Distributed robustness in cellular networks: insights from synthetic evolved circuits

Javier Macia[1] **and Ricard V. Solé**[1,2†]

(1) ICREA-Complex Systems Lab, Universitat Pompeu Fabra, Parc de Recerca Biomedica de Barcelona. Dr Aiguader 80, 08003 Barcelona, Spain
(2) Santa Fe Institute, 1399 Hyde Park Road, 87501 Santa Fe, New Mexico, USA

Evolved natural systems are known to display some sort of distributed robustness against the loss of individual components. Such type of robustness is not just the result of redundancy. Instead, it seems to be based on degeneracy, i. e. the ability of elements that are structurally different to perform the same function or yield the same output. Here we explore the problem of how relevant is degeneracy in evolved digital systems, and what types of network structures underlie the resilience of evolved designs to the removal or loss of a given unit. It is shown that fault tolerant circuits are obtained only if robustness arises in a distributed manner. No such reliable systems are reached just by means of just redundant organization, thus suggesting that reliable designs are necessarily tied to degeneracy.

Keywords: evolvable hardware, redundancy, degeneracy, robustness, fault tolerance

1. INTRODUCTION

One remarkable feature of many biological systems is the presence of a high degree of robustness against perturbations. Such robustness appears at multiple scales (Alon et al., 1999; Gibson 2002, Krakauer and Plotkin, 2002; Li et al., 2004; Jen, 2005; Wagner, 2006). Specifically, it is often found that temporal failure or permanent loss of some components has very often little or no impact on overall performance. In this context, such entities as a whole are able to cope with a changing world even under the loss of single units. A standard illustration of such robustness (or fault tolerance) is provided by gene knockouts through directed homologous recombination. In a large number of cases (close to 30%) little or no phenotypic effects are observed (Melton 1994). What is more surprising, it was shown that the mechanisms underlying such reliable behaviour are not based on redundancy. By redundancy, we refer to the presence of multiple copies of a given component: the failure of one of them would be compensated by another identical (isomorphic) copy. Instead, robustness in biology is largely associated to a distributed property, which has been dubbed either *degeneracy* (Edelman and Gally, 2001; Tononi et al., 1999) or *distributed robustness* (Wagner 2006). By degeneracy we refer to "the ability of elements that are structurally different to perform the same function" (Edelman and Gally, 2001; Tononi et al., 1999).

The problem of robustness was a hot topic in the 1950s, in parallel with the design of the first electronic computers. Back there, vacuum tube technology was rather unreliable and John von Neumann and others (von Neumann, 1952, Cowan and Vinograd 1963) explored the problem of how to design reliable computers from unreliable elements. Most of this work was based on digital designs described in terms of logic gates (figure 1a). The main conclusion from these studies was that a high degree of redundancy was required in order

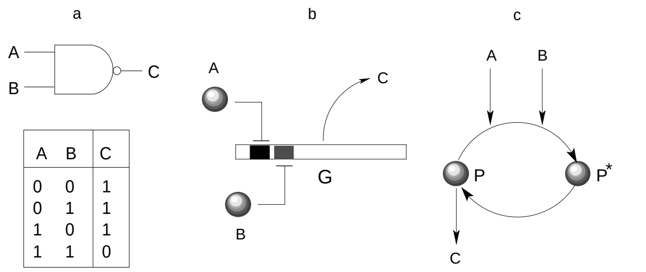

Figure 1. A logic gate, such as the NAND gate shown here, can be easily implemented using a molecular system. In (a) the standard symbol for the NAND gate (up) and its Boolean table representation (bottom) are shown. In (b) an example of the molecular implementation based on two proteins (A and B, which can be present or absent) activating two given genes whose proteins combine to form a dimer C. The presence (0) or absence (1) of such dimer molecule C defines the output of the gate. Moreover, signalling cascades also allow defining logic blocks. In (c) we give an example of such scenario for our NAND system. Here a protein P can be activated is two inputs A and B (external signals) are present. In that case it makes a transition $P \rightarrow P^*$ to an active form. If $C = [P]$ is the measured output, then a NAND gate is obtained.

†Author for correspondence (ricard.sole@upf.edu).

