The Relationship Between Evolvability and Robustness in the Evolution of Boolean Networks

David Shorten[1] and Geoff Nitschke[1]

[1]Department of Computer Science, University of Cape Town
dshorten@cs.uct.ac.za, gnitschke@cs.uct.ac.za

Abstract

Robustness and evolvability have traditionally been seen as conflicting properties of evolutionary systems, due to the fact that selection requires heritable variation on which to operate. Various recent studies have demonstrated that organisms evolving in environments fluctuating non-randomly become better at adapting to these fluctuations, that is, increase their *evolvability*. It has been suggested that this is due to the emergence of biases in the mutational neighborhoods of genotypes. This paper examines a potential consequence of these observations, that a large bias in certain areas of genotype space will lead to increased robustness in corresponding phenotypes. The evolution of boolean networks, which bear similarity to models of gene regulatory networks, is simulated in environments which fluctuate between task targets. It was found that an increase in evolvability is concomitant with the emergence of highly robust genotypes, where evolvability was defined as the population's *adaptability*. Analysis of the genotype space elucidated that evolution finds regions containing robust genotypes coding for one of the target phenotypes, where these regions overlap or are situated in close proximity. Results indicate that genotype space topology impacts the relationship between robustness and evolvability, where the separation of robust regions coding for the various targets was detrimental to evolvability.

Introduction

An open question in artificial and natural life is whether digital and natural organisms undergoing an evolutionary process are able to become better at adapting to the selective pressures presented to them, that is to become more *evolvable* (Wagner and Altenberg, 1996a). However, this question is complicated by multiple definitions of *evolvability*. In both evolutionary biology (Pigliucci, 2008, 2010; Wagner, 2008; Parter et al., 2008) and *Evolutionary Computation* (EC) (Tarapore and Mouret, 2014), for example, evolvability refers to either populations or individuals (Wilder and Stanley, 2015). Similarly, within EC (Eiben and Smith, 2003) numerous definitions and associated metrics have been proposed. For example, those that focus exclusively on solution fitness (Grefenstette, 1999; Reisinger and Miikkulainen, 2007) or variability of offspring (Lehman and Stanley, 2013). Tarapore and Mouret (2014) developed a metric which incorporated both the fitness and diversity of offspring. Reisinger et al. (2005) measured evolvability as the ability of genotypes with various representations to detect invariant patterns as commonalities in a changing fitness function.

Two significant definitions from the biological literature[1] which are pertinent to this work are those of *phenotypic accessibility* and *adaptability*. The former refers to the proportion of phenotypes that can be accessed through evolution (Wagner, 2008; Ciliberti et al., 2007a,b). This proportion is determined by the topology of the genotype space (that is, which other genotypes a given genotype can directly mutate to) and the G → P mapping (which phenotype each genotype codes for). Adaptability refers to the ability of organism to adapt to changes in the environment. It has the advantage of not imposing an experimenter-biased measure on the system, but merely asks whether evolution *delivers the goods* (Budd, 2006; Pigliucci, 2008). A prevailing hypothesis is that if the environment sufficiently varies over time, then organisms evolve the ability to be able to evolve suitable adaptations to such environmental changes faster (Wagner and Altenberg, 1996a; Wagner, 2008; Draghi and Wagner, 2008). Crombach and Hogeweg (2008) as well as Draghi and Wagner (2008) have demonstrated that computational models of *Gene Regulatory Networks* (GRNs) exhibit such *evolvability*.

The representation problem in EC addresses the issue of how to represent and adapt (mutate and recombine) genotypes in order that a broad range of complex solutions can be represented by relatively simple genotype encodings (Wagner and Altenberg, 1996b). The choice of representation and associated operators has a significant impact on the evolution of viable solutions and representations which facilitate evolution have been termed *evolvable* (Wagner and Altenberg, 1996b; Rothlauf, 2006; Simões et al., 2014). Similarly, in nature, genetic information defining the form and

[1] For a review of evolvability in biology, the reader is referred to Pigliucci (2008).

function of an organism is stored within its genotype, however, the developmental process which translates this information into phenotypes (the G → P map) is not well understood (Pigliucci, 2010). Yet, it has become clear that the G → P map is neither one-to-one nor linear (Gjuvsland et al., 2013). In many organisms and *Ribonucleic acid* (RNA) folding (Draper, 1992), it has been found that many genotypes can code for a single phenotype and that genetic change resulting from mutation is not proportional to phenotypic change (Pigliucci, 2010; Wagner, 2008; Parter et al., 2008).

