Anomalies in the Behaviour of a Modularity Inducing Problem Domain

Zhenyue Qin, Tom Gedeon and RI (Bob) McKay

Australian National University, Canberra ACT 2600, Australia
rimanucs@gmail.com

Abstract

Espinosa-Soto and Wagner (2010) introduced a domain with weak assumptions on biology and environment, where modular structures emerge under simple evolutionary processes.

We found a number of anomalous behaviours: modularity emerged in this domain, but could not dominate populations as observed in biology. Highly fit, modular solutions exist in the search space, can be readily found by a simple deterministic procedure (and presumably could dominate populations if found), but evolutionary search never found them, despite mutation biases that appear to favour those solutions. Moreover, emergence of modularity was promoted by stochastic dynamicity in the fitness function: a stochastic but fixed fitness function generated much less modular solutions.

Introduction

Adaptability is an essential property of both biological and artificial evolutionary systems (Yang et al., 2013). Biological organisms have already evolved it through modularity (Gerhart and Kirschner, 2007), giving hope for artificial systems to also generate modular, adaptable systems (Pfeifer and Scheier, 2001).

Lack of modularity is a key factor limiting scaling of artificial biological systems to higher complexity (Kashtan and Alon, 2005; Pfeifer and Bongard, 2006). Artificial neural networks are usually densely connected (Jain et al., 1996), where brains have modules taking different responsibilities – hippocampus for novel situations, amygdala for emotional controls (Coward, 2013). Thus it is important to understand the conditions under which modularity spontaneously emerged through biological evolution. Engineers may leverage them to generate modular systems, while understanding may help to winnow the evolutionary theories of biology.

Formally, modularity is the division of structures or functions into sub-units that perform autonomously (Schlosser and Wagner, 2004). A module is a group of elements which associate preferentially within the group (Newman, 2006; Espinosa-Soto and Wagner, 2010). Many biological activities and structures can be modeled as networks – animal brains, signaling pathways, etc. (Barabasi and Oltvai, 2004). A modular network can be partitioned into highly connected components, with only sparse connections between them (Freeman, 1977; Clune et al., 2013). Elements within a module preferentially undertake coherent functions independent of outside elements (Espinosa-Soto and Wagner, 2010; Larson et al., 2016). Such modules appear everywhere in biology (Coward, 2013), at multiple levels of organisation (Espinosa-Soto and Wagner, 2010; Coward, 2013). Modularity can promote the evolvability of organisms, i.e. the ability to rapidly adapt to novel environments (Pigliucci, 2008). Modular networks allow changes in one module without disturbing others; and modular structures can be reused and recombined to perform new functions (Espinosa-Soto and Wagner, 2010; Wagner and Altenberg, 1996).

Despite decades of research into modularity (Wagner et al., 2007), there is no consensus on its biological origin (Wagner and Mezey, 2004; Espinosa-Soto and Wagner, 2010). Three theories stand out, as their preconditions may commonly arise in nature (Wagner et al., 2007): modularly-varying evolutionary goals (Kashtan and Alon, 2005), biological parsimony pressures (Clune et al., 2013), and specialisations in gene activity patterns (Espinosa-Soto and Wagner, 2010). In the first, modular changes in environments generate an impetus toward modularity (Kashtan and Alon, 2005). Organisms whose environmental subcomponents change repeatedly show more modularity than those from stable environments (Parter et al., 2007). Fluctuations are omnipresent in real environments (Espinosa-Soto and Wagner, 2010; Yachi and Loreau, 1999). However it is unclear to what extent these fluctuations are modular (Espinosa-Soto and Wagner, 2010). While links in networks often incur costs, as Clune et al. assert, it is less clear that the cost is so uniform across the many forms of biological networks as to fully account for modularity's ubiquity.

Gene regulatory networks (GRNs) commonly regulate to preserve specific gene activation patterns against external disturbance – the target pattern may differ over time or location (Jones and Taylor, 1980). Espinosa-Soto and Wagner (2010) suggested this may promote modularity, and defined a GRN abstraction to test it. In the model, there

was initially a single target; over evolutionary time, additional modularly-structured targets were added. They used a mutation-only (crossover-free) evolutionary algorithm. Modularity was observed to emerge, though in contrast to biological systems, it was not seen to dominate populations. Subsequently, this work was extended by Larson et al. (2016), who examined the effects of different recombination mechanisms. Among other results, they first demonstrated an evolutionary impact from crossover hotspots, a phenomenon in which we were already interested.

