GENOMIC REGULATION MODELED AS A NETWORK WITH BASINS OF ATTRACTION

A.WUENSCHE
Santa Fe Institute,
1399 Hyde Park Road, Santa Fe, New Mexico 87501, USA
wuensch@santafe.edu

Many natural processes consist of networks of interacting elements which affect each other's state over time, the dynamics depending on the pattern of connections and the updating rules for each element. Genomic regulatory networks are arguably networks of this sort. An attempt to understand genomic networks would benefit from the context of a general theory of discrete dynamical networks which is currently emerging. A key notion here is global dynamics, whereby state-space is organized into basins of attraction, objects that have only recently become accessible by computer simulation of idealized models[12,13], in particular "random Boolean networks". Cell types have been explained as attractors in genomic networks[5], where the network architecture is biased to achieve a balance between stability and adaptability in response to perturbation[3]. Based on computer simulations using the software Discrete Dynamics Lab (DDLab)[15], these ideas are described, as well as order-chaos measures on typical trajectories that further characterize network dynamics.

1 Introduction

Processes consisting of concurrent networks of interacting elements which affect each other's state over time occur in a wide variety of natural systems, the dynamics depending both on the pattern of connections (wiring) and on the update rules for each element. The behavior of such complex feedback webs is difficult to treat analytically by classical mathematics using its tools of partial differential equations and continuous dynamics. Instead, a theoretical understanding of these networks depends on numerical simulations of idealized computer models known as discrete dynamical networks.

An example are cellular automata, a powerful yet simple class of network, characterized by a universal rule and uniform nearest neighbor wiring, which are used to study processes in physical systems such as reaction-diffusion[11], and self-organization by the emergence of coherent interacting structures[14]. By contrast, biological systems such as neural, immune or genomic networks require models where wiring and rules are unconstrained. These more general "random Boolean networks" (RBN) were proposed as models of genomic regulatory networks by Kauffman[5,6] and have roots in Ashby's work on dynamical systems and cybernetics[1].

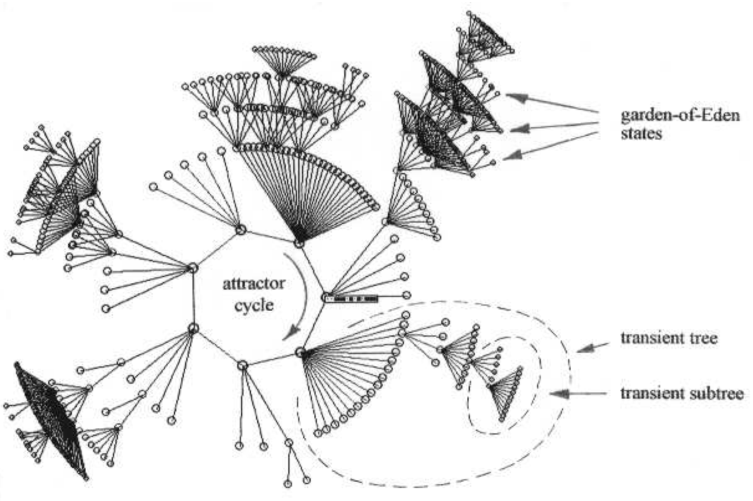


Figure 1: A basin of attraction (one of 15) of a random Boolean network (n=13, k=3) defined in figure 4. The basin links 604 states, of which 523 are garden-of-Eden states. The attractor has period 7, and one of the attractor states is shown as a bit pattern. The direction of time is inwards from garden-of-Eden states to the attractor, then clock-wise.

A key notion underlying the behavior of discrete dynamical networks is that they organize their state-space into a set of basins of attraction, object which sum up the network's global dynamics. The terminology is borrowed from classical continuous dynamical systems. Basins of attraction can be investigated by running networks forward from many initial states to discover the range of attractors present. More recently, exact representations of basins of attraction have become accessible, where new algorithms allow the network to be run "backwards" to disclose all possible historical paths[12,13,16].

2 Genomic regulatory networks

The cells of living 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 an identical set of genes, some of which are expressed others not. A cell type is defined by the particular subset of genes that are expressed. The gene expression pattern of a cell needs to be stable but also adaptable. What then is the mechanism that maintains the several hundred alternative stable patterns of gene expression of the various cell types making an organism?