These features of the G → P map have two important consequences on evolutionary dynamics. The first is that they allow for the emergence of robust genotypes. As many genotypes can code for one phenotype, it is possible that some number of a genotype's mutational neighbors code for the same phenotype as it does, thus affording the genotype a degree of mutational robustness (Wagner, 2008). The second consequence is that it becomes plausible that certain phenotypes are found in greater abundance in certain regions of the genotype space. This in turn results in genotypic mutations having non-random effects on phenotypes, opening up the possibility that the distribution of these effects is in some way advantageous (Hogeweg, 2012; Watson et al., 2014; Parter et al., 2008; Meyers et al., 2005; Gerhart and Kirschner, 2007; Pavlicev et al., 2010).

We hypothesize that both robustness and mutational bias facilitate evolvability, however, the complex relationship between robustness and evolvability makes this non-trivial to elucidate. Recent work has indicated that a certain degree of robustness has a large benefit on evolvability interpreted either as phenotypic accessibility (Wagner, 2008; Ciliberti et al., 2007a) or adaptation to a goal (Draghi et al., 2010). These studies are predicated on the assumption that a certain proportion of genotypes are non-viable and will never produce offspring. This constrains the portion of the genotype space that is accessible from a given genotype. That is, once robustness rises above a certain threshold, genotypes of a given phenotype get connected in large neutral networks which can be traversed in order to access a large variety of phenotypes and this access to variation allows for faster adaptation towards a stationary task target.

Mutational biases, however, can increase the likelihood of a given phenotype occurring in the neighborhood of a given genotype, thus facilitating adaptation towards it. Various recent studies have demonstrated that organisms evolving in environments fluctuating non-randomly are able to become better at adapting to these fluctuations, thus increasing their *evolvability* (Crombach and Hogeweg, 2008; Draghi and Wagner, 2008). It has been suggested that this is due to the emergence of biases in the mutational neighborhoods of genotypes (Hogeweg, 2012; Watson et al., 2014).

Hence, this study aims to examine a potential consequence of these biases, which is that increasing the bias towards a small number of phenotypes within a region of the genotype space will increase the robustness of the genotypes within that region. Experiments were conducted using the simulated evolution of boolean networks in a fluctuating environment. These boolean networks were comprised of nodes containing either NAND logic gates or threshold functions. The networks composed of threshold functions closely resemble models of gene regulatory networks used to demonstrate the emergence of evolvability in varying environments (Crombach and Hogeweg, 2008; Draghi and Wagner, 2008). The networks composed of NAND logic gates were similar to those used to demonstrate the emergence of modularity (Kashtan and Alon, 2005) as well as task performance speedup (Kashtan et al., 2007) in fluctuating environments.

Results indicate that an increase in evolvability is concomitant with the emergence of highly robust genotypes. Analysis of the genotype space elucidated that evolution finds regions containing robust genotypes coding for one of the target phenotypes and that these regions are either overlapping or situated in close proximity. It was further found that the greater the separation between robust regions, the greater the drop in fitness during target changes. This implies that less separated robust regions allows for greater evolvability to emerge.

Parameter	**Value Range**
Weights (w_{ij})	$[-2, 2]$
Thresholds (θ_{ij})	$[-3, 3]$
Number of nodes	20
Incoming connections per node	$[1, 10]$
Simulation iterations (maximum t)	6

Table 1: Parameters for the networks composed of nodes with threshold functions

Parameter	**Value Range**
Number of nodes	20
Incoming connections per node	2
Simulation iterations (maximum t)	6

Table 2: Parameters for the networks composed of nodes with threshold functions

Methods

Network Models

Networks were formed of nodes which could hold an activation value of either zero or one. Each node had an activation function which was either a threshold or a NAND function. The activations of other connecting nodes were used as the inputs to these functions, the specification of which nodes'

Operator	Description
`mutate_weight`	A new value for the weight of an edge is chosen from the allowed values.
`mutate_connection`	An incoming connection to a node is given a different source.
`mutate_threshold`	A new value for the threshold of a node is chosen from the allowed values.
`add_edge`	A new incoming edge is added to the node. It connects to a random node and has a random weight.
`delete_edge`	One of the node's edges is chosen at random and removed.

