Basins of attraction in network dynamics:
A conceptual framework for biomolecular networks [*]

Andrew Wuensche
Discrete Dynamics Inc.
7 Calle Andreita, Santa Fe, New Mexico 87506, USA,
wuensch@santafe.edu, www.santafe.edu/~wuensch/

Abstract

The state-space of a discrete dynamical network is connected into basins of attraction, mathematical objects that can be computed and shown as graphs for small networks. Multiple attractors explain how the same genetic regulatory network can maintain different stable patterns of gene activation, the cell types in multi-cellular organisms. Ideas of modularity suggest that the overall genetic network is actually made up of semi-independent sub-networks. Each sub-network also settles into one of a range attractors according to its current state, which if perturbed can cause the dynamics to jump to alternative attractors. A network's "memory", its ability to categorize, is provided by its separate basins of attraction, and also by the topology of the trees and sub-trees rooted on each attractor.

Based on computer simulations of cellular automata and random Boolean networks, using the authors software Discrete Dynamics Lab[23], this paper describes how basins of attraction might provide a conceptual framework for biomolecular networks.

1 Introduction

Some of the outstanding questions in genetics, evolution, evolvability and development, including notions of modularity, will involve unraveling and comprehending networks of interacting elements.

In differentiation and gene regulation, feedback makes the one way signaling pathway paradigm inadequate. It has been superseded by a dynamical network approach. The dynamics manifests itself in two ways. Firstly by the changing pattern of activation *on* network elements (genes, or neurons in a neural network). Secondly by the dynamics *of* the network, how the network architecture itself changes, its set of elements and how they connect, driven by evolution, development and learning.

Another issue is the quality of different network architectures in terms of connectivity, and also the updating logic. Of the possible types of network architectures and logical schemes, has nature selected some tiny subset? Are there characteristic biases which affect dynamics? Probably. This kind of ensemble approach to biological networks might complement the search for the detailed biomolecular interactions.

Recent studies have shown that network topologies in real cell signaling and metabolic networks share some universal features[1, 8, 16]. The connectivity appears to be "scale-free", where link frequency (the number of connections at each node) roughly follows a power law distribution. Most elements have few connections, and a few are highly connected. The "small world" average distance between pairs of elements depends on those few highly connect nodes, which if disconnected can break the network into separate components. Similar topologies have been found in many different contexts, both natural and artificial, ranging from ecosystems to the world wide web. Are gene regulatory networks of this type? Or are they broken into modules, small semi-independent sub-networks, responsible for useful adaptations and functions which are conserved over long stretches of the evolutionary tree[12]. Note that each module itself could also have a scale-free topology. This question goes to the heart of modularity and evolvability. Figure 1 illustrates these two types of hypothetical network. Although their link frequency profiles are similar (figure 2), their dynamics turn out to be very different. Results described in section 8 indicate that the modular network has more basins of attraction, which are relatively more stable, than the fully connected network.

[*] To appear in "Modularity in development and evolution", eds. G. Schlosser and G. P. Wagner, Chicago University Press.

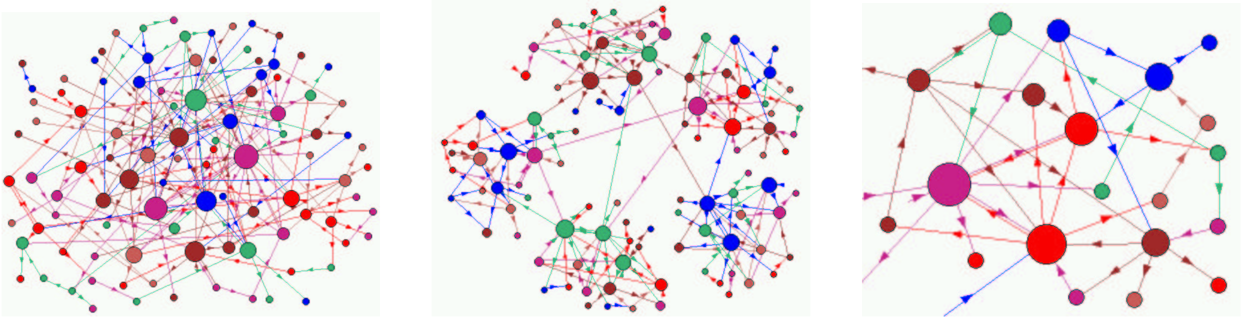


Figure 1: Hypothetical networks of interacting elements (size n=100) with an approximate power-law distribution of connections, both inputs (k) and outputs, which are represented by directed links (with arrows). Nodes are scaled according to k and average $k \simeq 2.2$. *left*: A fully connected network. *center*: A network made up of five weakly inter-linked n=20 sub-networks or modules. *right*: A detail of the top right sub-network. These are examples of random Boolean networks defined in section 4.

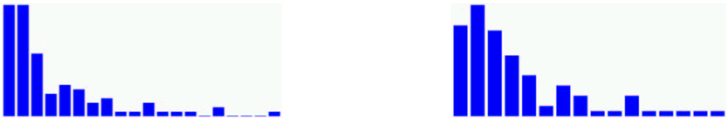


Figure 2: Histograms of link frequency (y axis) against link size (x axis), for inputs+outputs, in the networks in figure 1. The fully connected network (*left*), and modular network (*right*), have a similar link frequency profile. However their dynamics are very different, as described in section 8.

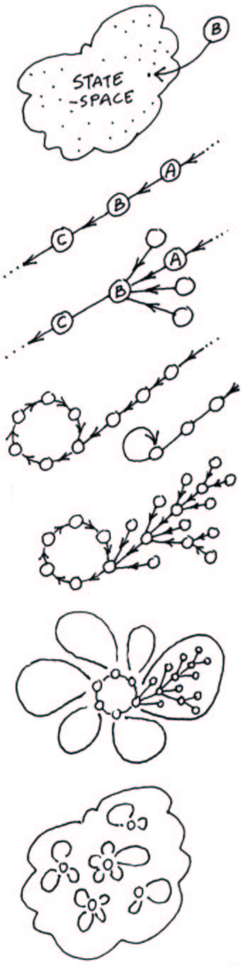
Dynamical networks in biology are found wherever one cares to look, from the brain to ecology. In the context of modularity in evolution and development the networks in question are genetic regulatory networks in cell differentiation, protein networks in cell signaling, networks of cells in tissues and organs, and of organs in the body. These networks overlap, making super-networks, and break down into sub-networks through many levels. The finer details of signaling pathways and network fragments are being discovered, both by painstaking experiment on Drosophila and other organisms[17], and by the new micro-array methods providing floods of data on the dynamics of gene expression patterns in development. There appears to be an urgent need for theoretical approximations and concepts to keep pace with the data[13].

Can methods from complex systems, network theory and discrete dynamics contribute? Cellular automata have provided models of pattern formation in biological organisms[2]. Boolean networks have provided models of genetic networks underlying cell differentiation based on basins of attraction[9, 5, 14, 26]. Basins of attraction can be computed and portrayed[21, 22, 23], revealing the global dynamics on small networks. Could this provide a conceptual framework for networks in biology?

Discrete dynamical networks, as abstract systems, manifest ubiquitous emergent properties which transcend any particular context, studied for their own sake, not just as models of something else. This paper will outline some of these properties for cellular automata, and for random Boolean networks, which are more general and probably have more biological relevance. The key emergent property is that the dynamics on the networks converge, thus fall into a number of "basins of attraction", which hierarchically categorize patterns of activation, "state-space", creating memory as a function of the network architecture. High convergence implies order, low convergence implies disorder or chaos. The most interesting phenomena occur at the transition, sometimes called the "edge of chaos"[11].

Basins of attraction sum up the network's global dynamics, analogous to Poincaré's "phase portrait" which provided powerful insights in continuous dynamics. For small systems, its possible to compute and draw basins of attraction, and measure their convergence and stability to perturbation. Analogous basins of attraction can be imagined in biological networks. That these networks are very simple might be a strength rather than a weakness when extrapolating the ideas to real systems, by the argument that if the simple network has particular emergent properties, a real biological network, infinitely more subtle and complex, should be capable of that *at least*. Also, a biologically faithful model might become so elaborate as to mask the underlying phenomena that one seeks to capture, which might be clear in a simple model.

2 Basins of Attraction



For a network size n, an example of one of its states B might be $1010\ldots0110$. *State-space* is made up of all 2^n states, the space of all possible bitstrings or patterns.

Part of a *trajectory* in state-space, where C is a successor of B, and A is a *pre-image* of B, according to the dynamics on the network.

The state B may have other pre-images besides A, the total is the *in-degree*. The pre-image states may have their own pre-images or none. States without pre-images are known as *garden-of-Eden* states.

Any trajectory must sooner or later encounter a state that occurred previously - it has entered an attractor cycle. The trajectory leading to the attractor is a *transient*. The period of the attractor is the number of states in its cycle, which may be just one - a point attractor.

