The Artificial Epigenetic Network

Alexander Phillip Turner

Ph.D.

The University of York
Department of Electronics

October 2013

Abstract

The term epigenetics refers to typically heritable biological mechanisms which facilitate stable yet reversible modifications of gene expression or phenotype state, without alteration of the underlying genetic code. More specifically, epigenetic mechanisms allow organisms to control which genes are active at a given time. In eukaryotes, epigenetic mechanisms have essential roles in gene regulation, cellular differentiation and genetic packaging. These epigenetic mechanisms give rise to functionality which DNA alone is generally incapable of providing.

This thesis takes inspiration from the fields of genetics and epigenetics, and builds a computational model which captures the beneficial properties of epigenetics *in silico*. This computational model is referred to as the artificial epigenetic network. The artificial epigenetic network can dynamically control which genes within the network are active at a given time, allowing certain groups of genes to become specialised towards specific aspects of a task. Hence, the artificial epigenetic network can contain many different regulatory circuits, each with specific properties. This gives the networks the ability to more readily express a wider range of dynamical behaviours, which were found to produce a number computational benefits. The artificial epigenetic network is applied to a diverse range of control tasks, each with varying dynamics, to ascertain how the functionality of the artificial epigenetic structures effects the functionality of the network. An emergent property is that the epigenetic structures can partition the network into functional units corresponding to the logical decomposition of the tasks, and control these units with a switch like behaviour. This provides an interface, where a user can gain control over the complex dynamics of the target domain via the activation or deactivation of these switches.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my greatest thanks to Mic for his guidance and friendship throughout the past three years; Freddie, for his companionship and willingness to explore the world; Vanessa for the adventures throughout my time in York; Ben and Andy M for being both hilarious people and great friends; Chris W for his inspiration; Chris A for always welcoming me back to my homeland; Mark, for introducing me to a different world. I would also like to thank Andy T, Leo and Susan for imparting their wisdom on me throughout this journey.

Finally I would like to thank my parents, Phillip and Anna for without them none of this would have been possible.

Declaration

Part of the work within this thesis has been previously published by the author (Turner et al. (2012, 2013*a*, *b*), Lones et al., 2013). All work within this thesis is original to the best knowledge of the author. Any work or research which has contributed to this thesis has been referenced and acknowledged as appropriate.

Hypothesis

This research is motivated by the idea that epigenetic structures in nature facilitate a wide range of genetic behaviours and that aspects of these behaviours can be captured within artificial gene regulatory networks. Specifically it is asserted that:

- Epigenetics in nature dynamically alters the activity of genes based upon internal and external environmental cues. These epigenetic changes can act as biological switches resulting in significant phenotypic changes within an organism in a time frame which would not be possible using gene regulatory networks or mutation alone.
- Epigenetic components are for the most part structurally separate from the underlying genetic structure. It is the combination of genes and epigenetics and their interactions which gives rise to certain beneficial behaviours.
- There have been many computational structures which have been inspired by natural networks such as artificial biochemical networks, artificial immune systems and neural networks. These computational models have captured useful traits from their biological counterparts, such as robustness, self organisation and adaptability when the resulting architecture is applied to control tasks that require a range of specific dynamics to solve.

Therefore it is hypothesised that an artificial epigenetic analogue can be added to a pre-existing artificial gene regulatory network, capturing certain beneficial properties of epigenetic structures, specifically the ability to abruptly and robustly change their phenotype, *in silico*, and in turn improving functionality.

Chapter 1

Introduction

Contents

1.1 Overview

The natural world has had significantly more time to evolve and adapt compared to the humans that inhabit it. In this time, organisms have evolved to methods to store (DNA), manipulate (genetic recombination) and process (biochemical networks) data. All organisms can be considered biological computers. The traditional computer has been designed to specifically process data, and is very adept at this. However, biological organisms have been evolved and have acquired emergent properties such as robustness, fault tolerance and adaptability. These traits are typically not present in computational hardware. Many computational architectures draw inspiration from nature to incorporate these typically evolved traits *in silico*, and many have been successful. However, many of these architectures take a limited view of biology as inspiration. This thesis focuses on creating more biologically realistic artificial gene regulatory networks, capturing a wider range of emergent properties found in their biological counterparts *in silico*.

1.2 Artificial Gene Regulatory Networks

Artificial gene regulatory networks (AGRNs) are computational models which are inspired by the genetic interactions that occur within cells. They fall into two distinct categories. The first aims to model gene regulation to better understand its functionality. The second builds abstract models which capture the biological properties of gene regulation *in silico*. This thesis focuses on the latter.

AGRNs model genes as abstracted computational units which are capable of taking a set of inputs, processing them and producing an output. A collection of interconnected genes forms the network. AGRNs are able to capture specific traits of the biological systems on which they are based such as robustness, self organisation and adaptability. These models have gained popularity because of their functionality, being utilised in fields such as chaos targeting, optimisation and the control of systems which express complex non-linear dynamics (Lones et al., 2010, 2012).

1.3 Evolutionary Algorithms

Evolutionary algorithms are a set of biologically inspired computational techniques which are used to optimise data structures for solving specific tasks. Within this thesis there is a particular focus upon genetic algorithms, a flexible evolutionary algorithm capable of evolving solutions to a diverse range of problems. Genetic algorithms are a population based algorithm which uses biologically inspired operators to artificially evolve candidate solutions towards a specific goal. Within this thesis, they are used to optimise AGRNs.

1.4 Epigenetics

Epigenetics refers to a set of biological structures and processes which are able to heritability modify gene expression without making changes to the underling genetic code. Epigenetic research has been growing in popularity through the decades as it has been found to play a key role within many genetic processes. Epigenetic structures are interesting from an engineering perspective because they are structurally separate entities to that of the genetic code, yet they cooperate to produce behaviours which genetic structures alone are not capable of within the same time frame (Veening et al., 2008).

1.5 The Artificial Epigenetic Network

The artificial epigenetic (AEN) network is a type of AGRN which takes inspiration from epigenetic structures. The AEN is the product of the work within this thesis. It consists of an epigenetic analogue which can dynamically modify the activity of genes within the network. The philosophy of this work is that by improving the richness of the computational gene regulatory network architectures by introducing epigenetic analogues, the biological properties and behaviours of epigenetics can be captured *in silico*.

1.6 Contributions

The work within this thesis has made the following contributions to knowledge:

- The development and implementation of a new epigenetically inspired artificial gene regulatory network.
- The demonstration that the principals outlined in this thesis on capturing complex biological traits *in silico* are capable of doing so.
- The demonstration that the AEN can function on a wide range of tasks, utilising its epigeneitc functionality.
- The realisation that the capturing of epigenetic traits *in silico* leads to many advantages, including increased objective performance.
- The demonstration that the epigenetic analogue can dynamically reconfigure the structure of the artificial gene regulatory network.
- The realisation that the artificial epigenetic network can reduce the complexity of its control to the point where the network dynamics can be externally controlled.
- The realisation that beneficial natural epigenetic characteristics can be captured *in silico.*

1.7 Thesis Organisation

This thesis is organised in three parts. Chapter 2 introduces the biological background upon which this thesis is based. Chapter 3 serves to bridge the gap between the biology and the computational models created within this thesis. Chapters 4 - 6 describe the field of artificial gene regulatory networks and the main contribution of this thesis, the artificial epigenetic network. In addition, they describe evolution and their computational counterparts, evolutionary algorithms. Chapters 7 - 11 describes the application of the artificial epigenetic network to a range of tasks, and the conclusions that can be drawn from this work. More specifically :

Chapter 2 Introduces the biological structures and functions of genetics, and uses this and an underpinning to provide an in-depth description of epigenetic structures and behaviours.

Chapter 3 Introduces the properties and features of biological systems.

Chapter 4 Describes the computational field of evolutionary algorithms.

Chapter 5 Reviews the field of artificial gene regulatory networks, their faithfulness to biology and their computational properties.

Chapter 6 Describes the artificial epigenetic network, its structure and the rationale behind its architecture.

Chapter 7 Is a preliminary guide to the experimental chapters.

Chapter 8 Presents experimental analysis of the application of the artificial epigenetic network to the control of dynamics within Chirikov's standard map.

Chapter 9 Presents experimental analysis of the application of the artificial epigenetic network to the coupled inverted pendulums task.

Chapter 10 Presents experimental analysis of the application of the artificial epigenetic network to the control of transfer orbits in gravitational systems.

Chapter 11 Summarises the work conducted through this thesis, drawing conclusions and suggesting future lines of research.

Appendix A Further experimentation with the artificial epigenetic network.

Chapter 2

The Structures and Processes Of Genetics And Epigenetics

Contents

This thesis describes the implementation of an epigenetically inspired computational network. In order to build up a picture of the functionality of epigenetics, the underlying biological processes and structures must be understood. The purpose of this chapter is to show how high level emergent biological properties and structures are a product of much smaller fundamental components. The first half of this chapter is used to build up an understanding of the biological fundamental structures and processes which are associated with genetics and gene regulation. Thereafter, this information will be used to paint an indepth picture of the structures, properties and behaviours of epigenetics.

2.1 Proteins

Proteins are the fundamental structures of life, the most versatile macromolecules facilitating almost all biological process in the natural world (Berg et al., 2002). Proteins are highly complex structures that exist on the micro (μ) scale and below, with the largest instance of a protein being titin at $\approx$1μm containing $\approx$ 30000 amino acids (Lu et al., 1998). At the other end of the scale, Trp-cage is less than $1nm$ in size and contains only 20 amino acids (Pitera & Swope, 2003). At this scale, the interactions of proteins with other entities is not the typical physical-physical interaction, but the complex product of very specific electrostatic charges. Hence, with the advances of x ray crystallography and being able to view the shape of proteins, it does not provide the complete picture as to how they operate, as their true method of operation is not a product of their shape alone.