Table 3: Mutation operators for the networks.

Parameter	Value
Population size	5000
Births per generation	5000
Tournament size	10
Recombination	None
Generations per task variant switch	10
Number of generations	500
Mutation	See table 3
NAND connection mutation rate	0.05
Threshold connection mutation rate	0.005
Node mutation rate	0.02

Table 4: Evolutionary Algorithm Parameters

	Target One	Target Two
Pair One	0000011001100000	0110111111110110
Pair Two	0000011001100000	0000111001100100

Table 5: Task target pairs used. The bit strings represent the desired outputs for each input permutation, where the permutations are ordered by their integer interpretation. Thus, for instance, the first bit of the string represents the desired output for the permutation 0000, the last bit for permutation 1111 and the fourth bit for permutation 0100.

Setup	Silhouette score
NAND, pair one	0.27 (0.08)
NAND, pair two	0.08 (0.05)
Threshold, pair one	0.06 (0.03)
Threshold, pair two	0.06 (0.05)

Table 6: Silhouette scores for generated genotypes in the neighborhood of the best genotype at then end of multiple runs for each setup. Standard deviations are in parentheses.

activations to use thus implied the connectivity of the network. Updates were done synchronously. The threshold function used is specified in equation 1.

$$s_i(t+1) = \begin{cases} 1: & \sum_j w_{ij} s_j(t) > \theta_i \\ 0: & \sum_j w_{ij} s_j(t) \leq \theta_i \end{cases} \tag{1}$$

Where, $s_i(t)$ is the activation of the ith node at simulation iteration t, w_{ij} is the connection weight of the directed edge from the ith to the jth node, and θ_i is the threshold of the ith node. If no such connection exists then $w_{ij} = 0$. Table 2 presents the parameters of the NAND networks and table 1 presents the parameters of the threshold networks.

Mutation Operators

Table 3 specifies the mutation operators used in the evolutionary algorithm applied to evolve the networks. The `mutate_weight` operators were applied to the GRN node connections, where for every connection, the operator was applied with a given probability (table 4). All other mutation operators acted on nodes' activation and threshold functions with a given probability (table 4). In the evolution of networks with NAND functions only the `mutate_connection` operator was used, whereas all the operators were used in the evolution of networks with threshold functions.

Evaluation

Networks were evolved to perform specific boolean functions with four inputs and one output. Each network was therefore run sixteen times, once for each of the possible input permutations. For each run, nodes, designated as the *input nodes*, had their activations set to the values of the given input permutation and their activations were clamped to these values for the duration of the run. The network was then simulated for the number of time steps specified in tables 1 and 2.

At the end of the simulation, the value of a designated *output node* was read and if it matched the output value of the target function, for the given input, the fitness of the network was incremented by one. Networks could therefore have fitness values in the range $[0, 16]$.

Each evolutionary run had a pair of target functions. The target against which networks were evaluated alternated between the members of this pair. The target pairs are specified in table 5. Target one of pair one is the function (a XOR b) AND (c XOR d) and target two of pair one is (a XOR b) OR (c XOR d). This was done to maintain consistency with previous work on evolving boolean networks in fluctuating environments (Kashtan and Alon, 2005; Kashtan et al., 2007).

Target one of pair two is the same as in pair one, however the second target was chosen to differ randomly in two outputs, as can be seen in table 5. This choice was made so as to ascertain the effect of alternating between targets that are more similar. Moreover, these various experiment parameters were chosen because preliminary testing showed that they were conducive to the emergence of evolvability.

Evolutionary Algorithm

The networks were evolved with an *Evolutionary Algorithm* (EA) using tournament selection and elitist survivor selection (Eiben and Smith, 2003). Also, the EA used mutation only (there was no recombination operator). At each generation, 5000 tournaments of ten genotypes were created. The winner of each tournament went on to produce a single child. The fittest 5000 of the 10000 genotypes composed of children and current generation's population went on to form the subsequent generation's population. Table 4 presents the EA parameters. The choices of selection operators as well as algorithm parameters were made as preliminary experiments showed that they were conducive to the emergence of evolvability.