Genes regulate each other's activity by coding for transcription factors, which may enhance or repress the expression of other genes by binding (possibly in combination) at particular sites. Though a particular gene directly regulates just a small set of other genes, those genes regulate other genes in turn, so a gene will indirectly influence the activity of many genes downstream. Conversely, a particular gene is indirectly influenced by many genes upstream. A gene may directly or indirectly contribute to regulating itself. The result is a genomic regulatory network, a complex feedback web of genes turning each other on and off. This may be interpreted as an idealized dynamical system of model genes with directional links or "wires" (transcription factors), updating their (on-off) state in parallel, according to the combinatorial logic of their inputs[8], Kauffman's random Boolean networks[5,6].

There is justified debate as to whether parallel (synchronous) updating, and the on-off characterization of genes, are valid idealizations when applied to real genomic 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. The RBN idealization is arguably a valid starting point for gaining insights into gene network dynamics.

When genomic regulatory networks are seen from a discrete dynamical network perspective, the problem of understanding how the range of stable cell types can exist with identical genes becomes more understandable. Cell types can be defined as the separate attractors or basins of attraction into which network dynamics settles from various initial states. Trajectories leading to attractors can be seen as the pathways of differentiation.

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. The definition of a cell type may be more correctly expressed as a set of closely related gene expression patterns, allowing an essential measure of flexibility in behavior. Too much flexibility might allow a perturbation to flip the dynamics into a different basin of attraction, a different cell type such as a cancer cell, or a bone cell to a fat

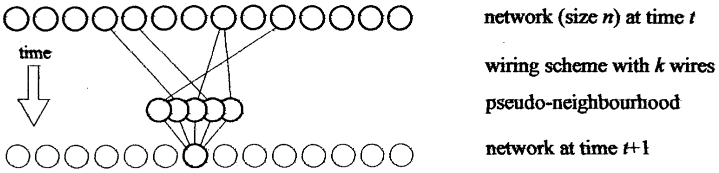


Figure 2: RBN architecture. Each element in the network synchronously updates its value according to the values in a pseudo-neighborhood, set by single wire couplings to arbitrarily located network elements at the previous time-step. Each network element may have a different wiring, rule and number of input wires k. The system is iterated.

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. This can be visualized as a balance between order and chaos with its particular notion of a phase transition.

3 Random Boolean Network Architecture

Each element (or model gene) in a RBN may have a different wiring and/or rule scheme (but not necessarily), and also a different number, k, of input "wires" from other network elements. Most work on RBN has assumed networks with uniform k, but recent investigations of genomic regulatory networks require setting up networks with particular proportions of different k inputs to match data from cell biology[3]. Mixed k networks are also required to check solutions to the inverse problem, where the network architecture is deduced from some degree of knowledge of the basin of attraction field[9].

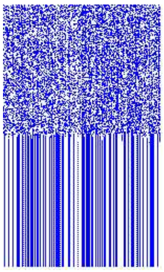
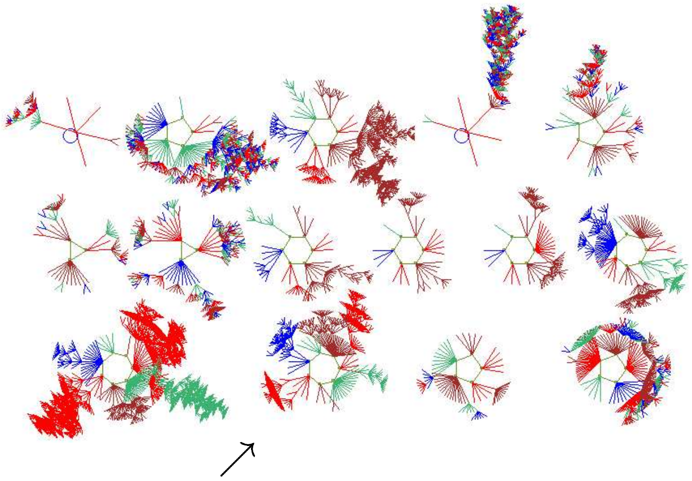


Figure 3: RBN 1d space-time pattern, n=150, k=5. Time proceeds from the top down. Wiring and rules were initially set at random resulting in chaotic dynamics. Middway in the space-time pattern the rules were adjusted to set canalizing inputs at 52% resulting in the rapid emergence of frozen elements, a sign of order.

