BioSystems 176 (2019) 17–26

Contents lists available at ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

Evolutionary acquisition of complex traits in artificial epigenetic networks

Alexander Turner[a,*], Andy Tyrrell[b], Martin Trefzer[b], Michael Lones[c]

[a] Department of Computer Science, University of Hull, UK
[b] Department of Electronic Engineering, University of York, UK
[c] School of Mathematical and Computer Sciences, Heriot-Watt University, UK

ARTICLE INFO

Keywords:
Artificial gene regulatory networks
Evolutionary dynamics
Computational optimisation

ABSTRACT

How complex traits arise within organisms over evolutionary time is an important question that has relevance both to the understanding of biological systems and to the design of bio-inspired computing systems. This paper investigates the process of acquiring complex traits within epiNet, a recurrent connectionist architecture capable of adapting its topology during execution. Inspired by the biological processes of gene regulation and epigenetics, epiNet captures biological organisms' ability to alter their regulatory topologies according to environmental stimulus. By applying epiNet to a series of computational tasks, each requiring a range of complex behaviours to solve, and capturing the evolutionary process in detail, we can show not only how the physical structure of epiNet changed when acquiring complex traits, but also how these changes in physical structure affected its dynamic behaviour. This is facilitated by using a lightweight optimisation method which makes minor iterative changes to the network structure so that when complex traits emerge for the first time, a direct lineage can be observed detailing exactly how they evolved. From this we can build an understanding of how complex traits evolve and which regulatory environments best allow for the emergence of these complex traits, pointing us towards computational models that allow more swift and robust acquisition of complex traits when optimised in an evolutionary computing setting.

1. Introduction

Genetic networks are the fundamental systems through which biological cells regulate their function and development, and this realisation has promoted a sustained effort to understand genetic networks through computational modelling and simulation (Hasty et al., 2001; De Jong, 2002; Hecker et al., 2009; Chen et al., 2010; Le Novère, 2015; Akutsu, 2016). Genetic networks can be modelled in various ways, depending on how the model is to be used. For example, Boolean networks (Albert and Thakar, 2014) are a popular formalism for capturing the qualitative dynamics of genetic networks, and continuous-valued models such as recurrent neural networks (Ling et al., 2013) and systems of differential equations (Le Novère, 2015) provide greater insight into quantitative dynamics. Regulatory interactions happen at various spatial and temporal scales within genetic networks, for instance transcription pre-initiation, transcript elongation and RNA interference (Duncan et al., 2014). In all these models, a genetic network is represented as a graph. This gives them sufficient generality to capture most of these regulatory processes. However, they do assume that the structure of the graph remains fixed, and consequently cannot model regulatory processes that change the underlying topology of the genetic network.

An example of such a process, which we consider in this paper, is chromatin remodelling (Narlikar et al., 2013; Lorch and Kornberg, 2015; Zaret and Mango, 2016; Adam and Fuchs, 2016), an epigenetic process that regulates physical access to the genes, and in doing so effectively modifies the topology by turning on and off different parts of the genetic network. The important role that chromatin remodelling, and epigenetic processes more generally, play within biological systems has become increasingly apparent over the last decade (Duncan et al., 2014; Koster et al., 2015). It is known, for instance, that chromatin remodelling is central to the process of cellular differentiation (Kurimoto et al., 2015; Adam and Fuchs, 2016), and hence to the development of multicellular organisms. More generally, epigenetic processes are instrumental for the evolutionary acquisition of complex traits (True et al., 2004; Petronis, 2010). However, exactly how cell fates and complex traits are acquired remains unclear (Adam and Fuchs, 2016), suggesting a need for computational models that can capture and simulate the interplay between epigenetic processes and genetic networks.

Despite some recent progress (Zentner and Henikoff, 2014), experimental data regarding the chromatin dynamics of cells remains

[*] Corresponding author.
E-mail address: alexander.turner@hull.ac.uk (A. Turner).

https://doi.org/10.1016/j.biosystems.2018.12.001
Received 16 May 2018; Received in revised form 30 November 2018; Accepted 3 December 2018
Available online 14 December 2018

relatively sparse, of low temporal resolution, and challenging to link to other transcriptional dynamics data (Coulon et al., 2013). This limits the potential for building, and hence studying, models of particular epigenetic circuits. As an alternative approach, we might look to previous work on modelling genetic networks, where significant understanding has come about through consideration of their dynamics at a more abstract systems level, rather than through the study of specific genetic circuits (Albert and Thakar, 2014; Abou-Jaoudé et al., 2016). For example, work on the dynamics of random Boolean networks has given insight into the nature of the dynamical states that correspond to stable cell types (Huang et al., 2005, 2009). Given the development of suitable models at the epigenetic level, it is conceivable that similar studies could lead to insights into the role of epigenetic processes within the dynamics of genetic networks (Bull, 2012, 2014).

However, a limitation of abstract systems-level studies is the absence of realistic evolutionary pressure driving the need to acquire complex traits. To address this, we consider another group of modelling approaches which take a quite different approach, using evolutionary algorithms (or other metaheuristics) to optimise genetic network models so that they carry out designated computational behaviours (Banzhaf, 2003; Lones, 2016). These behaviours vary from the relatively simple, such as the implementation of logic functions (Bull and Preen, 2009), to computationally challenging, such as controlling the movements of robots through complex environments (Taylor, 2004; Trefzer et al., 2010; Joachimczak et al., 2012; Fuente et al., 2013; Sanchez and Cussat-Blanc, 2014). Problems such as the latter require the evolution of behaviours such as homeostatic control and robust pattern generation, i.e. the same kind of traits that have been acquired by biological systems. In general, these models have had a more significant focus on artificial intelligence where the emphasis is on understanding or instigating the emergence of complexity in computational systems. The genetic network models optimised by these approaches are available for full inspection, and it is also possible to record and inspect a complete phylogenetic tree, showing exactly when and how adaptive traits were acquired. This makes them potentially very useful in the study of how complexity emerges in biological systems.

In this work, we consider how higher order structures can be used in the acquisition of complex traits in computational models for problem solving. In particular, we use an artificial epigenetic network called *epiNet* (Turner et al., 2015), which allows modification of the genetic network's topology through a mechanism inspired by chromatin remodelling. Using a simple evolutionary algorithm, we study the evolution of this model's genetic and epigenetic elements when exposed to selective pressure. Selective pressure, in this case, is induced by selecting model instances based on their ability to solve a series of artificial problems that require the model to acquire complex traits such as maintaining homeostasis, orchestrating a series of changes in a complex environment, and robustly generating patterns at the same time. This work has its roots in computational and engineering problems, it serves to provide an understanding of why computational analogues of epigenetic structures can be useful from these perspectives. By showing this, and how epigenetic structures optimise themselves, the hope is that tools such as this can be used in future work to better understand both computational systems and the biological systems by which they were inspired.

The paper is organised as follows: Sections 2–4 cover relevant material on genetic networks, chromatin remodelling and the EpiNet architecture. Section 5 describes the optimisation algorithm and the computational tasks. Section 6 presents results and discussion. Section 7 concludes.

2. Background

The majority of genes encode messenger RNA, which in turn describe the amino-acid sequences of proteins in biological organisms. In this sense, a gene can be considered a section of DNA used to code for a biological molecule which has a particular function (Turner, 2008). In order for gene expression to occur, a gene has to be transcribed and translated using a cell's processing machinery. This processing machinery is the functional product of other genes within the cell; hence, the genes form regulatory networks, regulating one another's expression. These genetic networks exist over many scales, from self contained regulatory processes such as the lac operon to organism wide networks governing the activity and development of the entire organism (Göndör, 2016).