Take a state on the attractor, find its pre-images (excluding the pre-image on the attractor). Now find the pre-images of each pre-image, and so on, until all garden-of-Eden states are reached. The graph of linked states is a *transient tree* rooted on the attractor state. Part of the transient tree is a subtree defined by its root.

Construct each transient tree (if any) from each attractor state. The complete graph is the *basin of attraction*. Some basins of attraction have no transient trees, just the bare “attractor”.

Now find every attractor cycle in state-space and construct its basin of attraction. This is the *basin of attraction field* containing all 2^n states in state-space, but now linked according to the dynamics on the network. Each discrete dynamical network imposes a particular basin of attraction field on state-space.

Figure 3: State-space and basins of attraction.

Figure 3 provides a summary of the idea of state-space and basins of attraction in idealized networks, sometimes called “decision making” networks. The dynamics depends on the connections and update logic of each element, which “decides” its next value based on the values of the few elements that provide its inputs, which might include self-input. The result is a complex web of feedback making the dynamics difficult to treat analytically, despite the extreme simplicity of the underlying network. In fact, although the dynamics are deterministic, the future is in general unpredictable. Understanding these systems relies chiefly on computer simulation.

A more precise definition of the network architecture is given in later sections. For the moment we will define just some essential concepts and properties, and also some terminology. At a given moment, each element in the network has a value, for simplicity 0 or 1 in a binary network, though the same arguments apply to a multi-state network. The pattern of 0s and 1s across the whole network is the network’s configuration, pattern of activation, or “state” at that moment in time, which can be represented as a bit-string. Time proceeds in discrete steps, “time-steps”. At each time-step, all signals transmitted by network links are processed simultaneously, that is synchronously, in parallel. This transforms a state at time-step t to another state at time-step $t + 1$, then another state at $t + 2$ and so on; the system is iterated. This process is deterministic, there is just one successor to each state, and continues indefinitely, while the network architecture itself (its wiring scheme and rule scheme) stays the same. That is, the source of the inputs (the “wiring”), and the logical function (the “rule”) that each element executes on its inputs to update its value,

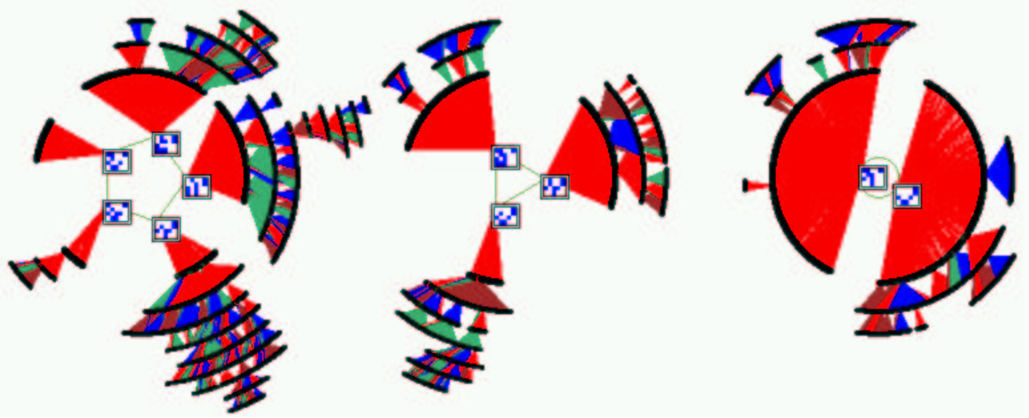
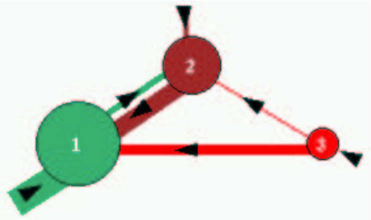


Figure 4: The basin of attraction field of the n=20 sub-network shown in detail in figure 1 (*right*). The rules (input logic) were assigned at random. State-space (size $2^{20} \simeq 1.05$ million) is partitioned into three basins of attraction. The attractor states are shown as 5x4 bit patterns. The table and diagram on the right show the probability of jumping between basins due to one-bit perturbations of their attractor states. P = attractor period, J = possible jumps ($P \times n$), and V% is the basin "volume" as a percentage of state-space. For example in basin 1, P=5, J=100 possible jumps, 15 of these jump to basin 2, and 85 back to itself, so basin 1 is relatively stable. Basin 3 has relatively few jumps back to itself so in unstable, it is also unreachable from the other basins. The diagram below the table, the "metagraph" (see section 5), shows the same data graphically. Node size reflects basin volume, link thickness percentage jumps, arrows the direction, and the short stubs self-jumps. The fraction of garden-of-Eden states in all three basins is 0.999+ indicating high convergence and order.

	1	2	3	P	J	V%
1:	85	15	.	5	100	61.78
2:	48	12	.	3	60	28.57
3:	32	6	2	2	40	9.65



do not change. The sequence of states is called a "trajectory". A "space-time" pattern representation of the dynamics is made by placing one-dimensional bit-strings, representing successive states, one below the other (time proceeds down) as in figures 5, where 0s and 1s are shown as white and black dots. This demonstrates how the pattern evolves from an initial state. It might stabilize, become periodic, chaotic, or show some interesting pattern formation.

Because the network is finite, we can define a "state-space" as the space of all possible bit-strings or states. There are 2^n unique states for a binary network of size n. We can say that a trajectory started from some initial state moves through its state-space. Because the state-space is finite, sooner or later the trajectory must encounter a state that occurred before. When this happens, because the system is deterministic, the trajectory must become trapped in a perpetual cycle of of repeating states, a state cycle, or "attractor". The number of time-steps between the repeats of a state is the attractor period, which could be just one, a "point attractor", where the space-time pattern has completely stabilized. Conversely, a chaotic space-time pattern might have a very large attractor period.

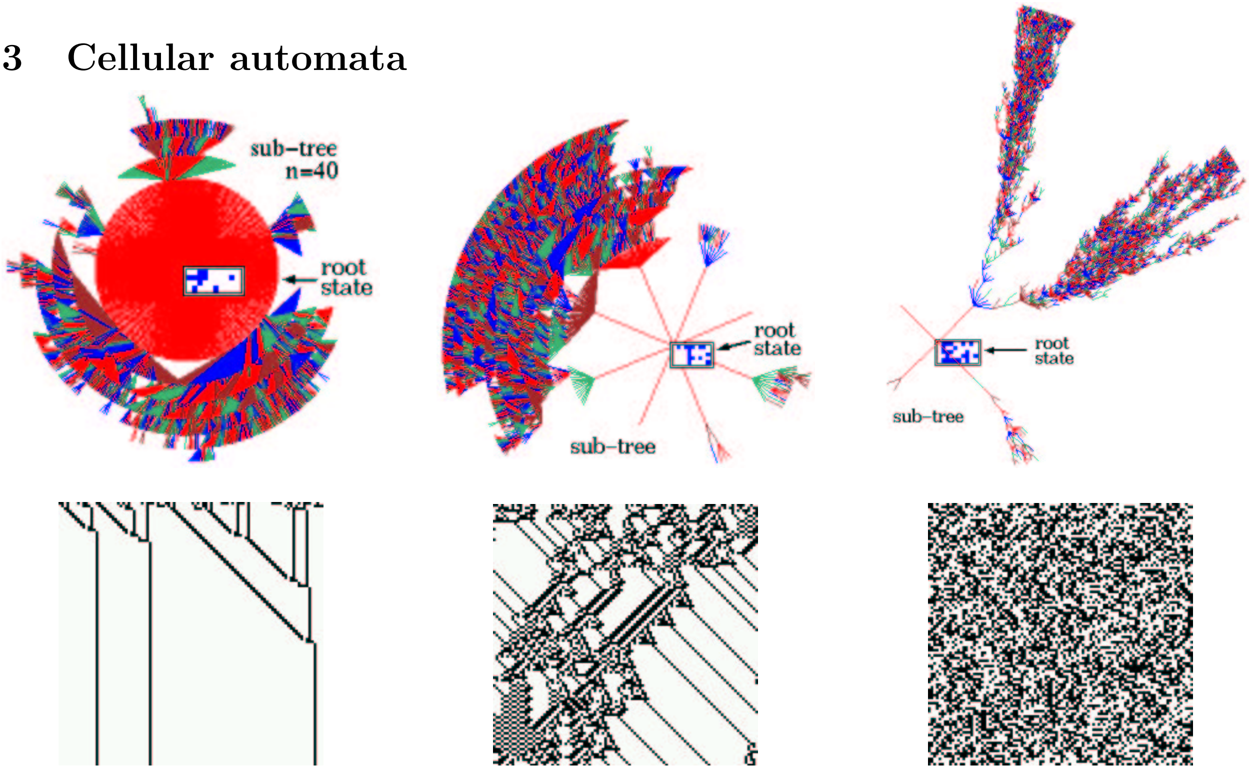
Each state has just one successor, but what about a state's immediate predecessors? It turns out that a state can have any number of these, called "pre-images", including none. The number of pre-images a state has is its "in-degree". The existence of in-degrees other than exactly one, and the existence of states outside the attractor, are conditions that must imply each other, otherwise the basin would be just a bare attractor cycle. States outside an attractor lie on trajectories that flow to the attractor, known as "transients". A state with zero in-degree is known as a "garden-of-Eden" state, and except for highly chaotic systems, most states, almost all for large networks, turn out to be garden-of-Eden states.

