Evolving Genetic Regulatory Networks Using an Artificial Genome

Jennifer Hallinan
Institute for Molecular Biosciences and School of ITEE,
The University of Queensland
j.hallinan@imb.uq.edu.au

Janet Wiles
School of ITEE & School of Psychology
The University of Queensland
j.wiles@itee.uq.edu.au

Abstract

Boolean models of genetic regulatory networks (GRNs) have been shown to exhibit many of the characteristic dynamics of real GRNs, with gene expression patterns settling to point attractors or limit cycles, or displaying chaotic behaviour, depending upon the connectivity of the network and the relative proportions of excitatory and inhibitory interactions. This range of behaviours is only apparent, however, when the nodes of the GRN are updated synchronously, a biologically implausible state of affairs. In this paper we demonstrate that evolution can produce GRNs with interesting dynamics under an asynchronous update scheme. We use an Artificial Genome to generate networks which exhibit limit cycle dynamics when updated synchronously, but collapse to a point attractor when updated asynchronously. Using a hill climbing algorithm the networks are then evolved using a fitness function which rewards patterns of gene expression which revisit as many previously seen states as possible. The final networks exhibit "fuzzy limit cycle" dynamics when updated asynchronously.

Keywords: genetic regulatory networks, asynchrony, fuzzy limit cycles

1 Introduction

Genetic regulatory networks developed using Reil's Artificial Genome (AG) model (Reil, 1999) reflect the behaviour of real genetic regulatory networks in many ways. They exhibit complex dynamic behaviour, ranging from rapid "freezing" at a point attractor in state space, through limit cycles of varying length and complexity, to apparently chaotic dynamics, depending on the connectivity and degree of inhibition of the network. However, these models suffer from several important deviations from biological plausibility. Time is modelled as discrete time steps, and gene activity is binary—a gene is either on or off at any given time step. This simplification is widely accepted, largely because Boolean networks do exhibit dynamic behaviour similar to that of biological cells.

A potentially more important factor in the modelling of GRNs is the connectivity distributions of the networks. Random Boolean Networks (RBNs) have an exponential connectivity distribution (Kauffman, 1993), while the connectivity distributions of AG models which are shaped by the choice of parameters such as gene and chromosome length, tend towards a Poisson distribution for input connectivity and a

Gaussian distribution for output connectivity. Neither of these patterns is biologically plausible; a wide variety of intercellular interaction networks have been shown to have scale-free connectivity (Jeong, 2000; Wagner & Fell, 2000; Solé & Pastor-Santorros, 2002).

A more striking departure from biological plausibility pertains to the gene update rules. In the standard AG, as in the classical Random Boolean Network (RBN), all genes are updated simultaneously, on the basis of their inputs from the previous time step. While computationally convenient, such synchronous updating does not occur in biological GRNs; factors such as mRNA and protein synthesis, degradation and transport times mean that the system is replete with delays of varying amounts, and genes are activated or inhibited in a fundamentally asynchronous manner. Unfortunately, with Boolean GRN models, the implementation of asynchronous gene update, whether deterministic or not deterministic, completely alters the network dynamics. Unless connectivity and inhibition levels are very high, most networks quickly move to a point attractor in state space and remain frozen there.

The existence of complex dynamic behaviour in network models is of interest because biological cells are assumed to function on the "edge of chaos", in the regime between totally frozen and chaotic dynamics. This region is characterized by the presence of limit cycle attractors with wide basins of attraction. Such attractors are widely assumed, following Kauffman (1993), to be models of cell types—each cell type is an attractor in gene expression phase space.

The biological implausibility of synchronous updating is widely recognised, and the effects of asynchrony have been examined in a variety of models, including globally coupled logistic maps (Abramson & Zamette, 1998), Conway's Game of Life (Blok & Bergersen, 1999), cellular automata (Schönfisch & de Roos, 1999) and random Boolean networks (Harvey & Bossomaier, 1997; Di Paolo, 2001).

Limit cycles *per se* do not generally exist in an asynchronously updated networks, although networks with such properties can be specifically handcrafted (Nehaniv, 2002). However, several authors have

demonstrated the existence of pseudo-periodic "loose" attractors in asynchronously updated networks (Harvey & Bossomaier, 1997; Di Paolo, 2001).