Proteins are highly specific molecules, with specific charges, bonds and structures that are defined by their underlying structure. As well as their specificity, proteins also demonstrate plasticity and elasticity which means they can robustly operate within a changing environment. It is because of these features and mechanistic behaviours that proteins are utilised and involved in nearly every natural process. Proteins are abundant within living systems, and it is the form and structure of proteins along with their concentrations within a systems that play a major role in defining their function and activity (Petsko & Ringe, 2004; Robinson et al., 2007).

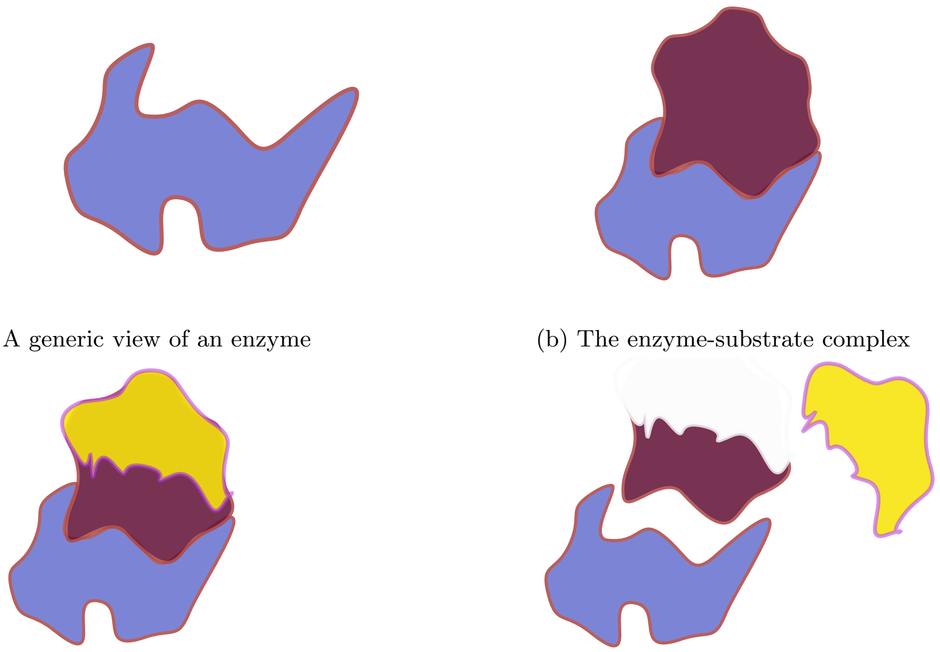
In terms of form, proteins can be classified into 1 of 3 groups : fibrous, globular or membrane (Stoker, 2011). Fibrous proteins are commonly associated with structure and tissues, globular with enzymes and catalysis and membrane with molecular transport and membrane control. In terms of function, proteins are generally classified as 1 of 4 key functions: binding, catalysis, switching and structure (Petsko & Ringe, 2004).

Amino acids are the building blocks of proteins. There are 20 different proteomic (being part of a naturally occurring protein) amino acids (Chou, 2009; Blom et al., 2004). Each of these amino acids has a basic biochemical structure. Within this structure, there is an "R" group, and it is the chemical composition of the R group with defines the type of amino acid (Jones, 2002), in turn defining how it interacts with other amino acids within the protein. For example, if the R group was of the methyl group CH_3, the amino acid would be alanine. Each amino acid has a specific structure and biochemical function. Amino acids can assemble together to form a linear polypeptide chain, which is known as the primary structure, the basis of all proteins.

2.1.1 Enzyme Catalysis

Enzymes are a specific form of proteins which facilitate many of the chemical reactions needed for life (Figure 2.1). In a basic sense, the role of enzymes is to manipulate molecules, typically breaking them apart or joining them together (catalysis). This is important to organisms, as not all molecules can be sourced from the environment, and therefore must be synthesised. Additionally, these chains of catalytic reactions frequently require conditions which are not possible within the organism. Enzymes alter the requirements for such reactions. One of the most significant abilities of enzymes is to reduce the amount of energy required for a chemical reaction to take place. This is vital for life as frequently, the energy needed for certain chemical reaction would perturb the internal environment of an organism, or be wholly impractical.

Of the many properties of proteins, the most significant are their catalytic efficiency, specificity and the relative simplicity by which catalytic activity can be regulated (Tsai, 2007). Enzymes are so highly efficient, that they can display rate enhancement (increase in speed of substrate synthesis) of between 10^{10} to 10^{25} times (Radzicka et al., 1995). To put this in perspective, the Orotidine 5'-phosphate decarboxylase enzyme can catalyse a substrate in 18



(a) A generic view of an enzyme

(b) The enzyme-substrate complex

(c) Substrate transformation

(d) Release of the products

Figure 2.1: An illustration of the induced fit hypothesis.

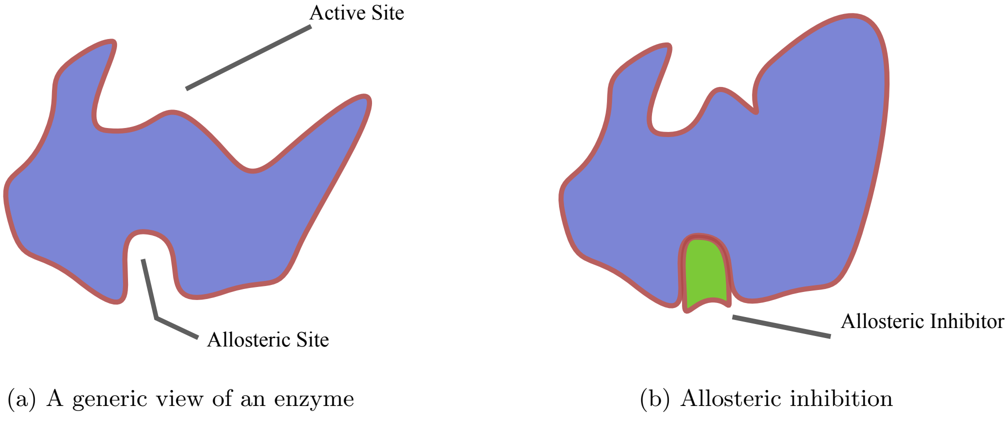
milliseconds, a process that would take 78 million years without the enzyme (Wolfenden & Snider, 2001).

Enzymes are complex structures, yet they are highly specific. Because of the highly intricate surface of the enzyme, it will only bind and operate under specific conditions with the correct molecules. Hence, enzymes can exist around many other molecules, but only have an effect on specific ones. The most widely accepted process in which enzymes operate is the induced fit hypothesis (Figure 2.1).

Within the structure of enzymes, there are very precise processes in place to control their activity such as allosteric control (Popovych et al., 2009), covalent modification (Nagai et al., 2011) and protein processing (Millevoi & Vagner, 2010). Allosteric control of an enzyme is through the allosteric site on the enzyme (Figure 2.2). Typically, if the allosteric site is occupied, the structure of the enzyme changes, altering which chemicals can interact with it.

2.1.2 Protein Binding

A key characteristic proteins have is the ability to bind with other molecules (Petsko & Ringe, 2004). The structure of individual types of protein allows for a high level of specificity within an environment. It is the diversity of binding that proteins can achieve that make them



(a) A generic view of an enzyme

(b) Allosteric inhibition

Figure 2.2: The effect of allosteric inhibition and the following change in the shape of the active site.

essential to the majority of all key process within nature.

One of the most ubiquitous binding proteins found in nature are antibodies. Antibodies are an essential part of immune systems in almost all natural systems (De Berardinis & Haigwood, 2004; Avrameas et al., 2009; Chia et al., 2010). Principally, antibodies are responsible for binding to antigens that a biological system defines as alien to it. Therefore, once an antibody has bonded to the alien artefact, it is much easier to locate, contain and augment according to the interests of the system (Phelps & Hassed, 2012).

2.1.3 Protein Switching

A property of proteins' structure is that in general, they are flexible - meaning that they can allow for small changes to their structure without degradation. This means that the protein can exist in two different states, and therefore posses the ability to act as a marker and messenger for certain events. These switches are key to maintaining homoeostasis (3.3.4), as they can react to specific changes within the environment such as pH change, temperature and energy levels and provide an intracellular message to coordinate the cell towards maintaining a positive state. An example of this, is the process in which proteins switch to begin cellular division. This requires a high level of coordination within the cell to organise and reproduce organelles, whilst ensuring the cell maintains homoeostasis (Halfmann et al., 2010; Robinson et al., 2011).

2.1.4 Structural Proteins

Structural proteins are typically fibrous and are responsible for a wide range of biological structures and tissues. They are ubiquitous in almost all of nature and are present in both the inner cellular structures such as cytoskeletons, and much larger multicellular structures such as vascular networks. There are many structural proteins, each with different attributes such as keratin, elastin and collagen. Each of these provide structure, with keratin being attributed to harder structures, elastin to structures which require absorbent elastic properties and collagen, which is used for connective tissue. It is the combination of these proteins which give rise to a diverse set of organisms over many levels of complexity (Critchley, 2009; Luger et al., 1997; Schalch et al., 2005; Schroeder Jr & Cavacini, 2010).