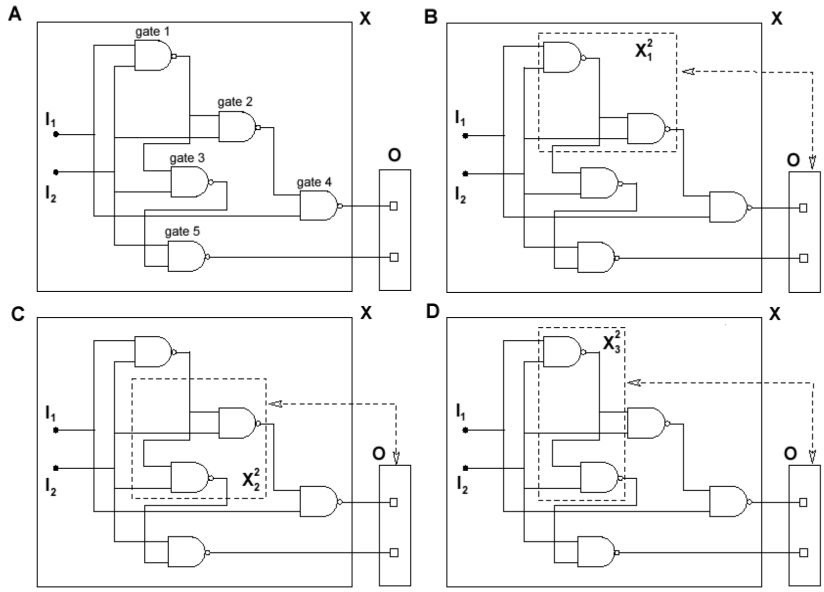

Figure 2. An example of a small digital circuit. Gates 1, 2 and 3 are the elements of the set X. Gates 4 and 5 define the output layer O. Figures B, C and D show three possible subsets of size two that can be constructed (see text).

to achieve such goal. Since nature seems to deal with faulty behavior by using non-redundant mechanisms, something different must be at stake. Exploring the problem of degeneracy involves a number of difficulties. Since redundancy does not explain robustness (at least not most of it) we cannot understand the problem in terms of repeated, dissociated pieces. In this paper we want to address this problem using evolved synthetic circuits performing simple computations.

Digital and switching circuits have been widely used in modeling gene and signalling networks (Kauffman 1969, 1993; Mendoza and Alvarez-Buylla, 1998; Mendoza et al., 1999; Wuensche and Lesser, 2000; Astor and Adami, 2000; Solé and Fernandez, 2003; Kauffman et al., 2003; Weiss et al., 2003; Simpson et al., 2004; Sauro and Khodolenko, 2004; Alvarez-Buylla et al., 2006; Willadsen and Wiles, 2007; Braunewell and Bornholdt, 2007) as well as other more general problems concerning the evolution of technology (Arthur and Polak, 2006). Evolved circuits provide a good framework where our questions can be explored in a sensible way (Koza, 1992; Miller et al. 2000). These systems, resulting from artificial evolution, allow to obtain new designs without direct human intervention, in many cases displaying a higher efficiency (Koza, 1992). Our goal here is the generation of digital circuits by evolution under different external conditions, with the purpose of exploring the emergence of fault tolerance and how it relates to redundancy and degeneracy. Although the digital metaphor has some limitations, it has been widely used in computational systems biology to address very diverse types of questions. It seems to define an appropriate level of description to the swithcing behavior found in many biological systems, from gene regulation to cell signaling (figure b-c). The study of the behavior of these models has shown to provide deep insight on the origins and importance of robustness (Bornholdt and Sneppen, 2000; Klemm and Bornholdt, 2005; Fernandez and Solé, 2007; Ciliberti et al., 2007). In this context, although previous studies have shown that robust structures emerge as a consequence of evolutionary rules, the exact origin of such robustness is often missing. Here, using the formal definitions introduced in (Edelman and Gally, 2001; Tononi et al., 1999) we show that robust designs are achieved by means of distributed robustness.

2. Measuring distributed robustness

A first step before we present our results on evolved networks is to properly define a set of quantitative measures of network redundancy and degeneracy. To this goal, we will make use previous measures from (Edelman and Gally, 2001, Tononi et al., 1999). These information-based measures can be quantified using statistical measures of entropy and mutual information (Ash, 1965; Adami 1998). We build these measures on a network X of Z interacting units where some computation is being performed. This is the case of logic circuits, such as the one shown in figure 2a. Here each unit is a NAND gate. The reason of choosing this particular gate is that it allows to buld *any* possible digital circuit. Several measures can be defined on X based on information theory. An appropriate combination of them allows to properly define robustness and measure it. The basic measure of information theory is the entropy, defined as (Ash, 1965; Adami, 1998):

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \tag{1}$$

for one single variable x, and similarly we have

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log p(x,y) \tag{2}$$

for two variables x and y. Here $p(x)$ is the probability distribution for the possible values of x and $p(x, y)$ is the joint probability distribution associated to (x, y)-pairs. From these basic measures, it is possible define the mutual information $I(X,Y)$, in terms of the entropies as:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \quad (3)$$

