RESYS Project

Differentiation of hematopoietic precursors in embryos

M. Jeremie Perrin
M2 BIM student
ENS Cachan

M. Corbin Hopper
M2 MPRI student

2019 – 2020

The Dataset

The Early Hematopoietic process

The biological process we are studying is one of differentiation of multipotent precursor cells *Hemangioblasts* into hematopoietic and endothelial cells alike. In the mouse embryo, there is emergence of blood cells at day 7 in the yolk sac and this is the marker of the beginning of the hematopoiesis. Both the Hematopoietic Stem Cells (HSC) and the Endothelial Stem Cells (EPC) originate from these multipotent precursor cells. The blood cells will be formed by differentiation of the HSCs while the vasculature (tissue of the blood vessels) will be formed through differentiation of the ESCs.

The Experiment

In the paper *Decoding the Regulatory Network for Blood Development from Single-Cell Gene Expression Measurements*, single-cell gene expression data was used in order to infer the regulatory networks at work during the hematopoietic differentiation process. It is known that blood development initiates at gastrulation from mesodermal cells, which initially have the potential to form blood, endothelium and smooth muscle cells. They showed that single-cell analysis of a developing organ coupled with computational approaches can reveal the transcriptional programs that control organogenesis.
In order to acquire the data necessary they sampled single-cells in *in vivo* mice embryos. The cells were sampled from the mesoderm and their potential to differentiate into blood cells was asserted thanks to expression of Flk1 and Runx1 expression. The sampling was done at four distinct time points.
Those four times points define groups of cells, which are not hemogeneous since the differentiation process is asynchronous. That is to say some cells begin their differentiation process earlier than others :

E7.00 At this time point the cells are labeled "PS"
E7.50 ———————————————— "NP"
E7.75 ———————————————— "HF"
E8.25 At this time points cells were categorized into two different set. Those cells which expressed *GFP* were labeled "4SG" and where considered as putative blood cells while those that did not where labeled "4SFG" and considered as putative endothelial cells.

At each time points gene expression of a set of genes was measured in each cell, these genes were selected by hand as they were known to play a role in the process. Forty-six genes were selected, out of those : four were housekeeping genes in order to assess the quelity of the measures. Nine were markers known to identify the different cell states and thirty-three were transcription known to play a role in the transcriptional program underlying the differentiation process.

Categorizing Genes

To be pertinent in our analysis we first need to categorize the genes into the different cell states they belong to. We use the literature to guide us in our task. In the article, the authors underlign groups of genes as being caracteristic of the two end states :

- For Hematopoietic Cells : Hbb-bH1, Gata1, Nfe2, Gfi1b, Ikzf1 (Ikaros) and Myb
- For Endothelial Cells : Erg, Sox7, Sox17, Hoxb4, Cdh5

In order to build our own gene categories, we have proceeded in the following way. Each gene is represented by 5 values, those are the mean expression at each sampling points. Before doing unsupervised clustering we wanted to see if the genes were spatially grouping in terms of when they were expressed and how much they were expressed. We therefore ran a Principal Component Analysis (see Fig. 1).Using this two dimensional representation of the data we were able to see that the genes characteristic of the different end stages grouped together. We separated the data into three clusters using K-Means algorithm (see again Fig. 1).