Some time ago I invented algorithms for finding the pre-images of a state directly, without having to exhaustively test the entire state-space[21, 22]. This allowed the efficient backwards tracing and reconstruction of the branching transients that flow into attractors, the "transient tree", where the "leaves" are garden-of-Eden states. Conversely, the flow towards attractors is convergent, like a river. High convergence implies order in space-time patterns; short, bushy, highly branching transient trees, with many leaves and small attractor periods. Conversely, low convergence implies chaos in space-time patterns; long, sparsely branching transient trees, with few leaves and long attractor periods. Figure 5 illustrates this, showing a transient sub-tree for three representative cellular automata rules, for order, complexity and chaos, and their corresponding space-time patterns. A simple measure of convergence, taken on a basin of attraction field, a single basin, a subtree, or on just part of a subtree for larger systems, is the density of garden-of-Eden states, and its rate of increase of with n. A more comprehensive measure is the in-degree frequency[25, 27].

The set of transient trees rooted on an attractor cycle is a “basin of attraction”. The dynamics of the network connects state-space into a number of distinct basins, the “basin of attraction field”, representing the systems global dynamics, as in figures 4, 8 and 10. State-space has now been partitioned, categorized, by the dynamics of the network into a number of separate basins of attraction. In addition the root of each subtree within a transient tree forms a sub-category. The precise way that states are linked into a basin of attraction field depends on the details of the network architecture.

Sub-trees and basins of attraction are portrayed as state transition graphs, vertices or nodes representing states are connected by directed edges. The direction of edges (i.e. time) flows inward from garden-of-Eden states to the attractor, and then clockwise around the attractor cycle, as indicated in figures 8 and 10. In the graphic convention[21, 23], the length of edges decreases with distance away from the attractor, and the diameter of the attractor cycle approaches an upper limit with increasing period.

3 Cellular automata



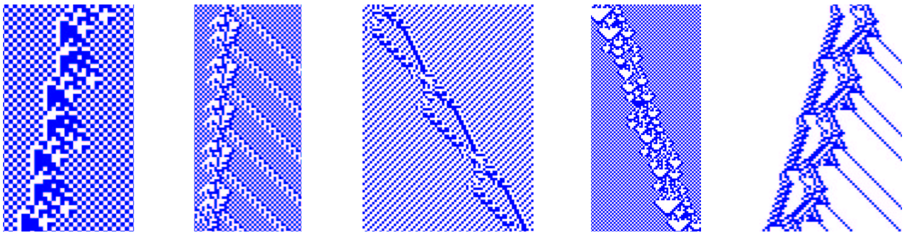
Ordered rule 01dc3610 Complex rule 6c1e53a8 Chaotic rule 994a6a65

Figure 5: Ordered, complex and chaotic dynamics of one-dimensional cellular automata are illustrated by the space-time patterns and subtrees of three typical k=5 rules (shown in hex, see section 4). The bottom row shows the space-time patterns from the same random initial state. The bit-strings (n=100) of successive time-steps (represented by white and black dots) are shown horizontally one below the other, i.e. time proceeds down. Above each space-time pattern is a typical sub-tree for the same rule. In this case n=40 for the ordered rule, and n=50 for the complex and chaotic rules. The root states were reached by first iterating the system forward by a few steps from a random initial state, then tracing the subtree backwards. Note that the convergence in the sub-trees, their branchiness or typical in-degree, relates to order-chaos in space-time patterns, where order has high, chaos low, convergence.

Cellular automata are networks whose elements form a regular lattice, possibly in one, two or three dimensions, with inputs from nearest neighbors (and possibly next nearest etc), and homogeneous logic. In fact the lattice is a consequence of the input connections. Cellular automata can be seen as a special case, a subset, of the more general system, random Boolean networks, and there are a variety of hybrid systems between the two. Cellular automata have provided models in computational and physical systems, and in biological systems, such as pattern formation, for example stripes on mammalian coats and patterns on nautilus shells[2], and in ecology, for example modeling forest fires. Large scale surface pattern can emerge from just local interactions, based on reaction-diffusion as in 7(*left*) and other mechanisms[20].

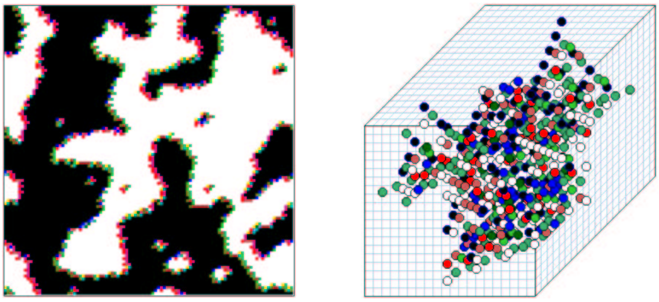
Another type of pattern formation which provides a striking example of self-organization is the emergence of coherent interacting structures, sometimes known as "particles" or "gliders"[3], as in figures 6 and 8(*top*). Only a small fraction of CA rules generate gliders. They are classified as complex, in contrast to ordered or chaotic[19]. Gliders can be embedded within a uniform or periodic background, and propagate at various velocities up to the system's "speed of light" set by the neighborhood diameter. Colliding gliders can annihilate, pass through each other, or produce new gliders. Compound gliders may emerge made up of sub-gliders re-colliding periodically, which can combine into yet higher order structures, and the process could unfold without limit in large enough systems. Once gliders have emerged, cellular automata dynamics can be described at a higher level, by glider collision rules as opposed to the underlying cellular automata rules[27].

Figure 7 shows examples of emergent patterns in two and three dimensional cellular automata. Figure 8 shows a large space-time pattern of a one dimensional complex cellular, and also the basin of attraction field of the same rule for a small system. Note that the basin symmetries can be explained by the regularity of cellular automata architecture [21]. In random Boolean networks these symmetries, and also gliders, are absent. Glider dynamics is said to occur at a phase transition in rule-space between order and chaos, relative to static rule parameters[11, 27]. Measures on space-time patterns allow rule space to be classified automatically, and complex rules to be identified[27]. There is a sense here of modularity. In cases where a bio-cellular or biomolecular substrate approximates a regular geometric lattice with local interactions, cellular automata might provide useful models.



(a)7e8696de (b)89ed7106 (c)89ed7106 (d)b51e9ce8 (e)b51e9ce8

Figure 6: Gliders, "glider guns" (which generate sub-gliders), and compound gliders in k=5 1d cellular automata. (c) is a compound glider made up of two independent gliders locked into a cycle of repeating collisions. (d) is a glider with a period of 106 time-steps. (e) is a compound glider-gun.



2d 100x100 triangular grid 3d 20x20x20

Figure 7: Examples of emergent patterns in 2d and 3d cellular automata. *left*: an evolved time-step of a 2d cellular automaton on a k=7 triangular lattice with a reaction-diffusion rule. *right*: a time-step of a 3d nearest neighbor (k=7) cellular automaton with a randomly selected rule, starting from a single central 1. View this as if looking up into a transparent box.