In their analysis of robustness, Tononi the al. (Tononi, et al. 1999) define the degeneracy D_Z of the system as:

$$D_Z(X) = \frac{1}{2}\sum_{k=1}^{Z}\left\langle I(X_i^k, O) + I(\hat{X}_i^k, O) - I(X,O)\right\rangle \quad (4)$$

where X_i^k represents the i-th subset of k elements which is possible to build from the Z elements of the network. The average <> is computed over all the possible subgroups X_i^k with size k in which we can divide the system. This balance of mutual information takes into account the possible overlapping of the information being processed by a subset $X_i^k \subset X$ formed by k elements and the rest of the network i. e. $\hat{X}_i^k = X - X_i^k$. In the case of two independent subgroups the balance is simply $I(X_i^k, O) + I(\hat{X}_i^k, O) - I(X,O) = 0$. This case corresponds to the lower bound of degeneracy. In any other case it measures the overlap between subgroups. The upper bound corresponds to the extreme case of full degeneracy. In this case the mutual information between one subgroup of the network and the output layer must be similar to the mutual information between the rest of the network and the output layer and similar to the mutual information between the total network and the output layer:

$$I(X_i^k, O) \approx I(\hat{X}_i^k, O) \approx I(X,O) \quad (5)$$

Building a network with such extreme degeneracy is likely to be impossible, but considering this situation it is possible to define an upper bound for degeneracy as:

$$D_Z(X) \leq \frac{Z}{2} I(X,O) = D_z^* \quad (6)$$

which is obtained from (4) using the condition given in (5).

Computing these measures is involved and computationally costly, but they are also well defined. In order to illustrate these computations, in appendix I we show some explicit calculations of these measures using the circuit of figure 2. For this example, the mutual information balance (MIB) for the subset X_1^2 (gates 1 and 2) is $\langle I \rangle = 0.21$. This value means that part of the computation performed by the network from the input to the output involves some overlap between the subset X_1^2 and the rest of the network $X - X_1^2$(gate 3). In other words, two subsets of the network which are structurally different perform, at least partially, the same computation process. To compute the degeneracy values with (4) we need to calculate the average value of the MIB for all the possible subsets for a given size (in the example if the size is 2 there are three possible subsets depicted in figures 1B, 1C and 1D) and perform the sum over all possible sizes, from $k = 1$ to $k = Z$. Such combinatorial scenario implies that, even for small networks we require a very large number of subset combinations to be considered. This limits the total size of circuits that can be used in our analysis, which in our study is limited to $Z = 15$.

Additionally, we can also estimate the redundancy of our system using the following definition (Tononi et al., 1999):

$$R_Z(X) = \left[\sum_{i=1}^{Z} I(X_i, O)\right] - I(X,O) \quad (7)$$

This expression measures the overlap of the information processed by a one basic element and the rest of the network. If the different elements of the network are independent then $R_z(X) = 0$. Otherwise, some of the elements of the network are redundant.

Finally, a complexity measure can be also defined as (Tononi et al., 1999):

$$C_Z = \frac{1}{2}\sum_{i=1}^{Z}\left\langle I(X_i^k, \hat{X}_i^k)\right\rangle \quad (8)$$

This expression measures the level of coherent integration of the different parts of the system (Tononi et al 1998). It takes into account the average mutual information between each possible subset and the rest of the network. If the different possible parts in which the network can be divided have a lower overlap in the information process means that the network has a low integration. In this case the network does not work as a whole, nut as a set of more or less independent parts. This case corresponds to a small values of C_Z.

3. Evolving circuits

Different evolutionary rules have been used for the synthesis of circuits (Higuchi et al., 1997; Yu and Miller, 2001; Banzhaf and Leier, 2006; Arthur and Polak, 2006). Some of these methods have obtained systems with high levels of fault tolerance and robustness. The circuits object of our study are formed by a network of NAND gates, and start from a randomly wired set. These circuits evolve until they are able to implement a certain N-input, M-output target binary function ϕ, being ϕ a member of the set of possible Boolean functions described as a mapping:

$$\phi_i : \Sigma^N \longrightarrow \Sigma^M \quad (9)$$

with $\Sigma = \{0,1\}$. In our study, we will use $N = 3$ and $M = 2$. The only topological limitation considered here is that there are no backward connections, in order to avoid temporal dependencies. In other words, only donwstream effects are considered. The evolutionary rules are follow previous work on circuit evolution (Miller et al., 1999, Miller and Hartmann, 2001) with additional selection constrains in order to canalise the evolution process. Two kinds of evolutionary process have been analyzed, as described below. Below we describe the two basic scenarios of circuit evolution.

3.1. Neutral evolution

The goal of this selection algorithm is just matching the desired target function with no further constrains.