We originally intended to extend this work, concentrating on two widespread biological phenomena, diploidy and crossover hotspots. To establish a baseline, we experimented with variants of standard genetic algorithms, yielding anomalous and difficult-to-explain results. This paper details some further results of our explorations, revealing further anomalies and leading to deep questions about our intuitions on both the structure of this problem, and biological evolutionary landscapes in general.

Methods

We use genetic algorithms as our evolutionary simulation tools. The GRN domain was originally proposed by Wagner (1996) and customised by Espinosa-Soto and Wagner (2010) and Larson et al. (2016). All simulation code was implemented in Java 1.8.0 and Python 2.7.10. Modularity was evaluated using the NetworkX package with the community API (Hagberg et al., 2008).

Model

Cells in an organism display heterogeneity in functionalities and morphologies, yet generally contain the same genes. This heterogeneity primarily arises from differing gene expression profiles resulting from differing gene-gene regulation (repression or activation). Cells interpret the same genetic material in different ways so that their behaviors and structures vary. These distinct interpretations are due to regulation, among other mechanisms via the activation and repression of genes by other genes, Such a GRN may be abstracted as a weighted directed graph with the weights limited to +1 (activation) and -1 (repression) (Wagner, 1996).

We used the Espinosa-Soto and Wagner (2010) representation: a GRN with N genes is represented as an N^2 adjacency matrix $A = a_{ji}$ with $a_{ji} \in \{-1, 0, 1\}$ with 0 representing independence of gene i from gene j. The gene activity pattern of this network at time t is a Boolean row vector $s_t = [s_t^0, ..., s_t^{N-1}]$. Gene i can either be active ($s_t^i = 1$) or inactive ($s_t^i = -1$). The state transition is modeled by:

$$s_{t+1} = \sigma[\sum_{j=1}^{N} a_{ji} s_t^j] \quad (1)$$

where $\sigma(x)$ equals 1 if $x > 0$ and -1 otherwise. For a more detailed explanation and justification of the model, please refer to the above paper.

Fitness

Biological GRNs are commonly able to maintain specific activation states in cells in the face of random external perturbations (Aderem, 2005). The abstraction by Espinosa-Soto and Wagner (2010) generated a set of P perturbations of the target, with each gene having 0.15 probability of mutating to the opposite state (they used $P = 500$, Larson et al. (2016) used $P = 300$). The GRN was recursively applied to each perturbation. Preliminary experiments indicated that it normally took fewer than 20 transitions to reach an attractor (Wagner, 1996). In that case, the Hamming Distance D between attractor and target state was returned; otherwise the maximum possible Hamming distance D_{max} was returned. In either case, the value $\gamma_i = (1 - D/D_{max})^5$ was computed for each perturbation i, with $1 \leq i \leq P$. Finally, the mean value $\bar{\gamma}$ over all γ_i was used to compute the fitness of the GRN g over a particular target t as:

$$f_t(g) = 1 - e^{-3\bar{\gamma}} \quad (2)$$

This process, of randomly sampling a set of perturbations of the target, and evaluating the GRN's ability to robustly return them to the target, was repeated each generation.

In stage one, the system evolved to regulate only the first target state. Subsequently, the fitness function rewarded regulation of newly introduced states, while maintaining pressure to regulate earlier ones, computing the overall fitness $f(g)$ as the arithmetic mean of $f_t(g)$ over all targets t.

We followed the strategy of Espinosa-Soto and Wagner (2010) (please see their paper for fuller detail), using only two targets: evolving for 500 generations with one target, then adding the second for a further 1500 generations. We based our choice of the number of perturbations (75) on a trade-off between the observation of Totten (2015) that 75–100 perturbations are sufficient for emergence of modularity, and the practical need to minimise runtime.