Experiments

For each of the two network types, evolution was run twenty times for 750 generations for each of two different task target pairs. Thus, four different evolutionary setups were each run twenty times. During evolution the goal was switched between the two targets of the pair every ten generations.

During each evolutionary run, at the end of each generation, if there was at least one genotype in the population with maximum possible fitness, then a test on the average robustness of such genotypes was run. That is, if at least one genotype had reached the current target, as specified in table 5, then such a test was run. This test consisted of randomly drawing 200 such genotypes from the population, allowing for a given genotype to be drawn multiple times, and then creating 15 mutated copies of this genotype, using the same EA mutation operator and parameters as specified in the Evolutionary Algorithm section. The average robustness of the maximum fitness genotypes was then computed as the proportion of these mutated copies which were also of maximum fitness.

Results and Discussion

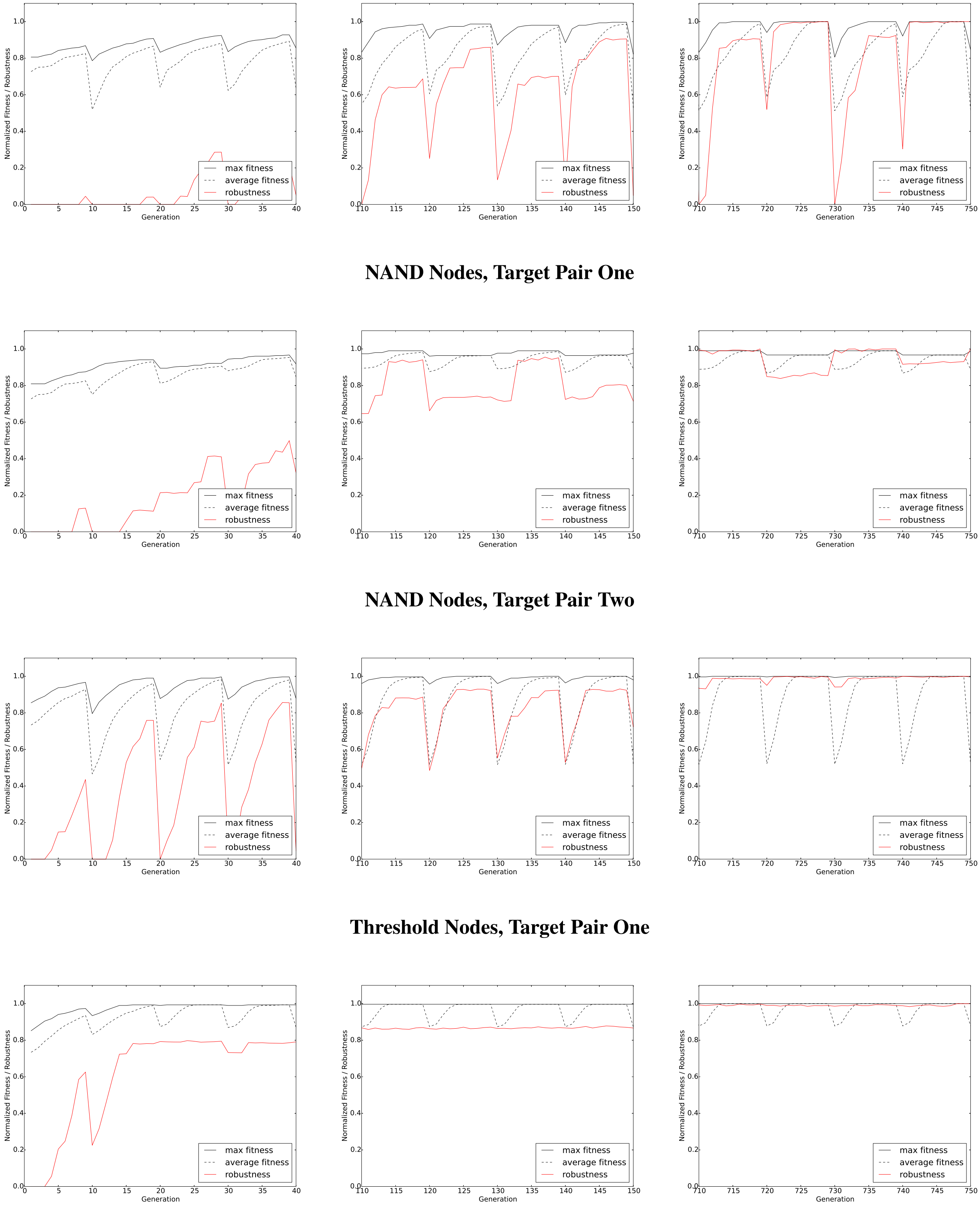
Figure 1 presents plots of the maximum fitness, average fitness and robustness of maximum fitness genotypes averaged over the 20 runs for each of the four setups. In early generations [0, 150] the average and maximum fitness respond more slowly to changes in the task target as compared to later generations [710, 750]. A further observation is that the average robustness of the maximum fitness genotypes that are present in the population increases over evolutionary time.