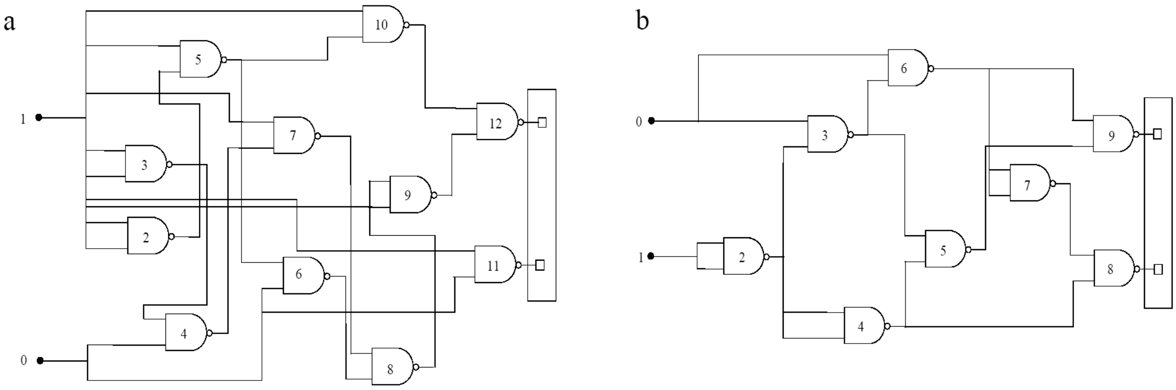

Figure 3. Examples of evolved circuits resulting from a process of evolutionary optimization. In (a) a circuit obtained by conditional evolution is shown. All gates are identical (NAND gates). Nodes 1 and 2 are the input nodes. The circuit outputs are located after nodes 11 and 12. This circuit has a very high fault tolerance value ($\rho = 0.944$). In (b) we show a circuit obtained by neutral evolution. This circuit implements the same logic function than the circuit in (a) but involves a smaller fault tolerance of $\rho = 0.54$.

The idea here is to see if circuits which just correctly perform the desired computation are also robust *for free*. The target functions ϕ are chosen at random: the set of outputs for each input combination are generated using 0 and 1 with equal probability. The steps of the algorithm are:

1. Create a random generated population formed by S individuals (candidate circuits) $X_1, ..., X_S$. Each one of these individuals are formed by Z randomly wired gates, with no backward connections. All circuits start with $Z = 5$ NAND gates. In each one of these nodes there is a NAND gate of two inputs and one output. This population constitutes the starting generation. Here, given the computational constraints, we fixed the number of candidate solutions to $S = 10$.

2. The behaviour of each individual solution is simulated for the 2^N possible inputs, comparing its outputs with the target function outputs. The fitness for the k-th circuit X_k is defined as:

$$F_k = 1 - \frac{1}{M2^N} \sum_{i=1}^{M2^N} |O_i^k - E_i^k| \qquad (10)$$

   where $\{O_i^k\}$ is the set of outputs of the circuit for the different inputs, and $\{E_i^k\}$ is the set of (expected) target function outputs.

3. The individual with greater fitness is chosen to create a new generation formed by the selected individual and $S-1$ random mutations of it. The random mutations (allways respecting th ebackward patterning) can be: (a) Elimination of an existing connection, with probability E_c, (b) Creation of a new connection with probability C_c, (c) Elimination of a node (gate removal) with probability I_n and (d) Creation of a new node (gate addition) with probability C_n Here we use $E_c = 0.8, C_c = 0.8, I_n = 0.3$ and $C_n = 0.6$.

4. Repeat step 2 until a fitness $F = 1$ is reached.

3.2. *Fault tolerant evolution*

This algorithm differs from the previous one in that selection of the most optimal individual of each generation introduces fault tolerance. Fault tolerance ρ is measured as:

$$\rho = 1 - \frac{1}{ZM2^N} \sum_{p=1}^{Z} \left(\sum_{i=1}^{M2^N} |O_i - O_i^p| \right) \qquad (11)$$

where $\{O_i\}$ is the output set of the circuit for the different inputs, and $\{O_i^p\}$ is the output set of the circuit under a perturbation of the p-th node. For each input all the gates are in a binary state 0 or 1. The perturbation consists on the inversion of the logical level of the gate located at the p-th node. This perturbation is applied for each node p.

In this case the external condition on the evolutionary process is prevailing. The selection process, in this case, does not stop when $F = 1$. Once $F = 1$ the individuals can continue evolving until ρ reaches a stable value. In this approach, we want to compare the final designs with those resulting from the neutral evolution approach. Looking at the final circuits generated, we can measure redundancy and degeneracy and see how they contribute to the network robustness.