2.2 Nucleic Acids

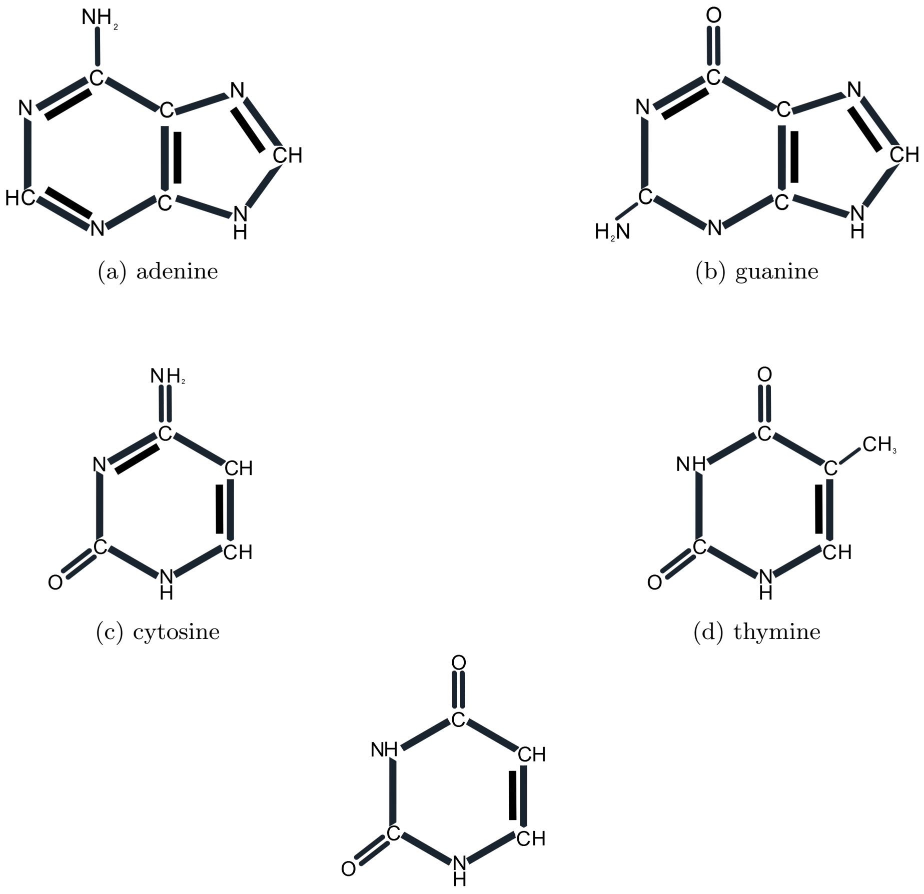
Nucleic acids are the structure in which genetic information is held in every single instance of life discovered (Krude, 2004). They are the method nature uses to encode data. The structure of nucleic acids allow for efficient access, manipulation and duplication of this data and importantly, the ability to retain this information over successive generations. Nucleic acids are long thread like macromolecules comprised of a repeating set of nucleotides.

2.2.1 Nucleotides

All nucleotides are the product of 3 components, a hetrocyclic base, pentose sugar and phosphate residue. There are 5 nucleotide bases, adenine, guanine, cytosine, thymine and uracil (Blackburn, 2006) (Figure 2.3e). Thymine is only found in DNA, and uracil in RNA. They serve many purposes within an organism, however in this section they will be viewed as the fundamental molecules of the genetic code, representing single units of genetic data.

2.2.2 DNA and RNA

Deoxyribonucleic acid (DNA) is a macromolecule which holds the genetic instruction set within almost all living organisms (Krude, 2004; McCabe et al., 2008). The structure of DNA is a double helix, using complementary base pairing, where adenine always binds to thymine, and guanine always binds to cytosine (Crick & Watson, 1953). This is referred to as complementary base pairing. The bonds between complementary bases are weak, allowing



(a) adenine

(b) guanine

(c) cytosine

(d) thymine

(e) uracil

Figure 2.3: The 5 bases of the genetic code adapted from (Strachan & Read, 2004).

the two strands to become detached with minimal energy. However, the DNA molecule is very stable both to physical perturbations and 'corrupt' bases due to large numbers of interactions.

RNA is similar to DNA; however, it typically consists of a single strand. There are exceptions to this, such as the double stranded RNA found in retroviruses, but in general RNA is just a single strand. Furthermore, RNA does not use the thymine base, and instead uses uracil. RNA is much easier to access, and also much more manoeuvrable; however it loses a lot of the structural integrity compared to DNA as it only has one strand. RNA holds the same amount of information as the equivalent length of DNA.

There are many variants of RNA. This chapter will focus on the following three: messenger RNA (mRNA); transfer rNA (tRNA) and ribosomal RNA (rRNA). Other variants include mircroRNA (section 2.7.4) and small interfering RNA. One of the key properties of RNA is that it is structurally more malleable than DNA, and as such, due to complimentary base pairing, can bind with itself. This allows it to form structures which can be utilised during protein synthesis.

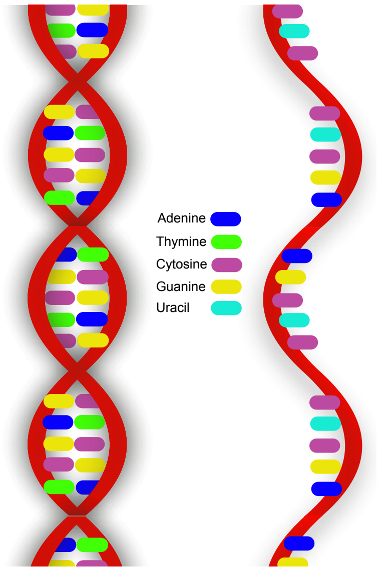


Figure 2.4: DNA double helix and corresponding RNA single strand.

DNA and RNA are the key components of life. They hold the data specifying the primary sequence of all known proteins, and act as a biological blueprint for every known organism. Within a living organism, DNA acts as a hard copy of genetic data akin to a biological memory holding the list of proteins and regulatory information within an organism.

2.3 Protein Synthesis

Protein synthesis is the process of creating the primary structure of a protein from a DNA or RNA template. It is an essential part of all organisms, and is a constant process due to changes in gene regulation and replacing degraded proteins.

2.3.1 Transcription

Transcription is the process of creating an RNA strand from DNA, and it is the first stage of protein synthesis. Initially, the bonds between the bases of DNA break apart, effectively unwinding the helical structure. Upon the breaking of the bonds, the genetic code can be accessed by the cellular machinery, specifically RNA polymerase. The initialisation of transcription is a complicated process, and usually begins at a short, specific sequence of DNA (Tora & Timmers, 2010). Proteins can then bind specifically to that site, to create a transcription complex. The purpose of this complex is to create a structure consisting of multiple proteins which has at the core of its function RNA polymerase. This is because RNA polymerase does not commonly bind with DNA, and the transcription complex has a much higher affinity to bond with the DNA (Harris et al., 2002). Transcription factors such as repressors can also be used to disrupt the creation of a transcriptional complex.

RNA polymerase acts as a structure in which to access the base pairs in a strand of unwound DNA, and match these up to the corresponding base pair on the newly formed RNA strand (Figure 2.5). This process is repeated until an end sequence has been reached. After this, DNA can then recombine to form its double helix structure.

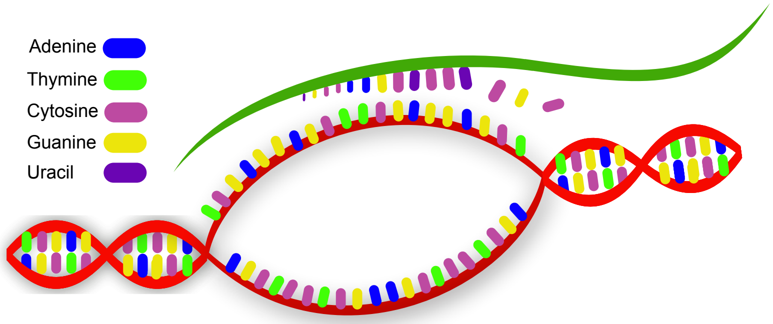


Figure 2.5: RNA being transcribed from DNA.

Once the RNA molecule has been synthesised, if it can bind with itself, it will begin to fold into a structure such as tRNA or the rRNA in the ribosome. There are also post transcriptional modifications that can occur to RNA such as RNA editing (Li et al., 2009; Nishikura, 2010) and RNA interference (Siomi & Siomi, 2009). RNA editing is the modification of the RNA strand via the insertion or deletion of bases or through deamination. The biological role of RNA editing is not fully understood, however the creation of protein variants and a regulator of gene expression have been hypothesised (Speijer, 2011). RNA interference is the process in which small RNA strands (such as microRNAs section 2.7.4) are created specifically to prevent transcription. One of the means by which this is achieved is by binding to longer RNA molecules to prevent translation (Hannon, 2002).

In order for protein synthesis to continue to the next stage of translation, the RNA must be non folded, and RNA in this state is known as mRNA.

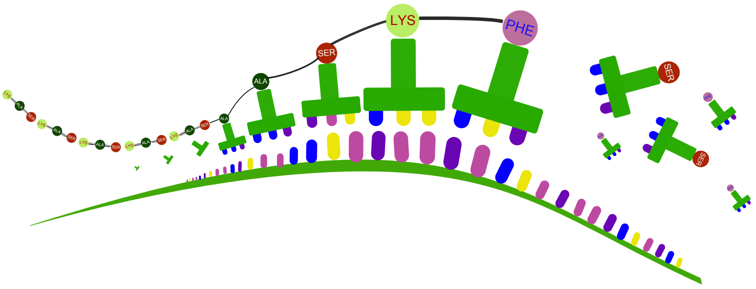


Figure 2.6: RNA being translated into an amino acid sequence.

