

Metadata of the chapter that will be visualized in SpringerLink

Book Title	Advances in Computational Intelligence Systems	
Series Title		
Chapter Title	Evolutionary Constraint in Artificial Gene Regulatory Networks	
Copyright Year	2019	
Copyright HolderName	Springer Nature Switzerland AG	
Corresponding Author	Family Name	Turner
	Particle	
	Given Name	Alexander P.
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Hull University
	Address	Cottingham Road, Hull, HU6 7RX, UK
	Email	alexander.turner@hull.ac.uk
Author	ORCID	http://orcid.org/0000-0002-2392-6549
	Family Name	Lacey
	Particle	
	Given Name	George
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Hull University
	Address	Cottingham Road, Hull, HU6 7RX, UK
Author	Email	
	ORCID	http://orcid.org/0000-0003-0494-2460
	Family Name	Schoene
	Particle	
	Given Name	Annika
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Hull University
Author	Address	Cottingham Road, Hull, HU6 7RX, UK
	Email	
	ORCID	http://orcid.org/0000-0002-9248-617X
	Family Name	Dethlefs
	Particle	
	Given Name	
	Prefix	
	Suffix	
	Role	
	Division	

Given Name	Nina
Prefix	
Suffix	
Role	
Division	
Organization	Hull University
Address	Cottingham Road, Hull, HU6 7RX, UK
Email	
ORCID	http://orcid.org/0000-0002-6917-5066

Abstract

Evolutionary processes such as convergent evolution and rapid adaptation suggest that there are constraints on how organisms evolve. Without constraint, such processes would most likely not be possible in the time frame in which they are seen. This paper investigates how artificial gene regulatory networks (GRNs), a connectionist architecture designed for computational problem solving may too be constrained in its evolutionary pathway. To understand this further, GRNs are applied to two different computational tasks and the way their underlying genes evolve over time is observed. From this, rules about how often genes are evolved and how this correlates with their connectivity within the GRN are deduced. By generating and applying these rules, we can build an understanding of how GRNs are constrained in their evolutionary path, and build measures to exploit this to improve evolutionary performance and speed.

Keywords
(separated by '-')

Artificial gene regulatory networks - Evolutionary dynamics - Computational optimisation



Evolutionary Constraint in Artificial Gene Regulatory Networks

Alexander P. Turner^(✉) , George Lacey , Annika Schoene ,
and Nina Dethlefs 

Hull University, Cottingham Road, Hull HU6 7RX, UK
`alexander.turner@hull.ac.uk`



Abstract. Evolutionary processes such as convergent evolution and rapid adaptation suggest that there are constraints on how organisms evolve. Without constraint, such processes would most likely not be possible in the time frame in which they are seen. This paper investigates how artificial gene regulatory networks (GRNs), a connectionist architecture designed for computational problem solving may too be constrained in its evolutionary pathway. To understand this further, GRNs are applied to two different computational tasks and the way their underlying genes evolve over time is observed. From this, rules about how often genes are evolved and how this correlates with their connectivity within the GRN are deduced. By generating and applying these rules, we can build an understanding of how GRNs are constrained in their evolutionary path, and build measures to exploit this to improve evolutionary performance and speed.

Keywords: Artificial gene regulatory networks
Evolutionary dynamics · Computational optimisation

1 Introduction

For a species to adapt over time to a specific environment, it must be able to incorporate meaningful change into its genome whilst selecting against maladaptive outcomes. There are certain regions within the DNA of organisms which code for functions which are essential to the survival of the organism, and are under strong purifying selection. Processes such as redundancy mean that essential processes within an organism are often coded by multiple genes [8, 25]. This reduces the likelihood of that function being removed due to a specific mutation. Conversely, there are sections of the DNA which are unlikely to provide any change to the organism regardless of if they are mutated, and such regions are often found to be void of modularity and redundancy [24]. In between these two examples is an evolutionary sweet spot, where mutations can cause meaningful changes to an organism without the risk of lethality. Hence, it is likely that there are constraints onto how evolution can progress and the direction it will take based upon an organisms genetic structure, and which areas allow for

useful variability [28]. If such constraints can be known, it may be possible to inform predictions about how a species will evolve, which is the theory behind evolutionary constraint.