In this project we use an evolutionary computation algorithm to evolve "interesting" dynamics in an asynchronously updated, AG-based GRN. "Interesting" behaviour is defined in terms of the number and length of limit cycles exhibited by the network. Networks show more complex dynamic behaviour after evolution than before, cycling through a limited number of states (a loose limit cycle) rather than moving quickly to a point attractor in state space.

2 Methods

2.1 The Artificial Genome

Our model is based on the Artificial Genome model (AG) developed by Reil (1999). A genome is generated at random, using equal proportions of *b* "bases". There are four bases in real DNA – adenine (A), thymine (T), guanine (G) and cytosine (C), so we used four bases, designated 0, 1, 2 and 3. The genome is then searched for instances of a gene marker string of length *l* (we used 0101) analogous to the TATA box to which biological transcription factors bind. The following *g* bases are then designated a gene. The region between the end of a gene and the beginning of the next 0101 string becomes the promoter region for the downstream gene.

Each gene is "translated" into a gene product by incrementing each base by 1. A gene with the sequence 012130 will therefore result in a product with the sequence 123201. All of the promoter regions in the genome are searched for matches with each gene product; if a match to the product of gene **A** is found in the promoter region of gene **B**, we say that gene **A** controls gene **B**. This control may be either excitatory—**A** promotes the transcription of **B**—or inhibitory. In this way a genetic regulatory network is constructed from the randomly generated genome (Figure 1).

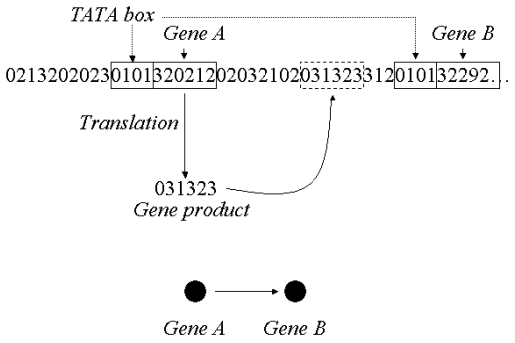


Figure 1. Reil's artificial genome model of a genetic regulatory network.

2.2 Update Scheme

Four network updating schemes were used—synchronous deterministic (SD), synchronous nondeterministic (SND) asynchronous deterministic (AD) and asynchronous nondeterministic (AND). In the deterministic update scheme each node is updated in turn in a fixed order, while in the nondeterministic schemes nodes are chosen for update at random, with replacement. The two synchronous update schemes are strictly equivalent, but both were included to facilitate comparison with the analogous asynchronous update schemes.

2.3 The Evolutionary Algorithm

There are two aspects to interesting behaviour in a network: the number of different limit cycles found, *n*, and the length of these limit cycles, *l*. To incorporate both of these variables, the fitness function for the EA, *f*, was defined as:

$$f = n(l / 2)$$

The values of *n* and *l* were estimated by generating a network and then running it 100 times from different random seeds, and counting the number and length of the observed limit cycles. Under these conditions *n* has a maximum value of 100 (a different limit cycle for each run of the network) and *l* has a maximum value of 1,000 (since networks which run for more than 1,000 steps without repeating a state are defined in this model as "chaotic"). The function *f* therefore is monotonically increasing with a maximum of 50,000 and as such should be an ideal candidate for a hill climbing algorithm.

Networks were generated at random. Each network was run 100 times from different initial states, and the number and length of observed limit cycles were counted. Each network was then evolved using the hill climbing algorithm described above, with a stopping criterion of 200 generations with no improvement in fitness or 1,000 generations with no improvement in fitness over the initial network. The resulting network was analysed as described for the initial network. Each network was evolved ten times, with a different random number seed each time. This procedure was repeated with mutation rates ranging from 0.00007 (approximately one mutation per chromosome) to 1.0 (every base mutated in each generation).

3 Results

3.1 Original GRNs

As a basis for comparison, we analysed the dynamics of the networks initially generated by the AG.