A genetic network can be seen as a graph, comprising a set of connected nodes where each connection has a weight used to define the strength of regulation that one node has upon another. Genetic networks are often modelled in this way. However, it should be noted that in reality these connections are transient and emerge from the stochastic physico-chemical interactions between numerous biomolecules. In this respect, genetic networks differ from another prominent biological network, the neural networks of the brain, where the nodes are physically connected via axons. This has a significant effect on the speed of information travel between genetic and neural networks within an organism. However, from an evolutionary perspective, it allows for the molecular interactions of genetic components to be less constrained and to explore interactions between a wider range of cellular components. This graphical representation highlights the similarities between genetic networks and biological neural networks. Indeed, there are many similarities: for example, in computational models it is commonplace for the nodes of both networks to be modelled as sigmoidal functions. However, there are also many differences. The principal difference is that there is no direct physical connection between the nodes in genetic networks. In biological neural networks, the nodes are physically connected, generally via axons; in genetic networks, the connections are transient and emerge from the physical–chemical interactions between numerous biomolecules. This has a significant effect on the speed of information travel between genetic and neural networks within an organism. However, from an evolutionary perspective, this allows for the molecular interactions of genetic components to be less constrained and to explore interactions between a wider range of cellular components.

Chromatin remodelling is a prominent example of this, and is a specific type of epigenetic process. Epigenetics refers to the study of cellular trait variations which occur as a result of factors which control gene expression (Turner, 2008; Göndör, 2016). Chromatin is the combination of structural proteins and DNA. The structural proteins within chromatin are called histones and are organised along with DNA into nucleosomes, and then through higher order folding into chromatin. Chromatin is capable of condensing the genetic material within a cell to the extent that 2m of DNA can fit inside a 2 μm nucleus. This was originally thought to be chromatin's principal purpose within a cell; however, in recent studies it has become clear that proteins can interact with the protein scaffolding of chromatin to change its structure (Turner, 2008; Göndör, 2016). This, in turn, changes which parts of the DNA are accessible within the protein complex, and facilitates regulatory control over gene expression. For example, given a set of environmental perturbations, a genetic network can modify itself to allow for the transcription of genes specifically designed to control these perturbations. This allows for a greater specific control over gene transcription, reducing the energy needs of the cell, and also reducing the scope for interference between biochemical processes.

3. Topological morphology

The original motivation for adding a chromatin remodelling analogue to an existing artificial gene regulatory network model (AGRNs) was that it would allow for topological self-modifications to occur throughout execution (Turner et al., 2017). In turn, this was expected to promote the emergence of complex behavioural traits, in a similar way to how chromatin remodelling within biological systems appears to allow cells to orchestrate more complex regulatory behaviours. This is a departure from standard forms of connectionist computation, where the structure of a network remains invariant during the course of execution. The intended effect of these topological modifications is to allow large changes in the dynamics of a network whilst it is executing. This is not something that has been explored within connectionist models, with the possible exception of models of neuromodulation (such as gas nets, Zhao et al., 2014) where smaller changes to the local dynamics can be induced by switching between different nodal functions.

In a sense, topological modification during execution provides a further layer of complexity when analysing a network's behaviour. In many applications this may not be an issue, since connectionist models are typically used as black boxes where only accuracy of the input–output mapping is a concern. However, in another sense, topological modifications provide extra information about what the network is doing, since changes in dynamics are mirrored by changes in topology, which can be readily observed (Turner et al., 2017). Moreover, from an evolutionary perspective, a single change in a single node can often lead to the acquisition, or removal of a dynamical regime within the network, vastly altering its functionality. In order to understand its behaviour, it is often required to track the dynamic topology of the network alongside its function. In this work, the emphasis is on understanding how modifications to the network during optimisation can lead to the acquisition of complex dynamical traits which in turn are capable of solving complex dynamical tasks.

4. EpiNet architecture

In the epiNet architecture, an analogue of chromatin (referred to more generically as an epigenetic molecule) is added to an existing AGRN, and a mechanism is introduced to allow other nodes to modify its activation state and positioning. The purpose of the epigenetic molecules is to control which genes from the AGRN are active, and hence contributing to the network's dynamics, at any given time (Fig. 1). The underlying AGRN architecture is similar to a recurrent neural network.

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

G is a set of genes $\{n_0 \ldots n_{|N|} : n_i = \langle a_i, I_i, W_i\rangle\}$ where:
a_i: 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 $0 \le w_i \le 1$, $|W_i| = |I_i|$.
L is a set of initial activation levels, where $|L_N| = |N|$.
$In \subset G$ is the set of genes used as external inputs.
$Out \subset G$ is the set of genes used as external outputs.

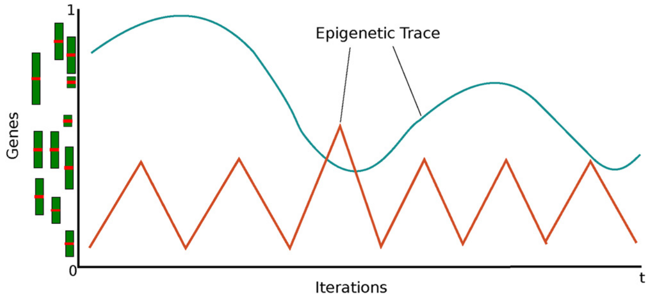

Fig. 1. A representation of epiNet executing over a set number of iterations. On the left of the figure, the genes can be seen. The genes remain statically positioned throughout execution. At each iteration, the epigenetic molecules take inputs from the genes and update their positions. The genes which are then selected to be executed are the ones closest in proximity to the epigenetic molecule on the *y* axis. The epigenetic trace shows the position of the epigenetic molecule over multiple time steps.

The architecture of epiNet can be defined by the tuple $\langle N, S, L, In, Out, A\rangle$, where:

E is a set of epigenetic molecules $\{s_0 \ldots e_{|S|} : e_i = \langle e_i, I_i, W_i, C_i\rangle\}$:
a^i: ℝ is the position of the epigenetic molecule.
$I_i^e \subseteq G$ is the set of inputs to the molecule.
W_i^e is a set of weights, where $0 \le w_i \le 1$, $|W_i| = |I_i|$.
$C_i^e \subseteq G$ is the set of genes controlled by the switch.
A is a set containing all active genes.

The genes within the network are invariant; however, their involvement within the network's dynamics at any particular time is determined by the epigenetic molecules, whose behaviour is akin to the local unwinding of DNA, allowing genes to become accessible. Specifically, genes become active when they are within a given distance of an epigenetic molecule, where proximity is determined using a Euclidean distance metric within a reference space (defined in Section 4.1). Furthermore, the epigenetic molecules are able to move around the network, covering and uncovering different groups of nodes as they do so, with the current position of an epigenetic molecule governed by the dynamical states of genes within the network, through the use of a weighted Sigmoid function. This epigenetic dynamism more closely reflects the biological dynamics of chromatin modification than earlier models which we have used (Turner et al., 2012, 2017). It also differentiates our approach from work by Bull (2012, 2014), who used an evolutionary algorithm to study the effect of adding epigenetic elements to a GRN model within abstract NK landscapes.

4.1. Encoding

During evolution, the connections I_i, I_i^s and C_i^s between components within epiNet are determined via an indirect encoding. Specifically, components are given locations within an indirect reference space. This is based upon earlier works in the AGRNs field (Reil, 1999; Banzhaf, 2003), which were motivated by the manner in which biological components interact through physical and chemical properties rather than their exact location within a DNA encoding. This means that interactions within the network are positionally independent, with a gene functioning identically regardless of where it occurs within the list of genes. In particular, this allows for the preservation of existing gene functions when other genes are introduced or removed from the network, adding to the evolvability of the networks.

A connection is specified using both a position and a proximity. Genes can be considered connected to each other when their position ± their proximity overlaps another gene's position. Epigenetic molecules exist within the same space as genes. Each epigenetic molecule has a defined extent within the reference space which it uses to connect to active genes. Using the expression values of these genes, it processes their weighted sum and its position is the result of that sum. The position of the epigenetic molecule specifies a region (its position ± its proximity) within the reference space where all genes within that region will become active. System-level inputs are mapped onto the *active* genes before execution and the outputs are mapped back after execution.