A useful statistic in this context is the correlation between the average robustness achieved after evolving towards one goal for the allotted ten generations and how quickly the population is able to adapt back to that goal after it has spent the next ten generations evolving towards the other goal. That is, we need to test the hypothesis that robustness and evolvability are correlated. In order to measure rate of adaptation, the average fitness two generations after the change back to the original goal was recorded.

We measure average absolute fitness, rather than the rate of fitness change (relative fitness) as an indicator of a population's evolvability. We elected to use absolute fitness, since measuring relative fitness would unfairly benefit genotypes whose fitness dropped the most after a change in the task environment. That is, given two populations, the one with the highest fitness a given interval after a task change is concluded to be the most evolvable. Also, this interpretation holds in the case that the other population suffers a greater fitness decrease after the task change, which might subsequently lead it to having a faster fitness increase.

Thus, for each of the four setups, the Pearson correlation (Freedman et al., 2007) was applied between the average robustness of the maximum fittness genotypes and the average fitness early into the evolution back to the goal. Specifically, the correlation measure was applied between robustness at the end of each period (where genotypes evolve towards the first target of the pair), and the average fitness two generations into the subsequent period evolving back towards this target. Results deriving from periods where no genotypes in the population were of maximum fitness (at the end of a period of evolution towards the first target) were excluded from the correlation computation. That is, robustness could not be calculated in these instances. It was found that in all four setups, there was a positive correlation between the robustness achieved and evolvability ($p < 0.01$). This supports our hypothesis that robustness and evolvability, defined as adaptability, are concomitant phenomena.

A further pertinent question was the structure of this robustness across genotype space. It is plausible that evolution could have found an area of robust genotypes for both targets interspersed.



NAND Nodes, Target Pair One

NAND Nodes, Target Pair Two

Threshold Nodes, Target Pair One

Threshold Nodes, Target Pair Two

Figure 1: Plots of the population average and maximum fitness as well as the average robustness of maximum fitness genotypes averaged over the 20 runs for each of the four setups. The left column contains plots from early generations [0, 40], the middle from slightly later generations [110, 150] and the right from generations near the end of the run [710, 750].

Alternatively it could be alternating between separate areas of robustness where these areas are adjacent or separated by a greater distance and connected by a strong fitness gradient. In order to ascertain this structure, visualizations of the genotype space were constructed using the t-SNE algorithm (Van der Maaten and Hinton, 2008).

For each of the four setups, evolution was run for 210 generations, where this number was chosen so that the population would be between robust regions (assuming their existence). At the end of these runs, mutated copies of the fittest genotype in the population were created using the same EA mutation operator and parameters as specified in the Evolutionary Algorithm section , although all mutation probability parameters were increased by a factor of two. This was because preliminary testing found smaller mutation probabilities caused the generation of sufficient target phenotypes to be too computationally expensive.

These mutated copies were created until there were 300 distinct genotypes which coded for each of the two target phenotypes. These genotypes were subsequently fed into the t-SNE algorithm which arranged them in a two-dimensional space to preserve distance in the higher dimensional (visualized) space. That is, genotypes which were closer together in the genotype space, formed clusters in the visualization.