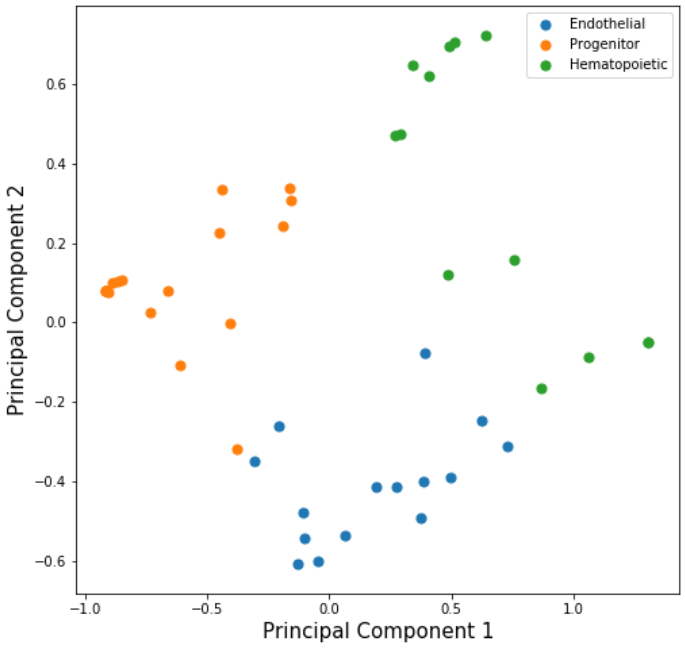

FIGURE 1 – Genes represented after PCA transformation and grouped by unsupervised clustering into three clusters.

We then plotted the actual mean expression values of the different genes grouped by their infered categories (see Fig. 2). We clearly see the distinction in between groups, the clear expression of genes categorized as Hematopoietic in the 4SG stage as well as the expression of Endothelial genes in the 4SFG stage.