Larson et al. (2016) applied a different approach to evaluating the fitness of networks, by sampling a static (but larger) set of perturbations at the beginning of each run, and using this same set of perturbed targets whenever network fitness was calculated. This method has important computational cost advantages, since the fitness value of a given GRN on a given target remains fixed from generation to generation, so that cacheing and hashing methods can be used to give substantial speedups However it converts the original stochastically dynamic fitness evaluation into a static, deterministic one. This has potential implications for search.

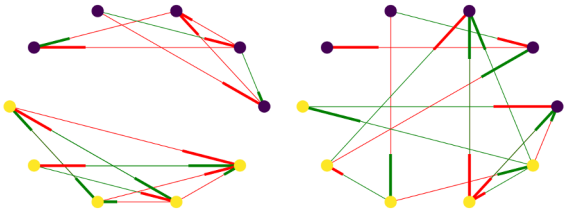
Evolutionary Simulations

Espinosa-Soto and Wagner (2010) imposed a mutation bias towards networks with a relatively low link density. A node u in the network has a probability $\mu = 0.05$ to mutate every

generation; if it does, it either loses or gains an interaction. The probability for u to lose an interaction is:

$$p(u) = \frac{4r_u}{4r_u + N - r_u} \tag{3}$$

where N is the number of genes in the target activation pattern, and r_u is the number of regulators of gene u (Espinosa-Soto and Wagner, 2010), i.e. the number of incoming edges. The probability an interaction is $1 - p(u)$. This bias acts to preserve the sparseness of the network, which computational biology research suggests is necessary for modularity to emerge.



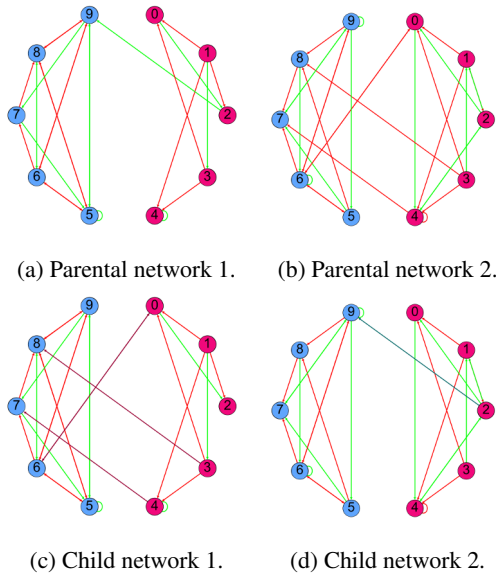
(a) A modular example (b) A non-modular example

Figure 1: Modular and Non-modular networks. Different node colors represent distinct modules (based on the modular changes in target values). Green/red edges mean activation/repression.

Espinosa-Soto and Wagner (2010) defined modules as components of targets that followed similar activation histories. In the most-used example, with two targets of length ten, the activations of the first five locations in both targets were identical, while the activations of the second five were inverted between the targets. Thus the modules treated as the connected components in the GRN involving nodes 1–5 and nodes 6–10. Figure 1 shows typical examples.[1]

Espinosa-Soto and Wagner (2010) used no crossover; Larson et al. (2016) used horizontal crossover, exchanging blocks of rows: when matrices A_1 and A_2 cross over at index i, the sub-matrices $A_1[0 : i-1, :]$ and $A_2[0 : i-1, :]$ remain unchanged, while the remainders are exchanged. However it ignores the diagonal symmetry of modules (if a_{ij} is in a module, then so is a_{ji}). We defined a 'diagonal crossover' using a diagonally symmetric interchange: given a random crossover point $[i, i]$, we preserve the sub-matrices $A_1[0 : i-1, 0 : i-1]$ and $A_1[i : 9, i : 9]$ (and also for A_2), while exchanging the rest. Compared with horizontal crossover, as Figure 2 illustrates, this should better preserve community structure.

[1]These color conventions are used throughout this paper. While color in the images conveys additional information, the key distinctions are still observable in black and white.



(a) Parental network 1. (b) Parental network 2.

(c) Child network 1. (d) Child network 2.