The metric used in the t-SNE algorithm was the Hamming distance between genotypes. This is because two networks which differ only in the wiring of one connection should always be considered to be the same distance apart, regardless of the numerical identification values assigned to the nodes. A similar argument can be made for weights and thresholds.

The resultant visualizations are displayed in figure 2. When NAND networks were being used on target pair one there was a visible separation between the two robust regions. However, there was less separation in the other runs.

In order to gain a more quantitative understanding of this separation, the *silhouette score* (Rousseeuw, 1987) for each data set was computed. The silhouette score is a measure of the efficacy of a clustering algorithm. Scores are in the range $[-1, 1]$ and a positive score indicates distinct clusters, a negative score indicates data points being placed in the wrong cluster and a score of zero indicates overlapping clusters. Thus, in this instance, the silhouette score measured the efficacy of the clustering of the genotype space, where corresponding phenotypes were the labels used in the silhouette score calculation.

In order to facilitate statistical comparisons, this process was run 20 times. The results of these runs are displayed in table 6. The score for the NAND networks on target pair one is larger than for the other setups. Furthermore, this difference is statistically significant ($p < 0.01$, Mann-Whitney U test with Bonferroni correction (Flannery et al., 1986)).

The results indicate that an increase in evolvability, is concomitant with an increase in the robustness of genotypes coding for the target phenotypes. This can be observed in the plots of fitness and robustness displayed in figure 1, where rapid adaptation to new targets coincided with increased robustness. The link between evolvability and robustness is further supported by a statistically significant correlation between evolvability and robustness.

These results also support the hypothesis that increased evolvability is driven by biases in regions of genotype space (Hogeweg, 2012; Watson et al., 2014) and the inference that a strong bias towards certain genotypes will increase the robustness of these genotypes. Furthermore, finding a genotype within a given region will be aided by an abundance of this genotype. Thus, as it can be argued that evolvability implies robustness and robustness implies evolvability, the position of this paper is that the two phenomena are linked and occur in tandem.

Visualizations of the genotype space, as shown in figure 2, demonstrated that the nature of these biases can be non-trivial. That is, evolution does not necessarily settle on a region which is uniformly biased towards both of the target phenotypes. Although this case was observed, the corresponding case of adjacent regions, each biased to one of the phenotypes was also seen. The implications of these visualizations were supported by the significant difference in the silhouette scores between these cases.

Furthermore, the difference between these cases had a strong influence on the evolutionary dynamics, with a greater non-uniformity of the bias corresponding with larger drops in population fitness during task target changes.

It is theorized that the separation of the robust regions in the evolution of NAND networks on target pair one is due to the fact that, in that setup, phenotypes of either target are less likely to occur in close proximity. Nodes in the NAND networks had fewer possible connections than those in threshold networks (tables 1 and 2). Moreover, their thresholds and connection weights were not subject to mutation. This meant that the genotype space was of a much lower dimension, reducing the chance of two given phenotypes being nearby. Furthermore, the targets in pair one were more dissimilar. Should this hypothesis be correct, the implication is that the evolution of robust boolean networks is facilitated by a high dimension genotype space.

Moreover, these observations have led to further research questions that are the subject of current work. For example, confirmation of the above hypothesis, how large the distance between separated regions can be and whether analogous structures emerge when evolution fluctuates between larger numbers of task targets. We are also investigating the relationship between these results and improvements in adaptation, facilitated by robustness, increasing the accessibility of the genotype space (Draghi et al., 2010).