Bio-inspired computational techniques have been designed to mimic biological systems *insilico* for decades [12, 13, 15, 17, 18, 20]. There have been two main reasons for doing this. Firstly, and most predominately is to develop computational systems which are intelligent and are more able to behave like biological systems. The front runner in this field is artificial neural networks [15]. The second is to build computational models to improve the understanding of biology. Previously, conclusions drawn about biological processes were typically derived from comparatively expensive wet-lab techniques [7, 9, 11]. Computational methods have helped to improve the time frame in which certain biological experiments can be conducted, whilst alleviating a range of ethical concerns [21]. Some models are able to both solve computational problems, whilst being able to provide evidence to support biological theory [19].

In this work we consider how an artificial gene regulatory network (GRN), a type of bio-inspired computational model which is based upon gene regulation can be used to quantify constraint during evolution based upon its topology [10]. This will be done by analysing the connectivity of the underlying genes within the GRN over its evolutionary path to ascertain if certain levels of connectivity between the genes are more likely to provide beneficial mutations. Based upon these findings, this work will then create a targeted evolutionary approach which focuses on mutating genes which are more likely to cause useful meaningful change without loss of functionality within the GRN. There are two main reasons for using a GRN in this capacity. Firstly, genes can evolve a wide range of behaviors, allowing the emergent GRN to possess a wide range of functionality, akin to genetic networks in biology [20, 23]. Secondly, the GRN is particularly robust to structural perturbations [20]. This allows for variation within the networks in a similar vein to a biological organism, a key property in evolvable systems [22, 23]. The GRN will be applied to two tasks, the inverted pendulum task and a pattern generating task, both of which are modified to run indefinitely to promote evolution over long time-spans. If a task is solved, the task will randomly select new parameters and the network will have to adapt to this new task again. In order for the GRN to be applied to a task, a simple evolutionary algorithm will be used, which allows for the evolution of the GRN to be closely monitored throughout experimentation.

2 Biological Evolution

Evolution can be considered the heritable change to a species over generations. The backbone of evolution is DNA, which holds the code responsible for describing the primary structure of proteins in an organism, as well as certain regulatory information. DNA can be organised into genes, which specify a unit of hereditary information. The process of evolution is necessary to allow species to adapt over

time, allowing organisms to thrive in varying environments. It is an evolutionary advantage to be able to allow for positive genetic variation whilst reducing lethality due to such variation.

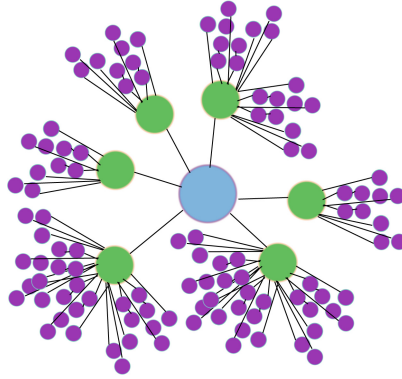


Fig. 1. An illustration of a hub, abstracted from [26] which shows how genes are interacting. If we are to look at this hub from a connectivity point of view, the blue center node is connected to all the green nodes, and indirectly to all the purple nodes. Hence, if the blue node were to mutate, the entire hub could fail to function. If for example one of the purple nodes was mutated, it would be unlikely to cease the functionality of the hub or cause a significant change. The green nodes sit in between the blue and purple ones, a mutation would mostly likely not cause the hub to fail, but would probably cause significant change in its functionality.

The ability for a species to be able to accept change yet maintain its function can be described as robustness [23], of which there are three key concepts. The first is modularity, which is the process of keeping functioning genes and genetic behaviors within a module meaning that the failure of a single module can be prevented from causing the failure of an entire organism. The second is redundancy, which as mentioned earlier specifies that a functioning unit is encoded by multiple genes. This means that more important functioning units are not coded for in exactly one place, and multiple mutations would be required to cause negative effects. The third is decoupling, and is the idea that the phenotype of an organism is the product of an indirect representation of that organism or functionality [23]. This provides separation from the function or an organism and low level genetic encoding, providing robustness and yet the opportunity to maintain variation in the species.