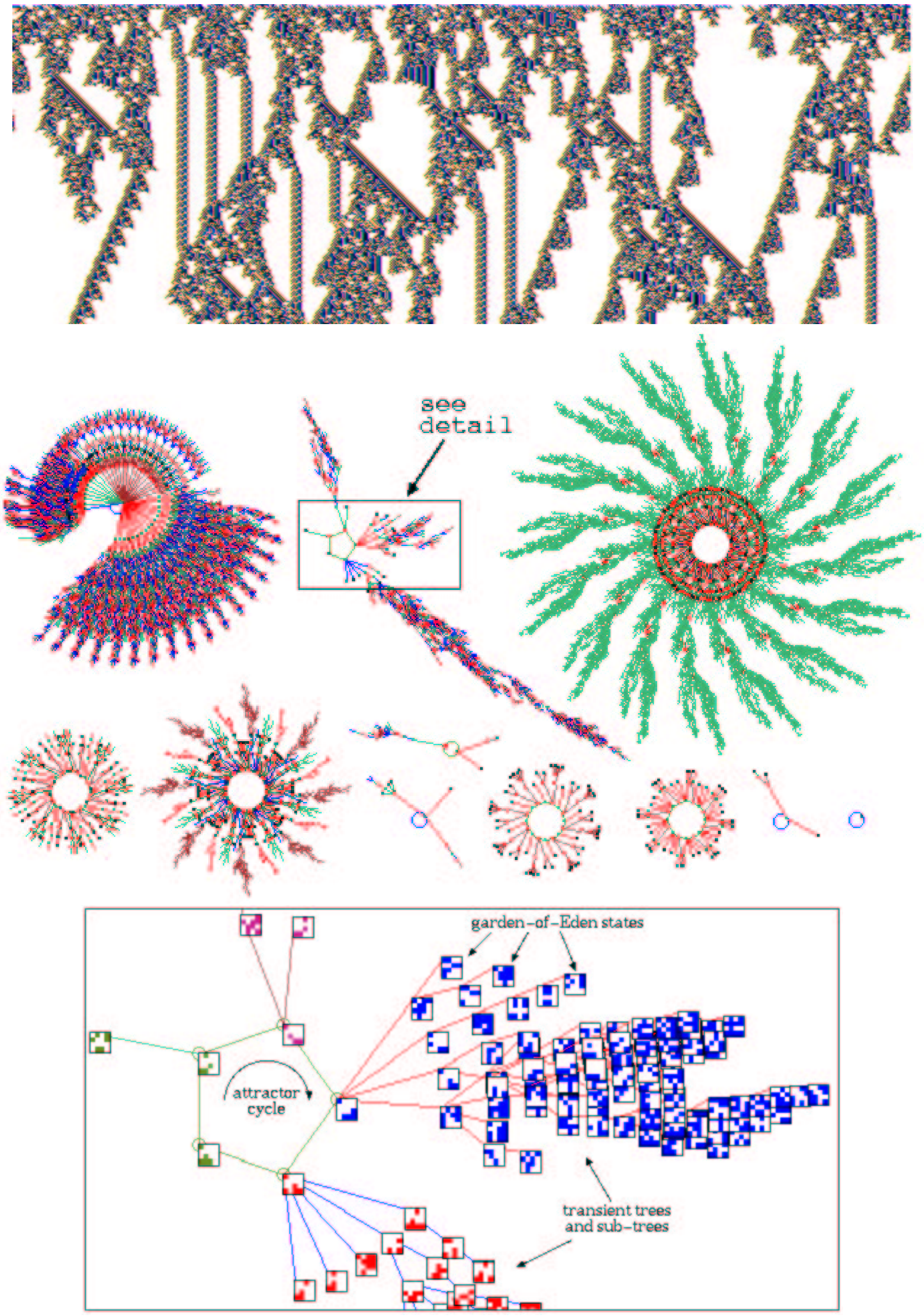


Figure 8: *top*: The space-time pattern of a 1d complex cellular automaton with interacting gliders[27], n=700, k=7, showing 308 times-steps from a random initial state. *center*: The basin of attraction field for the same rule, n=16. The 2^{16} states in state-space are connected into 89 basins of attraction, but only the 11 non-equivalent basins are shown, with symmetries characteristic of cellular automata. *bottom*: A detail of the second basin in the basin of attraction field, where states are shown as 4 × 4 bit patterns.

4 Random Boolean networks

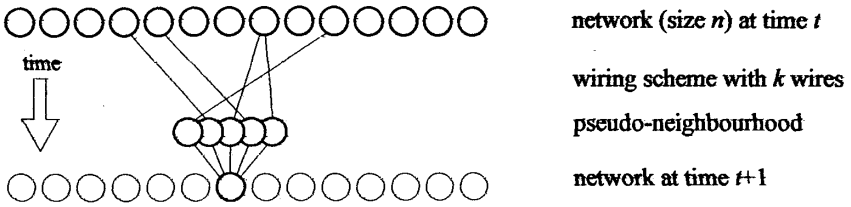


Figure 9: Random Boolean network architecture. Each element in the network synchronously updates its value according to the values in a pseudo-neighborhood, set by inputs (wiring) from anywhere in the network. Each network element may have a different number of input wires k, different wiring, and rule. The system is iterated.

A random Boolean network is a more general system than a cellular automaton. The connections, inputs, are nonlocal, and both inputs and logic can be different for each network element. The network seems more biological, for example in relation to neural networks, or genetic regulatory networks. The non-locality and heterogeneity prevent the sort of pattern formation possible with cellular automata, though one is of course free to introduce any measure of locality and homogeneity if appropriate, like taking inputs randomly from a local patch, or restricting, biasing, the range of logic.

The network consists of a set of elements, which could be model genes or neurons, taking inputs (0 or 1) from each other, and changing their value (0 or 1) according to some logical function on these inputs (the rule). The connectivity is usually sparse. The number of network elements, n, is usually much bigger that the typical number of inputs to each element k. The updating is synchronous, in discrete time-steps. The pattern of 0s and 1s across the network is a global "state", changing over discrete time. The network architecture itself does not change. Although there are various interesting possibilities for elaborating this model, having a larger range of values (or alphabet) than just 0 or 1, and various schemes for asynchronous updating, this discussion stays with the simple model.

Figure 9 shows one element's inputs "wired" between two consecutive time-steps, where the wires connect to a "pseudo-neighborhood", equivalent to a cellular automaton's real neighborhood. The number of possible input patterns on a pseudo-neighborhood of size k is 2^k. The most general expression of the Boolean function or rule is a look-up table (the rule-table) with 2^k entries, giving 2^{2^k} possible rules. Sub-categories of rules can also be expressed as simple algorithms, concise logical statements, totalistic rules[19], or threshold functions.

By convention[19] the rule-table, as in cellular automata, is arranged in descending order of the values of neighborhoods as binary numbers, and the resulting bit string converts to a decimal or hexadecimal (hex) rule number. For example, the k=3 neighborhoods are set out below in the conventional order, and the bitstring formed by the output of each neighborhood is the rule-table, or rule, in this case rule 30 in decimal, or rule "1e" in hex,

7	6	5	4	3	2	1	0	... neighborhoods, decimal
111	110	101	100	011	010	001	000	... all possible neighborhoods, binary
0	0	0	1	1	1	1	0	... outputs, the binary rule-table, equals 30 in decimal

The possible alternative network architectures, thus the behavior space of a random Boolean network, is very large, taking into account all possible wiring schemes and rule schemes, but there are also equivalence classes which reduce the number somewhat. In general, the number of effectively different random Boolean networks of size n, and thus basin of attraction fields, cannot exceed $(2^n)^{(2^n)}$.

The examples in figure 10 illustrate a small random Boolean network, its network architecture, basin of attraction field, and the stability of the attractors to perturbation. Note that because of the heterogeneity of wiring and rules, the basin topology lacks the symmetries characteristic of cellular automata.

el.	wiring	rule
12	10,1,7	86
11	6,2,9	4
10	10,10,12	196
9	2,10,4	52
8	5,6,8	234
7	12,5,12	100
6	1,9,0	6
5	5,7,5	100
4	4,11,7	6
3	8,12,12	94
2	11,6,12	74
1	6,5,9	214
0	12,9,6	188