4. Results

In figure 3 we show two examples of the synthetic circuits obtained from the two previous algorithms. Figure 3.a shows a circuit obtained by conditional evolution, with a final fault tolerance of $\rho = 0.944$. For the same Boolean function, figure 3.b shows the corresponding outcome of neutral evolution with a much lower fault tolerance of $\rho = 0.54$. As a general trend, we have found that evolved circuits from neutral selection are tipically

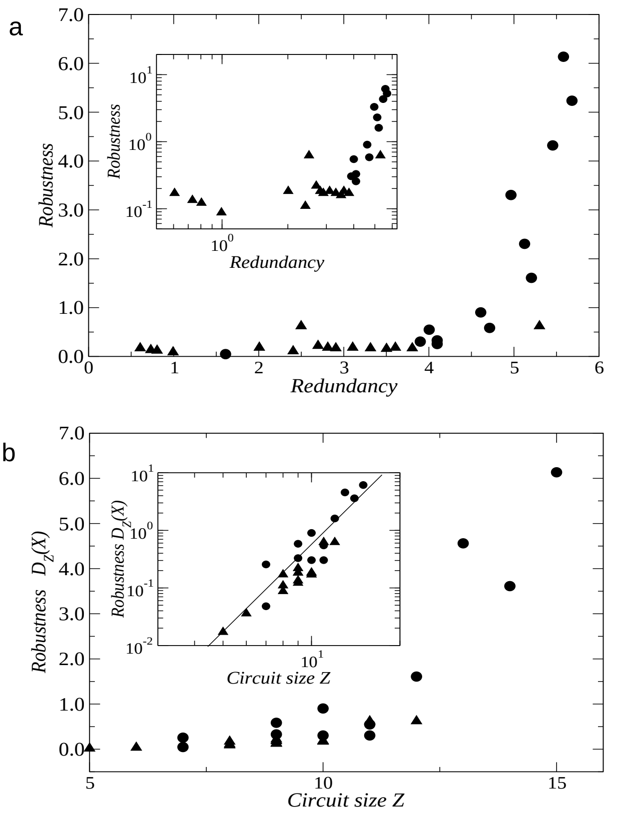

Figure 4. (a) robustness, as measured in terms of degeneracy D_z, increases with redundancy R_z in a nonlinear fashion. Here neutral and fault tolerant cases are indicated by means of triangles and circles, respectively. If neutral evolution is used, low levels of robustness are achieved, whereas conditional evolution leads to high robustness, provided that enough redundancy is at play. The inset shows the same results in log-log scale. In (b) we show the correlation between system size N and fault tolerance. Here we can clearly see that high levels of reliability are achieved by increasing system's size. The inset shows the same plot in log-log scale, where we can appreciate a scaling behavior $\rho \sim Z^{\gamma}$ with $\gamma \approx 3$.

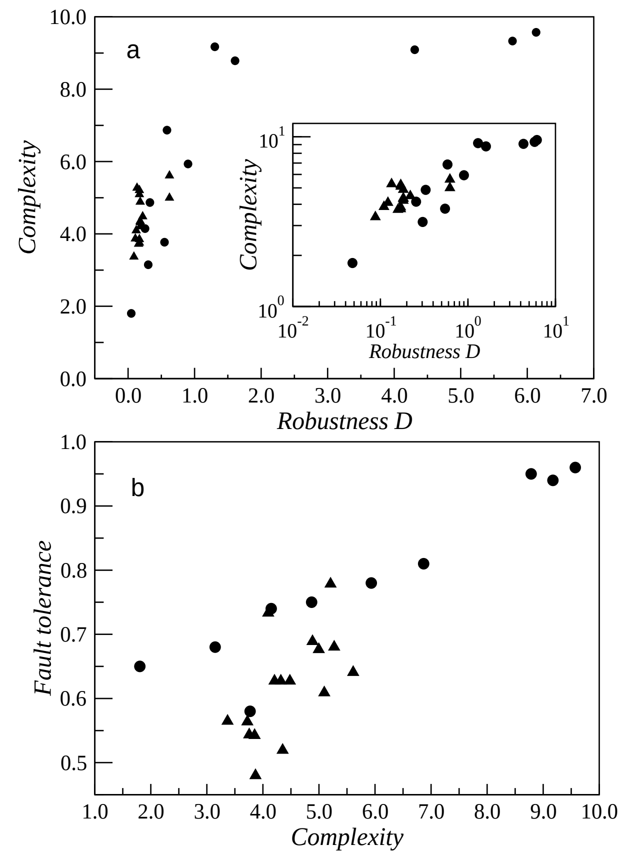

Figure 5. Correlations between circuit complexity and robustness (a) indicate that system's integration rapidly increases with D_z, reaching a plateau. On the other hand, fault tolerance seems well correlated with complexity, as shown in (b) thus indicating that an appropriate internat integration allows the system to be more reliable under failure of single units.

smaller than the corresponding circuits selected for conditional evolution. This is an expected result, since it seems clear that robustness against gate failure must require some kind of internal capacity of reorganization. In order to see what emerges in terms of robustness, we measured redundancy and degeneracy in a set of evolved circuits under both types of selection pressures.