2.1 Constraint

Evolutionary constraint is a hypothesis which suggests that the pathway of evolution is most likely to occur in certain directions [30, 31]. Indeed, such processes as convergent evolution and rapid adaptation are difficult to explain without

evolutionary constraint. Take Fig. 1 as an abstracted example of the evolution of a hub. If the center node is mutated, it will likely have a large effect on the hub, which although could be positive, could also be negative. If one of the outside nodes is mutated, it is much less likely to cause an effect on the hub as its connectivity is much lower. Evolutionary constraint proposes that there is a sweet spot between these two examples, such as the green nodes, which can produce meaningful change without risking non-functionality. The sweet spot guides the path of evolution allowing for processes such as convergent evolution [28, 30, 31].

3 The Artificial Gene Regulatory Network

The artificial gene regulatory network (GRN) is a computational model designed for problem solving which takes inspiration from gene regulation in nature. The GRN consists of a set of genes (nodes), each containing a parameterised regulatory function, typically a variable sigmoid function. Each gene has an expression level, which can either be used to interact with a task, or to update the expression of other genes. In this work, the expression level of a gene is calculated using the sigmoid function in Eq. (1), where s (sigmoid slope) $\in [0, 20]$, b (sigmoid offset) $\in [-1, 1]$. Each gene in the network has a number of connections and a weight, which are used as inputs to update a given genes' expression levels. This can be seen in Eq. 2, which specifies that a genes expression level is the weighted sum of the expression levels of the genes in which it is connected, and then this value is passed through that genes' regulatory function and this is the expression level for that gene (Eq. 1). The GRN is the emergent property of the behavior of its underlying genes.

$$f(n) = (1 + e^{-sx-b})^{-1} \quad (1)$$

$$x = \sum_{j=0}^n i_j w_j \quad (2)$$

Formally, this GRN architecture can be defined by the tuple $\langle G, L, \text{In}, \text{Out} \rangle$, where:

G is a set of genes $\{n_0 \dots n_{|N|} : n_i = \langle a_i, I_i, W_i \rangle\}$ where:

$a_i : \mathbb{R}$ is the activation level of the gene.

$I_i \subseteq G$ is the set of inputs used by the gene.

W_i is a set of weights, where $0 \leq w_i \leq 1$, $|W_i| = |I_i|$.

L is a set of initial activation levels, where $|L_N| = |N|$.

$\text{In} \subset G$ is the set of genes used as external inputs.

$\text{Out} \subset G$ is the set of genes used as external outputs.

For the GRN to interact with a task, a set of inputs relating to the state of the task are mapped onto the genes of the network. This is done by setting the expression level of a gene to a value representing the state of the task. Other genes in the network can then use the expression values of the genes which are mapped to the inputs to update their own expression values using the processes

shown above. The GRN can then execute by updating the expression values for all genes in the network. The expression value(s) of a sub set of genes are then mapped back to the task. These steps are repeated at each time step in the task.

4 Optimisation

In this work we want to focus on the evolutionary process and observe each step of the process to understand how it is functioning. Moreover, we wish to look at a single GRN throughout the evolutionary process rather than a population of them. To achieve this we use a 1+1 evolutionary algorithm which is similar to a hill climbing heuristic [14]. As the genes constitute a GRN, a single mutation of a network will focus on a given gene, for which its regulatory function, its connectivity and its weight are all able to be mutated. It would be possible to use a more classical evolutionary algorithm with a large population which would probably yield stronger objective performance of the GRN, however the focus on this work is on understanding evolutionary constraint in the GRN, and the objective performance of the networks is of lower priority. The algorithm detailing the optimisation process can be seen in Algorithm 1.

Algorithm 1. Optimising a GRN

```

1:  $P \leftarrow$  new random GRN
2: for number of evaluations do
3:   CLONE  $P$  AS  $Q$ 
4:   MUTATE( $Q$ )
5:   EVALUATE( $Q$ )
6:   if  $Q.\text{fitness} \geq P.\text{fitness}$  then
7:      $P = Q$ 
8:   end if
9: end for

```

5 Task Definitions

To focus on the evolutionary process, we have adapted two tasks to be in a never ending computational loop. These tasks can be solved in a classical fashion, but whenever they are solved the parameters of the task will change, which requires the GRNs to be continuously evolving. This is to provide an environment for the GRNs to adapt evolvable characteristics. Each of the tasks will have a set number of time-steps, and the objective performance of a GRN will be the amount of times it completes a given task. For each experiment, the GRN will contain 20 genes, and this will not be changed during the optimization process. The mutation process, as specified in Algorithm 1 will mutate 3 genes at every single time step.