Figure 2: Illustration of diagonal crossover

Modularity Metric

In most of this paper, we used the Q modularity metric defined in Newman and Girvan (2004). This measures whether there are more inter-module edges than would be expected from the total number of edges. Formally:

$$Q = \sum_{i}^{K} [\frac{l_i}{L} - (\frac{d_i}{2L})^2] \tag{4}$$

Q falls in the range $[-\frac{1}{2}, 1)$, the upper bound depending on the number of modules ($\frac{1}{2}$ for two, $\frac{3}{4}$ for three).

Preliminary Experiments: Modularity Surprises

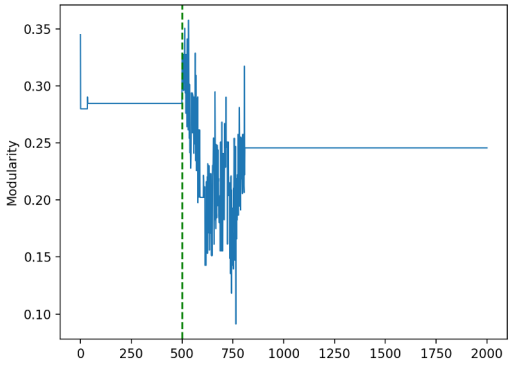


Figure 3: Modularity decreased after target change (marked by the vertical line).

Our baseline-setting experiments, using a standard genetic algorithm with an elite of 10 and tournaments of size 3, revealed surprising differences in the emergence of modularity from the results of Espinosa-Soto and Wagner (2010) and Larson et al. (2016). Recall that in their experiments, overall modularity increased after a second target was added to the fitness function. In our initial experiments (using the Louvain metric (Blondel et al., 2008) rather than the Q metric we use elsewhere), we instead observed a decrease immediately following the addition of the second target, with the overall modularity eventually stabilising to a level substantially below that of the first phase (see Figure 3).

Our settings differed from the previous work in the following ways:

1. Use of crossover (difference from the former only)
2. Tournament instead of proportional selection
3. Incorporation of elitism
4. Omission of the age–fitness Pareto mechanism (difference from the latter only, (Bongard, 2017))
5. Use of the Louvain metric

Of these differences, item 1 seems unlikely to explain our result since Larson et al. also used crossover, while items 3 and 4 both increase the relative eagerness of our search. The Louvain and Q metrics measure closely related properties, so seemed unlikely to be the cause. Item 2 is more complex, since a tournament of size 3 exerts relatively weak selection pressure, but the relative pressure of tournament and proportional selection varies with the stage of the algorithm. Proportional selection depends on relative differences in fitness, so it typically exerts fairly strong pressure in early stages of search, but as the population fitness converges and differences reduce, pressure weakens; by contrast, tournament selection, being dependent only on fitness rank order, exerts a relatively constant selection pressure throughout. In particular, when populations are relatively converged (as at the time of the target switch), we would expect even relatively small tournaments to be more selective than proportional.

Based on the above, we decided to test the joint effects of elitism and tournament selection. The results bore out this hypothesis: the same algorithm and settings, with elitism eliminated and proportional substituted for tournament selection, led to the emergence of modularity, see Figure 4.

Experiment Settings

Tables 1 and 2 show the gene activity patterns and the essential parameters of our evolutionary simulations. Unless otherwise specified, all experiments used the stochastic fitness evaluation of Espinosa-Soto and Wagner (2010). The only parameters that varied from the tables are the selection

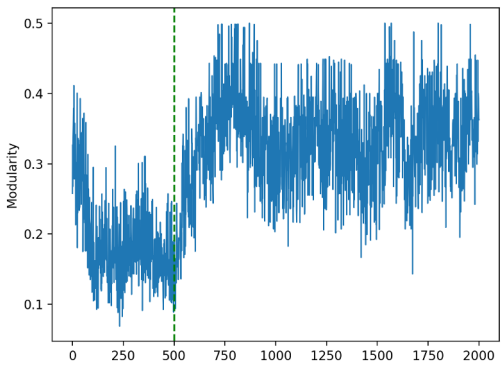


Figure 4: Without elitism, modularity increased after target change (marked by the vertical line).

