1 Random Boolean Networks

Let's consider a network of N nodes. The state of each node at a time t is given by $\sigma_i(t) \in 0, 1$ with $i = 1, ..., N$. The N nodes of the network can therefore together assume 2^N different states. The number of incoming links to each node i is denoted by k_i and is drawn randomly independently from the distribution $P(k_i)$. The dynamical state of each $\sigma_i(t)$ is updated synchronously by a Boolean function Λ_i:

$$\Lambda_i : \{0, 1\}^{k_i} \to \{0, 1\}$$

An update function specifies the state of a node in the next time step, given the state of its K inputs at the present time step. Since each of the K inputs of a node can be on or off, there are $M = 2^K$ possible input states. The update function has to specify the new state of a node for each of these input states. Consequently, there are 2^M different update functions. Fo example let's consider a network with $K = 1$, so all the functions Λ_i receives the input from one single node. We can see the different possible functions for all the possible networks with $K = 1$:

$$one : \Lambda_i(\sigma_j) = \begin{cases} 1 & \text{if } \sigma_j = 0 \\ 1 & \text{if } \sigma_j = 1 \end{cases}$$

$$zero : \Lambda_i(\sigma_j) = \begin{cases} 0 & \text{if } \sigma_j = 0 \\ 0 & \text{if } \sigma_j = 1 \end{cases}$$

$$copy : \Lambda_i(\sigma_j) = \begin{cases} 0 & \text{if } \sigma_j = 0 \\ 1 & \text{if } \sigma_j = 1 \end{cases}$$

$$invert : \Lambda_i(\sigma_j) = \begin{cases} 1 & \text{if } \sigma_j = 0 \\ 0 & \text{if } \sigma_j = 1 \end{cases}$$

Now, in general each element receives inputs from exactly K nodes, so we have a dynamical system defined from:

$$\sigma_i(t+1) = \Lambda_i(\sigma_{i_1}(t), \sigma_{i_2}(t), ..., \sigma_{i_K}(t)). \tag{1}$$

So, the randomness of these network appears at two levels: in the connectivity of the network (which node is linked to which) and the dynamics (which function is attributed to which node).

But, once the links and the functions Λ_i are fixed, the evolution of the configurations is deterministic and since we have finite N number of nodes, the evolution becomes periodic.

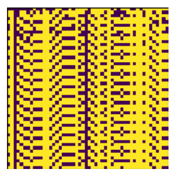

Figure 1: Example of the evolution of random boolean network with $N = 50$, $K = 1$.

2 Attractors

All nodes are updated at the same time according to the state of their inputs and to their update function. Starting from some initial state, the network performs a trajectory in state space and eventually arrives on an *attractor*, where the same sequence of states is periodically repeated. Since the update rule is deterministic, the same state must always be followed by the same next state. If we represent the network states by points in the 2^N-dimensional state space, each of these points has exactly one "output", which is the successor state. We thus obtain a graph in state space.

In a network where we have m attractors, we can define the size of an attractor ω_s with $s = 1, ..., m$ as the number of different states on the attractor. The basin of attraction of an attractor is the set of all states that eventually end up on this attractor, including the attractor states themselves. So we can define the size of the basin of attraction Ω_s as the number of states belonging to it. The graph of states in state space consists of unconnected components, each of them being a basin of attraction and containing an attractor, which is a loop in state space.

Let us consider an example with a network with $N = 4$ and $K = 1$ shown in Figure 2, which consists of 4 nodes:

If we assign to the nodes 1,2,3,4 the functions invert, invert, copy, copy, an initial state 1111 evolves in the following way:

$$1111 \to 0011 \to 0100 \to 1111$$

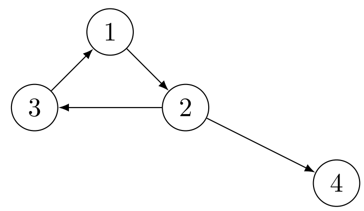

Figure 2: A small network with $N = 4$ and $K = 1$.

This is an attractor of period 3. If we interpret the bit sequence characterizing the state of the network as a number in binary notation, the sequence of states can also be written as

$$15 \rightarrow 3 \rightarrow 4 \rightarrow 15$$

The entire state space is shown in Figure 3:

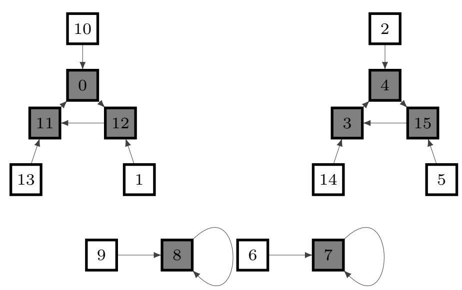

Figure 3: The state space of the network shown in Figure 2, if the functions copy, copy, invert, invert are assigned to the four nodes. The numbers in the squares represent states, and arrows indicate the successor of each state. States on attractors are shaded.

There are 4 attractors, two of which are fixed points (i.e., attractors of length 1). The sizes of the basins of attraction of the 4 attractors are $\Omega_1 = 6, \Omega_2 = 6, \Omega_3 = 2, \Omega_4 = 2$.

3 Perturbation

Since we are talking about gene networks for cell differentiation, we can make some assumptions:

We can consider attractors in the state space as gene regulatory networks of different cells, where different attractors represent cells of different type, and where for example cancer cells lay in one specific attractor. Now, if we suppose that different cell types lay in different attractors, we can suppose that the jump from an attractor to one other is given by a perturbation in the binary sequence of the genes. So for example we take the previos network, and consider that we are in the state 12 of the first attractor:

$$12 \to 11 \to 0 \to 12$$

And suddenly we change the state of the third node from 0 to 1:

$$1100 \to 1110$$

We change the system to have the state 14 and so we jump into the second attractor:

$$14 \to 3 \to 4 \to 15 \to 3$$

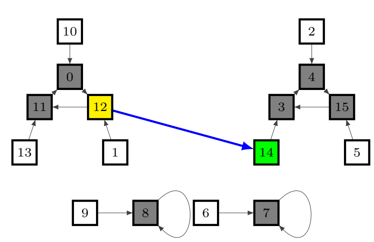

Figure 4: Jump from one attractor to one other in the state space

Now, if we consider all the possible perturbations in the binary sequence of the gene, we can get the frequencies of the transitions between the attractors, and what we obtain is an other network, where the nodes are the attractors and the frequencies of transitions can be put in the adjacency

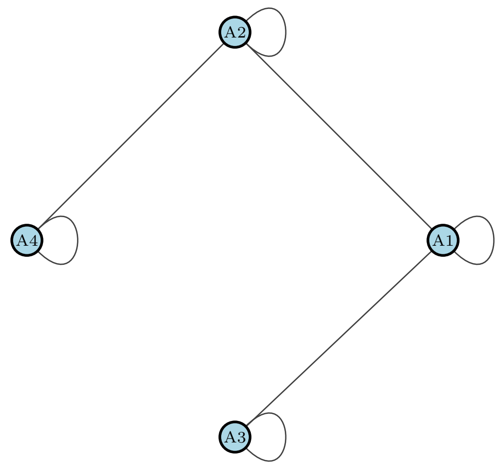

Figure 5: Example of the network in the attractor space

matrix of this network. So in this example we have four attractors, so the network will have the form in Figure 5.

So, having the adjacency matrix formed by the transistion rates from one attractor to one other we can construct a random walk in this network