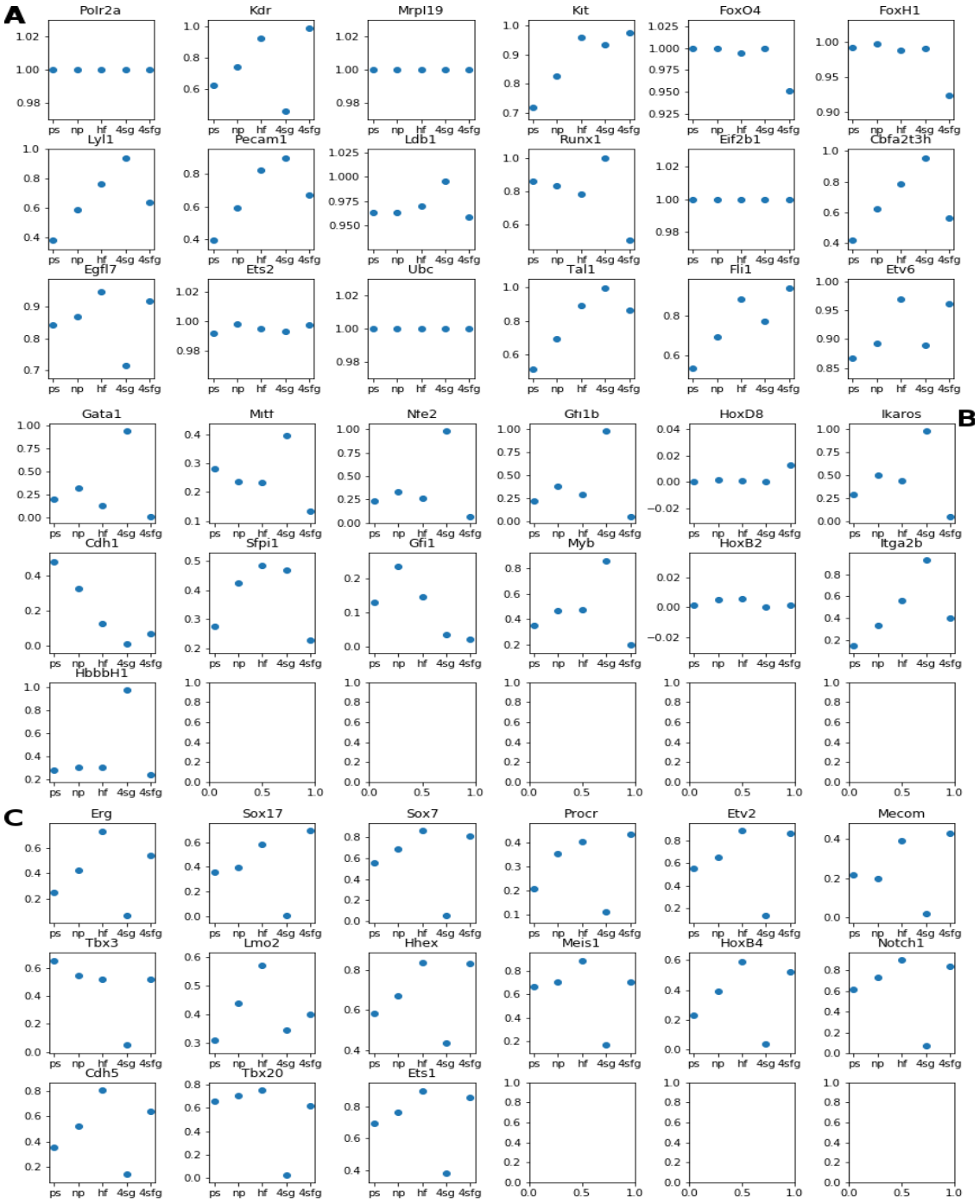

FIGURE 2 – Mean gene expression values throughout the experiment. **A** Precursor genes **B** Hematopoietic genes **C** Endothelial genes

Conditional Entropy Based Network Inference

Approach

One approach attempts to use conditional entropy to infer a directed graph. Although it is ultimately not robust on real data sets, it is used as a preprocessing step in the final network inference. Given that the hemapoeitic data lacks a clear temporal structure, inference about paths seem dubious. Instead, this approach considers each gene independently and appraises the most likely set of genes that regulate it. The goal is to minimize the conditional entropy of each vertex, given its input edges, relative to that of a random process. A predecessor is a vertex with a directed edge to its successor. However, the edges remain unsigned, the transfer function unspecified, and latent variables are not inferred.

Conditional entropy of a vertex given its predecessors corresponds to the amount of randomness in the vertex that is unexplained by its predecessors. The set of genes that regulates another gene should minimize the uncertainty about that gene relative to a random set of the same size. Given that biological processes are inherently stochastic, the approach should prevent certain spurious correlations. Say a network is organized such that $x \to y \to z$. Then x should not form an edge to z, since y would provide more certain information about z.

Given a graph G with vertices and edges (V, E). The goal is formally defined as follows :

$$\underset{E \in G}{\operatorname{argmax}} \sum_{v \in V} I(v; e_{in}) - \mathbb{E}[I(A; B)] \tag{1}$$

Where I(Y ;X) is the mutual information between the random variables Y and X. $\mathbb{E}[I(A; B)]$ refers to the expected mutual information of a subset of instances A, B that are the same size as v, e_{in},but drawn from random distributions. Over many random distributions, some small amount of distributions are expected to be correlated by chance. Equation 1 is then simplified as follows :

$$\sum_{v \in V} \underset{e_{in} \in E}{\operatorname{argmax}} I(v; e_{in}) - \mathbb{E}[I(A; B)] = \sum_{v \in V} \underset{e_{in} \in E}{\operatorname{argmax}} (H(v) - H(v|e_{in})) - \mathbb{E}[H(A) - H(A|B)]$$

$$= \sum_{v \in V} \underset{e_{in} \in E}{\operatorname{argmin}} (-H(v) + H(v|e_{in})) - \mathbb{E}[-H(A) + H(A|B)]$$

Where I(Y ;X) is the mutual information between the random variables Y and X. The entropy of v, $H(v)$, does not change over its set of input edges (and same for $H(A)$). Then equation 1 can be further rewritten as :

$$\sum_{v \in V} \underset{e_{in} \in v}{\operatorname{argmin}} H(v|e_{in}) - \mathbb{E}[H(A|B)] = \sum_{v \in V} \underset{e_{in} \in v}{\operatorname{argmin}} - \mathbb{E}[H(A|B)] \sum_{(i,j) \in (v, e_{in})} -p(i,j) \log_2 p(i|j) \tag{2}$$

Where H(Y|X) is the conditional entropy of the random variable Y, given X, and p(x,y) is the joint probability of events x and y occurring. One can sum over instances (say rows of a

matrix) $(v, e_{in})_0 ... (v, e_{in})_n$, instead of summing over outcomes $(i, j) \in (v, e_{in})$:

$$\underset{e_{in} \in v}{\operatorname{argmin}} - \mathbb{E}[H(A|B)] \sum_{k=0}^{n} - \log p(v_k | e_{in_k}) = \underset{e_{in} \in v}{\operatorname{argmax}} - \mathbb{E}[H(A|B)] \prod_{k=0}^{n} p(v_k | e_{in_k})$$

It then becomes clear that the approach maximizes the log-likelihood of the probability of observing the activity of vertices given their immediate predecessors.

