Visualising Random Boolean Network Dynamics

Susan Stepney
Department of Computer Science, University of York, UK
susan@cs.york.ac.uk

ABSTRACT

We propose a simple approach to visualising the time behaviour of Random Boolean Networks (RBNs), and demonstrate the approach by examining the effect of canalising functions for $K > 2$ networks.

Categories and Subject Descriptors

I.3.3 [**Computer Graphics**]: Picture/Image Generation—*Display algorithms*

General terms

Algorithms

Keywords

Random Boolean networks, RBNs, canalising functions, dynamical systems, attractors

1. INTRODUCTION

Random Boolean networks (RBNs) are a well-studied form of complex discrete dynamical systems [1, 2, 3, 4, 5]. Visualisation of the dynamics can aid understanding, but (unlike for 1D Cellular Automata, for example), there has been no satisfactory visualisation of RBN time behaviour. Here we propose a simple approach to visualising the time behaviour of RBNs and demonstrate the approach by examining the effect of canalising functions for $K > 2$ networks.

A Random Boolean Network (RBN) comprises N *nodes*. Each node i at time t has a binary valued state. Each node has K inputs assigned randomly from K of the N nodes (an input may be from the node itself); the wiring pattern is fixed throughout the lifetime of the network. The state of node i's neighbourhood at time t is a K-tuple of its input node states. Each node has its own randomly chosen local state transition rule. These nodes form a network of state transition machines. At each timestep, the state of each node is updated in parallel. The global dynamics is determined by the local rules and the connectivity pattern of the nodes. Kauffman [3, 4] investigates the properties of RBNs as a function of connectivity K. Despite all their randomness, "such networks can exhibit powerfully ordered dynamics" [3], particularly when $K = 2$.

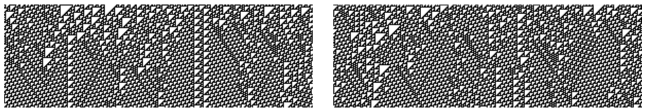

Figure 1: Visualisation of the time evolution of ECA rule 110, with $N = 300$, 100 timesteps, and two different random (50% "on", 50% "off") initial conditions

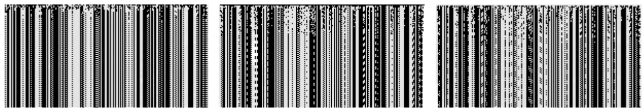

Figure 2: Visualisation of the time evolution of a typical $K = 2$ RBN, with $N = 200$, 100 timesteps, and initial condition (a) nodes randomised (50% "on", 50% "off") (b) all nodes "on", (c) all nodes "off"

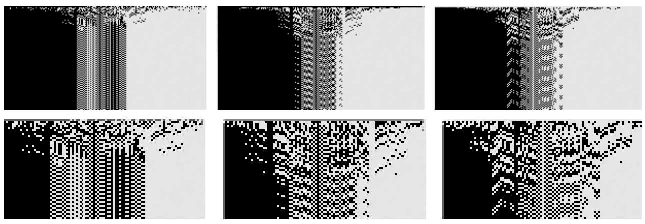

Figure 3: Visualisation of the time evolution of the $K = 2$ RBN with the nodes sorted to expose the frozen core. (a) top: the same RBNs as in fig.2. (b) bottom: zooming in on the central region.

2. VISUALISATION

Good visualisations can aid the understanding of complex systems, and can help generate new questions and hypotheses about their behaviours.

For 1D cellular automata (CAs), the global behaviour from a given initial state is conventionally visualised by drawing the global state at time t as a line of nodes (with colours corresponding to the local state), then drawing the state at $t + 1$ directly below, and so on (figure 1).

CAs have a regular topology, which is used when laying out the nodes for visualisation. RBNs have no such regular topology. If this approach is taken with their nodes laid out at random (as done, for example, in [5](fig.3) or [2](fig.2)), the structure of the dynamics is hard to discern (figure 2).

Kauffman [4](p.203) observes that $K = 2$ RBNs "develop

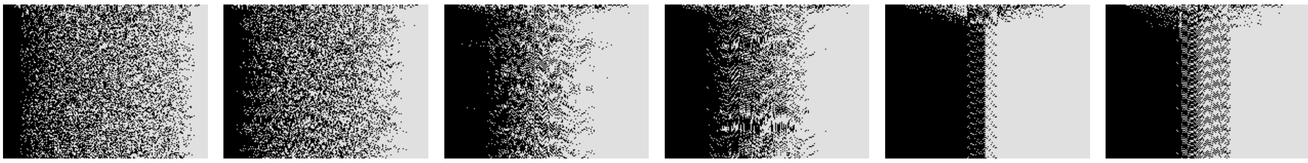

Figure 4: Visualisation of the time evolution of 6 typical $K = 3$ RBNs, with $N = 200$, and initial condition all nodes "off"; for 150 timesteps; columns have the following number of canalised nodes: (a) 94 = 47.0% (b) 128 = 64.0% (c) 181 = 90.5% (d) 184 = 92.0% (e) 190 = 95.0% (f) 198 = 99.0%

