different combinations of input signals and perturbations.

The success of Boolean networks in modeling protein-signaling systems is likely due to the fact that these systems are largely linear (i.e. there are no or few feedback loops) and that most of the processes they encompass take place on similar time-scales.

Obviously, predictions on a systems behavior would be difficult to make based on visual inspection alone and it would be equally complicated to obtain the necessary stoichiometric and kinetic information in order to model this process with a mechanistic ODE based model.

Boolean network models therefore provide an interesting option for the investigation of systems that are relatively complex and not yet understood in great mechanistic detail. Nevertheless, there is no guarantee that Boolean network models will yield meaningful results and the validity of the models always needs to be checked carefully against experimental data. Validation and optimization of Boolean networks with experimental data will be one of the topics of the next lecture.

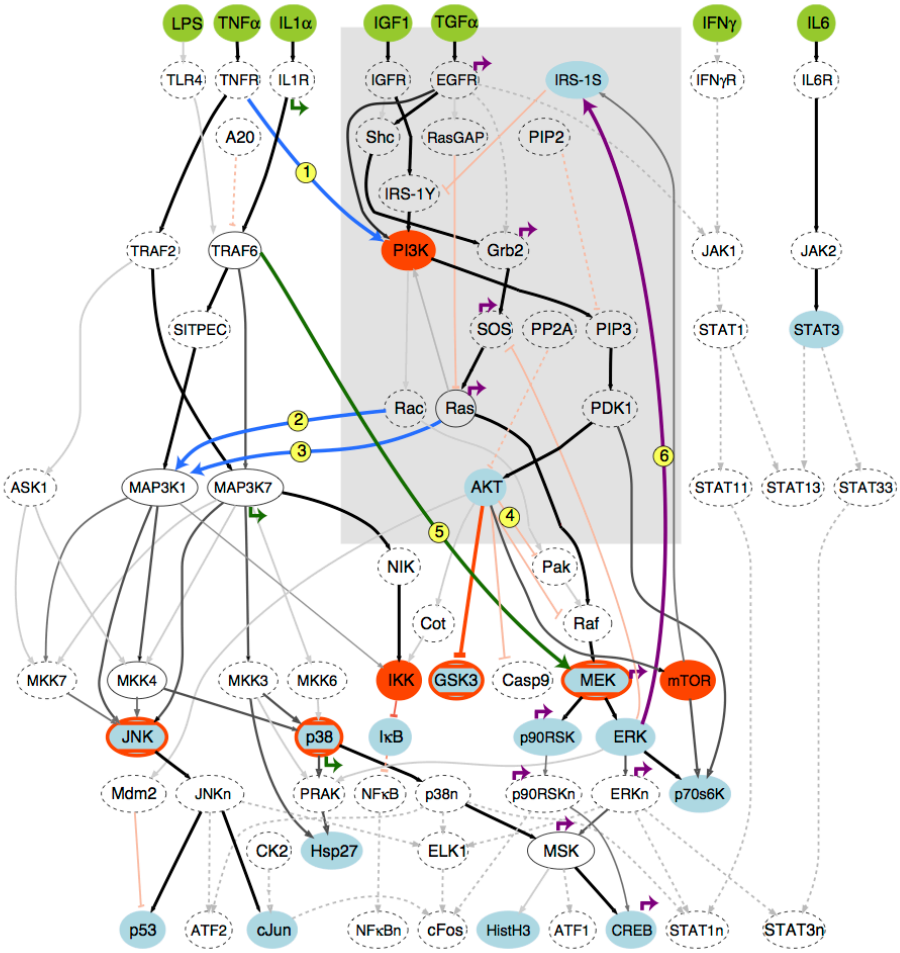


Figure 3 *Boolean network model of the immediate response of human cells to the input from seven different cytokine growth factors (shown in green at top). A Boolean network trained on experimental data was able to predict the response of the system (i.e. the state of the nodes shown in blue) to different combinations of input signals (green) and perturbations (i.e. nodes marked in red were inhibited by specific drugs). (image source Saez-Rodrigues Mo. Sys. Biol. 200*