RBN imply a value range of 2, where each element has a value of either 0 or 1 (an inactive or active gene), but the attractor basin approach would apply equally to networks with more than two values. The state of a network of



the basin in figure 1

the network

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

Figure 4: The basin of attraction field of a random Boolean network (n=13, k=3). The $2^{13} = 8192$ states in state space are organized into 15 basins, with attractor periods ranging between 1 and 7. The number of states in each basin is: 68, 984, 784, 1300, 264, 76, 316, 120, 64, 120, 256, 2724, 604, 84, 428. Figure 1 shows the arrowed basin in more detail. Right: the network's wiring/rule scheme.

n elements is the pattern resulting from the values assigned to each element, from a finite range of values v (usually v=2). Each element synchronously updates its value in discrete time steps. The value of each element C_i at time $t+1$ depends on its particular Boolean function f_i, applied to a notional or pseudo-neighborhood, size k, whose values depend on the wiring, as illustrated in figure 2. The resulting sequence of states, the network's trajectory, can be shown as a space-time pattern in one (or two) dimensions as in figure 3.

A pseudo-neighborhood of size k has 2^k permutations of values. The most general expression of the Boolean function or rule is a lookup 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 AND/OR/NOT logical statements, totalistic rules[11] or threshold functions. The number of effectively different rules is reduced by symmetries in the rule table[12]. By convention the rule table is arranged in descending order of the values of neighborhoods, and the resulting bit string converts to the decimal or hexadecimal rule number.

An example of a k=3 rule table for rule 30 (hex 1e),

```
 7   6   5   4   3   2   1   0  - neighborhoods, decimal
111 110 101 100 011 010 001 000 - neighborhoods, binary
 0   0   0   1   1   1   1   0  - outputs, the rule table
```

The system's parameters are set by specifying the wiring and rule for each of the n network elements. Once set, the networks wiring/rule scheme is usually fixed over time.

The i-th cell C_i has wiring connections $w_{i,k-1}, w_{i,k-2}, \ldots, w_{i,0}$. Connections are assigned to any of the n elements in the network, including C_i itself. Duplicate connections are allowed, giving n^k possible alternative wiring options. The rule for element i is f_i, assigned from the 2^{2^k} alternatives in rule-space. The time evolution of the i-th element is given by, $C_i^{t+1} = f_i(C^t_{w_{i,k-1}}, C^t_{w_{i,k-2}}, \ldots, C^t_{w_{i,0}})$. The number of all possible alternative wiring/rule schemes that can be assigned to a given RBN of size n and homogeneous connectivity k is given by[13], $S_{n,k} = (n^k)^n \times (2^{2^k})^n$. This may be compared with the number of possible basin of attraction fields[16], F^n, for a network of size n, which equals the number of alternative mappings, $Q \to Q$, for a set Q of size n, so $F^n = (2^n)^{(2^n)}$.

As $S_{n,k} > F_n$ for a fully wired network where $k = n$, which can achieve any mapping, there must be considerable redundancy in wiring/rule schemes so that many equivalent schemes generate identical basin of attraction fields. This is indeed the case. For example, if the pseudo-neighborhood of a network element is re-ordered, there is an appropriately re-ordered rule table that gives identical behavior, thus $k!$ equivalent wiring-rule schemes. Myers[7] has derived expressions for the number of effectively distinct maps, thus basin of attraction fields for RBN.

The software DDLab[15] allows a network size n (arranged in one or two dimensions) to be set up with any arrangement of wiring, rules, and k values. Wiring and rule parameters can be restricted or biased in various ways. For example wiring can be confined within a local neighborhood, and the rules can be tuned to a given fraction of canalising inputs[3,16] (see section 10).