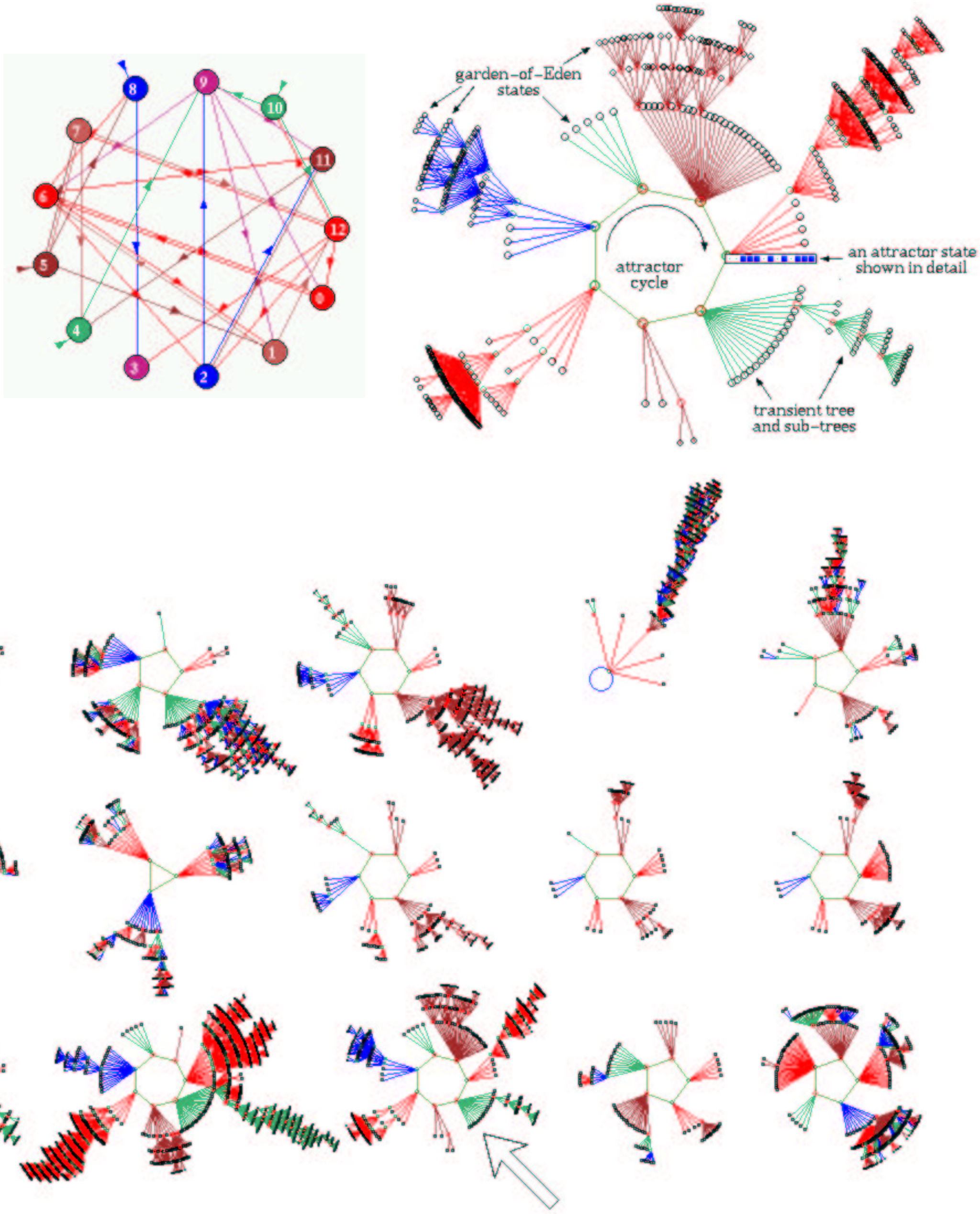

8
9
10
7
11
6
12
5
0
4
1
3
2
garden-of-Eden states
attractor cycle
an attractor state shown in detail
transient tree and sub-trees

Figure 10: A small random Boolean network, n=13, with homogeneous k=3 wiring, though some elements have more than one input from the same element. *top left*: The wiring/rule scheme, which fully defines the network.. *top center*: The network show as a directed graph with numbered nodes, self-links are short arrows sticking into nodes. *top right*: One of the basins of attraction. The basin links 604 states, of which 523 are garden-of-Eden states. The attractor period is 7, and one of the attractor states is shown in detail as a bit pattern. The direction of time is inwards from garden-of-Eden states to the attractor, then clock-wise. *bottom*: The basin of attraction field. The 2^{13} = 8192 states in state-space are organized into 15 basins, with attractor periods ranging between 1 and 7, and basin volume between 68 and 2724. The arrow points to the basin shown in more detail.

5 Jumping between basins due to attractor perturbations

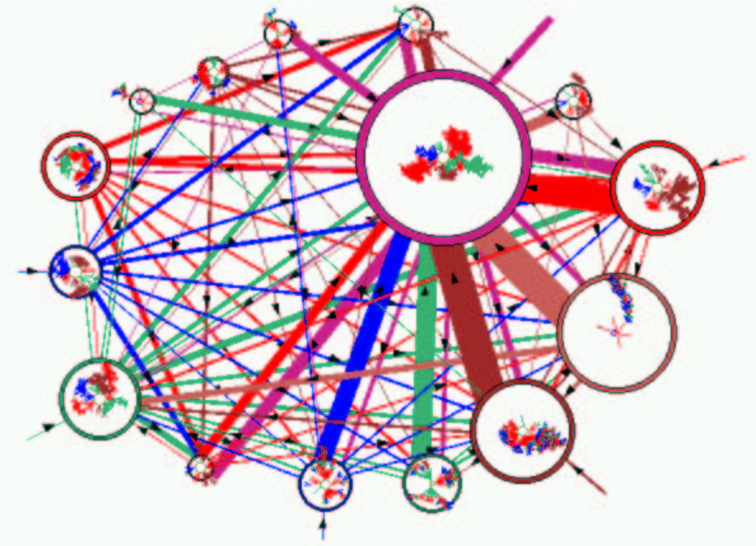


Figure 11: The meta-graph of the basin of attraction field in figure 10, showing the probability of jumping between basins due to single bit-flips to attractor states. Nodes representing basins (shown inside each node) are scaled according the number of states in the basin (basin volume). Links are scaled according to both basin volume and the jump probability. Arrows indicate the direction of jumps. Short stubs are self-jumps.

In a basin of attraction field, perturbations to network states will reset the dynamics, which may then jump to another basin, or to a different position in the same basin. Stability requires a high probability of returning to the same basin, whereas adaptability or differentiation requires appropriate jumps to other basins in response to specific signals. Perturbations or external signals are most likely to affect attractor states, because thats where the dynamics spends the most time.

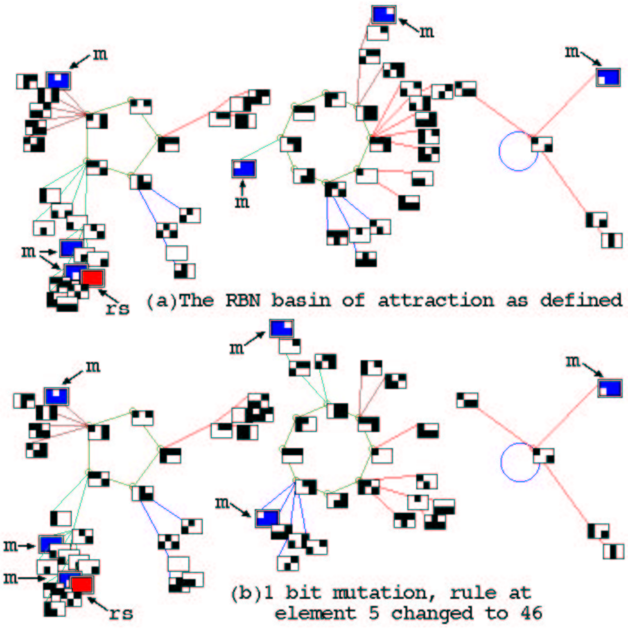
Taking single bit-flips, perturbations, to attractor states as the simplest case, the meta-graph of the basin of attraction field represents the probabilities of jumping between basins, and gives some insight into the stability and mobility between the basins of attraction. Figure 11 shows the meta-graph for the basin of attraction field in figure 10. Link thickness represents jump probability, and self-jumps are shown as short stubs. As the network was set up at random, not surprisingly the basins are unstable. The largest basin receives the most jumps, and sends jumps to other basins as well as back to itself.

6 Stability of basins due to mutations in the network architecture

Very fine mutations can be made to the wiring/rule scheme of a random Boolean network, such as moving one wiring input or flipping one bit in the rule of just one element. In general, the basin of attraction field is stable to such small mutations which result in slight changes to the overall topology. This is especially true for larger networks where single mutations become relatively less significant. However, even a single mutation can sometimes have a drastic effect, such as breaking an attractor cycle. The consequences of moving a connection wire are usually greater than a one-bit mutation in a rule[22].

In the software DDLab[23] there are "learning" algorithms that allow pre-images to be a automatically attached (learnt) or detached (forgotten) to/from a selected target state by mutations, moving wires or flipping bits in rules[22]. The aim is to "sculpt" a basin of attraction field to correspond to some desirable structure. The method is useful for fine tuning a network, but not for building a network from scratch, because of side effects, unplanned changes in other parts of the basin of attraction field.

One way to see the effect of network mutations is to highlight a set of states, and track them as the basin of attraction field changes for successive mutations. Figure 12 shows one such mutation for a very small network so that the pattern at each node can be seen. Larger networks are affected in analogous ways.


m
m
m
m
rs
(a)The RBN basin of attraction as defined
m
m
m
m
m
m
rs
(b)1 bit mutation, rule at element 5 changed to 46

el.	wiring	rule
5	2,4,5	62
4	5,0,1	61
3	4,3,5	108
2	2,5,0	5
1	4,2,1	64
0	3,1,2	231

Figure 12: The basin of attraction field of (a) the random Boolean network (n=6, k=3) as defined in the table above, and (b) the network following a 1-bit mutation to one of its rules. The attractors are stable to this mutation, though some differences in subtrees are evident. The all 1's reference state (rs) and the six states that differ from it by one bit (m), are indicated in the two fields.