In figure 4a we show the relationship between redundancy and degeneracy (our measure of robustness) for our evolved circuits under neutral and conditional evolution, respectively. The picture clearly shows that circuits evolved under neutral evolution (and thus properly computing the Boolean function ϕ) have diverse levels of redundancy but very small degeneracy (here indicated as robustness). Two relevant implications of this result can be obtained. The first is that an active selection for robustness is required in order to achieve reliable designs. The second is that circuits obtained under selection for fault tolerance can achieve large levels of robustness and that such robustness is distributed, but requires some amount of internal redundancy (as shown by the rapid increase of D_Z at high R_z values. Let us note that the circuit with greater fault tolerance have a degeneracy value of $D = 6.28$, closer to the upper bound value of $D^*_Z = 7.79$ obtained from (6).

These results would support the view that biological designs, which are expected to experience different sources of noise and perturbation, make use of degeneracy -instead of redundancy- in order to properly function. But it also suggests that redundancy must be at work in order to build the appropriate levels of robustness. Moreover, as shown in figure 4b, there is a growing trend relating fault tolerance and the system size achieved through the evolutionary dynamics. We have found that this trend is highly nonlinear, with a rapid increase in reliability as Z grows for conditional selection. This implies that high increases in robustness can be achieved by properly adding a single new element to the system.

In figures 5a-b we also compare our measures of robustness and fault tolerance with the internal organization of the circuits as measured in terms of complexity. In our circuits, we see that large levels of complexity and circuit integration closely follow high degeneracy levels. This is consistent with previous findings using neural networks (Tononi et al., 1998). Such high complexity levels indicate that circuits evolve towards structures in which the different parts act with greater levels of coherent integration.

As figure 5.a shows, an increase in the degeneracy implies an increase in the complexity of the circuits, although a saturation is observed beyond some point (probably due to the small circuit sizes, which does not allow further increases). Such relation does not happen with the redundancy (not shown): high redundancy levels are not consistent with more complex circuits. Similarly, complexity and fault tolerance are related, although the tendency towards higher fault tolerance with network integration is more obvious in circuits obtained through conditional evolution (Figure 5.b).

5. Discussion

The evolution of complex life forms seems to be inextricably tied to robustness. The problem of how complexity, modularity, adaptation and reliability are connected is an old one (Conrad, 1983) but far from being closed (Hogeweg, 2002; Lenski et al., 2003; Wagner et al., 2007). Capturing the details ofsuch relation is a difficult task, and theoretical approaches require strong simplifications that are typically based on a computational picture of the system under consideration. In this paper we have explored the possible origins of robustness in networks performing computations by using evolved artificial circuits. Although the approach taken is not a realistic biological implementation, it captures some of the logic of cellular networks at least at the level of computation. Nevertheless, our main goal was to determine the possible forms of reaching reliable systems under different selection pressures and understanding how robust designs can be obtained. Given the two potential origins of robust responses, namely either redundancy or degeneracy, we measured the resulting structures in order to determine what are the contributions of each to the observed levels of fault tolerance.

As found in biological systems, we can see that the origins of robustness against the failure of a given element is largely associated to a distributed mechanism of network organization. Both degeneracy and complexity (a measure of network integration and coherence) have been shown to reach high levels provided that redundancy is high enough. In this context, the use of an explicit measure of network reliability based on information exchanges between different subparts allows us to reach well-defined conlusions. Redundancy on the other hand is shown to have little relevance to the fault tolerant behavior of our circuits, but interestingly high redundancy might be needed in order to properly build degeneracy into the system. Understanding the mechanistic basis of these results will need further exploration.

Acknowledgments

We thank the members of the Complex Systems Lab for useful discussions. This work has been supported by the European Union within the 6th Framework Program under contracts FP6-001907 (Dynamically Evolving Large-scale Information Systems), FP6-002035 (Programmable Artificial Cell Evolution), the James S. McDonnell Foundation and by the Santa Fe Institute.

6. Appendix: computing redundancy and degeneracy in a case study

To illustrate the estimation of degeneracy as defined above we can use a simple logic network as the one shown in figure 1a. It is formed by five NAND gates.

Input	G_1	G_2	G_3	G_4	G_5	Output
00	1	1	1	1	1	11
01	1	0	0	1	1	11
10	1	1	1	0	1	01
11	0	1	1	0	0	00

Table 1. Different states for each element (here NAND gates) for the circuit showed in figure 1A

Table I shows all possible state for each gate for each input combination. Gates G_5 and gate G_6 define the output layer. To compute the degeneracy values we must to build all the subsets of NAND gates. It is possible to define three subsets of one gate each, X_i^1 $(i = 1, 2, 3)$, three subsets of two gates X_i^2 $(i = 1, 2, 3)$ as displayed in figures 1(b-d) and one subset of three gates X_1^3. Here we illustrate the calculation method for one of these subsets of size two, i. e. X_1^2. First, we must calculate the different entropy values using the standard definitions (1) and (2). Table II shows the different states for the subset X_1^2, shown in figure 1B.