In the first set of experiments the mutations will be randomly selected, so that at each time step, 3 genes are selected randomly for mutation, in which

any one of their parameters will be modified. In the second set of experiments, the mutations will be guided according to the distributions discovered in the first set of experiments, which will have provided rules and correlations as to which genes are most likely to induce positive change of the GRN. 50 runs will be conducted for the coupled inverted pendulums tasks, and 100 for the pattern generation task as it is less computationally expensive.

5.1 Coupled Inverted Pendulums

The inverted pendulum task [16] in this instance consists of a cart mounted to a 1-dimensional track within a finite space, with a pendulum hanging vertically downwards from the cart. The boundaries of the space must not be exceeded by the cart. The objective of the task is to move the cart in such a way that the pendulum swings from below the cart and can be balanced in equilibrium above it. This task was designed as an efficient proxy for evolving decentralised robotic controllers, and has served as a benchmark for many applications [27]. The inputs and outputs from the task can be seen in Table 1.

Table 1. Sensory inputs used for the inverted pendulums task. The values are re-scaled to $[0, 1]$ before they are used as inputs to a network.

ID	Sensor name	System to sensor mapping
S_0	Pendulum Angle 0	$\phi \in [0, 0.5\pi] \rightarrow [127, 0]$, 0 else
S_1	Pendulum Angle 1	$\phi \in [1.5\pi, 2\pi] \rightarrow [0, 127]$, 0 else
S_2	Pendulum Angle 2	$\phi \in [0.5\pi, \pi] \rightarrow [127, 0]$, 0 else
S_3	Pendulum Angle 3	$\phi \in [\pi, 1.5\pi] \rightarrow [0, 127]$, 0 else
S_4	Proximity 0	Distance left $\rightarrow [0, 127]$
S_5	Proximity 1	Distance right $\rightarrow [0, 127]$
S_6	Cart Velocity 0	$v \in [-2, 0] \rightarrow [127, 0]$, 0 else
S_7	Cart Velocity 1	$v \in [0, 2] \rightarrow [0, 127]$, 0 else
S_8	Angular Velocity 0	$w \in [-5\pi, 0] \rightarrow [127, 0]$, 0 else
S_9	Angular Velocity 1	$w \in [0, 5\pi] \rightarrow [0, 127]$, 0 else
A_i	Actuators 0	$A_i \in [0, 127]$, for $i \in 0, 1$
u	Motor Control 0	$2(A_0/127 - A_1/127) \rightarrow [0, 1]$

A single evolutionary run of the inverted pendulum task will consist of 1'000'000 evaluations, and at each point if the task is solved, the gravitational constant within the system will be randomly modified from 9.81 initially to a number between 7 and 12. This changes the dynamics of the tasks to an extent where it is likely that the GRN will have to evolve new behaviors to learn how to solve it again, yet maintains the overall order of the system. This is aimed at evolving the GRN to be able to solve many different variants of the same task, and to improve its evolutionary robustness.

5.2 Pattern Generation

The pattern generation task consists of 10 randomly selected Boolean values which the GRN has to emulate in order. This is done by executing the network over 10 time steps, and generating an output at each time step. If the output is above 0.5, this represents a true value, otherwise it is false. If the outputs from the network matches the Boolean values, the task is reset, and a new set of Boolean values is generated which must then be matched. The Boolean values required to solve the task are never given to the GRN, so the network has to enter a dynamical regime which is capable of recreating these values. For a single run, 10'000'000 evaluations will be conducted.

6 Results

The results showing the number of mutations depending on the connectivity of a given gene for each task can be seen in Figs. 2 and 3. Both of these graphs show that there is a similar characteristic regardless of which task the GRN is being optimised for. If we look at the mean mutations for a given gene (left graph of Figs. 2 and 3) it can be seen that genes which have 2 connections are more positively mutated than genes with any other number of connections. The distributions show that for the inverted pendulum task (Fig. 2) the median number of positive mutations for genes with a single connection was slightly higher than that for a gene with two connections. Both tasks show that after 2 connections, as the number of connections increase genes are less likely to get mutated and yield a positive outcome for the GRN.