5. Optimisation

In this work, we choose to work with a mutation only hill climbing heuristic, which is similar to an evolutionary strategy (Dong and Zhou, 2014). This is because we want to be able to precisely control the level of change at each optimisation step so that the evolutionary process can be accurately observed. Additionally we want to be able to focus on a single network rather than a population so that we can precisely monitor its development. Although objective performance of the algorithm may change using a mutation only heuristic, the outright performance of the networks is not a key focus of this work. The amount of mutation applied to the networks at each step is 5%. This means that for every instance of data representing and parameterising the networks, there is a 5% chance that that data instance will be mutated. The individual data within each of the genes and epigenetic structures are mutated using a Gaussian distribution, with its previous values set at the mean, which is a fairly standard approach for mutating real-valued encodings within the context of an evolutionary algorithm (and is also a reasonable model of normally-distributed mutations within a biological context).

Algorithm 1. Optimising epiNet

```
P ←new random epiNet
for $numberofevaluations$ do
    CLONE P AS Q
    MUTATE(Q)
    EVALUATE(Q)                              ▷ see Algorithm 2
    if Q.fitness >= P.fitness then
        P = Q
    end if
end for
```

Algorithm 2. Evaluating epiNet on a task

```
initialize control task
$a \leftarrow L$                                ▷ initialize epiNet state
repeat
    $cout \leftarrow$ state variables from controlled system
    In ← SCALE($cout$)                   ▷ scale inputs to $[0, 1]$
    for $i \in \{0, \dots, |E|\}$ do           ▷ update positions of epigenetic molecules
        $a_i^s \leftarrow$ SIGMOID($I_i^s \cdot W_i^s$)       ▷ modify epigenetic positions
    end for
    for $i \in \{0, \dots, |G|\}$ do     ▷ update genes closest to the epigenetic positions
        $a_i \leftarrow$ SIGMOID($I_i \cdot W_i$)
    end for
    $cin \leftarrow$ SCALE(Out)              ▷ scale outputs to range
    modify controlled system according to $cin$
until control task finished or timed-out
$fitness \leftarrow$ progress on control task objectives
```

5.1. Computational tasks

To best understand the emergence of complexity within the networks, we apply epiNet to three computational tasks: a coupled inverted pendulums control task (Hamann et al., 2011), a transfer orbit traversal task and a memory task. Each requires different dynamical properties to solve. These tasks are intended both to highlight the emergence of complex traits within the networks as well as being challenging enough to validate the model as a computational system. Additionally, the underlying AGRN model is applied to each of these tasks, so that the impact of adding the epigenetic layer can be measured.

5.1.1. Coupled inverted pendulums

The coupled inverted pendulums control task was designed as a proxy for a range of real world control tasks such as robotic control for legged robots (Hamann et al., 2011). It was designed to be able to test controllers in an environment that produces a range of complex behaviours, where is it difficult for controllers to encompass all the behaviours required to optimally solve the task.

The task consists of three carts on a 1-dimensional track, with a pendulum hanging below each cart. The objective is to move the carts on the track in such a way as to move the pendulums vertically above the cart, and keep them in that position. Additionally, the carts are inelastically tethered together so that all the carts' movements must cooperate to solve the task. If a tether is extended too far, the simulation stops. If carts hit one another, or leave the set boundaries, the simulation stops.

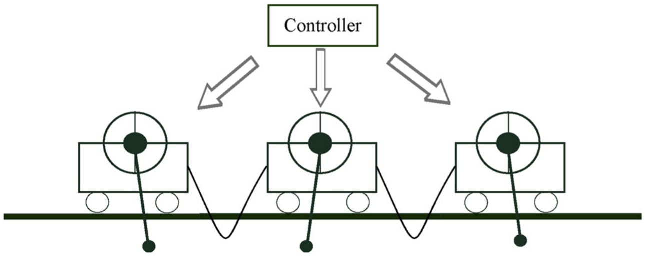

Fig. 2. Pendulum task. The objective of the task is to maneuver the carts from left to right in such a way as that the pendulums move from a downward position to an upright position and are maintained there. The carts are joined together so their movements are limited, and must be coordinated. Each cart has its own separate controller.

Each cart is controlled using an actuator which takes the difference between two inputs, allowing it to move towards or away from its neighbours (see Fig. 2). There is a single controller per cart, which is passed 10 state inputs, listed in Table 1. The controller produces 2 outputs which control the actuator of the cart. The overall fitness of the controller is defined as how many time steps each pendulum spends in the upright position. The parameters of the simulation are given in Tables 1 and 2. The simulation is conducted over 100,000 iterations, and 50 runs. To improve the realism of the simulation, there is a stochastic noise function attached to the state variables. This noise is randomly sampled from a normal distribution. To provide a more robust measure of fitness, each controller is evaluated 5 times. A mutant is only considered better, and therefore replaces its parent, if the mean and best scores over these evaluations are an improvement.

5.1.2. Multi-point traversal through an n-body system

In the second task, we consider the control of a multi-point traversal through an n-body system. The objective of this task is to guide a rocket's trajectory towards a fixed point, then change orientation in order to land as close as possible to a given location on a planetary surface, and have as low a velocity as possible upon landing. The fitness of a controller is calculated by equally weighting the distance from the target and the final velocity. The planet is of large enough size to have a significant gravitational effect on the dynamics of the rocket at all points. This simulation is time constrained; in order to be able to land on the planet, the first objective must be completed within a reasonable time frame. The simulation is conducted over 100,000 iterations and 50 runs. The input data for this task can be seen in Table 3.

Once the first stage of the task is completed, a fault is injected which

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

ID	Sensor name	System to sensor mapping
S_0	Pendulum Angle 0	$\o \in [0, 0.5\pi] \rightarrow [127, 0]$, 0 else
S_1	Pendulum Angle 1	$\o \in [1.5\pi, 2\pi] \rightarrow [0, 127]$, 0 else
S_2	Pendulum Angle 2	$\o \in [0.5\pi, \pi] \rightarrow [127, 0]$, 0 else
S_3	Pendulum Angle 3	$\o \in [\pi, 1.5\pi] \rightarrow [0, 127]$, 0 else
S_4	Proximity 0	Distance left → [0,127]
S_5	Proximity 1	Distance right → [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	$2(A_0/127 - A_1/127) \rightarrow [0, 1]$

Table 2
Physical parameters of the coupled inverted pendulums task.

Parameter	Value
Pendulum length	0.5 m
Max. positive acceleration	7.0 ms^{-2}
Min. positive acceleration	8.5 ms^{-2}
World width	2 m
Tether length	0.35 m
Proximity sensor range	1.0 m
Cart width	0.1 m
Time steps (t)	3000

Table 3
Physical parameters of the multi-point traversal task.

Parameter	Value
Starting Rocket Position	0; 6,571,000; 10,000
First Target Position	−700,000; 6,671,000; −20,000
Planet position	0;0;0
Planet mass	5.972E24 kg
Planet radius	6,371,000 m
Rocket Mass	2000 kg
Rocket Acceleration	(x ± 50 ms^{-2}) (y ± 50 ms^{-2}) (z ± 50 ms^{-2})
Time steps	5000 per target
Integration Step	0.05

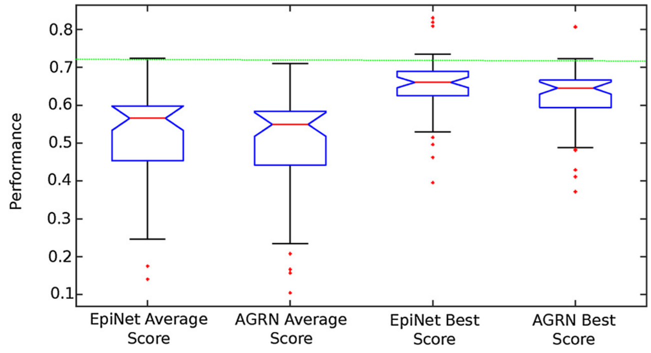

Fig. 3. Best and average results of epiNet against the AGRN for the pendulum task. The horizontal green line denotes the point at which the optimum behaviour is achieved.