Input	$X_1^2(G_{1,2})$	$\hat{X}_1^2(G_3)$	$X(G_{1,2,3})$	$O(G_{4,5})$
00	11	1	111	11
01	10	0	100	11
10	11	1	111	01
00	01	1	011	00

Table 2. Different possible states for the subset X_1^2 and the rest of the network $\hat{X}_1 = X - X_1^2$

In Table III we give the probabilities associated to the different possible input combinations and the associated entropy calculated directly from the possible states given in Table II.

Finally, different mutual information values can be computed from the previous measures. The results are summarized in table IV.

REFERENCES

Adami, C. 1998. *Introduction to Artificial Life.* Springer Verlag, New York.

Entropy	Probability distribution		Entropy value
$H(X_1^2)$	$p(11) = 2/4$ $p(10) = 1/4$	$p(01) = 1/4$	1.04
$H(\hat{X}_1^2)$	$p(1) = 3/4$ $p(0) = 1/4$		0.56
H(X)	$p(111) = 2/4$ $p(100) = 1/4$	$p(011) = 1/4$	1.04
H(O)	$p(11) = 2/4$ $p(01) = 1/4$	$p(00) = 1/4$	1.04
$H(X_1^2, O)$	$p(11, 11) = 1/4$ $p(01, 11) = 1/4$	$p(10, 01) = 1/4$ $p(11, 00) = 1/4$	1.39
$H(\hat{X}_1^2, O)$	$p(1, 11) = 1/4$ $p(0, 11) = 1/4$	$p(1, 01) = 1/4$ $p(1, 00) = 1/4$	1.39
H(X,O)	$p(111, 11) = 1/4$ $p(100, 11) = 1/4$	$p(111, 01) = 1/4$ $p(011, 00) = 1/4$	1.39

Table 3. Entropy values for the different elements necessary for mutual information calculations. The probability distributions are directly obtained from Table 2.

Mutual Information (I)	I value
$I(X_1^2, O)$	0.69
$I(\hat{X}1^2, O)$	0.21
$I(X, O)$	0.69
$I(X_1^2, O) + I(\hat{X}_1^2, O) - I(X, O)$	0.21

Table 4. Mutual information calculations from entropy values of table 2

Alon, U., Surette, M. G., Barkai, N. and Leibler, S. 1999. Robustness in Bacterial Chemotaxis, *Nature* **397**, 168-171.

Ash, R.B. 1965. *Information Theory.* Dover. New York.

Banzhaf, W. and Leier, A. 2006. Evolution on Neutral Networks in Genetic Programming. In: *Genetic Programming - Theory and Applications III.* T. Yu, R. Riolo and B. Worzel (Eds.), Kluwer Academic, Boston, MA, pp 207-221.

Bornholdt, S. and Sneppen, K. 2000. Robustness as an Evolutionary Principle, *Proc. R. Soc. London B* **267**, 2281-2286.

Braunewell, S. and Bornholdt, S. 2007. Superstability of the yeast cell-cycle dynamics: Ensuring causality in the presence of biochemical stochasticity, *J. Theor. Biol.* **245**, 638-643.

Ciliberti, S. Martin, OC, and Wagner, A. 2007. Robustness can evolve gradually in complex regulatory gene networks with varying topology. Comp. Biol. **3**, e15.

Conrad, M. 1983. *Adaptability.* Plenum Press, New York.

Cowan, J. D. and Vinograd. 1963. *Reliable computation in the presence of noise.* MIT Press, Cambridge MA.

Edelman, G.M., Gally, J. A. 2001. Degeneracy and complexity in biological systems. *Proc. Natl. Acad. Sci. USA* **98**, 13763-13768.

Fernandez, P. and Solé, R. V. 2007. Neutral fitness landscapes in signaling networks. *J. Roy. Soc. Interface* **4**, 41-47.

Gibson, G. 2002. Developmental evolution: getting robust about robustness. *Curr. Biol.* **12**, R347-349.

Greensted, A. J. and Tyrrell, A. M. 2003. Fault Tolerance via endocrinologic based communication for multiprocessor systems. The 5th International Conference on Evolvable Systems: From Biology to Hardware. Trondheim, Norway.

Higuchi, T., Iwata M. and Liu W. (Eds.). 1997. *Proceedings 1st International Conference on Evolvable Systems: From Biology to Hardware.* Lect. Notes Comp. Sci, **1259**, pp. 327-343. Springer-Verlag, London.