4 Basins of Attraction

The idea of attractor basins in discrete dynamical networks is summarized in figure 5. Given an invariant network architecture and the absence of noise, a discrete dynamical network is deterministic, and follows a unique trajectory (a sequence of network states) from any initial state. This non-random walk through a finite state-space must reach an attractor. When a state that has occurred previously is encountered, the dynamics become trapped in a perpetual cycle of repetitions which defines the attractor (state cycle) and its period (minimum one, a stable point), whereas transients are sequences outside the attractor but leading to it.

These systems are dissipative. A state may have multiple predecessors (including none) but just one successor resulting in a convergent tree-like topology of the flow in state-space. The set of transient trees rooted on each attractor state defines a basin of attraction with a topology of trees rooted on cycles (figure 1). Network architecture thus organizes state-space into a set, or field, of basins of attraction, representing the systems global dynamics (figure 4).

5 Computing Pre-images

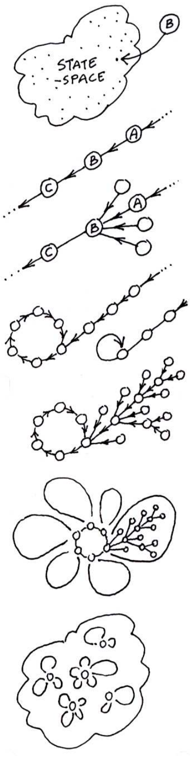
The capability of constructing attractor basins depends on reverse algorithms for directly computing the predecessors (also known as pre-images) of network states[12,13,16]. This allows the network's dynamics, in effect, to be run backwards in time. Backward trajectories will, as a rule, diverge. These algorithms open up a window on the precise structure of the basins of attraction of discrete dynamical networks.

Different reverse algorithms apply to networks with different qualities of connectivity. The most computationally efficient algorithm applies to 1d networks with cellular automata type connections, although the rules for each element may differ. An alternative algorithm is required for RBN with their non-local connections and possibly mixed k. This algorithm also applies to cellular automata, which are just a sub-class of RBN.

These methods are in general orders of magnitude faster than the brute force method, constructing an exhaustive map resulting from network dynamics. The exhaustive method rapidly becomes computationally intractable with increasing network size so is limited to very small systems, but applies to all network types and also allows the attractor basins of random maps to be constructed[16]. The three independent methods together form a valuable reality check on the correctness of the computed pre-images and attractor basins. These methods are applied in DDLab to compute and portray attractor basins. Previous investigations of attractor basins relied only on statistical methods. These are also implemented in DDLab and are appropriate for large networks (see section 10).

6 Constructing Attractor Basins

To construct a basin of attraction containing a particular state or "seed", the network is iterated forward from the seed state until a repeat is encountered in the trajectory. This identifies the attractor cycle as the sequence of states from the state that was repeated.



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 predecessor (*pre-image*) of B, according to the dynamics of the network.

The state B may have other pre-images besides A, and the pre-images 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 occured previously - it has entered an *attractor cycle.* The trajectory leading to the attractor is known as a *transient.* The *period* of the attractor is the number of states in its cycle, which may be only one - this is known as 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 at all, 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 of the network. Each discrete dynamical network imposes a particular basin of attraction field on state-space.

Figure 5: State space and attractor basins.

Once the attractor cycle is known (and drawn as a circle of nodes), the transient tree (if it exists) rooted on each attractor state is constructed in turn. Using one of the reverse algorithms, the pre-images of the attractor state are computed, but the pre-image lying on the attractor itself is deliberately excluded to prevent endlessly tracing pre-images backwards around the attractor cycle. The reverse algorithm is then re-applied repeatedly, to the pre-images of pre-images, until all the "leaves" of the transient tree have been reached. These are states without pre-images, known in automaton terminology as "garden-of-Eden" states. In this way the transient tree is completely specified.

In a similar way, just a subtree or a fragment of a subtree may be constructed rooted on a state. However, because a state chosen at random is very likely to be a garden-of-Eden state, to construct a subtree it is usually necessary to run the network forward by at least one step from the initial random state and use the state reached as the subtree root. Running forward by more steps will reach a state deeper in the subtree so allow a larger subtree to be constructed.

7 Portraying Attractor Basins

Attractor basins are portrayed as directed graphs as presented in[12]. Global states are represented by nodes, by a bit pattern in 1d or 2d, or as the decimal or hex value of the state. The nodes are linked by directed arcs. Each node will have zero or more incoming arcs from its pre-image nodes (in-degree), but because the system is deterministic, exactly one outgoing arc (one out-degree).