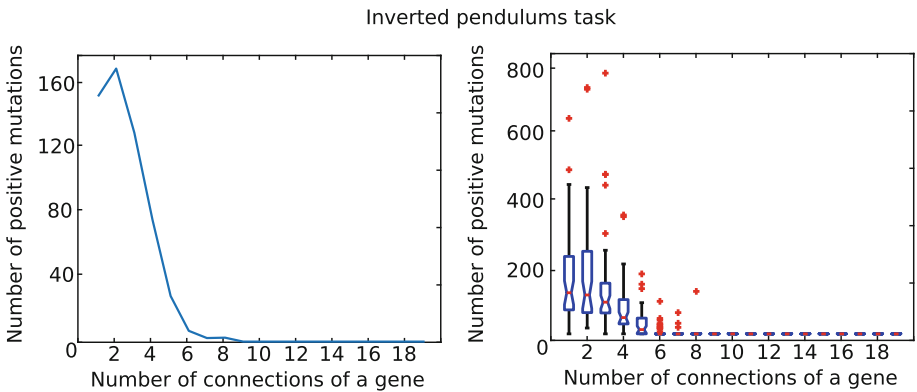


Fig. 2. The data for inverted pendulums task (mean left, distributions right). Given a gene with a certain number of connections, how often was that gene positively mutated. The results show that genes with 2 connections achieved the most positive mutations.

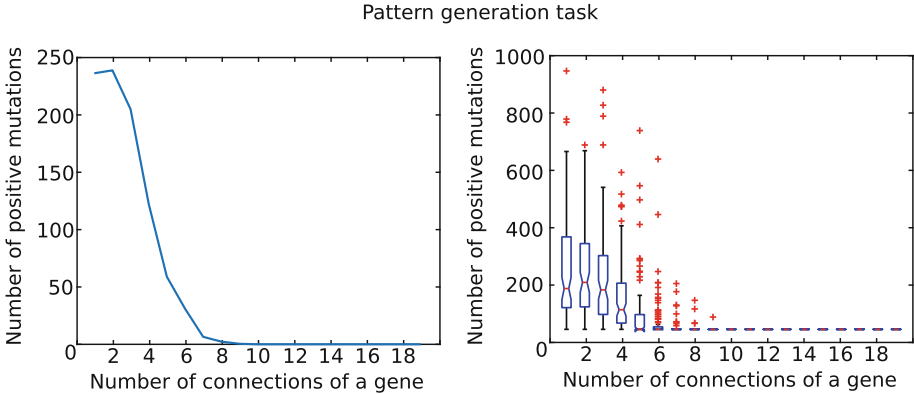


Fig. 3. The data for the pattern generation task (mean left, distributions right). Given a gene with a certain number of connections, how often was that gene positively mutated. Similar to the results for the inverted pendulums task, the results show that genes with 2 connections achieved the most positive mutations.

The raw data in Figs. 2 and 3 doesn't take into account the distribution of gene connectivity within the networks. This is important, as if there are more genes with 2 connections in the networks by default (which may be a property of the networks and their parameters) then they are more likely to be positively mutated for the simple reason that there are more of them. The data in Figs. 4 and 5 modifies the raw data in Figs. 2 and 3 by taking into account how many genes there are with a given number of connections in each run. What can be seen by doing this is that the general trend of the raw data (Figs. 2 and 3) still persists. There is a clear trend in both tasks for the mutation of genes with a given number of connections. For the inverted pendulum task, the mutation of genes with 2 connections resulted in proportionality the most positive benefit for the network, and in the pattern generation task, mutating genes with 3 connections was the most beneficial.

It is possible that the raw data was affected by individual run bias, where a single run may have produced a disproportionate number of mutations, favoring a specific gene and thus could affect the overall results. When the results were normalised for each run, so no one run could produce such a bias, the trends seen in Figs. 4 and 5 still persisted.

This information suggests that because certain connectivity of genes are favored in particular situations, there are genes which are more likely to yield positive mutations for the GRN than others. This suggests that we can predict based upon the distributions seen, the likelihood of certain genes being positively mutated over others, suggesting that multiple GRNs develop similar evolutionary patterns, which is the basis of convergent evolution.