Hogeweg P. 2002. Computing an organism: on the interface between informatic and dynamic processes. *Biosystems*, **64**, 97-109.

Jen, E., editor, 2005. *Robust Design*, Oxford U. Press. New York.

Kauffman, S. A. 1962. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* **22**, 437-467.

Kauffman, S. A. 1993. *The origins of order.* Oxford U. Press, New York.

Kauffman, S. A., Peterson, C., Samuelsson, S. and Troein, C. 2003. Random Boolean network models and the yeast transcriptional network. *Proc. Natl. Acad. Sci. USA* **100**, 14796-14799.

Klemm, K. and Bornholdt, S. 2005. Topology of biological networks and reliability of information processing, *Proc. Natl. Acad. Sci. USA* **102**, 18414.

Koza, J.R. 1992. *Genetic Programming: On the programming of computers by means of natural selection.* MIT Press.

Krakauer, D. and Plotkin, J. 2002. Redundancy, antiredundancy and the robustness of genomes. *Proc. Natl. Acad. Sci. USA* **99**, 1405-1409.

Lenski, R.E., Ofria, C., Pennock, R.T. and Adami, C. 2003. The evolutionary origin of complex features. *Nature* **423**, 139-144.

Li, F., Long, T., Lu, Y., Ouyang, Q. and Tang, C. 2004. The yeast cell-cycle network is robustly designed. *Proc. Natl. Acad. Sci. USA* **101**, 4781-4786.

Melton, D. W. 1994. Gene targeting in the mouse. *Bioessays* **16**, 633-638.

Mendoza, L. and Alvarez-Buylla, E.R. 1998. Dynamics of the genetic regulatory network for Arabidopsis thaliana flower morphogenesis. *J. Theor. Biol.* **193**, 307-319.

Mendoza, L., Thieffry, D. and Alvarez-Buylla, E.R. 1999. Genetic control of flower morphogenesis in Arabidopsis thaliana: a logical analysis. *Bioinformatics* **15**, 593-606.

Miller, J. and Hartmann, M. 2001. Evolving messy gates for fault tolerance: some preliminary findings. in: *Proceedings 3rd NASA Workshop on Evolvable Hardware.* pp. 116-123.

Miller, J., Thompson, A., Thompson, P. and Fogarty, T. (Eds.) 2000. Proc. Third International Conference on Evolvable Systems: From Biology to Hardware, *Lect. Notes Comput. Sci.* 1801. Springer-Verlag. Berlin.

Sauro, H. H. and Khodolenko, B. N. 2004. Quantitative analysis of signaling networks. *Prog. Biophys. Mol. Biol.* **86**, 5.43.

Simpson, M. L., Cox, C. D., Peterson, G. D. and Sayler, G. S. 2004. Engineering in the Biological Substrate: Information Processing in Genetic Circuits. *Procs. IEEE* **92**, 848-863.

Solé, R. V., Fernandez, P. and Kauffman, S.A. 2003. Adaptive walks in a gene network model of morphogenesis: insights into the Cambrian explosion. *Int. J. Dev. Biol.* **47**, 685-693.

Tononi, G., Edelman, G. M. and Sporns, O. 1998. Complexity and coherence: integrating information in the brain. *Trends Cogn. Sci.* **2**, 474-484.

Tononi, G., Sporns, O. and Edelman, G. M. 1999. Measures of degeneracy and redundancy in biological networks. *Proc. Natl. Acad. Sci. USA* **96**, 3257-3262.

von Neumann, J. 1952. Probabilistic logics and the synthesis of reliable organisms from unreliable components. California Institute of Technology. Lecture Notes pp. 43-98.

Wagner, A. 2006. *Robustness and Evolvability in Living Systems.* Princeton U. Press. Princeton.

Wagner, G., Pavlicev, M. and Cheverud, J. M. 2007. The road to modularity. *Nat. Rev. Genet.* **8**, 921-931.

Weiss, R., Basu, S., Hooshangi, S., Kalmbach, A., Karig, D., Mehreja, R. and Netravali, I. 2003. Genetic circuit building blocks for Cellular Computation, Communications, and Signal Processing, *Natural Comput.* **2**, 47-84.

Willadsen, K. and Wiles, J. 2007. Robustness and state-space structure of Boolean gene regulatory models. *J. Theor. Biol.* **249**, 749-765.

Wuensche, A. and M.J.Lesser. 1992. *The Global Dynamics of Cellular automata.* SFI Studies in the Sciences of Complexity, Addison-Wesley.

Yu, T. and Miller, J. F. 2001. Neutrality and the Evolvability of Boolean Function Landscape. *Lecture Notes in Computer Science* **2038**, 204-217. Springer. New York.