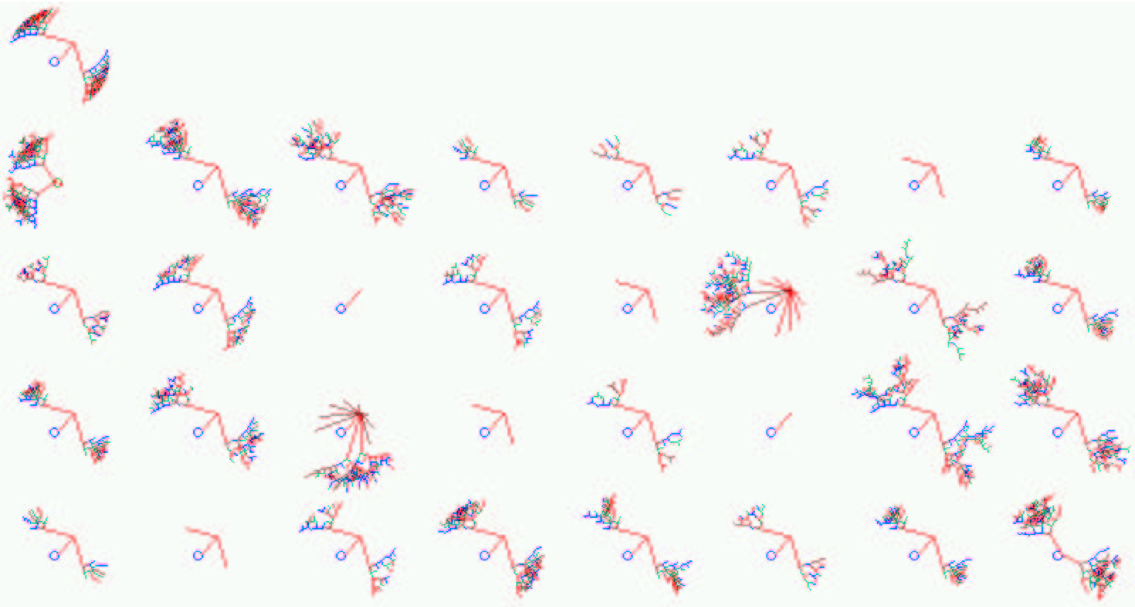
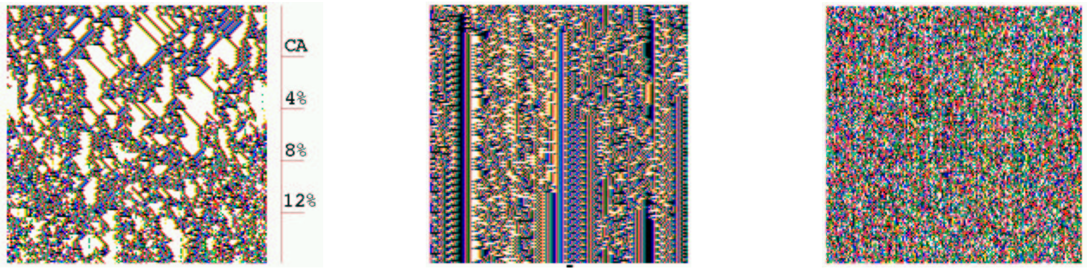


Figure 13: 32 mutant basins of attraction of the k=3 rule 195 (n=8, seed all 0s). *top left*: The original rule, where all states fall into just one very regular basin. The rule was first transformed to its equivalent k=5 rule (f00ff00f in hex), with 32 bits in its rule-table. All 32 one-bit mutant basins are shown. If the rule is the genotype, the basin of attraction can be seen as the phenotype.

Figure 13 shows the effects of all possible one-bit mutations to a cellular automaton rule, where all states fall into one very regular basin. As a cellular automaton has just one rule, effectively the same bit in each element's rule was mutated, so these mutations are more drastic than in figure 12. If the rule is seen as the genotype and the basin of attraction the phenotype, the relationship of the mutants to the original is striking. This is not atypical of mutations of cellular automata [21]. The phenotype is more sensitive to the mutation of some bits than others.

7 Order/chaos measures for large random Boolean networks



scrambling CA wiring | CA wiring, mixed rules | random Boolean network

Figure 14: Space-time patterns for intermediate one-dimensional architecture, from cellular automata (CA) to random Boolean networks. n=150, k=5, 150 time-steps from a random initial state. *left*: Starting off as a complex cellular automaton (as in figure 5), 4% (30/750) of available wires are randomized at 30 time-step intervals. The coherent pattern is progressively degraded. *center*: A network with cellular automata wiring but mixed rules, vertical features are evident. *right*: A random Boolean network, with no bias, shows extremely chaotic dynamics.

In contrast to cellular automata, glider dynamics and coherent pattern formation in general cannot occur in random Boolean networks because of their irregular architecture. Figure 14(*left*) shows glider dynamics degrading as local wiring is progressively scrambled.

One order-chaos measure for random Boolean network space-time patterns is the balance between "frozen", stabilized, regions and changing regions[10], as in figure 15(*top left*). Stable regions are characteristic of networks with low connectivity, $k \leq 3$, because rules which induce stability are relatively frequent in these rule-spaces. To induce stability (i.e. order) for $k \geq 4$, where chaotic rules become overwhelmingly predominant, biases on rules must be imposed. One way is to skew the proportion of 0s and 1s in a rule's look-up table away from an even distribution. Another is to set a high proportion of "canalizing" inputs, C. Although these biases overlap to a certain extent, a strong case has been made by Harris *et.al.*[6] that the latter type of bias is the most significant in control rules governing the activity of eukaryotic genes.

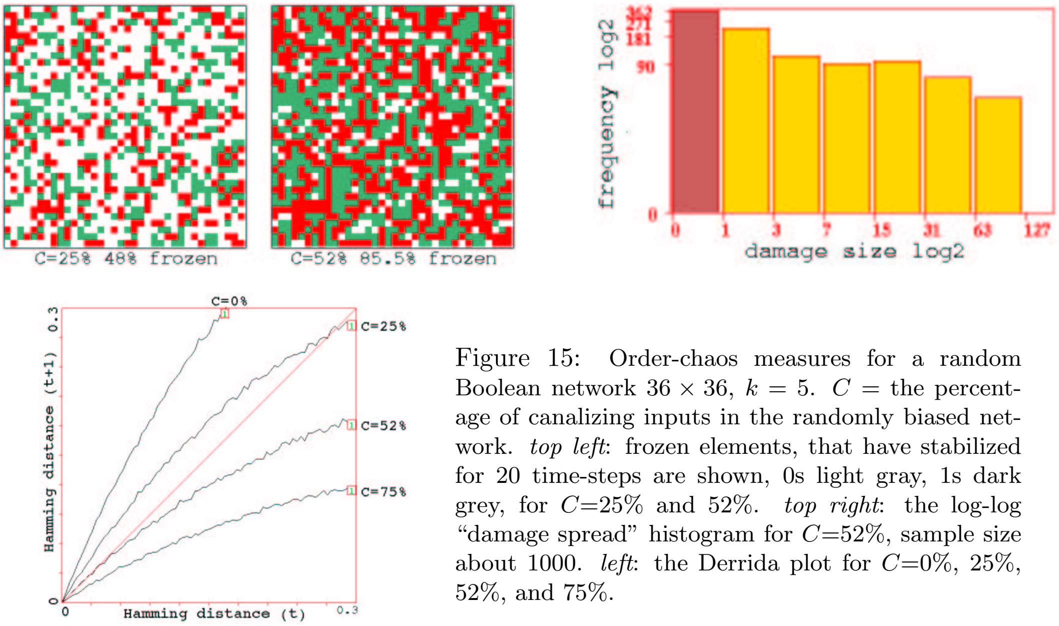


Figure 15: Order-chaos measures for a random Boolean network 36 × 36, k = 5. C = the percentage of canalizing inputs in the randomly biased network. *top left*: frozen elements, that have stabilized for 20 time-steps are shown, 0s light gray, 1s dark grey, for C=25% and 52%. *top right*: the log-log "damage spread" histogram for C=52%, sample size about 1000. *left*: the Derrida plot for C=0%, 25%, 52%, and 75%.

In a rule's look-up table, an input wire is said to be canalizing if a particular input value (0 or 1) determines the output irrespective of the other inputs. A rule's degree of canalization can range from 0 to k (for the same output). For the network as a whole, the percentage of all inputs that are canalizing, C, can be randomly set. A random Boolean network's order-chaos characteristics for varying C, applied especially for large networks, are captured by the measures illustrated in figure 15, and described below[4, 6, 23].

The "Derrida plot", analogous to the Liapunov exponent in continuous dynamics, measures the divergence of trajectories based on normalized Hamming distance H_t, the fraction of bits that differ between two patterns. Pairs of random states separated by H_t are independently iterated forward by one time-step. For a sample of random pairs, the average H_{t+1} is plotted against H_t, and the plot is repeated for increasing H_t. A curve above the main diagonal indicates divergent trajectories and chaos, below - convergence and order. A curve tangential to the main diagonal indicates balanced dynamics.

A related measure is the distribution of "damage" resulting from a single bit-flip at a random position in a random state, for a sample of random states. The bit-flip might cause an avalanche of difference, damage, relative to the original space-time pattern. The size of the resulting damage is measured once it has stabilized. A histogram is plotted of damage size against the frequency of sizes. Its shape indicates order/chaos in the network, where a balance between order and chaos approximates to a power law distribution. Results according to these measures for $k = 5$, indicate a balance at $C \approx 52\%$.