A major factor in the dynamics of the GRNs is the average connectivity of the nodes in the network. The average connectivity of a network <k> is determined by the interaction between the chromosome length, *c*, number of bases, *b*, TATA box length, *t*, and gene length, *g*, given by

$<k> = G / L$

where G is the expected number of links, given by

$G = c / b^t$

and L is the expected number of links (matches to the complement of a gene string) given by:

$L = (c / b^g) * GL$

We generated networks of different connectivity by manipulating these parameters. For each degree of connectivity, different proportions of the links were set to be inhibitory. During node update, inhibition was assumed to override activation, so if a node had both inhibitory and excitatory inputs it would always be off.

For each combination of connectivity and degree of inhibition 100 networks were randomly generated. Each network was run 100 times and the number and length of limit cycles reached was recorded (Figure 2; not all connectivities shown). Networks which ran for more than 1000 time steps without revisiting a previously visited state were assumed to have chaotic dynamics.

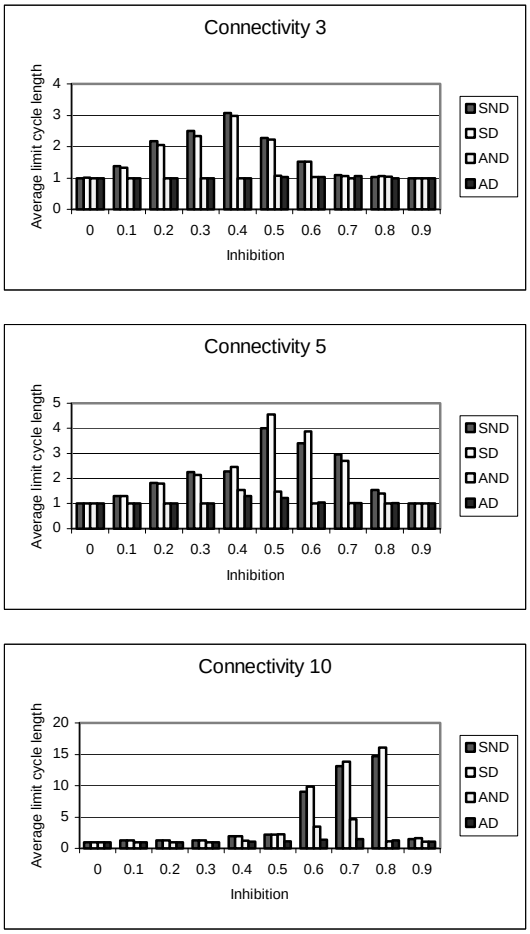


Figure 2. Average limit cycle length for networks of different degrees of connectivity and inhibition. Each graph is the average of 100 networks run 100 times from different starting points.

It is immediately apparent from Figure 2 that only networks with synchronous updating exhibit interesting dynamics. The average limit cycle length of asynchronous networks remains stubbornly at 1, except at very high levels of connectivity and inhibition. In almost all networks, asynchronous updating leads to the network, after a short transient period, freezing to a single state and remaining there. At connectivity levels of 10 and above a few limit cycles of 2 or 3 states arise, but this is still the exception to the rule.

We then selected a network topology for further investigation. Networks with an average connectivity of 4 and inhibition of 0.4 display dramatically different behaviour under different update rules. Synchronously updated networks have a large number of different limit cycle attractors of different lengths, while the same networks, asynchronously updated, almost always freeze quickly to a point attractor, where they remain. These networks were therefore selected for the application of the hill climbing algorithm.

The hill climbing algorithm works on a "mutant-champ" basis. A population consists of a single individual. At each time step each base of the genome is mutated with probability *mrate*; should a mutation occur, the base to which it is converted is selected with uniform probability. The mutated child replaces the parent if its fitness, defined as discussed above, is greater than or equal to that of its parent. Evolution continues until there has been no increase in fitness for 200 generations (or 1,000 generations if there has been no improvement over the fitness of the original individual).

An example of the typical dynamics of a network, before and after evolution is shown in Figure 3. In this figure genes are arranged along the x axis and time proceeds down the page; each row of a matrix represents the state of the network at a single time step. A black cell represents a gene which is on, and a white cell one which is off.