Table 1: Gene Activity Patterns
1

Target Pattern	Generation where Pattern is added
+1 -1 +1 -1 +1 -1 +1 -1 +1 -1	0
+1 -1 +1 -1 +1 +1 -1 +1 -1 +1	500

type (tournament) and size, and the elite size. The detailed explanations of these parameters are given in Table 3.

The evaluation metrics for experiments are the eventual fitness values and Q scores from the last generation. All are significance tested using Wilcoxon's Signed-Rank Test.

Experiments and Results

Diagonal Crossover Promotes Modularity

We ran trials comparing horizontal, diagonal and no crossover, in all cases without elitism. As Tables 4 and 5 show, diagonal crossover generated significantly higher fitness and Q score than horizontal crossover, which in turn generated significantly higher Q score, though non-significantly lower fitness, than absence of crossover.

This Boolean model was proposed in Wagner (1996), who showed that random recombination made no difference to evolution of stability. Our experiments suggest that more structured forms of recombination (which occur in biology) can contribute to evolvability. Diagonal crossover can preserve underlying network modules. Although horizontal crossover did not preserve community structures as well as diagonal, its partitioning is still based on a modular structure, and thus partially preserves modularity.

Greediness Reduces Modularity

Elitism Reduces Modularity

Table 2: Parameters of the Evolutionary Simulations

Edge Size	Number of Perturbations	Per-location Perturbation Rate
20	75	0.15
Mutation Rate	**Population Size**	**Selection Type**
0.05	100	Proportional
Reproduction Rate	**Maximum Generation**	**Elite Size**
0.9	2000	0 or 10
Trials per Treatment	**Significance Test**	
40	Wilcoxon Signed Rank	

Elitism increases greediness of search by deterministially retaining the fittest individuals. We compared an elite of 10 with no elite, finding significantly lower fitness and modularity in the former case (Tables 6 and 7).

Comparing Best Fitness and Modularity between Proportional Selection and Tournaments of Various Sizes Tables 8 shows final generation best fitness increased with decreasing tournament size, with proportional selection falling between tournaments of sizes 2 and 3, though none of these differences reached significance (Table 9). Modularity showed a similar pattern, although with proportional selection yielding lower modularity than tournaments of size 3; however only the differences with tournaments of size 10 were significant.

Further Analysis and Discussion

Emergence rather than Dominance

Modularity in biological networks is remarkably robust; specific GRN modules cross not merely species boundaries, but are shared across kingdoms, having survived billions of years. They do not merely appear in populations, they are generally common to all individuals in a species (Schlosser and Wagner, 2004). By contrast, modularity in this problem domain is a delicate flower. It does not appear at all if selection pressure is strong, and its emergence is heavily dependent on evolutionary details; not merely the dynamic fitness variation emphasised in the problem definition, but the details of whether stochasticity in the fitness evaluation is static or dynamic (see below).

What should we make of this? One possible reaction is to follow Clune et al. (2013) and ascribe biological modularity emergence to direct linkage costs, so that further inquiry is unnecessary. However our further confirmation that this domain does promote the emergence of modularity, albeit somewhat fitfully, deserves further investigation.

Table 3: Explanations of simulation parameters

Target Patterns	patterns that are perturbed, and towards which gene regulatory networks evolve
Target Addition Generations	the generations where new targets are introduced
Edge Size	the initial number of edges in the gene regulatory network
Perturbation Number	the number of perturbed versions of each gene activity pattern
Perturbation Rate	the expected proportion of corrupted genes in a pattern
Mutation Rate	the probability of a GRN node to gain or lose an interaction
Population Size	the number of individuals in the population
Selection Type	the type of selection used, and where tournament, the size of the tournament
Reproduction Rate	the proportion of the population reproduced without change, vacancies being filled by the selection mechanism
Maximum generation	the generation when the simulation will terminate

Table 4: Final Generation Best Fitness and Q Score with No, Horizontal and Diagonal Crossover

	No Crossover	Horizontal	Diagonal
Fitness	0.9476	0.9446	0.9488
Q Score	0.1961	0.2919	0.3386

We looked more deeply at the diagonal crossover data of Table 4; we assumed that the data might reflect some fitness advantage from modularity (i.e. that over many runs, fit but highly modular individuals might be fitter than their non-modular cousins). So from each diagonal crossover run, we collected the fittest individuals among those that had the highest modularity value, and conversely, the least modular among those that had the highest fitness. Averaged over all runs, the latter were fitter than the former (Table 10) – the less modular networks could more robustly recover the unperturbed pattern. Thus the failure of the modular networks to dominate was less surprising: they could not do so because fitter, non-modular ones would take over.