2.3.2 Translation

Translation is the process of synthesising a polypeptide sequence from an mRNA strand. This is the final stage of protein synthesis. The process of translation occurs within a ribosome, a complex structure which is essential in protein synthesis. The ribosome straddles the RNA strand and reads the base information from it. The RNA strand is read in groups of three, which are known as codons. The process of translation begins at a specified start codon, and ends at a stop codon. Upon reaching a start codon, the ribosome attracts tRNA. tRNA is a structure which consists of folded RNA which can attach to an amino acid on one side, and expose a base sequence of length three (an anti codon) on the other. This means that in cooperation with a ribosome, the tRNA molecule can specifically bind to a codon on the RNA strand, and line up an amino acid chain (Figure 2.6). The ribosome will then move along the RNA strand until a stop codon is found. The polypeptide chain can then break free and will begin folding into secondary and tertiary structures.

2.3.3 Posttranslational Modifications

Posttranslational modifications (PTM) can be applied to the polypeptide chain after translation. The principle reason for postranslational modifications is to generate a more diverse selection of proteins than that explicitly encoded in the genome. The human genome contains around 30,000 genes. However, estimates of the number of proteins in the human body is said to be an order of 10-100 times greater than that of gene encoded proteins alone (Walsh, 2006). This increase in diversity is because of PTMs. There are considered to be two key methods in which the primary structure of the protein can be modified: firstly by covalent addition to individual amino acids in the chain; secondly, by the hydrolysis of amino acids. These PTMs result in changes in the way the polypeptide chain folds into a mature protein, in turn creating protein variants.

2.4 Genes

A gene is a functional unit of hereditary information within a living organism, considered to be a region of DNA that specifies the primary structure of a protein which serves a specific function (Sarkar & Plutynski, 2008; Turner et al., 2013*b*). Every single living organism is a product of the genes held within its genetic structure.

Genes are structured similarly between all organisms, a phenotypic currency. This is why, within limits, genes which have never occurred naturally in an organism can be artificially introduced and still function (Lai et al., 2002).

Although genes are very similar in their form, principally a section of DNA specifying a sequence of amino acids, their organisation differs significantly between prokaryotes and eukaryotes. The difference between the two classes of organism is that prokaryotes lack a nucleus, and in turn are limited in complexity. Prokaryotes typically organise their DNA in the form of operons, in which a group of genes are located within a single regulatory promoter, meaning that they are all transcribed together (Dworkin et al., 2006; Hickey et al., 2007; Miller & Reznikoff, 1978). An example of this can be seen in the lac operon (Figure 2.7). This is advantageous in ways such as being able to complete a regulatory circuit via transcribing a single length of DNA, which in turn is more efficient. However, there is a lack of specificity in that operons have to transcribe all genes together, which under environmental perturbations may not be optimum.

The genetic operation in eukaryotes is organised differently. A primary reason for this is that eukaryotic genomes contains a majority of non-protein coding DNA. In prokaryotes, the relationship between the amount of non-protein coding DNA and protein coding within the genome is linear (Ahnert et al., 2008), which cannot be said for eukaryotes as gene regulation is more complex. This is highlighted in Figure 2.8. Eukaryotic genes also have higher order genetic structures such as chromatin which is used to package DNA into the nucleus of a cell.

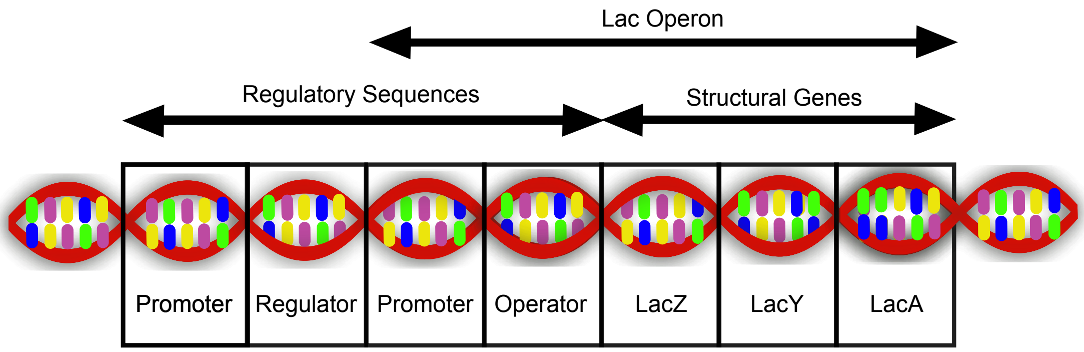


Figure 2.7: An overview of the structure of the lac operon.

2.4.1 Gene Regulation

Gene regulation is the process in which organisms control the rates of gene expression to maintain an optimum state. Gene regulatory circuits vary in size and complexity and exist over many levels of abstraction. Some of the smallest gene regulatory circuits have been fully mapped, a key example of this is the lac operon (Jacob & Monod, 1961) (Figure 2.7), found in the bacteria *Escherichia coli*. The lac operon is designed to metabolise lactose dynamically, depending on the amount of lactose in the environment. There are two control mechanisms the lac operon uses to do this, one in response to lactose, the other to glucose. The first works by responding to lactose in the environment via a regulatory lactose repressor. If lactose is not present, the repressor will bind to the operator which is downstream of the lac genes. This significantly inhibits expression of the genes. In the presence of lactose, allolactose will bind to the repressor, modifying its shape, and inhibiting its ability to bind to the repressor. The second is in response to glucose, and via an intertwined genetic and metabolic pathway (section 2.5.2). The transfer of glucose into the cell requires phosphorylation, and this in turn removes the phosphate group from certain proteins. These unphosphoralated proteins then bind to a protein designed to induce permeability in the cell wall, and in turn, prevent it doing this. These two control structures allow the *Escherichia coli* to control with high precision, the expression of certain genes in tune with the environment.

Gene regulation is the product of many interlinked structures and processes. In this chapter, transcription factors, RNA editing, RNA interference and repressors have all been shown to affect gene regulation. These structures and processes are heavily linked with higher order epigenetic structures such as chromatin modifications, which have been shown within this chapter to have a significant effect on gene regulation. It is the collective work between these structures over different levels that produce the emergent property of gene regulation.

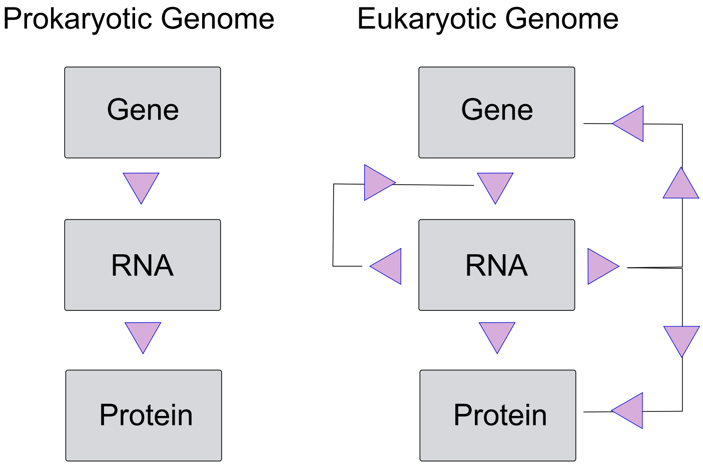


Figure 2.8: A comparison of the differences between the central dogmas of prokaryotic and eukaryotic gene regulation, which provides insights to the structures of their respective genomes. Adapted from (Mattick, 2001).

Although prokaryotes' genetic structure is more simplistic than that of eukaryotes, there is still much to be discovered about how it works. This is because although gene regulation can be broken down into small networks (not all of which are known) it is the emergent properties of these network interactions that create the interesting robust structures found in nature. Genome wide regulatory information is difficult to compile as there are so many other factors involved in gene regulation such as metabolism and environmental changes. Moreover, gene regulation does not exist on a single level of abstraction, there are networks of networks of genes which all have to function together to maintain homoeostasis.

2.5 Biochemical Networks

Biochemical networks are the underpinning of the functional and structural complexity within biological organisms (Lones et al., 2010). Biochemical networks interact to form higher order structures capable of expressing complex dynamical behaviours. This is refereed to as self-organisation, the idea that unconnected components with uncoordinated behaviour can result in ordered controlled behaviour on higher levels of abstraction. Biochemical networks are ubiquitous throughout biology, operating on the smallest scales, through to individual communication between individuals of the same species. It has been hypothesised that the high level emergent properties of biochemical networks arise solely as an emergent property of the underlying components (Bhalla & Iyengar, 1999). Biochemical networks have been regarded as computational devices within all living cells (Bray, 1995).

From a broad perspective, there can be seen to be three main biochemical networks within a living organism, gene regulatory networks, metabolic networks and cell signalling networks. Although these are separate entities, they rarely operate separately, and are very much intertwined.

2.5.1 Gene Regulatory Networks

Gene regulatory networks are the product of gene interactions, and these interactions along with the environment define the cell's state. Previously in this chapter there has been a focus on the genetic code (section 2.2) , protein synthesis (section 2.3) and protein function (section 2.1.1 - 2.1.4).

For all of these processes and structures there are methods of control in place to regulate and change the end result. It is the regulation of the processes involved in protein synthesis and protein function that determine the genetic expression levels within the cell.The regulatory nature of the cell is difficult to capture as there is a seemingly overwhelming number of possible steps in even the simplest methods of control. A given protein can bind to another protein, inhibit or excite DNA exposure, it can bind to a protein's allosteric site, it can modify the cell wall and change the environment. With all these operations going on constantly it is difficult to ascertain the underlying genetic circuits within the cell. The size of gene regulatory networks can vary immensely, from the lac operon (Figure 2.7) to complex transcriptional networks (figure 2.9).