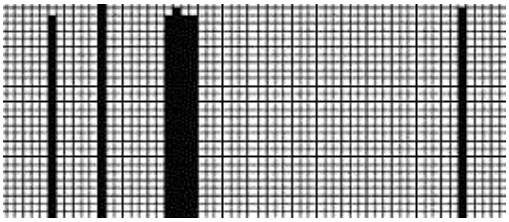
3.2 Evolved GRNs

In a synchronous network all nodes update simultaneously and each row in a gene expression diagram represents a single, identifiable "time step". Time is not so easily modelled in an asynchronous network. A time step consisting of a single gene update is obviously not equivalent to a time step in a synchronous network; it is, in effect, slicing time much more finely. A time step may alternatively be defined as occupying *n* node updates, where *n* is the number of nodes in the network (Schönfisch & de Roos, 1999). Depending on the details of the algorithm used, however, not all nodes will necessarily have been updated in a single time step under the latter definition. We used a time step of *n* updates in our diagrams, in order to maximise comparability with synchronous dynamics.

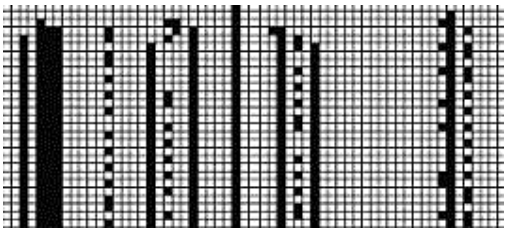
The original AG network, whose dynamics are shown in Figure 3a, has typically asynchronous behaviour. At the first time step all nodes are off, except for a single

node selected at random and set on. In the next time step *n* nodes have been updated, and three turn themselves on. By the third time step five nodes are on and they remain on for the rest of the run (not all of the run is shown here).

The dynamics of the same network after evolution are shown in Figure 3b. Once again a single, randomly chosen node is set to on in the first time step, and its activity propagates through the network. Unlike the original network, however, the network does not settle to a point attractor after an initial transient period. Instead genes continue to turn on and off in a pattern which looks almost like a limit cycle for the duration of the run. Once again, the full duration of the run is not shown in Figure 3.



a.



b.

Figure 3. Dynamics of evolved networks. a. Mutation rate 1 per generation. b. Mutation rate 0.001

Inspection of the gene expression diagrams suggests that evolved networks do exist in a different state space from the originally generated networks. In order to characterise this new state space, we ran 100 different networks 100 times each, with a different random number seed each time, and recorded the number of times each run revisited a previously seen state, and the number of time steps since that state was previously visited.

Figure 4 summarises the results of the evolutionary process on the average number and length of limit cycles, for a variety of mutation rates.

Figure 3 illustrates, and Figure 4 confirms, that the evolutionary process produces networks which, instead of freezing to a point attractor, tend to revisit previously seen states; while their behaviour is not chaotic, neither is it frozen; the system is dynamic, and confined to a small part of the state space available to it. Figure 5 shows the average number of states visited by a network before and after evolution, for various mutation rates.