We wondered whether this could arise from insufficient complexity in the targeted gene activity patterns. The num-

Table 5: Wilcoxon Ranked Sign Values for Table 4

	Fitness P	Q Score P
No $<$ Horz		1.7090e-6
Horz $\leq$ No	0.0882	
Horz $<$ Diag	0.0006	0.0019

Table 6: Final Generation Best Fitness and Q Score with and without Elites

	No Elite	Elite Size 10
Fitness	0.9488	0.9472
Q Score	0.3386	0.2735

ber of genes in patterns might be too simple, or the number of targets might be too few, to reflect natural environments. Perhaps modular networks might give great performance on complex tasks, but worse than non-modular ones for simple tasks. Using the basic set-up of subsection 'Diagonal Crossover Promotes Modularity', we ran more complicated evolutionary simulations using patterns comprising 15 nodes and three sub-patterns (for which the maximum modularity score is 0.75, rather than the 0.5 for two sub-patterns), encountering a sequence of seven different targets. Evolution was extended to 35,000 generations and during the final epoch from $(26000 \rightarrow 35000)$ generations, it was evolving to robustly recover all seven targets. We repeated the preceding analysis; the results in Table 11 resemble those of Table 10. Overall, the number and complexity of targets could not resolve the issue: less modular networks still recovered the target more robustly than more modular networks. However the differences in modularity were smaller, suggesting that modularity dominance might emerge with sufficiently complex problems – perhaps beyond the bounds of computational feasibility with current techniques.

Are Inter-Module Edges Critical to High Fitness?

Finally convinced that inter-module edges were essential to robust target recovery, we decided to validate this by an extreme meaure. From the 40 runs in the first experiment of this section, we extracted the least modular individuals among those having the maximum fitness value found in the run, yielding 40 fairly non-modular but highly fit individuals. We then simply removed all inter-module edges. Naturally the resulting networks were perfectly modular, but we expected them to be highly unfit. This expectation was borne out in 16 cases. But in the majority (24) this crude

Table 7: Wilcoxon Ranked Sign Values for Table 6

	Fitness P	Q Score P
Elite 10 $<$ No Elites	0.0003	0.0022

Table 8: Final Generation Best Fitness and Q Score for Proportional Selection and Different Sized Tournaments

	Proport	Tourn Size 2	Tourn Size 3	Tourn Size 10
Fitness	0.9488	0.9404	0.9404	0.9371
Q Score	0.3386	0.3697	0.3623	0.2783

Table 9: Wilcoxon Ranked Sign Values for Table 8

	Fitness P	Q Score P
Proportional $>$ Tourn Size 2	0.7401	0.0467
Tourn Size 3 $<$ Size 2	0.9313	0.7881
Tourn Size 10 $<$ Size 3	0.0164	0.0015
Tourn Size 10 $<$ Proportional	0.0227	0.0054

operation resulted in an *increase* in fitness. Figure 5 shows a real example of this process. The 'Before removal' GRN had a fitness of 0.9472. After removing $\approx 20\%$ of its edges to make it completely modular, its fitness rose to 0.9502. So the inter-modular edges were not merely inessential to the GRN's fitness, they were an impediment. Which leaves the question: since the resulting completely modular, very high fitness solutions were available to the search algorithm, why could it not find them?