One of the problems in modelling networks of gene regulation is that they exist over many levels of abstraction. Simple circuits can be fully understood, but they interact with other simple circuits, which produce behaviours that interact with larger circuits. However, there are specific traits that all gene regulatory networks have. They are dynamic, robust and self adaptive. These traits emerge from underlying elements such as genes. With this in mind, people have attempted to model gene regulation as a product of simple underlying components. Research has shown that randomly initiated network models consisting of only Boolean components can show self organisation, robustness and adaptivity over time (Kauffman, 1969).

Gene regulatory networks are the archetypal biochemical network, which interact with every aspect of an organism. This is partly down to the abundant nature of proteins, which must be synthesised from genes and partly down to genes being the defining method of holding biological data.

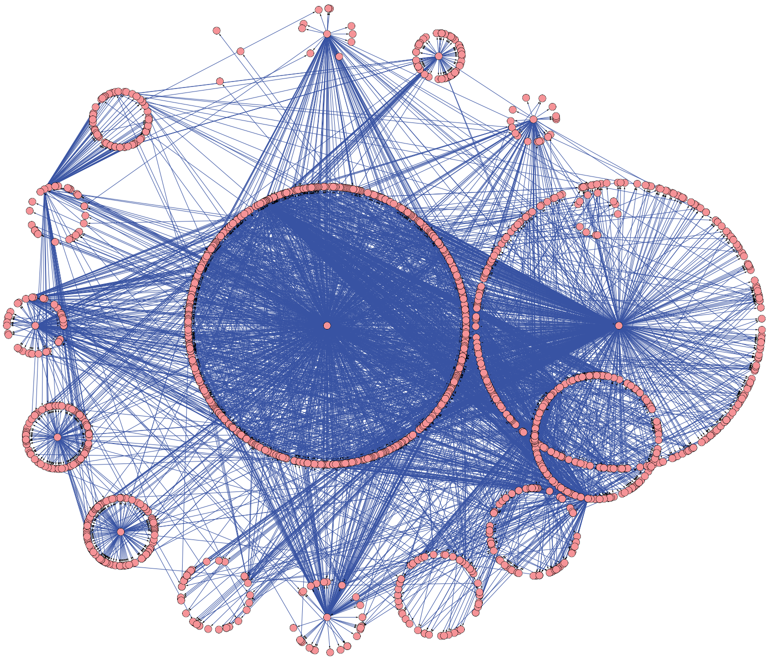


Figure 2.9: A visual description of the genes and iterations in a transcriptional network. Genes are represented by orange circles and causal relationships are represented by directed edges with black arrows. (Chen et al., 2007).

2.5.2 Metabolic Networks

The purpose of a metabolic network is to facilitate the generation of a chemical product. This usually consists of a number of intertwined pathways. Metabolic networks are essential to organisms because they allow the synthesises of products that are not currently available within the environment. One pervasive example of this is the creation of energy from various sources. The biological carrier of energy is adenosine triphosphate (ATP) (Coffee, 2004), which releases energy by breaking its bonds, which then creates adenosine diphosphate, and if repeated, adenosine monophosphate. ATP is a robust structure which can easily be transported around the cell to where it is most needed.

Metabolic networks may synthesise a plethora of products, and in turn, consist of many metabolic pathways. Natural diversity suggests that due to varying environments, metabolic networks should vary accordingly. However, research has shown that large scale organisation is seemingly identical across a range of species, and moreover, adheres to the design principles or robust and fault tolerant scale free networks (Jeong et al., 2000).

2.5.3 Cell Signalling Networks

Cell signalling networks are bidirectional communication links between cell and environment. They take an internal signal and propagate that signal outside the cell wall. In addition they take an external environmental signal and sense that signal by using plasma membrane receptors and receptor tyrosine kinases (Kholodenko, 2006). These networks are not a simple transmit / receive model, but are in fact information processors, encoders, and integrators. These environmental signals can be taken from the environment, into the cell and then processed and passed to gene regulatory networks in which gene expression values can adapt to make sure the cell is best suited to the environment. Frequently this would include metabolic networks too.

Research has recently shown that distinct spatial temporal activation of the same repertoire of signalling results in different protein pathways being activated (Hoffmann et al., 2002; Kholodenko, 2006). This suggests that cell signalling networks interact on levels of abstraction which may have previously been overlooked. Spatio-temporal elements are not commonly associated with gene regulatory networks or metabolic networks.

2.6 Epigenetics

Epigenetics is the study of systems that exist and operate in conjunction with, but on a different level of organisation than the genetic code. The 'epi' of epigenetics is Greek and means above or over (*Online Etymology Dictionary*, 2013), which fits accordingly with the theme of epigenetics. Epigenetic systems interact with DNA in order to regulate the expression of genes. In an abstracted sense, the genes can be seen as the instruction set of an organism, and the epigenetics are the control. This paints an interesting view of the evolution of such systems and how the partition between instruction and control exists in the biological world.

Epigenetics' functionally is wholly dependent on the structures and processes described previously within this chapter, and operates in conjunction with these to create a higher level biological control system. This chapter describes epigenetic structures and how they interact and modify biological processes such as gene regulation (section 2.4.1), cellular differentiation (section 2.8.2) and homoeostasis (section 3.3.4), and moreover, demonstrates why using epigenetics to do this is beneficial within an organism.

2.6.1 Definitions of Epigenetics

Since the inception of epigenetics, the scientific community has never pinned down an exact definition of what epigenetics is (Holliday, 2006; Berger et al., 2009; Riddihough & Zahn, 2010). Moreover, there is consistent debate about what biological mechanisms can be held under the umbrella of epigenetics. As epigenetics is a key component of this thesis, it is imperative that a definition is put into place so that other ideas and work can be referenced in terms of it.

There are aspects of epigenetics that are agreed upon :

- Epigenetics is contained within the study of structures or mechanisms which act in conjunction with DNA to alter gene expression or phenotypes without the modification of the underling genetic code (Berger et al., 2009; Riddihough & Zahn, 2010; Allis et al., 2007).
- Epigenetic controls are reversible (Jaenisch & Bird, 2003; Feinberg et al., 2006; Tollefsbol, 2010).
- Epigenetic modifications are stable (Goldberg et al., 2007; Berger et al., 2009).

However, the differences of opinion occur when talking about two key aspects of epigenetics :

- Whether or not an epigenetic mechanism has to be heritable.
- What biological mechanisms or structures are truly epigenetic.

In order to build up an accurate definition, we must look at the above statements. On the face of it, there is an overwhelming amount of evidence supporting the fact that epigenetic mechanisms are heritable (Allis et al., 2007; Holliday, 2006; Jones & Takai, 2001; Egger et al., 2004; Berger et al., 2009; Riddihough & Zahn, 2010; Jaenisch & Bird, 2003; Feinberg et al., 2006; Goldberg et al., 2007). However, in the case of non-dividing terminally differentiated neurons in the central nervous system the role of heritability is no longer present. Yet, there is epigenetic content in the neurons which is essential to memory formation (Levenson & Sweatt, 2005). Hence, there is at least one exception to the rule, and thus, in this instance with the evidence considered, it would not be accurate to define that an epigenetic mechanism must be heritable.

With the above research in mind, we can incorporate this into a more detailed definition of epigenetics given current understanding. The definition which fits current viewpoint on epigenetics succinctly is as follows:

> ”Epigenetics is the set of typically heritable biological mechanisms which facilitate stable yet reversible modifications of gene expression or phenotype without alteration of the underlying genetic code.”

Given that epigenetics is still in its infancy it is difficult to distinguish whether or not certain structures can be considered truly epigenetic. Although not key to the definition specifically, it is an aspect which must be addressed. In particular, histone modifications are commonly attributed to being an epigenetic mechanism (Tollefsbol, 2010; Allis et al., 2007) (section 2.7.1). There are however instances in which research has shown that it is not the modification of the histones that hold on to the hereditary information during mitosis, but certain proteins which stay associated with aspects of the DNA. This proposes that it is specific proteins that rewrite the histone’s code, effectively acting as an epigenetic marker and in turn being part of post-transcriptional modifications (Petruk et al., 2012). However, demonstrating cases where histone proteins are not seen as epigenetic structures does not necessarily detract from the majority of research that opposes this position, and it is not clear what is the exception and what is the rule. There is currently not enough research to discount histones as an epigenetic structure; this thesis will treat them as so.

2.6.2 History of Epigenetics

Epigenetics has a stunted scientific history in comparison to epigenetics, and was only accepted as a scientific theory after decades of debate. The reasons for this span back to pre-Darwinian times, where different theories of evolution coexisted up until the publication of the “origin of species” (Darwin, 1859), where natural selection and later Mendelian inheritance gradually became the most accepted theories of evolution (Darwin, 1859; Rudolph & Stewart, 1998). Around the 1920s, Paul Kammerer presented many accounts of research which demonstrated Lamarckian inheritance, which went against the scientific dogma at the time (Kutschera & Niklas, 2004). Most famously, the work revolved around modifying the behaviour and phenotypes of the midwife toad in time scales which would have been impossible through natural selection and genetic mutation alone.

At the time Kammerer’s work was disregarded, with many sources claiming the work was either unrepeatable, or fraudulent. Up until this day, the work has not been repeated, however, it has developed greater scientific merit over the years, especially since the inception of epigenetics, with many now regarding Kammerer as its forefather (Vargas, 2009).

Following Kammerer's work, the next instances of epigenetic-like work came in a series of symposia at Cold Spring Harbour, the first of which was in 1941 (Gottschling, 2004). The term epigenetics was coined and in print in 1957 by Conrad Waddington (Tollefsbol, 2010). Ever since, the field of epigenetics has been growing and expanding to the point at which now, it is scientifically entwined with genetics (Allis et al., 2007).