Inverted pendulum task

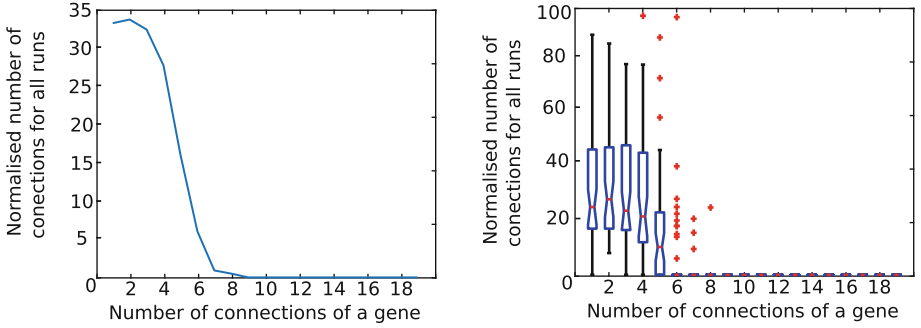


Fig. 4. The data from the inverted pendulum task (Fig. 2) divided the number of genes with a given connection which exist within the GRN. This is to prevent bias associated with genes of a certain connectivity being more common in a GRN.

Pattern Generation Task

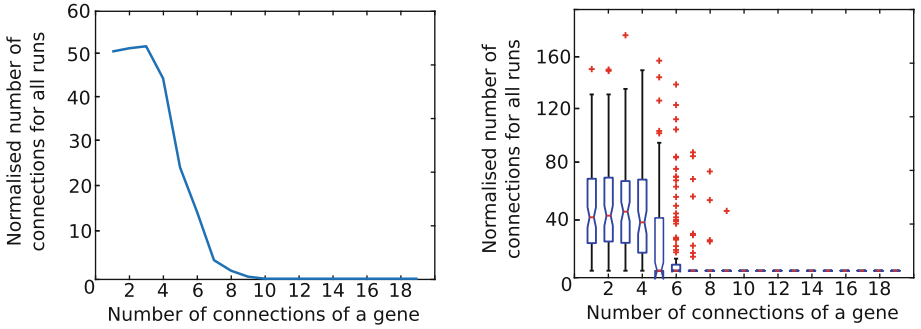


Fig. 5. The data from the pattern generation task (Fig. 3) divided the number of genes with a given connection which exist within the GRN. This is to prevent bias associated with genes of a certain connectivity being more common in a GRN.

6.1 Applying Mutational Distributions

We apply the distributions in Figs. 2 and 3 on a new set of experiments, where the genes are mutated according to their connectivity of these distributions. The results for this can be seen in Fig. 6. It can be seen that if these distributions are followed for both tasks, the GRNs evolve to solve the task more frequently, indicating that it is a positive evolutionary strategy. The difference between using random mutations and mutating genes according to the distribution is statistically significant for both tasks (For the pendulums the significance was $2.037e-5$, and for the pattern generation task the significance was 0.0422. Both

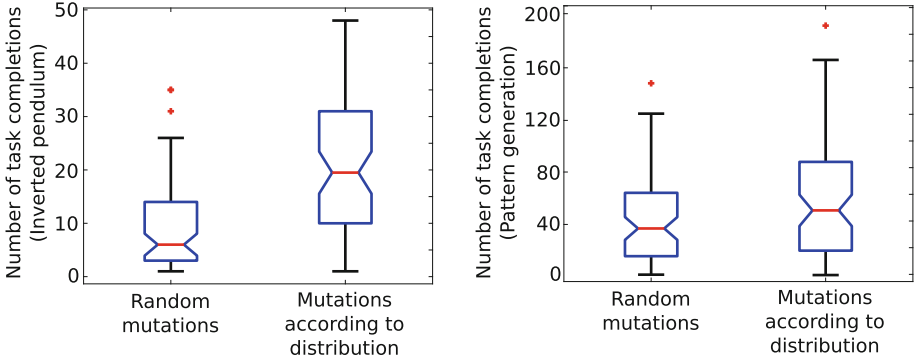


Fig. 6. The results of the inverted pendulum task and the pattern generation tasks when the distributions from Figs. 2 and 3 are used throughout evolution respectively. The graph shows that modifying the genes according to a distribution yields significantly better ($2.037\text{e-}5$) results for the inverted pendulum task, and significantly better results for the pattern generation task (0.0422). Both used the Wilcoxon rank-sum test.

used the Wilcoxon rank-sum test). This suggests that there are evolutionary pathways in GRNs which are more likely to yield positive results than random mutations alone, suggesting that certain properties of convergent evolution are present in the networks.