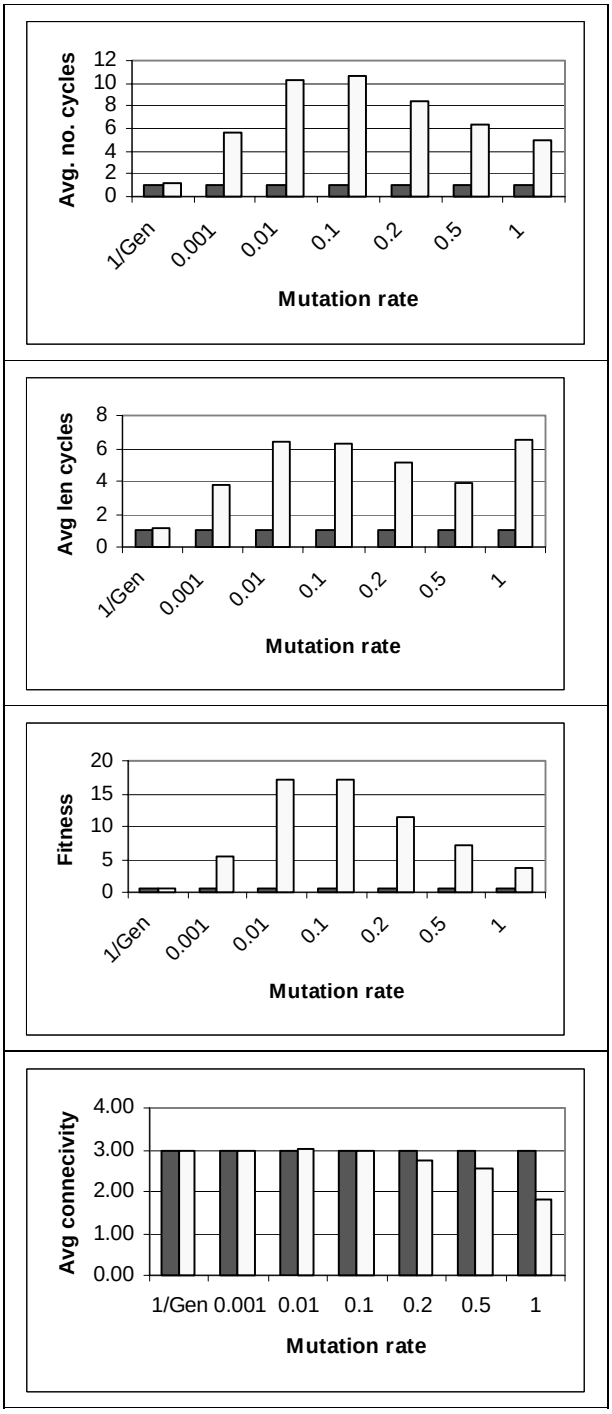


Figure 4. Effect of the evolutionary algorithm on measures of network structure and dynamics. In each case the dark bar represents the behaviour of the network before evolution, while the light bar is the same network after evolution. Each network was evolved 100 times using a different random number seed each time. a. Average number of times a prior state is revisited; b. Average time taken to first revisit a state; c. Average fitness of the network; d. Number of nodes in the network; e. Number of links in the network; f. Average connectivity of the network.

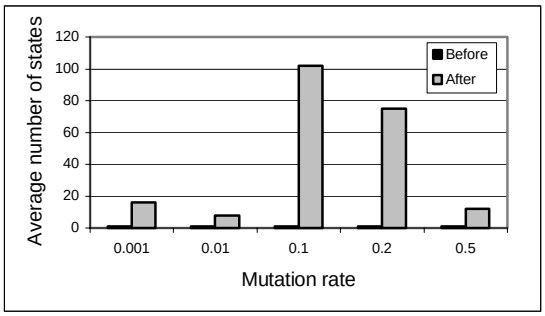


Figure 5. Average number of states visited by a network in 1,000 time steps, before and after evolution.

Similarly, Figure 6 shows the average number of times each state was visited, for a variety of mutation rates.

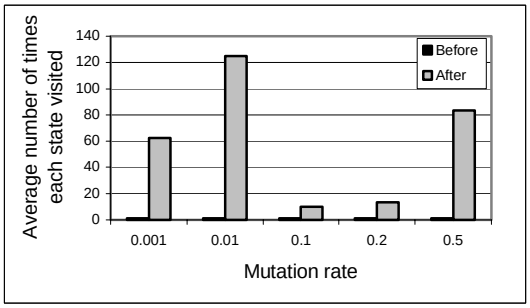


Figure 6. Average number of times each state visited by a network in 1,000 time steps, before and after evolution.

The networks after evolution display much more interesting dynamic behaviour, despite the fact that they are updated asynchronously. This is an encouraging result, since it indicates that biologically plausible (i.e. asynchronously updated) networks can, in fact, exhibit "fuzzy limit cycle" behaviour.