Conclusion

This study demonstrated that, in the simulation of evolving boolean networks in a fluctuating task environment, an

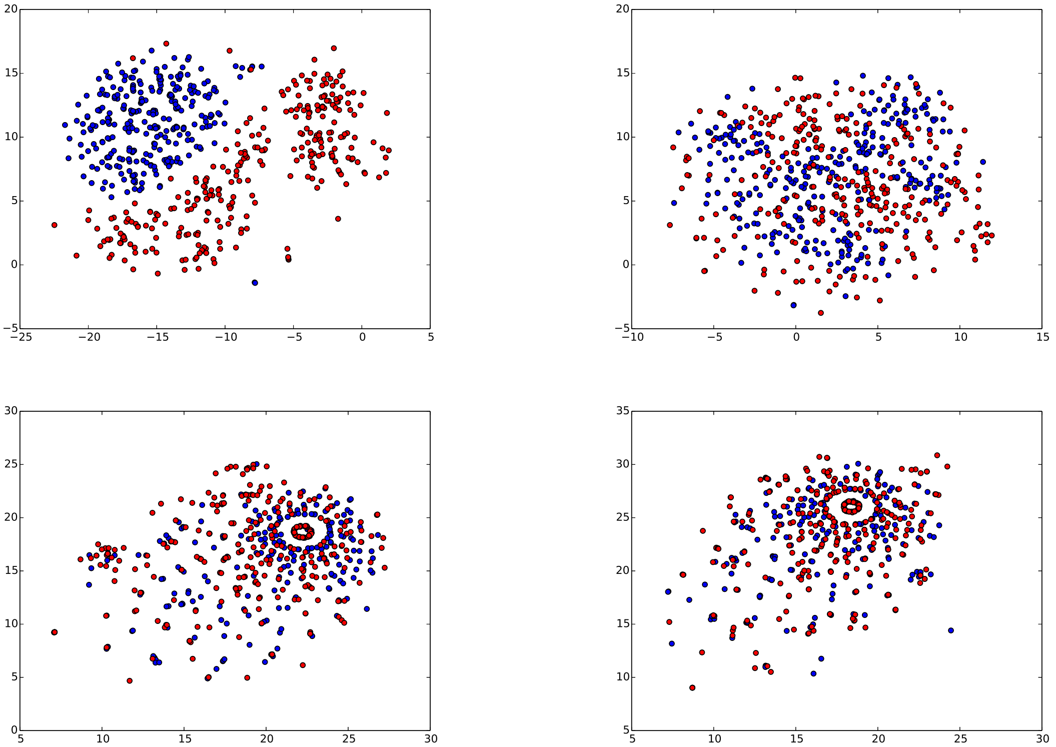


Figure 2: Visualizations of genotype space in the region of the best genotype at the end of an evolutionary run. Red dots are for genotypes that code for task target one and blue dots are for genotypes that code for target two. The positioning of the dots was determined using the t-SNE algorithm (Van der Maaten and Hinton, 2008) which aims to preserve the distance between the dots in the higher dimensional space. The left column contains visualizations based on target pair one and the right for target pair two. The top row contains visualizations for NAND networks and the bottom for threshold networks.

increase in *evolvability* was concomitant with an increase in the *robustness* of genotypes coding for the target phenotypes. These results support the hypothesis that evolvability, defined as adaptability, is driven by biases in the genotype space, however, visualizations of the genotype space showed unexpected structure in the nature of these biases. It was found that instead of biases towards different task targets occurring in the same region, in some instances they were separated into adjacent regions. These results contribute to a growing body of work (Wagner, 2008; Ciliberti et al., 2007a; Draghi et al., 2010) demonstrating the process by which robustness facilitates evolvability. However, elucidating the exact nature of the robustness, evolvability relationship remains the subject of ongoing research. Moreover, the impact of experiment parameters, notably the frequency of environment change, are yet to be determined.

References

Budd, G. E. (2006). On the origin and evolution of major morphological characters. *Biological Reviews*, 81(04):609–628.

Ciliberti, S., Martin, O., and Wagner, A. (2007a). Innovation and robustness in complex regulatory gene networks. *Proceedings of the National Academy of Sciences*, 104(34):13591–13596.

Ciliberti, S., Martin, O., and Wagner, A. (2007b). Robustness can evolve gradually in complex regulatory gene networks with varying topology. *PLoS Computational Biology*, 3:e15.

Crombach, A. and Hogeweg, P. (2008). Evolution of evolvability in gene regulatory networks. *PLoS Comput Biol*, 4(7):e1000112.

Draghi, J. and Wagner, G. P. (2008). Evolution of evolvability in a developmental model. *Evolution*, 62(2):301–315.