Originally, we suspected this anomaly might result from the lower edge density of the 'after removing' networks – perhaps they were too much below the edge density targeted by the biased mutation operator, so that this soft constraint eliminated them from search. Further investigation revealed that on average, they still retained approximately 30 edges, which is above the targeted density of the mutation operator (equation 3), so far from being difficult to reach, the biased mutation operator favored moving toward them.[2]

In order to further understand this phenomenon, that

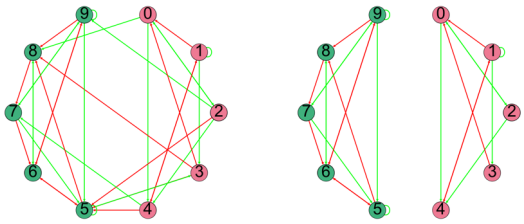
[2]We found this result so surprising that we suspected a bug in our implementation of the mutation bias. We turned off selection completely, giving every individual the same fitness. The average size stabilised at a little over 22 edges (because of asymmetry in the mutation distribution - a GRN must have at least zero edges, but has a finite probability of more than 40 - a small deviation above the mutation target of 20 is unsurprising).

Table 10: Modularity dominance for data underlying Table 4

Generation Range	Modularity	Fitness
(500, 2000)	0.5000	0.9482
	0.1736	0.9502

Table 11: Modularity dominance for extended runs from more complex environments

Generation Range	Modularity	Fitness
(26000, 35000)	0.5506	0.9100
	0.4151	0.9419



(a) Before removal (b) After removal

Figure 5: Illustration of inter-module connection removal.

our evolutionary simulations could not find a path to the trimmed networks, we recorded all fitness values that could be obtained by removing one inter-module edge in turn, until all have been deleted, and plotted them as graphs. Figure 6 is typical. We could usually see a steadily improving fitness as edges were deleted, along a path that was favored by the biased mutation operator, yet our genetic algorithm could not find these paths.

Fitness and Modularity benefit from Dynamic Fitness Evaluation

One important difference between the work of Larson et al. (2016) and of Espinosa-Soto and Wagner (2010) is the former's sampling perturbations only once across a run. Thus there is a single change in the fitness function (the changed target after generation 500), while for the latter, the changing perturbation sample varies the fitness evaluation every generation. Preliminary experiments had already shown that static fitness evaluation substantially decreased both ultimate fitness and modularity (Qin et al., 2018); we decided to try to improve the performance of the static version by limiting diagonal crossover to location $[5, 5]$ (i.e. the module boundary) – remarkably unsuccessfully, as lines 1 and 3 of Table 12 show. Unlike the fitness difference, the modularity difference was highly significant ($p = 2.6879 * 10^{-5}$).

To further understand this phenomenon, we collected the gene regulatory networks of the final generation, and mutated each network 499 times to generate neighbours. We measured the fitness values of these neighbours with the target perturbations from this generation, and determined their maximum. In this fashion, we would have 40 neighbourhood-maximum fitness values for both dynamic and static fitness evaluation. We repeated this process for

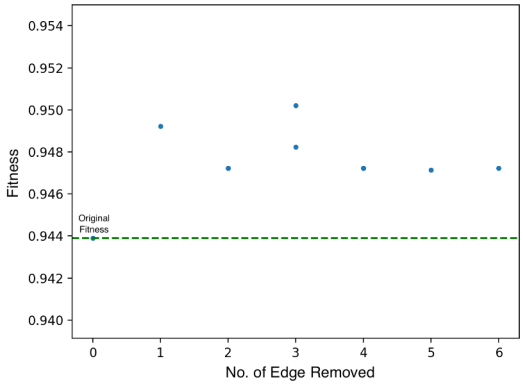


Figure 6: Fitness Values along all Inter-module Edge Removal Paths from a High Fitness, Low Modularity Network that results in Increased Final Fitness.

Table 12: Fitness and Q Scores for Neighbours of Final Generation Fittest Individuals in Static and Dynamic Environments

	Dynamic	Static
Final Fitness	0.9461	0.9323
Best Neighbour Fitness	0.9410	0.9323
Final Q	0.3374	0.1851
Best Neighbour Q	0.3791	0.2223

the modularity Q score. As Table 12 shows, the fittest individuals in the final generation for the dynamic problem were generally local optima, whereas for the static problem, they were generally on a fitness plateau, with equally fit neighbours, and a substantially lower fitness than found in the dynamic problem. In both scenarios, there were neighbours of substantially higher modularity than the original individual, but overall modularity, both of the final generation best individual and of its neighbours, were much higher in the dynamic problem than in the static. Again, fitness differences between dynamic and static were not significant, but modularity differences were ($p = 2.6956 * 10^{-5}$).