2.7 Epigenetic structures

2.7.1 Histones

Histones are fundamental to gene regulation, and also to the scaffolding which creates higher order genetic structures such as chromatin and chromosomes (Allis et al., 2007; Tollefsbol, 2010). There are a total of five histone groups. H2A, H2B H3 and H4 are core histones, and H1/H5 are linker histones with only H1 appearing in the human body. Histone proteins have tails which can be modified, and in turn change how they bond with DNA and the other surrounding histone proteins. Histones almost exclusively exist in eukaryotes, with the exception of those species in the Archaea domain (Griswold, 2008).

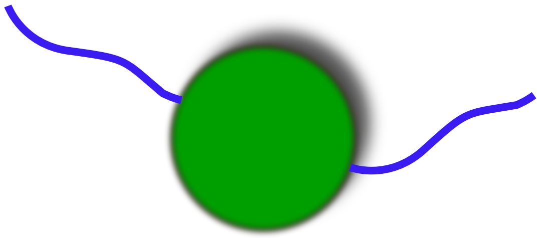


Figure 2.10: A simplified model of the histone, showing the core proteins (green) and the protruding tails.

There are, however, instances of homologous structures appearing in prokaryotes (Slesarev et al., 1998) which have different functionally to eukaryotic histones. The primary purpose of histones is as a type of DNA packaging. A single histone (Figure 2.10) serves little purpose, it is only when they are in the form of an octamer (eight histones arranged in a cuble like

Posttranslational Modifications	Transcriptional Role	Histones Modified
Acetylation	Activation	H3,H4,H2A,H2B
Phosphorylation	Activation	H3
Methylation	Activation Repression	H3 H4
Ubiquitylation	Activation Repression	H2B H2A
Sumoylation	Repression	H3,H4,H2A,H2B

Table 2.1: A table showing the effect of histone tail modifications of gene expression. Adapted from (Kouzarides & Berger, 2007; Chuang & Jones, 2007).

structure) is it possible to form a genetic structure.

There are a range of post-transcriptional modifications to histone proteins that ultimately lead to regulation of gene expression. Examples of these are in Table 2.1.

Histone Octamer

This histone octamer consists of 8 core histone proteins. The basic form of this can be seen in Figure 2.11. Each histone consists of 2 of the H2A, H2B, H3, and H4 histone proteins. When the histones are arranged in an octamer, it creates a surface which allows DNA to bond to it.

For each histone ocatamer, 147 base pairs (bp) of DNA are toroidally coiled around approximately 1.67 superhelical turns (Kaplan et al., 2008; Richmond & Davey, 2003; Luger et al., 1997). Each octamer is connected to another via approximately 80bp of DNA called linker DNA. The combination of DNA and histone proteins is referred to as the nucleosome. The modifications to histone tails and the effect this has on gene expression can be seen in Table

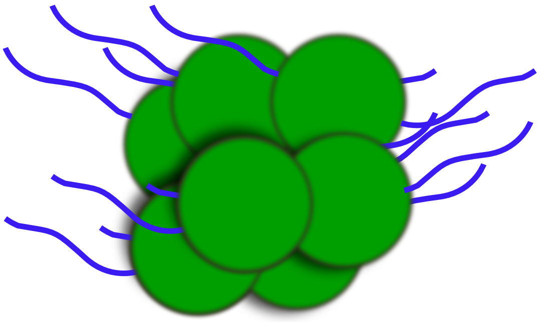
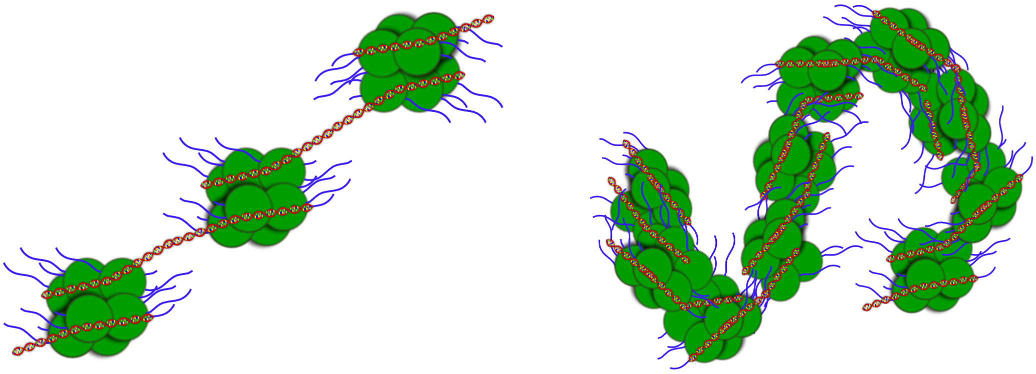


Figure 2.11: A simplified model of the histone octamer, showing how the single histones (Figure 2.10) come together to form the octamer.



(a) Non compact histones

(b) Compact histones

Figure 2.12: Nucleosomes and how the compactness of the histones allows access to the underlying DNA structure.

2.1.

Nucleosomes

The Nucleosome is a higher order structure of genetic material. It consists of the histone octamers and a further histone protein (H1/H5) intertwined with DNA. The H1/H5 is known as a linker histone, and facilitates the binding of the octamer with the DNA. The fundamental purpose of this is to exploit the fine structure of DNA to condense it. Nucleosomes are the first stage of DNA condensation that happens in the eukaryotic genome. Nucleomes are the constituent repeating units of chromatin, a higher order genetic structure. These structures can be seen in Figure 2.12.

2.7.2 Chromatin

Chromatin is the next higher order genetic structure above nucleosomes, but below chromosomes. Chromatin exists in one of two states, either heterochromatin or euchromatin. This differentiation pertains to the density of the chromatin fiber, with heterochromatin considered tightly packaged, and euchromatin loosely packaged.

Euchromatin is a bead and string like structure, similar to that in Figure 2.12a, but with many more nucleosomes. Euchromatin is more accessible to cellular machinery and more associated with active gene expression (Hwang et al., 2001). Hetrochromatin is more condensed, as illustrated in Figure 2.12b. Hetrochromatin is condensed into a 30 μm fiber, which is ultimately condensed into the chromosome. Its structure is hypothesised to be helical, but the scientific community is unsure about the exact structure, with the possibility of there being many variants of the heterochromatin structure (Schalch et al., 2005; Robinson & Rhodes, 2006). Because of its structure, heterochromatin is less associated with active gene regulation.

2.7.3 DNA Methylation

DNA methylation is one of the principal epigenetic mechanisms by which a cytosine or adenine base in DNA is methylated (Turner et al., 2013*b*) (Figure 2.13). Methylation usually occurs around high densities of cytosine bases. These locations are known as CpG islands, and it is estimated that in mammalian cells, 1% of all the bases are methylated (Kim et al., 2009). An example of a methylated set of CpG island can be seen in Figure 2.13.

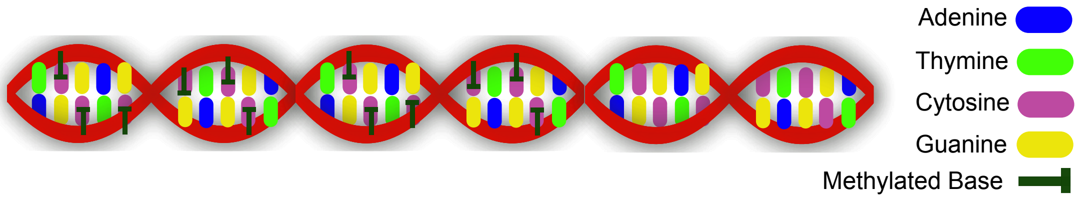
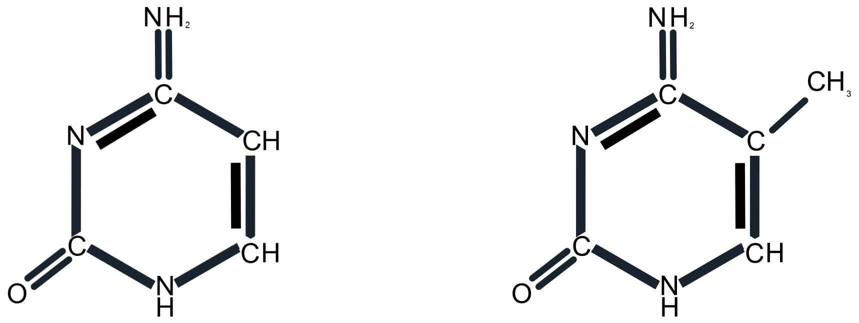


Figure 2.13: A simplified model DNA methylation of the cytosine base in DNA.

DNA methylation acts as a marker, and this results in the modification of gene expression by either physically preventing transcription of the DNA or by using that marker as a binding point to recruit proteins which effect the higher order genetic structures (Phillips, 2008). DNA methylation is usually regarded as having a more long term stable effect on the regulation of gene expression compared to that of other epigenetic mechanisms (Jones & Takai, 2001). Cellular differentiation has been highly linked with DNA methylation (Meissner et al., 2008; Huang & Fan, 2010) and, in turn, providing cells with a heritable identity.

2.7.4 MicroRNA

MicroRNA (miRNA) is a very small section of RNA of around 22bp which has been shown to have a profound effect in the regulation of gene expression (Chuang & Jones, 2007) and is a structure involved in RNA interference (section 2.3.1). They are transcribed in a similar way to RNA using a polymerase enzyme. miRNAs go through set stages and modifications *in vivo* until they have reached a mature state.