It remains for $\mathbb{E}[H(A|B)]$ to be simplified. Let A be a vector of length n and B a matrix of dimensions n by m. This corresponds to the case where the output of one vertex is being evaluated, relative to that of a subset of its predecessors. If A, B are the same type of distribution, then :

$$\mathbb{E}[H(A|B_m)] = \mathbb{E}[H(A, B_m) - H(B_m)] = \mathbb{E}[H(b_1, ..., b_m, b_{m+1}) - H(b_1, ..., b_m)]$$

If there is enough data $\mathbb{E}[H(b_m)]$ should approach the entropy of n random vectors. In other words, as n approaches infinity, the probability that all m vectors are different approaches 1, and so :

$$\mathbb{E}[H(A|B)] \approx \log 2(m+1) - \log 2(m)$$

Let $m = |e_{in}|$. Then equation 2 can be rewritten as :

$$\sum_{v \in V} \underset{e_{in} \in v}{\operatorname{argmin}} H(v | e_{in}) - \log_2(\frac{m+1}{m}) \tag{3}$$

Algorithm

An algorithm is designed for equation 3 and summarized in 3. The maximum likelihood approach has already been aggressively simplified by only considering the immediate neighbors of each node. An efficient algorithm should still avoid iterating over all possible edge combinations for each node. Since $H(Y|x_1, ..., x_m) \leq H(Y|x_1, ..., x_{m-1})$, one can safely start with all edges and iteratively remove them if :

$$H(Y|x_1, ..., x_{m-1}) - H(Y|x_1, ..., x_m) \leq \mathbb{E}[H(a|b_1, ..., b_{m-1}) - H(a|b_1, ..., b_m)]$$

$$H(Y|x_1, ..., x_{m-1}) - H(Y|x_1, ..., x_m) \leq \log_2(\frac{m}{m-1}) - \log_2(\frac{m+1}{m}) \tag{4}$$

$$H(Y|x_1, ..., x_{m-1}) - H(Y|x_1, ..., x_m) \leq \log_2(\frac{m^2}{m^2-1})$$

If none of the current edges satisfy equation 4, then the algorithm is done with that node. In contrast, one could start with an empty graph and adds edges, repeating a search through all edges only if a new edge has been added the last round. However, this approach would miss certain structures such as XOR, since $H(Y|x_1) = H(Y|x_2) = H(Y)$ would cause the algorithm to stop, despite the fact that $H(Y|x_1, x_2) > H(Y)$.

Infer Graph
Begin with a fully connected graph $G(V, E)$.
For each vertex $v \in V$:
    if $H(v) = 0$: remove v
For each vertex $v \in V$:
    Infer Node (v)

Infer Node (v)
Let w be the set of all predecessors of v, such that $(w_i, v) \in G$.
For each directed edge (u,v) :
    Let w\u be the set w, excluding vertex u.
    If $H(v|w\backslash u) - H(v|w) \leq \mathbb{E}[H(Y|X) - H(Y|X\backslash x)]$: 4
        Remove edge (u,v)
        Infer Node (v)

FIGURE 3

The algorithm also preprocesses nodes, such that if $H(x_i) = 0$, the node is removed. Such a gene hardly various across different cells, let alone across different anatomical stages. These genes often correspond to housekeeping genes.

Severe Limitations of The Approach

The approach works on simple binary problems.
For example, if $Y = (x_1\, NAND\, x_2\, NAND\, x_3)\, AND\, (x_4\, OR\, x_5)$, the algorithm correctly predicts that all x have a directed edge to y and there are no other edges in the graph. However, for problems such $Y = x_1\, XOR\, x_2$ it also produce edges from the output Y to both the inputs x_1, x_2. This is due to the symmetric nature of the problem, since $x_1 = Y\, XOR\, x_2$ as well.