In the graphic convention for drawing attractor basins[12], the length of transition arcs decreases with distance away from the attractor, and the diameter of the attractor cycle approaches an upper limit with increasing period. The forward direction of transitions is inward from garden-of-Eden states to the attractor, which is the only closed loop in the basin, and then clockwise around the attractor cycle (figure 1 shows an example). Typically, the vast majority of states in a basin of attraction lie on transient trees outside the attractor, and the vast majority of these states are garden-of-Eden states.

A given network defines its basin of attraction field, from which characteristic measures may be taken, such as the number of attractors, attractor periods, size of basins, characteristic length of transients and characteristic branching within trees (the density of leaves, in-degree frequency). These are also measures of convergence indicating the degree of order-chaos in the dynamics, where high leaf density, high branching, short transients, and small attractor cycles indicate order, and the converse indicates chaos.

8 Perturbations

The precise structure of attractor basins is of interest as it reflects the stability of cell types to perturbation, both to the current state of the network and to the network parameters themselves. A set of similar states can be specified, for example that differ by one bit from a reference state (a Hamming distance of one). The distribution of the set across the basin of attraction field indicates the network's response to a one bit perturbation to its current state of activation. The dynamics might remain in the same basin or flip to a different basin; remain in the same sub-tree or flip to a different sub-tree. Figure 6 gives

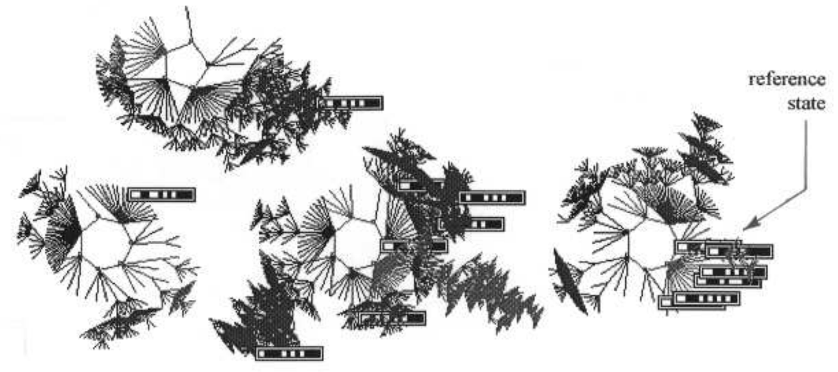


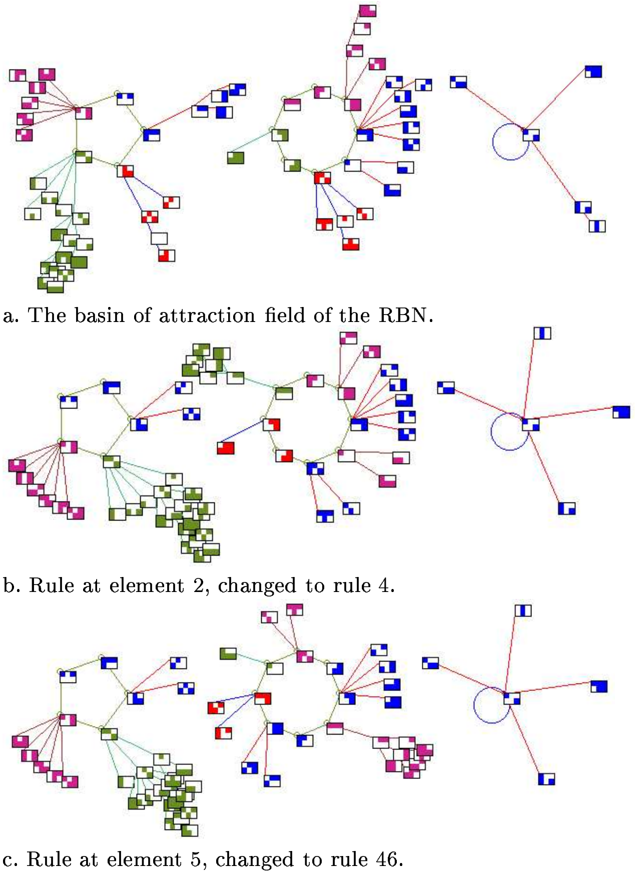
Figure 6: The distribution of a set of similar states differing by one bit from the reference state indicated, are highlighted in basins of attraction. The network is the same as in figures 1 and 2. Only the basins which contain the set of states and the reference state are shown, i.e. basins 2, 11, 12 and 13 in figure 2. Following a one bit perturbation of the reference state, the dynamics may stay in the same basin or flip to one of the other basins with a predictable probability.