These and other methods are applied in the context of random Boolean network models of genetic regulatory networks. The conjecture is that evolution maintains genetic regulatory networks marginally on the ordered side of the order-chaos boundary to achieve stability and adaptability in the pattern of gene expression which defines the cell type[6].

8 Attractors in fully connected and modular networks

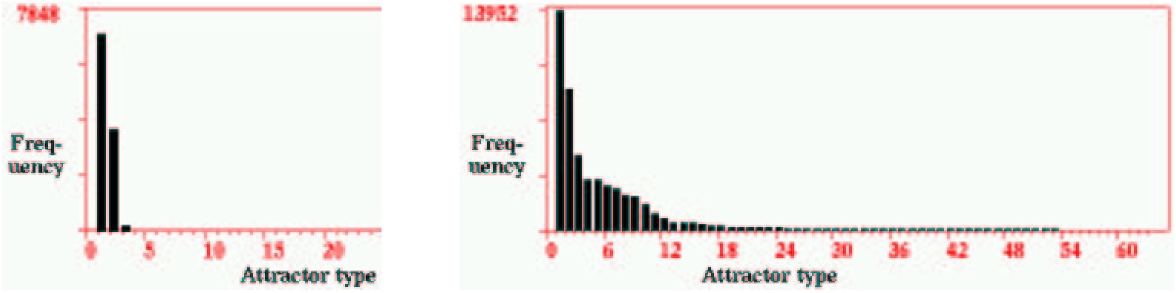


Figure 16: Attractor frequency histograms, showing the frequency (y axis, scale indicated top left) of falling into different attractors (x axis, sorted by basin volume) from a large sample of random initial states. These examples are for the "scale-free" networks in figure 1, where n=100. The frequency of each attractor is a statistical measure of basin volume, the fraction of state-space occupied by each basin. *left*: The fully connected network: the number of attractor types stabilized at just 3 after 10000+ runs. Their periods are 2046, 553, 380, with average transient length 605, 673, 97. *right*: The modular network: the number of attractor types stabilized at 53 after 50000+ runs, though about 2/3 of these represent very small basins. The 3 most frequent basins have periods 30, 14, 2, with average transient length 54, 46, 47.

For large networks, too big to generate the basin of attraction field, information on the field, the attractors, and basins of attraction, can be still be found by statistical methods[18], though for excessively chaotic networks, transients and attractor periods can be too long for the method to be practical. The network is run forward from many random initial states looking for state repeats to identify the different attractors, noting the periods and attractor states. Although this does not reveal subtree topology, the frequency of falling into different attractors is a statistical measure of basin volume, the fraction of state-space occupied by each basin, so all but the smallest basins are likely to be found in a large sample of initial states.

Figure 16 shows the attractor frequency histograms of the "scale-free" networks in figure 1, one fully connected, the other modular, but with similar link frequency and random rules. Figure 17 shows the meta-

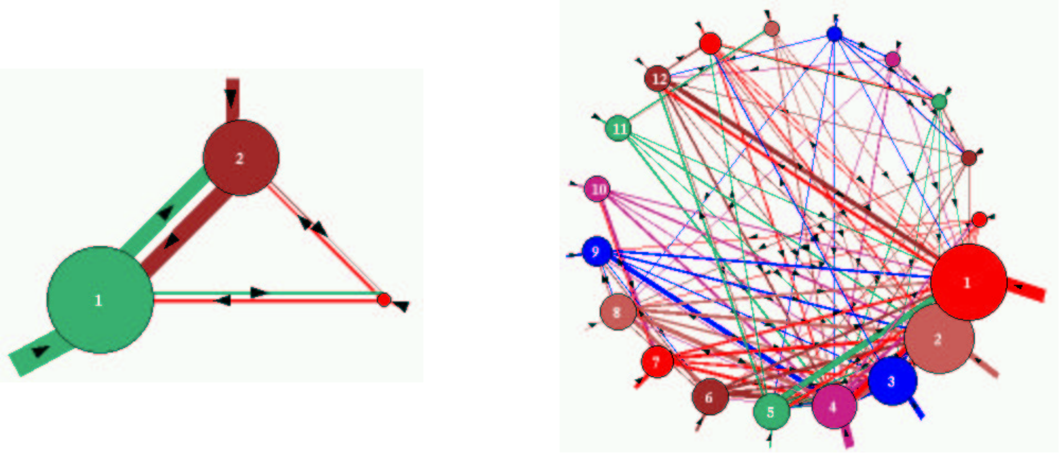


Figure 17: The meta-graphs of the attractor frequency histograms in figure 16, showing the probability of jumping between basins due to single bit-flips to attractor states. Nodes representing basins are scaled according to basin volume. Links are scaled according to both basin volume and the jump probability. Arrows indicate the direction of jumps. Short stubs are self-jumps. *left*: The fully connected network: the percentage of self-jumps is 60%, 36% and 6% respectively. *right*: The modular network: (for the 19 largest basins only, out of 53) the percentage of self-jumps is 41%, 20% and 31% respectively for the 3 largest basins. 11 of the smallest basins (not shown) are unreachable.

graphs of the attractor frequency histograms, showing the probability of jumping between basins due to single bit-flips to attractor states. The meta-graph for the modular network shows the 19 largest basins only.

The attractor frequency histograms and data indicate that the modular network has markedly more basins, with smaller attractor periods, and shorter transients, than the fully connected network. Other experiments on similar networks confirm the same tendency. The jump probabilities indicate that basins in the modular network are more stable than in the fully connected network, because the fraction of self-jumps for a typical basin is significantly greater than basin volume. This is not the case for the fully connected network, where self-jumps for the two biggest basins correlate closely with basin volume.

These results point to significant differences between the basin of attraction fields of modular and fully connected network architectures, where both are scale-free. Breaking a network into weekly linked modules increases both the number and stability of basins. Conversely, adding more links between the modules reduces both the number and stability of basins. The modules in the modular network are behaving like discrete coupled oscillators, perturbing each other between their alternative sub-attractors (see figure 4). This is where an explanation of the differences might be found.

Typical transient length and attractor period usually give some indication of the order/chaos characteristics of a network. The smaller attractor periods, and shorter transients, found in the modular network, would usually be a sign of a greater degree of "order". However, the order/chaos measures in section 7 do not show a significant difference. These are preliminary results, based on a limited sample. Further research is needed to better understand the behavior of these types of scale-free, modular, random Boolean networks.

9 Genetic regulatory networks

The cells of multicellular organisms differentiate within the developing embryo into the various cell types that form tissues by a process that is regulated at the molecular level by DNA sequences, encoding genes that produce proteins that regulate other genes. All eukaryotic cells in an organism carry essentially the same set of genes, some of which are expressed others not. A cell type depends on the particular subset of active genes, where the gene expression pattern needs to be stable but also adaptable. What then is the mechanism that maintains the many alternative patterns of gene expression of the various cell types making a multicellular organism?

Genes regulate each other's activity by coding for transcription factors, which may enhance or repress the expression of other genes by binding, often in combination, at particular sites. A given gene may directly regulate just a small set of other genes, those genes regulate other genes in turn, so a gene may indirectly influence the activity of many genes downstream. Conversely, a gene is indirectly influenced by many genes upstream. A gene may directly or indirectly contribute to regulating itself. The result is a genetic regulatory network, a complex feedback web of genes turning each other on and off.

This has been modeled as an idealized dynamical system, Kauffman's random Boolean network, where model genes are connected by directed links (transcription factors), updating their (on-off) state in parallel, according to the combinatorial logic of their inputs[9, 10, 14]. Depending on the network's current state, and biases on the connections and logic, the dynamics on such a network, as has been shown, can settle onto a number attractors, which are interpreted in this context as cell types. This provides a mechanism for explaining the apparent paradox that the same genome can create and maintain a variety of cell types and sub-types, skin, muscle, liver, brain, etc. In stem cells, trajectories leading to attractors have been interpreted as pathways of differentiation because small perturbations can nudge the dynamics towards different attractors.

The model has been criticized for being too simple, that parallel (synchronous) updating and the on-off characterization of genes are implausible idealizations when applied to real genetic networks, given that transcription is asynchronous and driven at different rates. However, gene activity at the molecular scale consists of discrete events occurring concurrently. Variable protein concentrations can be accounted for by genes being on for some fraction of a given time span. More elaborate models have been proposed but are harder to interpret. The random Boolean network idealization is arguably the simplest possible abstract model that is computationally tractable, well understood in fields outside biology, yet which still reflects the essential qualities of the dynamics of genetic networks.