Table 4
The average optimisation steps refers to how many positive mutations occurred from start to finish of the task over all networks, regardless of how well the task was performed. The average mutation size specifies how many changes on average were made to the network for each positive optimisation step. What can be seen is that for both networks, the n-body task had the most mutations both in frequency and the amount of changes per step, with the memory task having the least. In terms of the average mutation size, there is little difference between the AGRN and epiNet. However, for the average optimisation steps, for both the pendulum and n-body task, epiNet required less mutations *and* performed better than the AGRN (show figures). In the memory task, the AGRN had slightly fewer optimisation steps than epiNet, but again, epiNet outperformed the AGRN.

Task	Total average optimisation steps		Average mutation size	
	AGRN	EpiNet	AGRN	EpiNet
Pendulums	453.46	322.34	5.78	6.18
N-body	7249.19	3692.14	21.16	21.54
Memory	42.04	51.71	5.24	5.47

reduces the power to 10% on the y thruster. Hence, as well as navigating to the target, the controller has to react to a change in how the rocket interacts with the environment. The forces exerted on the spacecraft are calculated using Eq. (1), where m is the mass of a body and q is a vector of length 3 specifying the position of an object in 3-dimensional space. To improve the tractability of the system, the planet's position remains static. The acceleration is then calculated using Newton's second law of motion. The equations are simulated using leapfrog integration.

$$m_j q_j = G \sum_{k \neq j} \frac{m_j m_k (q_k - q_j)}{|q_k - q_j|3} \tag{1}$$

An evolved controller receives 5 inputs: a vector representing the target position, the admittance to the target and the rocket's speed. The controller is required to generate 3 outputs, which correspond to the power of the rocket thrusters in each dimension.

5.1.3. Network sequence memory

The final task is a sequence learning task that tests the memory and recall capacity of the network architectures, and particularly how the epigenetic layer effects a network's ability to encode new knowledge whilst preserving its existing dynamics. The objective is to recall as many Boolean values as possible from a sequence of 50 values. The networks do not take any inputs, relying on their internal dynamics to generate the appropriate sequence of output states. The single output of a network is mapped to [0, 1]; if the value is less than 0.5, its output state is false, otherwise it is true. Fitness is measured by the edit distance from the target sequence. The simulation is conducted over 100,000 iterations and 200 runs (since this task is much faster to evaluate than the previous two). A new target sequence is randomly generated at the start of each run.

6. Results and discussion

For the two control tasks, the objective is to optimise an epiNet instance so that it functions as a closed loop controller which is capable of guiding the dynamics of the simulation in a specified manner. For each time step of the simulation, its state is fed into the network by setting the activation levels of the input genes. The network then executes, generating one or more outputs which are then mapped back to the simulation. For the memory task, the epiNet instance functions as an open-loop system, generating states which are then fed into the fitness function to evaluate. In both cases, the plots used to describe the behaviour of the networks contain two components. First, the epigenetic trace specifies the location of the epigenetic molecules over a given number of time steps within the reference space. As the epigenetic molecules move, the genes which are active also change. Hence, modification of the epigenetic position results in changes to the topology of the network. The second component describes the dynamics of the network by plotting the expression of each of the genes over a given time frame.

6.1. Coupled inverted pendulums

To solve this task optimally requires multiple dynamical behaviours: each pendulum has to be swung into an upright position; each cart's controller must cooperate with its neighbours to achieve this; once in the upright position, the controller must keep it there. The transition between swinging and stabilising is an important behavioural inflection point, and in this context is considered a complex trait (many algorithms failed to reach this point, Hamann et al., 2011). Hence, we are particularly interested in whether, and to what degree, the epigenetic layer contributes to its acquisition. In the results for this task (shown in Fig. 3), this optimal behaviour occurs when the fitness is above 0.715. Although objective performance is not a key concern in this work, it can

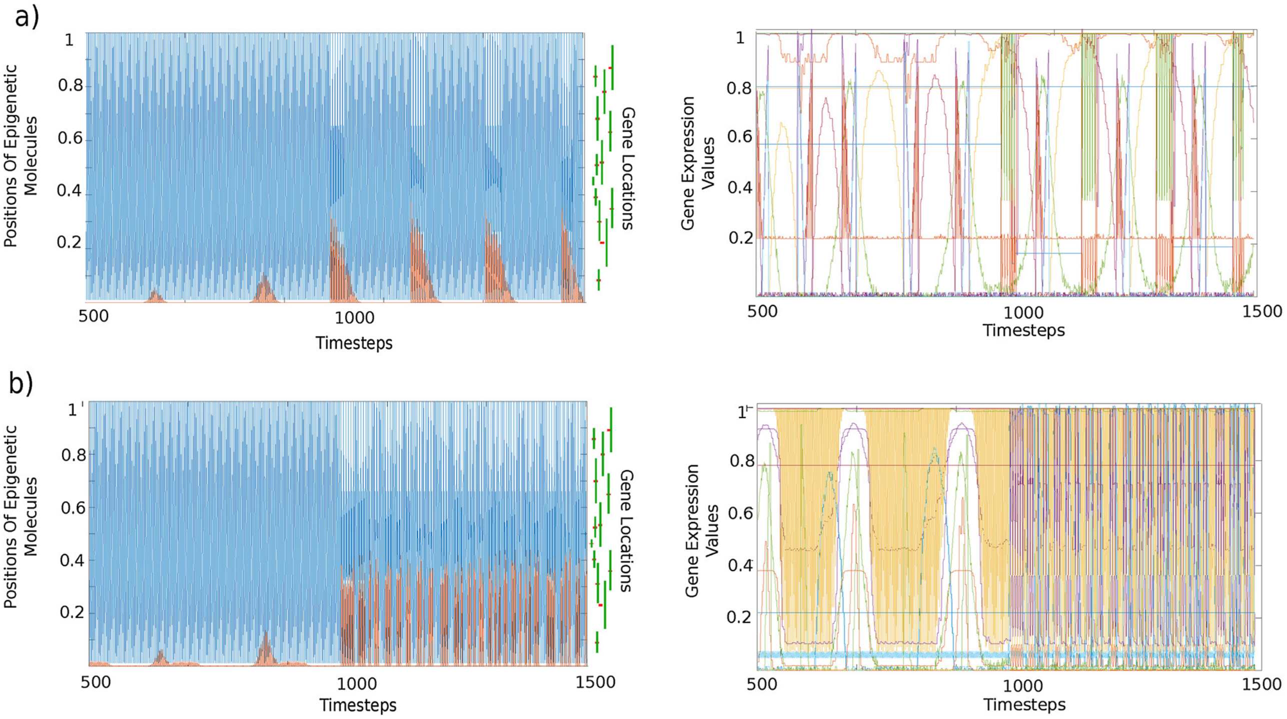

Fig. 4. The dynamics of epiNet before and after the optimum behaviour was acquired. Figure a shows the position of the epigenetic molecules within epiNet and the expression of every gene over 1500 time steps of the coupled pendulum simulation before the optimum behaviour was acquired. Figure b shows the same but after the optimum behaviour was acquired. The positions of the genes in reference to the epigenetic molecules can be seen to the right of the plots. In this instance the optimum behaviour was acquired by a mutation to the positions within the reference space of 2 genes, mutations to the regulatory functions of a further 3 genes, and a mutation to the regulatory function of one of the epigenetic molecules. Prior to the pendulums being in the upright position (time steps 500-900), these mutations do not produce a pronounced change in the dynamics of the epigenetic molecules, but a significant change is reflected in the gene expression values. After 900 time steps, the dynamics of the epigenetic molecules and gene expression values show pronounced change. In particular, an epigenetic molecule oscillates at a high frequency only when the pendulum is in the upright position, producing a behaviour capable of keeping it there.