(a) Cytosine

(b) 5-Methylcytosine

Figure 2.14: The methylation of the cytosine base in DNA.

miRNAs work by binding to mRNA either fully complementary or partly complementary. These are both hypothesised to down regulate gene expression.

Although miRNAs appear to operate separately to other epigenetic mechanisms, research has shown that DNA methylation and histone modifications are highly interlinked with the operation of miRNAs (Bao et al., 2004; Maison et al., 2002; Chuang & Jones, 2007). Moreover, there is research which has given evidence to the theory that DNA methylation and histone modifications can regulate the expression of miRNAs (Saito et al., 2006). The precise ways in which miRNAs operate is still not fully understood, but from the growing amount of research it is assumed that their relationship to other epigenetic mechanics is cyclical.

2.8 Biological Advantages Of Epigenetic Mechanisms

The descriptions in the previous sections has viewed epigenetics on a molecular level. But to be able to view how these molecular changes create high level phenotypic changes, epigenetics is best viewed at the level of the organism.

2.8.1 Genetic Packaging

If we review the sections dealing with the histones through to chromatin (sections 2.7.1 : 2.7.2), it can be seen that these structures are a packaging for DNA. The reason as to why DNA needs packaging is that, in the example of humans, 2m of DNA need to be held within a nucleus of around 6μm. It is the higher order folding of DNA around histones, into nucleosomes and into chromatin, which allows all this genetic material to fit within a nucleus (Alberts et al., 1994; Bushman, 2002). Because of how DNA is packaged into chromatin, it means that an organism can hold significantly more genetic data that what would be possible without higher order structures, and also maintain control over it. The presence of a nucleus is one of the determining factors between prokaryotes and eukaryotes, and the presence of chromatin like structures paints a picture as to why eukaryotes can be much more genetically and phenotypically complex.

2.8.2 Cellular Differentiation

With eukaryotes being able to store more genetic data, they can encompass different cellular phenotypes within their genetic data. This can be achieved by gene silencing, and is why humans can have approximately 210 different cell types, yet all cells share the same DNA (Strachan & Read, 2004). The inactivation of certain genes can lead to different cellular phenotypes which are specialised for certain functions (Lister et al., 2011; Lee et al., 2004). There is a wealth of information that suggest that cellular differentiation is significantly effected by epigenetic processes (Khavari et al., 2010; Veening et al., 2008; Lunyak & Rosenfeld, 2008) however, the mechanisms behind cellular differentiation are not fully understood, and it has been hypothesises that ATP-dependent chromatin remodeling enzymes might play a role in cellular differentiation (Khavari et al., 2010).

2.8.3 Genetic Memory

Epigenetics gives the genetic code a rudimentary form of memory in which to plan future events based on experience (Bonasio et al., 2010). Because epigenetic mechanisms are reversible, it means that a previously visited state can be revisited via the modification of the epigenome alone. This means large changes to gene expression and phenotype can occur in a much faster time frame than DNA mutations alone would allow, giving the organism a level of phenotypic plasticity. In addition, mutation and modification to the DNA is not reversible, meaning any changes are permanent which can be of detriment for the organism.

An example of this is bet hedging in bacteria. The idea of bet hedging in reference to bacteria means that bacteria can alter its phenotype according to its environment to maximise survival rates. The underlying DNA will remain the same, which means that phenotypic wide changes are available to certain types of bacteria in times scales that would be impossible through DNA mutations. A further example is that of the aforementioned cellular differentiation. Typically, terminally differentiated cells multiply thousands of times throughout their life span, and they must transfer this memory of what the cell is. It is hypothesised that this is achieved via DNA methylation or lack thereof and histone modifications (Levenson &

Sweatt, 2005). It is also hypothesised that cellular memory directly translates to organism wide memory which is associated with the central nervous system.

2.8.4 Higher Order Gene Regulation

The higher order genetic structures, specifically chromatin and how it modifies its structure, gives a higher level of genetic control than would otherwise be possible. It is because of this that a level of genetic memory becomes useful because it can be accessed only when needed. This is a sense that is promoted in epigenetics, that there is a wealth of possible genetic states available, but most of which are infrequently used. However, when the internal and external environment dictate a specific change is required, the epigeneome can modify gene expression with high levels of speed and precision. An example showing how higher order genetic structures influence gene regulation can be seen in Figure 2.15.

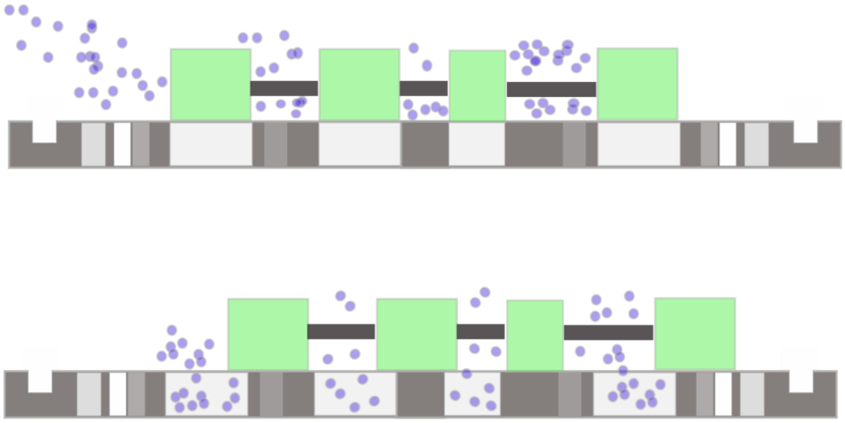


Figure 2.15: A simplified model showing how a small chromatin modification can regulate many genes, demonstrating a reduced dimensionality controller. Reducing the dimensionality occurs when a series of actions can be performed via the alteration of a more simplistic component. In this illustration, there are four genes and four chromatin molecules. In the top image, the chromatin molecules are blocking access to the genes. Via a single modification, the chromatin molecules can shift (as they are linked together), allowing access to four genes. The chromatin shift can be as small as a single modification. Hence, a single modification can effect the expression of many genes, reducing the dimensionality. These genes are typically organised together, so that a single switch can activate a biological process.

Previously in section 2.4.1, gene regulation has been described in reference to genes alone. Highly robust regulatory units have been found using genes alone, however, epigenetic mechanisms provide a richer range of structures in which to influence regulatory behaviours. This translates to a greater range of regulatory behaviours which are only available to organisms which contain higher order epigenetic structures.

2.9 Summary

This chapter has three principle aims. Firstly, to provide an underpinning of the field of genetics, and to demonstrate how this underpins that of epigenetic functionality. Secondly, to demonstrate the types of epigenetic mechanisms that have been discovered, and how they function to control gene regulation and facilitate a phenotypic plasticity. Thirdly to promote the idea that when looking at the natural world through epigenetic-tinted goggles, it appears separated into instruction (DNA) and control (epigenome). For all sakes and purposes, the epigenome is as ubiquitous as the genome, with all living organisms having aspects of epigenetic control. The higher order genetic structures typically found in eukaryotes such as chromatin also demonstrate that the genome is partitioned into genetic blocks which the epigenome can control. This creates an interesting characteristic, that chromatin modifications are a reduced dimensionality controller than that of the genome modifications alone.

Epigenetics is one of the principal underpinnings of this thesis, and this chapter amalgamates epigenetic mechanisms with the biological structures and processes described earlier in the chapter to generate an understanding on how epigenetics functions from the ground up. Moreover, this chapter has highlighted specifically why epigenetics in beneficial in nature, and this information is used as inspiration for the artificial epigenetic network, which is the focus of this thesis.

The following chapter describes the qualitative properties of biological systems and how these relate to specific traits such as evolvability and robustness.

Chapter 3

Properties And Characteristics Of Biological Systems

Contents

The purpose of this chapter is to describe the underlying principles of complex biological systems. This chapter focuses on two key aspects which are closely linked, evolvability and robustness. By generating an understanding of evolvable and robust systems in biology, it can provide a solid underpinning on which to base computational analogues of biological systems to best extract these beneficial properties.

3.1 Evolution

In its most basic form, evolution is the process of a single entity undergoing a change. Structures such as mountains, rivers and glaciers all change over time, and their present form can be considered to have evolved. However, in this chapter, were are concerned with biological evolution based on populations. Charles Darwin's book "The Origin Of Species" (Darwin, 1859) is considered by many to be the foundation of the theory of evolutionary biology. Biological evolution is the change in characteristics of a species or individual over time and has resulted in the diversification of species on Earth today. Evolution describes the progression of positive traits within a species, not as on overseeing power, but because positive traits are most likely to be preserved within a population. Richard Dawkins (Dawkins, 2006) subscribes to this view by describing genes as selfish, emphasising that genes have no ability to in fact *be* selfish; but they behave as if they were. This provides the backbone to the theory of natural selection. Natural selection, in terms of a population of species, is the preservation of certain replicating entities, in which each of these entities have an influence over their probability of being replicated. This probability is dependent on how good these entities are at surviving, living, and passing on their DNA, which in turn, is a product of their genetic structure. An entity's genetic structure is a biological mapping of a phenotype; hence, the phenotypes that are most successful have a higher probability of passing on their DNA, and as such those phenotypic traits will percolate through the population.

There are three general components to an biologically evolvable system. The entity, its genetic representation and the mechanisms in which to facilitate variation. The variation of an entity can occour in many different ways, however, the most pervaisive over all organisms is that of mutation. Mutation is the pseudo-random change in genetic information (typically a single nucleotide base). There are several possibilities of how this can occur; these include damage to the physical structure of DNA and interaction with viruses. Mutation is a constant process that creates different phenotypes and genetic data which was not part of the previous population. This allows lineages to be constantly adapting, acquiring new previously unknown phenotypic traits. This is one of the principal methods by which bacteria can generate resistance to antibiotics in such a relatively short time frame (Wilson et al., 2011; Lenski, 2010). In addition to this, there is genetic recombination, a method to produce variation which occurs in two main forms, vertical and horizontal gene transfer.