7 Conclusions

In this paper we investigated how the connectivity of genes within artificial gene regulatory networks (GRNs) influences the likelihood that a mutation with either be beneficial or detrimental. We looked at two different experiments, the inverted pendulum task and the pattern matching task. Both showed that genes with a certain connectivity (2 and 3 connections) are more likely to yield improvements to the network when mutated. We then showed that there are performance benefits when such distributions are used to guide the evolutionary process, focusing on genes which are most likely to provide beneficial mutations according to their connectivity. This significantly improved the performance of the GRNs for both tasks.

This work shows that for a given GRN and task there may be a more likely evolutionary path depending on the topology of the network and the connectivity of its genes. Although convergent evolution is a multifaceted topic in genetics, this work shows that a computational version of convergent evolution is present in GRNs, and exploiting it speeds up the evolutionary process. In future work it is possible that GRNs could be used as a platform to better understand convergent evolution in nature.

References

1. Good, B.H., Desai, M.M.: Fluctuations in fitness distributions and the effects of weak linked selection on sequence evolution. *Theor. Popul. Biol.* **85**, 86–102 (2013)
2. Klingenberg, C.P.: Cranial integration and modularity: insights into evolution and development from morphometric data. *Hystrix Ital. J. Mammal.* **24**(1), 43–58 (2013)
3. Ryan, C.J., Roguev, A., Patrick, K., Xu, J., Jahari, H., Tong, Z., Beltrao, P., Shales, M., Qu, H., Collins, S.R., et al.: Hierarchical modularity and the evolution of genetic interactomes across species. *Mol. Cell* **46**(5), 691–704 (2012)
4. Lauring, A.S., Frydman, J., Andino, R.: The role of mutational robustness in RNA virus evolution. *Nat. Rev. Microbiol.* **11**(5), 327–336 (2013)
5. Rigato, E., Fusco, G.: Enhancing effect of phenotype mutational robustness on adaptation in *escherichia coli*. *J. Exp. Zool. Part B Mol. Dev. Evol.* **326**(1), 31–37 (2016)
6. Lehman, J., Stanley, K.O.: Evolvability is inevitable: increasing evolvability without the pressure to adapt. *PloS One* **8**(4), e62186 (2013)
7. Pigliucci, M.: Is evolvability evolvable? *Nat. Rev. Genet.* **9**(1), 75–82 (2008)
8. Li, J., Yuan, Z., Zhang, Z.: The cellular robustness by genetic redundancy in budding yeast. *PLoS Genet.* **6**(11), e1001187 (2010)
9. Tokuriki, N., Tawfik, D.S.: Protein dynamism and evolvability. *Science* **324**(5924), 203–207 (2009)
10. Pavlicev, M., Wagner, G.P.: A model of developmental evolution: selection, pleiotropy and compensation. *Trends Ecol. Evol.* **27**(6), 316–322 (2012)
11. Graves, C.J., Ros, V.I., Stevenson, B., Sniegowski, P.D., Brisson, D.: Natural selection promotes antigenic evolvability. *PLoS Pathog* **9**(11), e1003766 (2013)
12. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge (1998)
13. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Heidelberg (2013)
14. Droste, S., Jansen, T., Wegener, I.: On the analysis of the (1+1) evolutionary algorithm. *Theor. Comput. Sci.* **276**(1–2), 51–81 (2002)
15. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
16. Hamann, H., Schmickl, T., Crailsheim, K.: Coupled inverted pendulums: a benchmark for evolving decentral controllers in modular robotics. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pp. 195–202. ACM (2011)
17. Dowsland, K.A., Thompson, J.M.: Simulated annealing. In: *Handbook of Natural Computing*, pp. 1623–1655. Springer (2012)
18. Castro, L.N.D., Timmis, J.: *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer Science & Business Media (2002)
19. Schwab, J.D., Siegle, L., Kühlwein, S.D., Köhl, M., Kestler, H.A.: Stability of signaling pathways during aging—a boolean network approach. *Biology* **6**(4), 46 (2017)
20. Turner, A.P., Caves, L.S., Stepney, S., Tyrrell, A.M., Lones, M.A.: Artificial epigenetic networks: Automatic decomposition of dynamical control tasks using topological self-modification (2016)