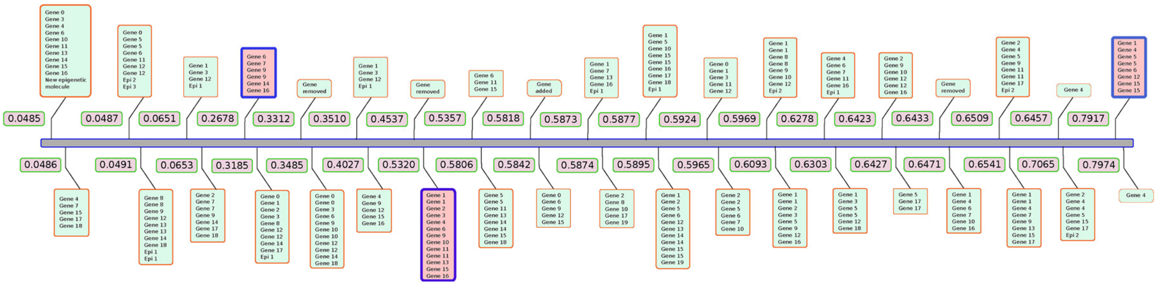
Gene 0 Gene 3 Gene 4 Gene 6 Gene 10 Gene 11 Gene 13 Gene 14 Gene 15 Gene 16 New epigenetic molecule
Gene 0 Gene 5 Gene 5 Gene 6 Gene 11 Gene 12 Gene 12 Epi 2 Epi 3
Gene 1 Gene 3 Gene 12 Epi 1
Gene 6 Gene 7 Gene 9 Gene 9 Gene 14 Gene 16
Gene removed
Gene 1 Gene 3 Gene 12 Epi 1
Gene removed
Gene 6 Gene 11 Gene 15
Gene added
Gene 1 Gene 7 Gene 13 Gene 16 Epi 1
Gene 1 Gene 5 Gene 10 Gene 15 Gene 15 Gene 16 Gene 17 Gene 18 Epi 1
Gene 0 Gene 1 Gene 3 Gene 11 Gene 12
Gene 1 Gene 1 Gene 8 Gene 8 Gene 9 Gene 10 Gene 12 Epi 2
Gene 4 Gene 6 Gene 7 Gene 11 Gene 16 Epi 1
Gene 2 Gene 9 Gene 10 Gene 12 Gene 12 Gene 16
Gene removed
Gene 2 Gene 4 Gene 5 Gene 9 Gene 11 Gene 11 Gene 17 Epi 2
Gene 4
Gene 1 Gene 4 Gene 5 Gene 5 Gene 6 Gene 12 Gene 15 Gene 15
0.0485
0.0487
0.0651
0.2678
0.3312
0.3510
0.4537
0.5357
0.5818
0.5873
0.5877
0.5924
0.5969
0.6278
0.6423
0.6433
0.6509
0.6457
0.7917
0.0486
0.0491
0.0653
0.3185
0.3485
0.4027
0.5320
0.5806
0.5842
0.5874
0.5895
0.5965
0.6093
0.6303
0.6427
0.6471
0.6541
0.7065
0.7974
Gene 4 Gene 7 Gene 15 Gene 17 Gene 18
Gene 8 Gene 8 Gene 9 Gene 12 Gene 13 Gene 13 Gene 14 Gene 18 Epi 1 Epi 1
Gene 2 Gene 7 Gene 7 Gene 9 Gene 14 Gene 17 Gene 18
Gene 0 Gene 1 Gene 2 Gene 3 Gene 8 Gene 12 Gene 12 Gene 14 Gene 17 Epi 1
Gene 0 Gene 0 Gene 3 Gene 6 Gene 9 Gene 10 Gene 10 Gene 12 Gene 12 Gene 14 Gene 18
Gene 4 Gene 9 Gene 12 Gene 15 Gene 16
Gene 1 Gene 1 Gene 2 Gene 3 Gene 4 Gene 6 Gene 9 Gene 10 Gene 11 Gene 11 Gene 13 Gene 15 Gene 16
Gene 5 Gene 5 Gene 11 Gene 13 Gene 14 Gene 15 Gene 18
Gene 0 Gene 6 Gene 9 Gene 12 Gene 15
Gene 2 Gene 8 Gene 10 Gene 17 Gene 19
Gene 1 Gene 2 Gene 5 Gene 6 Gene 12 Gene 13 Gene 14 Gene 14 Gene 15 Gene 15 Gene 19
Gene 2 Gene 2 Gene 5 Gene 6 Gene 7 Gene 10
Gene 1 Gene 1 Gene 2 Gene 3 Gene 5 Gene 9 Gene 12 Gene 16
Gene 1 Gene 3 Gene 5 Gene 5 Gene 12 Gene 18
Gene 5 Gene 17 Gene 17
Gene 1 Gene 4 Gene 6 Gene 7 Gene 10 Gene 16
Gene 1 Gene 1 Gene 4 Gene 7 Gene 9 Gene 13 Gene 15 Gene 17
Gene 2 Gene 4 Gene 4 Gene 5 Gene 15 Gene 17 Epi 2
Gene 4

Fig. 5. Evolutionary pathway from random initialisation to the acquisition of complex behaviours. Every positive mutation detailed through the evolutionary process. Each box details the mutations required to improve on the previous instance, along with the score of that instance. Each mutation is listed as a mutation to a specific gene. The actual data which is mutated is not listed, but was not specific to a type of data within the gene. There are three complex behaviours which can be seen (pink boxes, blue borders). The first complex behaviour is being able to swing the carts consistently. The second is swinging the carts with significantly more force, so that the pendulums are spinning around each cart. The third is being able to catch the pendulums in the upright state and balance them there.

be seen from the figure that epiNet outperforms the baseline model in the coupled inverted pendulum task. The results are significant, with 8 epiNet instances able to generate an optimum balancing behaviour compared to only 1 AGRN instance.

Table 4 summarises the average number and sizes of positive mutations that took place during the evolution of a controller, showing information for each task and architecture. For this task, it is notable that evolved epiNet controllers underwent on average 29% the number of positive mutations that AGRN controllers underwent, whilst at the same time achieving a higher average fitness. Given that the size of the mutations were similar for both architectures (last column in Table 4), this suggests that the positive mutations for epiNet led to more significant changes in behaviour. A likely explanation for this is that the positive mutations are causing (either directly or indirectly) changes to the epigenetic dynamics of the system, i.e. causing topological modifications that result in larger behavioural changes.

An example of this can be seen in Fig. 4, which shows the dynamics of an evolved network before and after a mutation led to the acquisition of the optimum behaviour. The figure shows the positions of the epigenetic molecules and the expression values of every gene. The plots detail 1500 of the 3000 time steps of the task, and cover the transition between swinging the pendulum and balancing it. It can be seen by looking at the gene locations (shown to the right of the epigenetic dynamics plots) that there is a clustering of genes generally at the higher and lower positions of the reference space. When looking at the positions of the epigenetic molecules throughout execution, it is clear that one of them moves throughout the entirety of the reference space. The second only moves within the range [0, 0.43]. Before the swinging

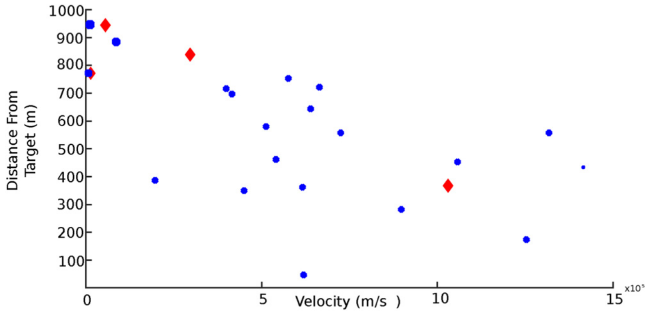

Fig. 6. Multi-point traversal through an n-body system. The performance of all controllers which were capable of navigating to both targets. The red diamonds represent AGRN controllers, and the blue circles represent epiNet controllers. The best performances was achieved by epiNet in both the velocity upon landing and the distance from the target.

behaviour is achieved, the second molecule only moves between 0 and 0.13. However, as soon as the swinging behaviour is achieved, the network (b) with the balancing behaviour shows abrupt rapid movements between 0 and approximately 0.4. These epigenetic dynamics correspond with rapid selection of varying genes which generate the required behaviour to keep the pendulum in the upright position, and lead to significant changes in the dynamics of gene expression. Before the final mutation, these epigenetic dynamics did not occur, and the network was not able to produce the optimum behaviour.