Harvey & Bossomaier (1997) cite Wuensche (1994) as saying that asynchronous updating "introduces an element of non-determinism…that might render any genetic search in a space of such networks very difficult". This does not, in fact, appear to be the case. The networks in this study evolve towards semi-orderly dynamic behaviour under almost all the conditions examined. The exceptions occur when very low or very high mutation rates are used. With very low mutation rates, on average one mutation per genome, most networks drift randomly in a sea of neutral mutations and never manage to find a path to a region of higher fitness. The occasional network was observed to increase in fitness slightly, but it was not a general pattern. Very high rates, such that nearly every base in the genome would be mutated every generation, are highly disruptive to the evolutionary process, producing an uncorrelated fitness landscape on which gradual adaptation simply cannot occur. Every intermediate mutation rate produced networks with the desired behaviour.

The evolutionary algorithm used here is the simplest one possible—a greedy hill climber. It is effective because the fitness function is designed in such a way as to produce a monotonically increasing fitness landscape, in which it is easy to find paths to increased fitness.

The algorithm also allows for the exploitation of neutral mutations. A child network is accepted to replace the parent if its fitness is greater than or equal to that of the parent. By accepting mutations of equal fitness it permits the system to explore paths of neutral fitness between fitness peaks and hence to explore more of the fitness landscape than would otherwise be possible. The importance of neutral mutations to biological evolution has been a topic of ongoing contention for several decades (e.g. Ohta, 1973; Brookfield, 2000; Newman & Englehart, 1998), but it has been suggested that neutrality or near-neutrality is important to the evolution of complexity in biological GRNs (Zuckerkandl, 1997; Gojobori, Tateno & Ohta, 1997).

Given that networks can be evolved fairly readily to exhibit complex dynamics under asynchronous updating, it then becomes of interest how this feat is achieved. What are the differences between evolved and unevolved networks?

The basic structure of the network changes very little with evolution. The number of nodes (genes) decreases slightly, with a consequent slight increase in average connectivity, but the diameter of the network (the longest of the shortest paths between every pair of nodes) and the cluster coefficient (a measure of local clustering) are essentially unchanged. both before and after evolution most of the nodes are part of a single large connected component (57 out of 61 nodes prior to evolution; 54 out of 58 post evolution). This is characteristic of networks exposed to relatively low mutation rates; as the mutation rate is increased the number of nodes in the network tends to decline. It is easy for mutation to disrupt a gene since the sole marker of a gene is the presence of a 0101 sequence. Any mutation to this string effectively eliminates that gene, and consequently increases the length of the promoter region for the next gene, making it more probable that then gene will match another gene product. Despite this, the network statistics tend to remain very similar before and after evolution.

It seems that the change in dynamics after evolution is not due to dramatic changes to the overall statistics of the networks. We then examined the connectivity distributions of the evolved networks, in order to determine whether the novel behaviour of the networks was attributable to changes in the pattern of connectivity. Figure 7 shows the input connectivity distribution before and after evolution, and Figure 8 shows the corresponding output degree distribution.

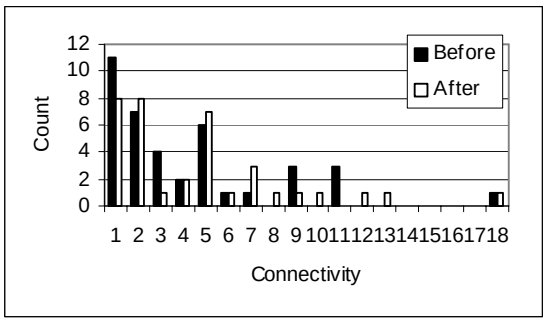


Figure 7. Input degree distribution of the network in Table 2 before and after evolution.

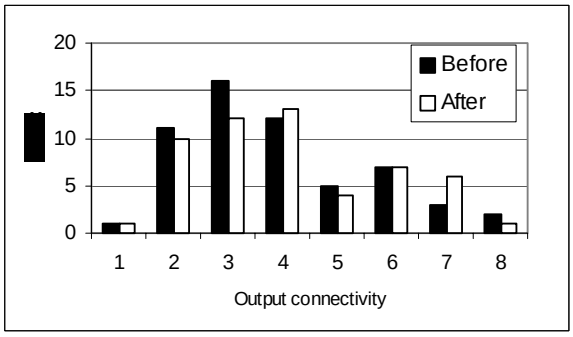


Figure 8. Output degree distribution of the network in Table 2 before and after evolution.