Draghi, J. A., Parsons, T. L., Wagner, G. P., and Plotkin, J. B. (2010). Mutational robustness can facilitate adaptation. *Nature*, 463(7279):353–355.

Draper, D. (1992). The rna-folding problem. *Acc. Chem. Res.*, 25(4):201–207.

Eiben, A. and Smith, J. (2003). *Introduction to Evolutionary Computing*. Springer, Berlin, Germany.

Flannery, B., Teukolsky, S., and Vetterling, W. (1986). *Numerical Recipes*. Cambridge University Press, Cambridge, UK.

Freedman, D., Pisani, R., and Purves, R. (2007). *Statistics*. International student edition. W.W. Norton & Company.

Gerhart, J. and Kirschner, M. (2007). The theory of facilitated variation. *Proceedings of the National Academy of Sciences*, 104(suppl 1):8582–8589.

Gjuvsland, A., Vik, J., Beard, D., Hunter, P., and Omholt, S. (2013). Bridging the genotype-phenotype gap: what does it take? *J Physiol.*, 591(8)::2055–2066.

Grefenstette, J. (1999). Evolvability in dynamic fitness landscapes: A genetic algorithm approach. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE.

Hogeweg, P. (2012). Toward a theory of multilevel evolution: long-term information integration shapes the mutational landscape and enhances evolvability. In *Evolutionary Systems Biology*, pages 195–224. Springer.

Kashtan, N. and Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *Proceedings of The National Academy of Sciences*, 102(39):13773–13778.

Kashtan, N., Noor, E., and Alon, U. (2007). Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences*, 104(34):13711–13716.

Lehman, J. and Stanley, K. (2013). Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PloS one*, 8(4):e62186.

Meyers, L. A., Ancel, F. D., and Lachmann, M. (2005). Evolution of genetic potential. *PLoS Comput Biol*, 1(3):e32.

Parter, M., Kashtan, N., and Alon, U. (2008). Facilitated variation: how evolution learns from past environments to generalize to new environments. *PLoS Comput Biol*, 4(11):e1000206.

Pavlicev, M., Cheverud, J. M., and Wagner, G. P. (2010). Evolution of adaptive phenotypic variation patterns by direct selection for evolvability. *Proceedings of the Royal Society of London B: Biological Sciences*, page rspb20102113.

Pigliucci, M. (2008). Is evolvability evolvable? *Nature Reviews Genetics*, 9(1):75–82.

Pigliucci, M. (2010). Genotype phenotype mapping and the end of the genes as blueprint metaphor. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1540):557–566.

Reisinger, J. and Miikkulainen, R. (2007). Acquiring evolvability through adaptive representations. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1045–1052. ACM.

Reisinger, J., Stanley, K., and Miikkulainen, R. (2005). Towards an empirical measure of evolvability. In *Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pages 257–264. ACM.

Rothlauf, F. (2006). Introduction. In *Representations for Genetic and Evolutionary Algorithms*, pages 1–7. Springer Berlin Heidelberg.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Simões, L. F., Izzo, D., Haasdijk, E., and Eiben, A. E. (2014). Self-adaptive genotype-phenotype maps: neural networks as a meta-representation. In *Parallel Problem Solving from Nature–PPSN XIII*, pages 110–119. Springer.

Tarapore, D. and Mouret, J.-B. (2014). Comparing the evolvability of generative encoding schemes. In *Proceedings of ALife*, pages 1–8. MIT Press.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85.

Wagner, A. (2008). Robustness and evolvability: a paradox resolved. *Proceedings of the Royal Society of London B: Biological Sciences*, 275(1630):91–100.

Wagner, G. and Altenberg, L. (1996a). Complex adaptations and the evolution of evolvability. *Evolution*, 50(1):967–976.

Wagner, G. and Altenberg, L. (1996b). Perspective: Complex adaptations and the evolution of evolvability. *Evolution*, pages 967–976.

Watson, R. A., Wagner, G. P., Pavlicev, M., Weinreich, D. M., and Mills, R. (2014). The evolution of phenotypic correlations and developmental memory. *Evolution*, 68(4):1124–1138.

Wilder, B. and Stanley, K. (2015). Reconciling explanations for the evolution of evolvability. *Adaptive Behavior*.