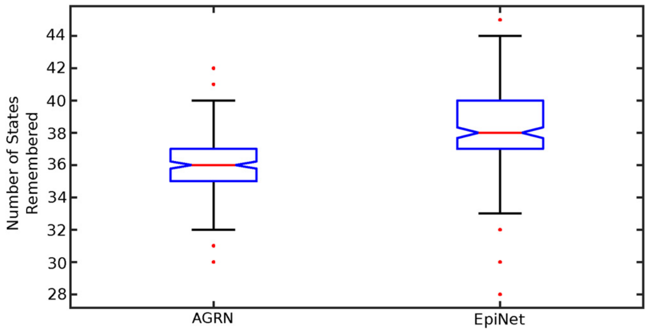

Fig. 8. Memory results. The difference in performance between epiNet and the AGRN when applied to remembering Boolean states. The differences are statistically significant and indicative that epiNet is able to hold more memory states than that of the AGRN.

Fig. 5 shows a full evolutionary pathway from random initialisation to the acquisition of complex behaviours. This emphasises that positive mutations were generally the result of multiple synchronised changes to the network. However, most of these changes are to genes rather than epigenetic elements, and where epigenetic elements are targeted, there is rarely more than one mutated at the same time. The fact that most mutations are genetic suggests that changes to the epigenetic dynamics

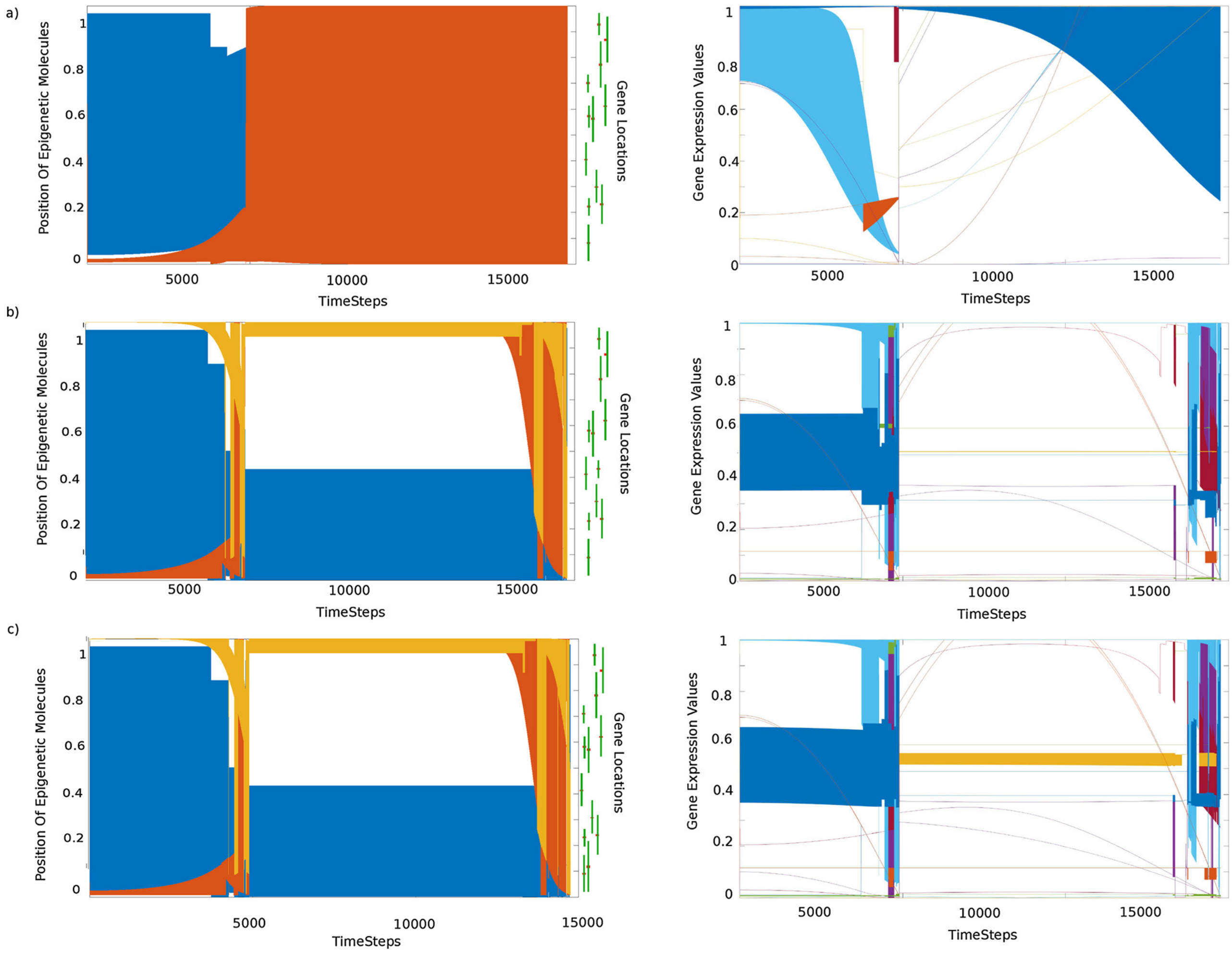

Fig. 7. N-body dynamics. The dynamics of 3 epiNet controllers from the same optimisation lineage. The first (a) shows networks capable of navigating to the first point, but not landing on the planet. (b) shows the epiNet controller the instance before it was capable of navigating to the first point and landing on the planet, and (c) is a controller capable of navigating to both points and landing on the planet.