In a cell type's gene expression pattern over a span of time (i.e. its space-time pattern), a particular gene may, broadly speaking, be either on, off, or changing. If a large proportion of the genes are changing, chaotic dynamics, the cell will be unstable. On the other hand, dynamics that settles to a pattern where a large proportion of the genes are permanently on or off (frozen) may be too inflexible for adaptive behavior. Cells constantly need to adapt their gene expression pattern in response to a variety of hormone and growth/differentiation factors from nearby cells. A cell type is probably a set of closely related gene expression patterns, not just on the attractors, but shifting around within the basin of attraction, allowing an essential measure of flexibility in behavior. Too much flexibility might allow a perturbation to flip the dynamics too readily into a different basin of attraction, to a different cell type such as a cancer cell, or from a bone cell to a fat cell. The conjecture is that the appropriate dynamical regime has evolved to find a delicate balance between stable on/off regions and dynamically changing regions[6].

The basins of attraction in random Boolean networks are idealized models for the stability of cell types against mutations, and also for their response to perturbations of the current state of gene activation. Jumping between basins due to these effects was described in sections 5 and 6. If a particular pattern of gene expression undergoes a perturbation, the dynamics may jump to a different subtree in the same basin, which would temporarily adapt the pattern, or it may be flipped to another basin, a different cell type. In this case the basin of attraction field remains unchanged. Alternatively, the network itself may undergo a mutation, resulting in an an altered basin of attraction field. This model provides a mechanism for both the stability and adaptability of gene expression.

10 Memory

Attractors classify state-space into broad categories, the network's content addressable "memory" [7]. Furthermore, state-space is categorized along transients, by the root of each subtree forming a hierarchy of sub-categories. This notion of memory far from the equilibrium condition of attractors greatly extends the classical concept of memory by attractors alone[22, 24].

It can be argued that in biological networks such as neural networks in the brain or networks of genes regulating the differentiation and adaptive behavior of cells, basins of attraction and subtrees must be just right for effective categorization. The dynamics need to be sufficiently versatile for adaptive behavior but short of chaotic to ensure reliable behavior, and this in turn implies a balance between order and chaos in the network.

A current research topic, known as the "inverse problem", is to find ways to deduce network architecture from usually incomplete data on transitions, such as a trajectory. This is significant because it could help to infer the genetic regulatory network, modeled a random Boolean network, from data on successive patterns of gene expression in the developing embryo[15]. In pattern recognition and similar applications in the area of artificial neural networks, solutions to the inverse problem would provide "learning" methods for random Boolean networks to make useful categories[22, 24].

11 Conclusions

In the context of random Boolean network models of genetic regulatory networks, basins of attraction represent a kind of modular functionality, in that they allow alternative patterns of gene expression in the same genome, providing a mechanism for cell differentiation, stability and adaptability.

Although the entire genome can be viewed as a vast genetic network, the notion of modularity suggests that the network is in fact built from functionally semi-independent, weekly inter-linked, sub-networks. Each sub-network may itself consist of further sub-networks in a sort of nested hierarchy. Figure 1 gave a hypothetical example. Section 8 gave some preliminary results indicating that the dynamics of "scale-free" modular networks are markedly different from their fully connected cousins, having more basins with smaller attractors and transients, and which are relatively more stable.

There is evidence that genetic regulatory sub-networks, highly adapted to useful functions, are conserved by evolution, and that examples of essentially the same sub-network are redeployed across different species, genera, and even families[12]. Examples of sub-networks, self-contained transcriptional circuits in regulated eukaryotic genes, are being found by experiment, together with the biases on their connections and logic[6].

Each sub-network module could behaves semi-independently within its own basin of attraction field, but cross-talk, perturbations, between sub-networks, and inter-cellular signals, would allow adaptive jumps between basins of attraction, or to transients within the same basin, shifting the dynamics away from the equilibrium at the attractor, constantly adapting to the metabolic environment.

There are many open questions stemming from this discussion. From the network theory perspective, a better understanding is needed of the dynamics of modular random Boolean networks and related systems, their basins of attraction, the stability of basins, and jump probabilities between basins. How does the dynamics change as more links are added between interacting sub-networks? What biases on connectivity and logic are required for networks models to reflect data coming from biology?

From the biological perspective, data is accumulating on the make up of genetic and other biological networks. What kind of biases might exist in general, in connectivity, modularity and logic. How does the biological network strike the right balance between stability and adaptability. Are genetic regulatory networks "scale-free"? And if modular, to what extent are the modules interlinked? In short, what are the general principles that describe the quality of these networks.

The hope is that the ideas, methods and tools for studying network dynamics in idealized networks can provide a conceptual framework for comprehending networks in biology.

Acknowledgments: Thanks to Ricard Sole, Steve Harris, Gerhart Schlosser and Gunter Wagner for comments and suggestions. Also to Pietro Speroni di Fenizio for suggesting the idea of analyzing jumps between basins. Thanks to the Santa Fe Institute where a significant part of the research was done. The simulations and figures in this paper were made with DDLab (Discrete Dynamics Lab)[23]. The software and documentation is available at `www.santafe.edu/~wuensch/ddlab.html` and `www.ddlab.com`.

References

[1] Albert,R., H. Jeong, and A-L. Barabási. 2000. Error and attack tolerance in complex networks. *Nature*, vol 406, 378-381.

[2] Ball, P. 1999. *The Self-Made Tapestry, Pattern Formation in Nature.* Oxford University Press.

[3] Conway,J.H. 1982. What is Life? In *Winning ways for your mathematical plays*, Berlekamp,E, J.H.Conway and R.Guy, Vol.2, chap.25, Academic Press, New York.

[4] Derrida,B., and D.Stauffer. 1986. Phase transitions in Two-Dimensional Kauffman Random Network Automata. *Europhys.Lett.* 2, 739.

[5] Glass,L., S.A.Kauffman. 1973. The Logical analysis of continuous, non-linear biochemical control networks. *J.Theor.Biol.* 39:103.

[6] Harris,E.S., B.K.Sawhill, A.Wuensche, and S.Kauffman. 1997. Biased Eukaryotic Gene Regulation Rules Suggest Genome Behaviour is Near Edge of Chaos. *Santa Fe Institute Working Paper* 97-05-039, to appear in *Complexity.*

[7] Hopfield,J.J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of NAS 79*: 2554-2558.

[8] Jeong,H., S.P.Mason, A-L.Basabási. 2001. Lethality and centrality in protein networks. *Nature* 441, 41-42.

[9] Kauffman,S.A. 1969. Metabolic stability and epigenisis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22, 437-467.

[10] Kauffman,S.A. 1993. *The Origins of Order.* Oxford University Press.

[11] Langton,C.G. 1990. Computation at the Edge of Chaos. *Physica D 42*, 12-37.

[12] Relaix, F., and M.Buckingham. 1999. From insect eye to vertebrate muscle: redeployment of a regulatory network. *Genes & Dev* 13, 3171-78.

[13] Solé,R.V, I.Salazar-Cuidad, S.A.Newman. 2000. Gene network dynamics and the evolution of development. *Trends Ecol. Evol.* 15, 479-480.

[14] Somogyi,R., and C.Sniegoski. 1996. Modeling the Complexity of Genetic Networks. *Complexity*, Vol.1/No.6, 45-63.

[15] Somogyi,R, S.Fuhrman, M.Askenazi, A.Wuensche. 1997. The Gene Expression Matrix. In *Proceedings of the World Congress of Non-Linear Analysis*, (WCNA96), 30(3):1815-1824.

[16] Vogelstein,B., D.Lane, A.J.Levine. 2000. Surfing the p53 network, *Nature* 408, 304-310.

[17] Wolpert, L. 1998. *Principles of Development*, Oxford University Press.

[18] Walker,C.C., and W.R.Ashby. 1966. On the temporal characteristics of behavior in certain complex systems. *Kybernetik 3*, 100-108.

[19] Wolfram,S. 1984. Universality and complexity in cellular automata, *Physica 10D*, 1-35.

[20] Wolfram,S., ed. 1986. *Theory and Application of Cellular Automata*, World Scientific.

[21] Wuensche,A., and M.J.Lesser. 1992. *The Global Dynamics of Cellular Automata*, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley.

[22] Wuensche,A. 1994. The Ghost in the Machine. In *Artificial Life III*, ed C.G.Langton, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley.

[23] Wuensche,A. 2001. *The DDLab Manual* and *Discrete Dynamics Lab* software.
Online at `www.santafe.edu/~wuensch/ddlab.html` and `www.ddlab.com`

[24] Wuensche,A. 1996. The Emergence of Memory. In *Towards a Science of Consciousness*, eds. S.R.Hameroff, A.W.Kaszniak, A.C.Scott, MIT Press.

[25] Wuensche,A. 1997. Attractor Basins of Discrete Networks, *CSRP* 461, Univ. of Sussex (D.Phil thesis).

[26] Wuensche,A. 1998. Genomic regulation modeled as a network with basins of attraction. In *Pacific Symposium on Biocomputing'98*, World Scientific.

[27] Wuensche,A. 1999. Classifying Cellular Automata Automatically, *Complexity* Vol.4/no.3, 47-66.