an example. An analogous statistical measure is the distribution of "damage spread" resulting from many single bit flips in a large network[16].

The effect of perturbation to network architecture, mutating the wiring and/or rules, can be assessed by the affect on the basin of attraction field, analogous to the role of genotype and phenotype. Figure 7 shows the consequences of mutating a single bit in the rule table of just one element of a small RBN, n=6, k=3. The basin of attraction field is affected to varying degrees. The example networks are small so that the pattern at each node can be shown. Larger networks are affected in analogous ways, although in general the consequences of a one bit mutation become relatively smaller with increasing network size. However a particular one bit mutation may cause drastic consequences whatever the size, such as breaking an attractor cycle. Note that the consequences of moving a connection wire is usually greater than a one bit mutation.

9 Memory

Attractors classify state-space into broad categories, the network's "content addressable" memory in the sense of Hopfield[4]. 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



a. The basin of attraction field of the RBN.

b. Rule at element 2, changed to rule 4.

c. Rule at element 5, changed to rule 46.

the network

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 7: The consequences of mutating a single bit in the rule table of just one element of a small RBN, n=6, k=3. The state at each node is shown as a 3×2 bit pattern. Above: Network architecture.

of attractors greatly extends the classical concept of memory by attractors alone[13]. Although such a discussion would seem only to relate to the area of neural networks, it also applies to genomic regulatory networks in the sense that they must "remember" and return to their cell types (and sub-types) as they are continually perturbed by internal and external factors.

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, the topology of attractor basins, *the network's memory*, 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.

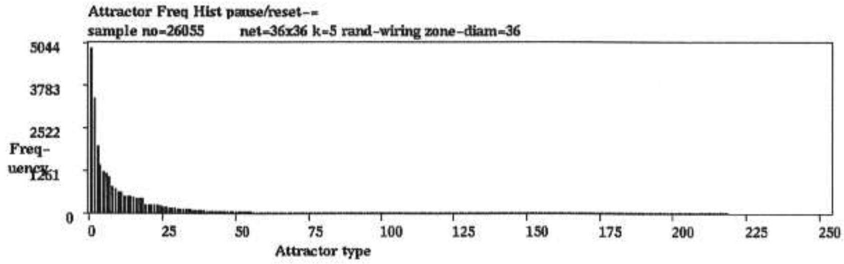


Figure 8: A histogram of attractor types against the frequency of each type sorted by frequency for a 2d RBN 36×36, k=5, with fully random wiring and a fraction of canalising inputs C=52%. The data was assembled by running the network forward from a sample of about 26,000 random initial states and recording the fraction of different attractors reached, about 220 types in this sample. The frequency of arriving at each basin of attraction indicates its relative size, the number of states it contains.

10 Attractor basins of large networks

Time and memory constraints limit the size of networks when computing basin of attraction fields, with a practical upper limit of about n=26, though DDLab allows $n \leq 31$ given sufficient memory. Single basins of attraction and subtrees can be computed for much larger networks. However, data on the basin of attraction field can be assembled for large networks by statistics on forward dynamics where the demands on memory are much smaller.

The number and relative size of the different basins in the basin of attraction field can be inferred from a histogram of the frequency of reaching different attractor types from many random initial states. Other data such as attractor length and average run-in length are also available. In the example in figure 8 the rules scheme is randomly biased to give a fraction of canalising[3] inputs of 52%. Canalization occurs when a particular input (0 or 1) on a connection determines a gene's value irrespective of its other inputs. That connection is then said to be canalising. This tunes the network to achieve a balance between order and chaos according to various dynamical measures (see below).