Unfortunately, the approach does not perform well on benchmark tests. When the algorithm is tested on *Lizards*, *Coronary*, and *Asia* datasets of BNLearn, it removes all edges. It appears that the algorithm is not robust to noise. The benchmarks have few nodes, which encounter high thresholds in the algorithm and are not sufficiently correlated to remain connected. Returning to the simple binary problems, randomly flipping 10% of the bits severely impairs the inference accuracy, even when the input was repeated several times for redundancy.

Application to Hematopoietic Regulatory Network

Given the drawbacks of the approach, it is used as a preprocessing tool instead of inferring the hematopoietic regulatory network alone. The large amount of nodes result in lower thresholds for the expected change in conditional entropy. As such the algorithm should not be as harsh as when applied to the benchmarks. Indeed, most genes remain and the graph remains dense after running the algorithm. 4 genes were removed since their entropy alone was 0. The algorithm then continued to remove 7 more genes, reducing the total to 35. Although graph remains too dense, the algorithm trimmed some edges, resulting in an average in-degree and out-degree of 28. This reduction simplified the subsequent application of the MIIC algorithm to infer the hematopoietic regulatory network.

Using MIIC

We then turned to MIIC to infer a network from the dataset provided, now that we've categorized the genes and trimmed those that did not provide enough information. We downloaded the tool from gitHub and compiled the sources. We present the resulting graph in Fig. 4.

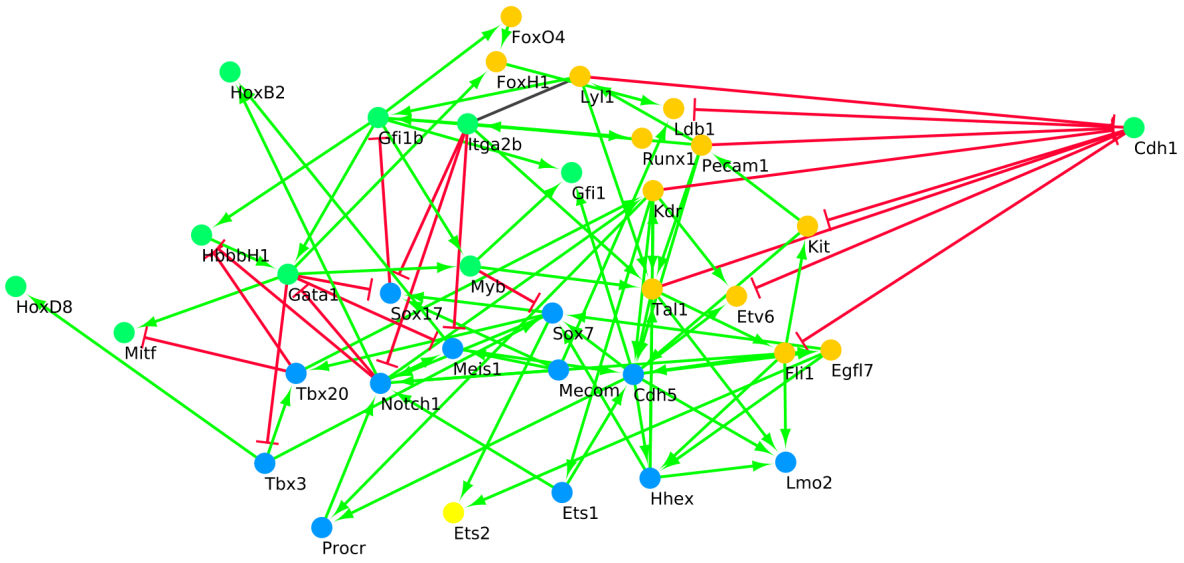

FIGURE 4 – Network Inferred by MIIC. Precursor genes, Hematopoietic genes and Endothelial genes