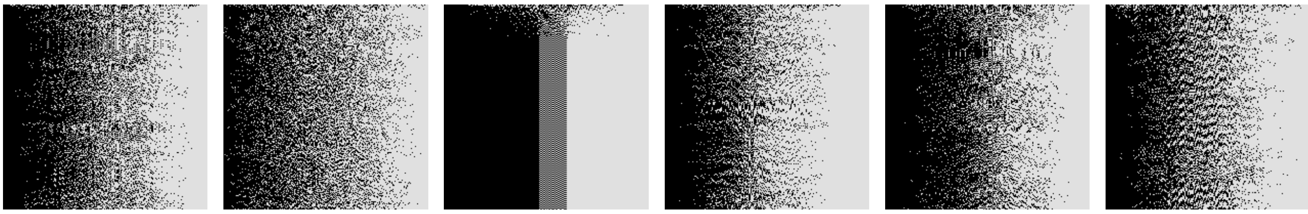

Figure 5: Visualisation of the time evolution of 6 typical $K = 4$ RBNs, with $N = 200$, and initial condition all nodes "off"; for 200 timesteps; all functions canalising

a connected mesh, or *frozen core*, of elements, each frozen in either the 1 or 0 state." We use this to determine an order for placing the nodes in the visualisation. Nodes frozen in the 1 or 0 state are placed towards the edges of the figure; nodes that are changing state are placed towards the centre: see figure 3. The different transient behaviours and attractors are now clearly visible; for example, it is clear that these show three different attractors, with three different periods.

The algorithm is as follows. For a given RBN, to determine the order of drawing the nodes in the visualisation, do the following: (1) Pick a representative number of timesteps, t (for example, the number to be used in the subsequent visualisations). (2) Set the RBN into a given initial state (for example, all zeroes). (3) Run it for t timesteps, counting how many times each node is on. Repeat steps 2 and 3 for other suitable initial conditions (for example, all ones), accumulating the counts. (4) Sort the nodes by the total number of times they were on in these runs.

Thus frozen core nodes are the edges, since they are in a constant state (after transient behaviour has died out), whilst the nodes with cycling states are in the centre. Additionally, the frozen core nodes with shorter transient behaviour are closer to the edges than those with longer transient behaviours. Similarly, nodes with cycling states are sorted according to the amount of time they spend in one state or the other, with those half the time in each state towards the centre. This highlights the attractor structure.

Note that the precise order depends on the various initial states chosen. In the examples given here, for simplicity, the network was run just from the all zeroes and from the all ones state to determine the sort order. In figure 3, it can be seen that in the all ones initial state (middle column) the central node is always on, whilst in the all zeroes initial state (right column) it is always off. (This implies it is a node with a self-connection.) Hence, when these two cases are combined, it is on for an average of half the time, and so ends in the centre.

To illustrate the technique further, we visualise the effect of canalising functions on the time behaviour of $K > 2$ networks. Kauffman [4](p.203) defines a canalising function as "any Boolean function having the property that it has at least one input having at least one value (1 or 0) which suffices to guarantee that the regulated element assumes a specific value (1 or 0)." Kauffman argues that the canalising functions are important for establishing the frozen core and ordered dynamics of $K = 2$ networks. The proportion of canalising functions decreases rapidly with increasing K. Kauffman [4](p.206) states that "networks with $K > 2$ restricted to canalyzing functions ... [have] orderly dynamics in the entire network".

Visualisations of the effect of canalising functions on the time behaviour are show in figures 4 and 5. Clearly for $K = 3$ (figure 4), increasing the proportion of canalising functions does make transients and attractors shorter, and establish an "orderly dynamics". However, for $K = 4$, even with all functions canalising, change in the chaotic behaviour is marked in only a minority of cases. The effect does not appear to be as strong as Kauffman suggests.

In summary: we have introduced a very simple algorithm to allow the time behaviour of RBNs to be visualised in a manner that exposes the transient behaviour, and the structure of the frozen core and cycling nodes.

Acknowledgements

Thanks to Leo Caves, Adam Faulconbridge, Julian Miller, and Jon Timmis for comments on an earlier draft.

3. REFERENCES

[1] B. Drossel. Random boolean networks. In H. G. Schuster, editor, *Reviews of Nonlinear Dynamics and Complexity*, volume 1. Wiley, 2008. arXiv:0706.3351v2 [cond-mat.stat-mech].

[2] C. Gershenson. Introduction to random boolean networks. In *Workshop and Tutorial Proceedings, ALife IX*, pages 160–173, 2004.

[3] S. A. Kauffman. Requirements for evolvability in complex systems. *Physica D*, 42:135–152, 1990.

[4] S. A. Kauffman. *The Origins of Order*. OUP, 1993.

[5] A. Wuensche. Genomic regulation modeled as a network with basins of attraction. *Pacific Symposium on Biocomput.*, pages 89–102, 1998.