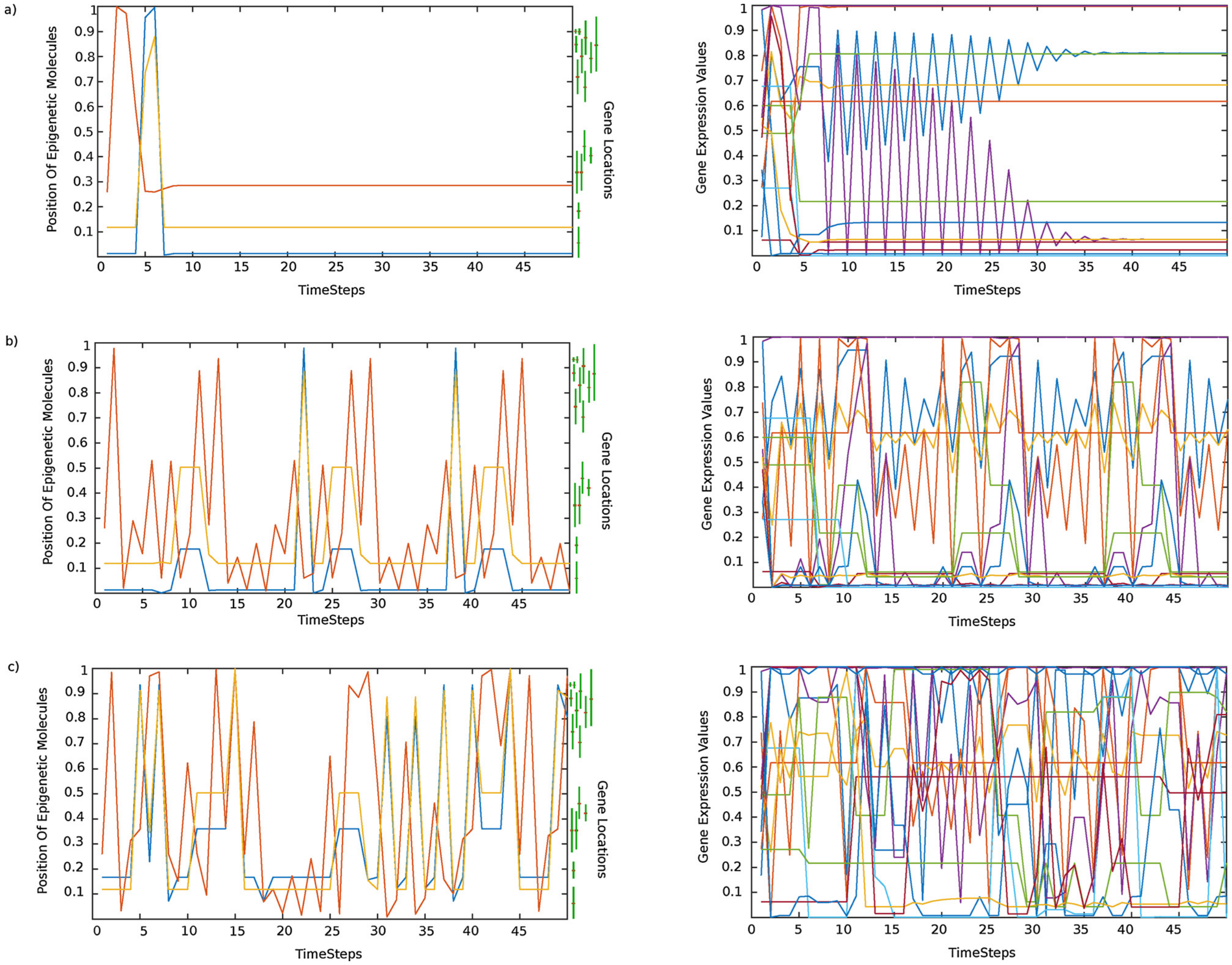

Fig. 9. Memory Dynamics. The dynamics associated with epiNets of the same optimisation lineage which are capable of remembering (a) 30 Boolean states, (b) 40 Boolean states and (c) 45 Boolean states respectively. The differences between networks (b) and (c) are quite subtle. Network (b) appears to have periodic repeating sets of epigenetic positions, which almost exactly translates to repeating dynamics of the network as seen by the changes in gene expression, especially after time step 10. After 8 mutations in network (c), the dynamics are less periodic and do not have a recognisable pattern, and are able to correctly recall 5 more states than network (b).

are generated indirectly through changes to the genes that regulate them. This makes sense, given that direct changes to epigenetic elements would likely to lead to comparatively large changes in the network's behaviour.

Although there was no general trend regarding the type of mutation that caused the optimal behaviour to first appear within the evolved epiNet controllers, it always followed from a modification to an existing gene or epigenetic molecule rather than through the addition or deletion of either of these. However, all optimal networks underwent several genetic or epigenetic deletions or insertions before the optimum behaviour was acquired. The smallest number of optimisation steps from random initialisation to optimum behaviour was 27, with 7 modifications resulting in the acquisition of the optimum behaviour. The smallest number of mutation events required to acquire the optimum behaviour was 3. The average mutation size, that is, the average number of simultaneous modifications which resulted in an improvement of the network, was 6.18.

6.2. Multi-point traversal through an n-body system

This task measures a network's ability to control a trajectory between multiple points whilst responding to a changing environment, again requiring it to switch between dynamical regimes during the course of execution. Although not all runs led to acquisition of the target behaviours, 20 epiNet controllers were successfully evolved, compared to only 4 AGRNs. Fig. 6 shows the behavioural characteristics of these successful controllers, showing that in general the epiNet controllers achieve a better balance between target error and minimising final velocity.

Similar to the previous task, Table 4 shows that epiNet controllers underwent considerably less positive mutations that the AGRNs in order to reach their final behaviour. Again, this suggests that epigenetic changes play an important role in the acquisition of the behaviours required to solve this task. Fig. 7 shows an example of the change in epigenetic and gene expression dynamics as an evolving network acquires increasingly complex behaviour. In particular, a large change in epigenetic dynamics can be seen between (a) and (b), and this appears to set the scene for a very subtle change in epigenetic dynamics (and a more significant change in gene expression) that led to the acquisition of the optimal behaviour in (c).

It is interesting to observe that the number of simultaneous changes that occur during positive mutations is relatively high for this task (Table 4) in comparison to the other two. This is not due to differences in the sizes of the networks, so is presumably a characteristic of the problem being solved. A possible explanation is that the fitness function is non-continuous, in that controllers that do not achieve certain behaviours can receive zero fitness. However, it is also worth considering that many of the component changes will be neutral and not have an effect upon fitness, so the actual number of co-occurring changes required may be much lower than this figure suggests.

6.3. Network sequence memory

This task is quite different to the previous two. Rather than moving between a small number of dynamical regimes in a context-sensitive manner, this task requires the dynamics to generate a large number of expression states in a fixed sequence, meaning that the acquisition of traits is a more gradual process. Also, there are no external inputs, so the dynamics must be created and sustained internally.

As shown in Fig. 8, epiNet solutions again outperformed the baseline AGRNs, recalling significantly more states and finding better solutions overall. Notably, this was the one task in which epiNet solutions underwent more positive mutations than AGRNs during their evolution, although this is likely to be offset somewhat by the fact that epiNet controllers evolved more complex behaviours on average. Despite this, changes in epigenetic dynamics still appear to be important in the development of complex traits: Fig. 9, for example, shows that the epigenetic dynamics become increasingly complex as the fitness of a solution increases. It was common for the dynamics of AGRN solutions in particular to settle into an attractor (similar to Fig. 9a) and this hints that topological changes play an important role in solving this task by maintaining complex, constantly changing, dynamics. This may explain the different mutational pattern seen in this task, with the epigenetic layer playing a general role in stimulating dynamics rather than switching between behaviours.

7. Conclusions

In this paper we have investigated how complex behaviours arise within EpiNet, a form of artificial genetic network that captures the important role of biological epigenetic processes such as chromatin modification, allowing for dynamical topological modification during execution. Using a simple optimisation algorithm, we studied how EpiNet instances evolve over time, identifying when and how they acquire the complex traits required to solve three different challenging computational tasks. Although not the focus of this work, EpiNet was shown to outperform a standard artificial genetic network on all tasks, showing the important role that epigenetic elements play within the acquisition of complex traits.

In this work, we focused on observing the points of evolutionary optimisation just before and after complex traits were acquired within the networks, and the underlying causes of this increase in behavioural complexity. One of the significant findings was that it was often not a single mutation which caused a complex behaviour to arise, but rather a collection of mutations occurring at the same time. These behaviours almost always emerged as a result of mutating existing elements of the model, rather than adding or removing genes or epigenetic molecules. It was, however, common for genes and epigenetic molecules to be added and removed throughout the optimisation process, but not at the point a complex behaviour arose.

Our results suggest that the epigenetic components of epiNet play an instrumental role in reducing the amount of optimisation effort required to acquire complex traits. This has clear implications for the field of artificial genetic networks, and demonstrates the benefits of modelling regulatory processes other than direct transcriptional regulation. Additionally these results, which are underpinned by biological theory, give support to the idea that work such as this has the potential be used in the future to inform evolutionary theory. If computational epigenetic structures allow for the faster acquisition of complex traits, could the same be said for biological models? Abstract level studies of epigenetic processes such as this could also play an important role in our understanding of the role of epigenetic processes in biological regulatory systems, which are very difficult to study directly due to challenges such as data sparsity.

Acknowledgment

The authors acknowledge the support of the EPSRC through the platform grant EP/K040820/1.

References

Abou-Jaoudé, W., Traynard, P., Monteiro, P.T., Saez Rodriguez, J., Helikar, T., Thieffry, D., Chaouiya, C., 2016. Logical modeling and dynamical analysis of cellular networks. Front. Genet. 7, 94.

Adam, R.C., Fuchs, E., 2016. The Yin and Yang of chromatin dynamics in stem cell fate selection. Trends Genet. 32 (2), 89–100.

Akutsu, T., 2016. Mathematical models and computational methods for inference of genetic networks. Evolutionary Computation in Gene Regulatory Network Research. pp. 30.

Albert, R., Thakar, J., 2014. Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions. Wiley Interdiscip. Rev.: Syst. Biol. Med. 6 (5), 353–369.