21. Ribba, B., Grimm, H.P., Agoram, B., Davies, M.R., Gadkar, K., Niederer, S., van Riel, N., Timmis, J., van der Graaf, P.H.: Methodologies for quantitative systems pharmacology (QSP) models: design and estimation. *CPT Pharmacometrics Syst. Pharmacol.* **6**(8), 496–498 (2017)
22. Kirschner, M.: Beyond Darwin: evolvability and the generation of novelty. *BMC Biol.* **11**(1), 1 (2013)
23. Kitano, H.: Biological robustness. *Nat. Rev. Genet.* **5**(11), 826–837 (2004)
24. Palazzo, A.F., Gregory, T.R.: The case for junk DNA. *PLoS Genet.* **10**(5), e1004351 (2014)
25. Pennisi, E.: Encode project writes eulogy for junk DNA (2012)
26. Zotenko, E., Mestre, J., O’Leary, D.P., Przytycka, T.M.: Why do hubs in the yeast protein interaction network tend to be essential: reexamining the connection between the network topology and essentiality. *PLoS Comput. Biol.* **4**(8), e1000140 (2008)
27. Silva, F., Duarte, M., Correia, L., Oliveira, S.M., Christensen, A.L.: Open issues in evolutionary robotics. *Evol. Comput.* **24**(2), 205–236 (2016)
28. Valero, K.W.: Aligning functional network constraint to evolutionary outcomes. *bioRxiv* (2018)
29. Qin, C., Sun, Y., Dong, Y.: A new method for identifying essential proteins based on network topology properties and protein complexes. *PloS One* **11**(8), e0161042 (2016)
30. Pervouchine, D.D., Djebali, S., Breschi, A., Davis, C.A., Barja, P.P., Dobin, A., Tanzer, A., Lagarde, J., Zaleski, C., See, L.-H., et al.: Enhanced transcriptome maps from multiple mouse tissues reveal evolutionary constraint in gene expression. *Nat. Commun.* **6**, 5903 (2015)
31. Foote, A.D., Liu, Y., Thomas, G.W.C., Vinař, T., Alföldi, J., Deng, J., Dugan, S., van Elk, C.E., Hunter, M.E., Joshi, V., et al.: Convergent evolution of the genomes of marine mammals. *Nat. Genet.* **47**(3), 272 (2015)

Author Queries

Chapter 3

Query Refs.	Details Required	Author's response
AQ1	This is to inform you that corresponding author has been identified as per the information available in the Copyright form.	
AQ2	Per Springer style, both city and country names must be present in the affiliations. Accordingly, we have inserted the country name “UK” in affiliation. Please check and confirm if the inserted country name is correct. If not, please provide us with the correct country name.	
AQ3	References [1–6, 29] are given in the list but not cited in the text. Please cite them in text or delete them from the list.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

<i>Instruction to printer</i>	<i>Textual mark</i>	<i>Marginal mark</i>
Leave unchanged	... under matter to remain	Ⓟ
Insert in text the matter indicated in the margin	⧵	New matter followed by ⧵ or ⧵ [Ⓢ]
Delete	/ through single character, rule or underline or ⎯ through all characters to be deleted	⧻ or ⧻ [Ⓢ]
Substitute character or substitute part of one or more word(s)	/ through letter or ⎯ through characters	new character / or new characters /
Change to italics	— under matter to be changed	↵
Change to capitals	≡ under matter to be changed	≡
Change to small capitals	≡ under matter to be changed	≡
Change to bold type	~ under matter to be changed	~
Change to bold italic	≈ under matter to be changed	≈
Change to lower case	Encircle matter to be changed	≡
Change italic to upright type	(As above)	⧻
Change bold to non-bold type	(As above)	⧻
Insert 'superior' character	/ through character or ⧵ where required	Y or Y under character e.g. Y or Y
Insert 'inferior' character	(As above)	⧵ over character e.g. ⧵
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	Y or Y and/or Y or Y
Insert double quotation marks	(As above)	Y or Y and/or Y or Y
Insert hyphen	(As above)	⎯
Start new paragraph	┐	┐
No new paragraph	┐	┐
Transpose	┐	┐
Close up	linking ○ characters	○
Insert or substitute space between characters or words	/ through character or ⧵ where required	Y
Reduce space between characters or words		↑