Figure 8 shows that the distribution of output connectivity is also shifted towards a more even, scale-free pattern, although this occurs to a lesser extent than in the input connectivity. It appears that the connectivity distribution of the network determines its dynamics under asynchronous node update. A network with a scale-free pattern of connectivity appears to be capable of producing rich dynamic behaviour under conditions of asynchronous updating in a way which a randomly connected—or, indeed, a self-organized AG network with Poisson input distribution and Gaussian output distribution—is not.

4 Conclusions

The artificial genome model of genetic regulatory networks is more biologically plausible than older models such as random Boolean networks, and has some interesting, and potentially useful, properties. One drawback of the AG model, however, is that the network dynamics rely on synchronous updating, which is biologically implausible; when a more realistic asynchronous updating scheme is employed the dynamic behaviour collapses to a single point attractor under almost all conditions.

We have demonstrated that a simple evolutionary algorithm applied to an AG-generated network can exhibit complex and interesting dynamics even under an asynchronous scheme. This dynamic behaviour appears to arise from a connectivity distribution very different from the uniform connectivity pattern usually assumed in computational models of genetic regulatory networks, and much closer to the scale free pattern observed in many naturally evolved biological networks.

5 References

Abramson, G. & Zanette, D. H. (1998). Globally coupled maps with asynchronous updating. *Physical Review e* **58**(4): 4454 - 4460.

Blok, H. J. & Bergersen, B. (1999). Synchronous versus asynchronous updating in the "game of Life". *Physical Review e* **59**(4): 3876 - 3879.

Brookfield, J. F. Y. (2000). Evolution: What determines the rate of sequence evolution?. *Current Biology* **10**: R410 - R411.

Di Paolo, E. A. (2001). Rhythmic and non-rhythmic attractors in asynchronous random Boolean networks. *BioSystems* **59**: 185 - 195.

Gojobori, T., Tateno, Y. & Ohta, T. (1997). Networks and evolution of molecular information. *Journal of Molecular Evolution* **44**(Suppl 1): 1.

Harvey, I. & Bossomaier, T. (1997). Time out of joint: Attractors in random Boolean networks. In P. Husbands & I. Harvey (eds.) *Proceedings of the Fourth European Conference on Artificial Life (ECAL97)*. Cambridge, MA, 67 – 75, MIT Press.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N. & Barabasi, A.-L. (2000). The large-scale organization of metabolic networks. *Nature* **407**: 651 - 654.

Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution.* New York: Oxford University Press.

Nehaniv, C. L. (2002). Evolution in Asynchronous Cellular Automata, In R.K. Standish, M.A. Bedau, and H.A. Abbass (eds.) *Proceedings of the Eighth International Conference on Artificial Life*, MIT Press, pp. 65-78.

Newman, M. E. J. & Engelhardt, R. (1998). Effects of selective neutrality on the evolution of molecular species. *Proceedings of the Royal Society of London*, Series B 265: 1593 - 1599.

Ohta, T. (1973). Slightly deleterious mutant substitutions in evolution. *Nature* **246**: 96 - 98.

Reil, T. (1999). Dynamics of gene expression in an artificial genome: Implications for biological and artificial ontogeny. In Floreano, D., Mondada, F. & Nicoud, J. D. (eds.) *Proceedings of the 5th European Conference on Artificial Life*. New York, NY, 457 – 466, Springer Verlag.

Schonfisch, B. & de Roos, A. (1999). Synchronous and asynchronous updating in cellular automata. *BioSystems* **51**: 123 - 143.

Sole, R. & Pastor-Santorros, R. (2002). *Complex networks in genomic and proteinic*. Santa Fe Institute Working Paper 02-06-026.

Wagner, A. & Fell, D. (2000). *The small world inside large metabolic networks*. Santa Fe Institute Working Paper.

Zuckerkandl, E. (1997). Neutral and nonneutral mutations: The creative mix - Evolution of complexity in gene interaction systems. *Journal of Molecular Evolution* **44**(Suppl 1): S2 - S8.