Conclusions

The modularity-inducing problem of Espinosa-Soto and Wagner (2010) is clearly important for understanding the evolution of network modularity, in that it relies on much weaker assumptions about the world that provides the evolutionary context (namely that some parts of the fitness target change independently of others) than do other explanations. It also provides potential insights into other aspects of biological evolution. The results of Larson et al. (2016) reveal

connections between modularity and recombination, and of particular importance, the first results known to us to suggest an evolutionary advantage to the ubiquitous phenomenon of recombination hotspots. Our own results suggest connections to homologous recombination, and to stochastic variation in fitness landscapes. They also reveal some puzzles: why don't modules dominate populations in this environment as they do in nature? Why can't the many different algorithms we tried find the very high fitness, completely modular solutions that we know to exist? Why does stochastic sampling generate fitter and more modular solutions than static sampling? We have ended up with more (but more detailed) questions than we started with. Clearly, further research and new analyses are needed...

Acknowledgements

Some of the outcomes / conclusions of the research in this paper were presented as a poster at GECCO 2018, and as a two-page summary in the proceedings (Qin et al., 2018). All underlying data is available at `https://goo.gl/6icgJm`, and code at `https://github.com/ZhenyueQin/Project-Maotai-Modularity`.

References

Aderem, A. (2005). Systems biology: its practice and challenges. *Cell*, 121(4):511–513.

Barabasi, A.-L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101–113.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Bongard, J. (2017). Personal communication.

Clune, J., Mouret, J.-B., and Lipson, H. (2013). The evolutionary origins of modularity. *Proceedings of the Royal Society B: Biological Sciences*, 280(1755):20122863.

Coward, L. A. (2013). Towards a theoretical neuroscience. In *Towards a Theoretical Neuroscience: from Cell Chemistry to Cognition*, pages 389–393. Springer.

Espinosa-Soto, C. and Wagner, A. (2010). Specialization can drive the evolution of modularity. *PLoS computational biology*, 6(3):e1000719.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.

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

Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL).

Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44.

Jones, P. A. and Taylor, S. M. (1980). Cellular differentiation, cytidine analogs and dna methylation. *Cell*, 20(1):85–93.

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

Larson, A., Bernatskiy, A., Cappelle, C., Livingston, K., Livingston, N., Long, J., Schwarz, J., Smith, M., and Bongard, J. (2016). Recombination hotspots promote the evolvability of modular systems. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 115–116. ACM.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.

Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

Parter, M., Kashtan, N., and Alon, U. (2007). Environmental variability and modularity of bacterial metabolic networks. *BMC evolutionary biology*, 7(1):169.

Pfeifer, R. and Bongard, J. (2006). *How the body shapes the way we think: a new view of intelligence*. MIT press.

Pfeifer, R. and Scheier, C. (2001). *Understanding intelligence*. MIT press.

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

Qin, Z., Gedeon, T., and McKay, R. I. B. (2018). Why don't the modules dominate? In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '18, pages 121–122, New York, NY, USA. ACM.

Schlosser, G. and Wagner, G. P. (2004). *Modularity in development and evolution*. University of Chicago Press.

Totten, M. (2015). Exploring the evolution of modularity in gene regulatory networks.

Wagner, A. (1996). Does evolutionary plasticity evolve? *Evolution*, 50(3):1008–1023.

Wagner, G. P. and Altenberg, L. (1996). Perspective: complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976.

Wagner, G. P. and Mezey, J. (2004). The role of genetic architecture constraints in the origin of variational modularity. *Modularity in development and evolution*, pages 338–358.

Wagner, G. P., Pavlicev, M., and Cheverud, J. M. (2007). The road to modularity. *Nature Reviews Genetics*, 8(12):921–931.

Yachi, S. and Loreau, M. (1999). Biodiversity and ecosystem productivity in a fluctuating environment: the insurance hypothesis. *Proceedings of the National Academy of Sciences*, 96(4):1463–1468.

Yang, X.-S., Cui, Z., Xiao, R., Gandomi, A. H., and Karamanoglu, M. (2013). *Swarm intelligence and bio-inspired computation: theory and applications*. Newnes.