11 Order-chaos measures of large networks

The wiring/rule architecture of RBN can be biased to produce the percolation of "frozen" genes, that have become fixed on or off over time[6]. These biases have been correlated with data on biological genomic networks[3] and are understood to confer stability to the cell type, while the remaining unstable genes provide adaptability. The dynamics is seen as being marginally on the ordered side of an order-chaos phase transition in the space of all possible network architectures according to a number of measures. In particular, increasing the degree of canalisation of network rules moves behavior across the transition, though the characteristics of network wiring also plays a role. Order-chaos measures[16] listed below may be studied with DDLab, based on statistics on large networks as well as exact data on small networks.

- Measures on the topology of attractor basins and subtrees such as their in-degree distribution.
- The Derrida plot[2], analogous to the Liaponov exponent in continuous dynamics, measures the divergence/convergence of trajectories from pairs of initial states separated by a varying Hamming distance.
- The distribution of "damage spread", where the difference between trajectories from pairs of initial states differing by one bit is monitored, and the distribution of this damage size is measured.
- The percolation of frozen regions, where the fraction of genes in a network that have stabilized on or off, or that continue to flicker is identified. This allows the distribution of "frozen skeletons" to be measured, characteristic frozen patterns that are treated as quasi-attractors.
- The distribution of attractor types as shown in figure 8.
- Input-entropy over time relating to single genes, which measures the frequency with which neighborhoods in the gene's rule-table are activated over a moving window of time-steps, and the entropy calculated.

12 Conclusion

Attractor basins of discrete dynamical networks provide key insights into a network's global dynamics. Idealizations of genomic regulatory networks are modeled as random Boolean networks, where the range of biases in network architecture can be tested against theories and data from cell biology. A broad range of characteristics, parameters and measures of global dynamics, as well as dynamics along particular trajectories, are now open to investigation, including the number of attractors (cell types), their size distribution and topology, stability and adaptability to perturbations of both patterns of activation and network architecture, issues of the network's memory and the inverse problem[9], and order-chaos measures on network dynamics. The software Discrete Dynamics Lab[15] is a tool for these investigations. DDLab continues to be developed with new capabilities in response to current research ideas.

References

1. Ashby,W.R., (1956) “An Introducion to Cybernetics”, Chapman & Hall.
2. Derrida,B., and D.Stauffer, (1986) “Phase transitions in Two-Dimensional Kauffman Random Network Automata”, Europhys.Lett. 2, 739.
3. 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.
4. Hopfield,J.J. (1982) “Neural networks and physical systems with emergent collective computational abilities”, Proceedings of the National Academy of Sciences 79:2554-2558.
5. Kauffman,S.A., (1969) “Metabolic stability and epigenisis in randomly constructed genetic nets”, Journal of Theoretical Biology, 22, 437-467.
6. Kauffman,S.A., (1993) “The Origins of Order”, Oxford University Press.
7. Myers,J.E., (1997) “Random Boolean Networks - Three Recent Results”, to be appear in Complexity.
8. Somogyi,R., and C.Sniegoski, (1996) “Modeling the Complexity of Genetic Networks: Understanding Multigenetic and Pleiotropic Regulation”, Complexity, Vol.1/No.6,45-63.
9. Somogyi,R, S.Fuhrman, M.Askenazi, A.Wuensche., (1997) “The Gene Expression Matrix”, in Proceedings of the World Congress of Non-Linear Analysis, in press.
10. Walker,C.C., and W.R.Ashby, (1966) “On the temporal characteristics of behavior in certain complex systems”, Kybernetik 3, 100-108.
11. Wolfram,S., ed. (1986) “Theory and Application of Cellular Automata”, World Scientific.
12. Wuensche,A., and M.J.Lesser. (1992) “The Global Dynamics of Cellular Automata”, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley.
13. 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.
14. Wuensche,A., (1994) “Complexity in One-D Cellular Automata”, Santa Fe Institute Working Paper 94-04-025.
15. Wuensche,A., (1996) “Discrete Dynamics Lab (DDLab)”, http://www.santafe.edu/ wuensch/ddlab.html
16. Wuensche,A., (1997) “Attractor Basins of Discrete Networks”, CSRP 461, Univ. of Sussex (D.Phil thesis, 1996).