Banzhaf, W., 2003. On the dynamics of an artificial regulatory network. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J. (Eds.), Advances in Artificial Life, vol. 2801 of Lecture Notes in Computer Science. Springer, pp. 217–227 (Chapter 24).

Bull, L., Preen, R., 2009. On dynamical genetic programming: random Boolean networks in learning classifier systems. In: Vanneschi, L., Gustafson, S., Moraglio, A., De Falco, I., Ebner, M. (Eds.), Genetic Programming, vol. 5481 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 37–48.

Bull, L., 2012. Evolving Boolean networks with structural dynamism. Artif. Life 18 (4), 385–397.

Bull, L., 2014. Evolving Boolean regulatory networks with epigenetic control. Biosystems 116, 36–42.

Chen, L., Wang, R., Li, C., Aihara, K., 2010. Modeling Biomolecular Networks in Cells: Structures and Dynamics. Springer Science & Business Media.

Coulon, A., Chow, C.C., Singer, R.H., Larson, D.R., 2013. Eukaryotic transcriptional dynamics: from single molecules to cell populations. Nat. Rev. Genet. 14 (8), 572–584.

De Jong, H., 2002. Modeling and simulation of genetic regulatory systems: a literature review. J. Comput. Biol. 9 (1), 67–103.

Dong, W., Zhou, M., 2014. Gaussian classifier-based evolutionary strategy for multimodal optimization. IEEE Trans. Neural Netw. Learn. Syst. 25 (6), 1200–1216.

Duncan, E.J., Gluckman, P.D., Dearden, P.K., 2014. Epigenetics, plasticity, and evolution: how do we link epigenetic change to phenotype? J. Exp. Zool. Part B: Mol. Dev. Evol. 322 (4), 208–220.

Fuente, L.A., Lones, M.A., Turner, A.P., Caves, L.S., Stepney, S., Tyrrell, A.M., 2013. Adaptive robotic gait control using coupled artificial signalling networks, hopf oscillators and inverse kinematics. In: 2013 IEEE Congress on Evolutionary Computation (CEC). IEEE. pp. 1435–1442.

Göndör, A., 2016. Chromatin Regulation and Dynamics. Elsevier Science.

Hamann, H., Schmickl, T., Crailsheim, K., 2011. Coupled inverted pendulums: a benchmark for evolving decentral controllers in modular robotics. In: Proc. 13th Annual Conf. Genetic and Evolutionary Computation, GECCO'11. Dublin, Ireland. pp. 195–202.

Hasty, J., McMillen, D., Isaacs, F., Collins, J.J., 2001. Computational studies of gene regulatory networks: in numero molecular biology. Nat. Rev. Genet. 2 (4), 268–279.

Hecker, M., Lambeck, S., Toepfer, S., Van Someren, E., Guthke, R., 2009. Gene regulatory network inference: data integration in dynamic models – a review. Biosystems 96 (1), 86–103.

Huang, S., Eichler, G., Bar-Yam, Y., Ingber, D.E., 2005. Cell fates as high-dimensional attractor states of a complex gene regulatory network. Phys. Rev. Lett. 94 (12), 128701.

Huang, S., Ernberg, I., Kauffman, S., 2009. Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. Semin. Cell Dev. Biol. 20, 869–876.

Joachimczak, M., Kowaliw, T., Doursat, R., Wróbel, B., 2012. Evolving morphologies and controllers for soft-bodied multicellular animats using gene regulatory networks and artificial embryogenesis. In: Doncieux, S., Jin, Y., Mouret, J.-B. (Eds.), Evo-Devo-Robo: Evolutionary Robotics and Developmental Robotics at the Fourteenth International Conference on Genetic and Evolutionary Computation, GECCO'12 Companion Proceedings. ACM, New York, NY, USA, pp. 357–360.

Koster, M.J., Snel, B., Timmers, H.T.M., 2015. Genesis of chromatin and transcription dynamics in the origin of species. Cell 161 (4), 724–736.

Kurimoto, K., Yabuta, Y., Hayashi, K., Ohta, H., Kiyonari, H., Mitani, T., Moritoki, Y., Kohri, K., Kimura, H., Yamamoto, T., et al., 2015. Quantitative dynamics of chromatin remodeling during germ cell specification from mouse embryonic stem cells. Cell Stem Cell 16 (5), 517–532.

Le Novère, N., 2015. Quantitative and logic modelling of molecular and gene networks. Nat. Rev. Genet. 16 (3), 146–158.

Ling, H., Samarasinghe, S., Kulasiri, D., 2013. Novel recurrent neural network for modelling biological networks: oscillatory p53 interaction dynamics. Biosystems 114 (3), 191–205.

Lones, M.A., 2016. Computing with artificial gene regulatory networks. Evolutionary Computation in Gene Regulatory Network Research. pp. 398.

Lorch, Y., Kornberg, R.D., 2015. Chromatin-remodeling and the initiation of transcription. Q. Rev. Biophys. 48 (4), 465–470.

Narlikar, G.J., Sundaramoorthy, R., Owen-Hughes, T., 2013. Mechanisms and functions of ATP-dependent chromatin-remodeling enzymes. Cell 154 (3), 490–503.

Petronis, A., 2010. Epigenetics as a unifying principle in the aetiology of complex traits and diseases. Nature 465 (7299), 721–727.

Reil, T., 1999. Dynamics of gene expression in an artificial genome – implications for biological and artificial ontogeny. In: Floreano, D., Nicoud, J.-D., Mondada, F. (Eds.), Proc. 5th European Conf. on Artificial Life, ECAL'99, No. 1674 in Lecture Notes in Artificial Intelligence. Springer, Berlin/Heidelberg, pp. 457–466.

Sanchez, S., Cussat-Blanc, S., 2014. Gene regulated car driving: using a gene regulatory network to drive a virtual car. Genet. Program. Evol. Mach. 15 (4), 477–511.

Taylor, T., 2004. A genetic regulatory network-inspired real-time controller for a group of underwater robots. In: Groen, F., Amato, N., Bonarini, A., Yoshida, E., Kröse, B. (Eds.), Proceedings of the Eighth Conference on Intelligent Autonomous Systems (IAS-8). IOS Press, pp. 403–412.

Trefzer, M., Kuyucu, T., Miller, J., Tyrrell, A., 2010. Evolution and analysis of a robot controller based on a gene regulatory network. In: Tempesti, G., Tyrrell, A., Miller, J. (Eds.), Evolvable Systems: From Biology to Hardware, vol. 6274 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 61–72.

True, H.L., Berlin, I., Lindquist, S.L., 2004. Epigenetic regulation of translation reveals hidden genetic variation to produce complex traits. Nature 431 (7005), 184–187.

Turner, A.P., Lones, M.A., Fuente, L.A., Stepney, S., Caves, L.S., Tyrrell, A.M., 2012. Using artificial epigenetic regulatory networks to control complex tasks within chaotic systems. In: International Conference on Information Processing in Cells and Tissues. Springer. pp. 1–11.

Turner, A.P., Trefzer, M.A., Tyrrell, A.M., 2015. Modelling epigenetic mechanisms to capture dynamical topological morphology: applications in edge detection. In: 2015 IEEE Symposium Series on Computational Intelligence. IEEE. pp. 1229–1235.

Turner, A.P., Caves, L.S., Stepney, S., Tyrrell, A.M., Lones, M.A., 2017. Artificial Epigenetic Networks: Automatic Decomposition of Dynamical Control Tasks Using Topological Self-modification.

Turner, B., 2008. Chromatin and Gene Regulation: Molecular Mechanisms in Epigenetics. Wiley.

Zaret, K.S., Mango, S.E., 2016. Pioneer transcription factors, chromatin dynamics, and cell fate control. Curr. Opin. Genet. Dev. 37, 76–81.

Zentner, G.E., Henikoff, S., 2014. High-resolution digital profiling of the epigenome. Nat. Rev. Genet. 15 (12), 814–827.

Zhao, Z., Zhang, X., Wu, X., Li, X.-Y., Han, J., Gasnet:, 2014. Efficient residential building gas leak monitoring via opportunistic networking. In: 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). IEEE. pp. 163–171.