Discussion The network clearly represents the hematopoietic process as seen by the repression links infered between the genes classified as hematopoietic and those classified as

endothelial. The Cdh1 gene behaves as a regulator (through repression) of the other precursor genes, if we look back at its gene expression levels we see that the expression drops throughout the differentiation process. This gene was not taken into account in the paper *Learning causal networks with latent variables from multivariate information in genomic data.* (PLoS computational biology 2017, Verny et al.). It leads us to believe that it was wrongly categorized as an hematopoietic gene, that it should rather be categorized as a precursor gene.Our inference has several difference from previous application of MIIC to this hematopoeitic dataset. For instance, it includes CDH5 as an endothelial gene with an in-degree of 6 and an out-degree of 7. CDH5 has been previously linked to the process of endothelial development. For instance, one study showed that CDH5 is necessary for endothelial polarity during embryonic development (*CCM1 regulates vascular-lumen organization by inducing endothelial polarity.* Lampugnani et al NCBI 2010).

Their Network Inference Method

The network inference tool used in the article was specially developed for their purpose. And indeed it achieves good results. Moreover it is explanatory, in the sense that it provides the update function of each gene as a boolean fucntion. However, the approach also risks being overly ambitious since the search space is extremely large. We were first tempted to implement it but we finally decided, the amount of work was excessive. Still, we found it interesting to describe their method.

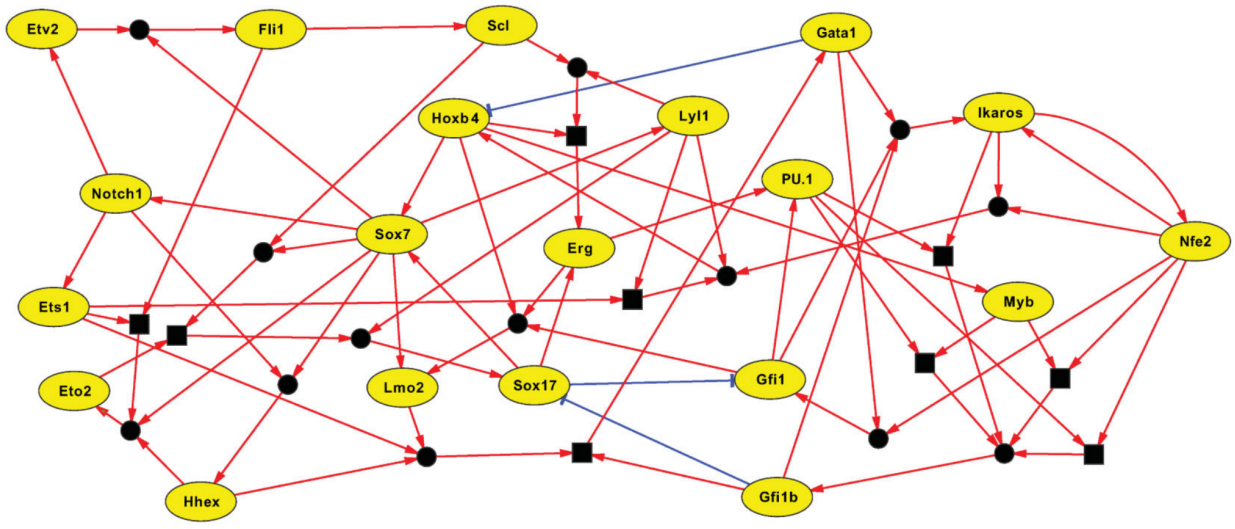

FIGURE 5 – Network constructed by the SCNS toolkit

State-transition graph The number of cells is great enough to consider we have all the states the cells can be in. Every pair of states that differ in the expression of exactly one

gene are connected to form the state-transition graph.

Network Inference Then they search for the direction of the edges as well as the Boolean update function for each of the genes. To do so, they translate their problem into a SAT problem.

Discussion One thing interesting about this type of boolean network is that you can find its stable state. A stable state will be assigning a value of expression (ie expressed or not) to each gene such that all boolean equalities are satisfied. We would expect the final cells (those that have already specified) to be stable state of the network. If the 4SG cells are indeed a stable state (or some of them at least) then it does tell that the network is accurate. Also one such network would allow one to simulate the evolution of a cell, it would be interesting to know how well this network fares in doing so.