3.1.1 Vertical Gene Transfer

Vertical gene transfer (Figure 3.1) is a method of passing genetic data from parent to offspring (sexual recombination). It is most commonly associated with eukaryotes and it serves as a way of taking DNA from different sources (parents within the same species) and producing a

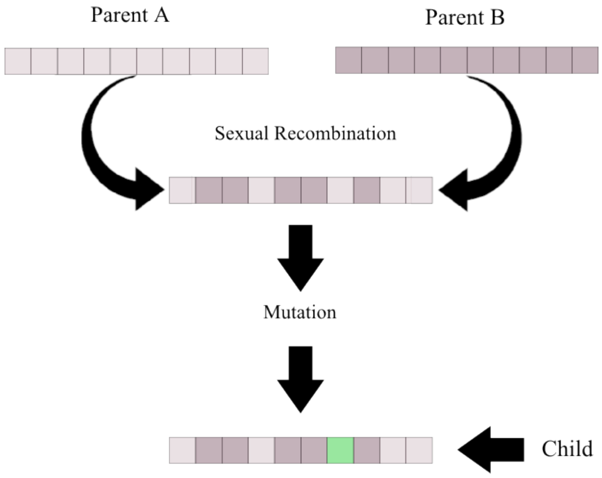


Figure 3.1: Vertical gene transfer from parent to child. Initially there are two parents. Recombination and mutation can create a child with properties of both their parents, modified by random change.

viable pheonotype with some traits from both parents. The purpose of doing this is to create children which are most likely to survive by having similar traits to their parents, yet not be identical to either. Hence the term "survival of the fittest", where only the fittest individuals tend to breed and in turn only the fittest individuals pass on their genetic material. The general consequence of vertical gene transfer is that over successive generations, the newest generation will be fitter than the last. This means that the species tends to be continually adapting, becoming better optimised within its environment.

3.1.2 Horizontal Gene Transfer

Horizontal gene transfer is the method of genetic exchange used between prokaryotes and single celled eukaryotes; however, there are also examples of horizontal gene transfer in multicellular eukaryotic species (Ros & Hurst, 2009). In particular, bacteria use processes such as conjugation to exchange DNA with another organism (Figure 3.2). Bacteria can also incorporate DNA directly from the environment, without it necessarily being passed from another organism.

Horizontal gene transfer occurs on a much smaller time scale than vertical evolution. Bacteria using horizontal gene transfer are capable of multiple DNA transfers over a single generation (cell division). Research which looked at the evolvability of 12 separate *E.coli* populations since 1998 has shown that they all developed certain mutations, and one population evolved to metabolise citric acid (Blount et al., 2008). During this time it is thought that *E.coli* have experienced every possible point mutation, of which only 100 reached fixation in any

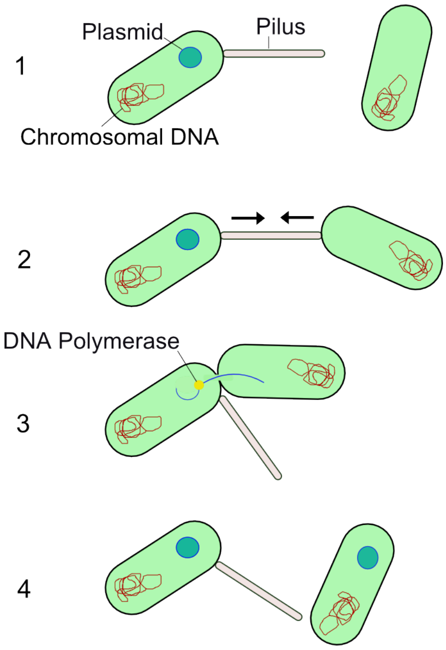


Figure 3.2: The four main steps bacteria use to pass genetic material between each other via bacterial conjugation. The first stage is for the bacteria to use its pilus to survey its surroundings. In the second stage, the bacteria finds another bacteria and attaches its pilus to it. Then, the two bacteria are drawn together and they form a channel between each other in which to pass genetic material. Once the genetic material is transferred, the bacteria detach from one another.

population. From this it can be seen that horizontal gene transfer (Cooper et al., 2003; Blount et al., 2008) is an effective method of optimising smaller less complex organisms.

3.2 Evolvability

In order for evolution to be a viable process, the entities within a population must be evolveable. This trait, known as *evolvability*, is a description of an entity's ability to evolve within an environment towards a positive gain (Kirschner & Gerhart, 1998). Specifically, evolvabilty is described as an entities ability to reduce the lethality of mutations, whilst minimising the number of mutations required to generate meaningful phenotypic variability (Kirschner & Gerhart, 1998). Evolvable systems are able to accept change without critical failures, and moreover, they are also able to hold onto existing traits which are beneficial. This also flows into another important facet of evolvability, and that is the ability to pass on change. In this sense evolvability can be seen as an evolved characteristic, because evovlable entities are more able to adapt to new environments, adopt variations and in turn survive.

Evolvability is intrinsically linked with robustness which is described in the following section.

The two of which have a complex yet pervasive relationship throughout biology. (Wagner, 2008) states that :

> “Understanding the relationship between robustness and evolvability is key to understand how living things can withstand mutations, while producing ample variation that leads to evolutionary innovations. ”

It is because of this that evolvability is difficult to discuss without referring and defining robustness.

3.3 Robustness

Robustness, as defined by (Kitano, 2004) is a property that allows a system to maintain its functions against internal and external perturbations. Robustness is distinct from homoeostasis (section 3.3.4) because it is concerned with maintaining the functionality of the system rather than states of that system (Kitano, 2007). Robustness, much like evolvability, is a facet of a biological system which is not the product of a single structure or behaviour. It is the complex interactions between systems, structures and behaviours out of which emerges the higher level behaviour that is robustness.

Robustnes is observed over many different levels of abstraction from biochemical networks to ecosystems. Within this thesis there is a focus on the creation of biologically inspired computational networks, and in a similar way, these computational structures must be both evolvable and robust. It has been shown that robust systems, regardless of their underpinnings can be designed to be robust and evolvable (Kitano, 2004; Lones, 2004; Gershenson et al., 2005). Within this work, there is an emphasis on staying faithful to biological underpinnings wherever possible to best allow for the emergence of beneficial complex behaviours.

Although robustness cannot be attributed to one specific factor, there are multiple facets of biological systems which are seen to be key contributors to robustness. These are modularity, redundancy, and decoupling (Kitano, 2004, 2007; Lones, 2004; Gershenson et al., 2005; Ancel & Fontana, 2000).

3.3.1 Modularity

Modularity is a mechanism whereby perturbations can be contained to minimise the effect on the whole system (Kitano, 2004). A module can be seen as a functioning unit which is separable from other entities. The identification of such modules is derived by looking at functional, evolutionary or topological criteria (Hintze & Adami, 2008). A key reason for modularity being a positive evolutionary trait is that the failure of modules does not correspond to failure of the organism. Biochemical networks, be they neural, metabolic, genetic or signalling tend to have high levels of modularity (Newman, 2006) (Figures 2.9 and 3.4).

3.3.2 Redundancy

A further aspect of robustness is redundancy, which specifies that a functioning unit is encoded by more than one gene. This positively affects robustness, because other systems can replace a failed system However, this negatively effects resource requirements (Kitano, 2007). This allows the evolution of one system, without the risk of critical failure. The robustness of an organism, including modularity and redundancy exists over many levels of abstraction, from the lowest biochemical networks, to humans, who have redundant back-ups, such as two kidneys. Additionally, these levels increase above individuals, towards large social and computational networks (Lones, 2004).

Redundancy can be incorporated into many systems, and is typically designed for systems which are safety critical. In (Kitano, 2004), the analogy of autopilot systems in aviation using three different systems to ensure it remained operational. These three systems had the same purpose, but were designed differently to ensure common mode failrue would not prevent the functionality of the entire system. This is a key philosophy as it shows that certain elements of robustness can be hard coded into place, and this means these elements can be captured outside of a biological setting.

3.3.3 Decoupling

Decoupling in biology is the idea that the phenotype of an organism or a structures functionality is the product of an indirect representation of that organism or functionality. One of the key examples of this is the decoupling of genotype and phenotype (Kitano, 2004, 2007; Lones, 2004). This seperates low level variation from high level functionalities (Kitano, 2004). This provides a form of robustness in which to accept change yet maintain diversity.

This concept has been incorporated into artificial gene regulatory networks, and it has shown to improve the computational evolvability of such networks (Reil, 1999).

3.3.4 Homoeostasis

Homoeostasis within an organism refers to the maintenance of an adaptive balance of an internal environment, and is a key requirement for any living organism (Muehlenbein, 2010). In order for homoeostasis to be feasible, there has to be a dynamic response by an organism to external perturbations such as environmental change. Gene regulation is the primary control system that facilitates homoeostasis.

Homoeostasis is a complex process involving many different biochemical networks, and is a constant process. On the small scale, every time an organism absorbs food, that food needs to be digested before it can be utilised. This requires alterations of gene expression to ensure the correct concentrations of enzymes are present. On a much larger scale, the immune system is capable of significant changes upon detection of any substance which it considers alien. This can prompt an organism-wide change in gene expression to best deal with this threat. This is done whilst trying to ensure all other systems cooperate in such a way to keep a homoeostatic environment possible. Homoeostasis does not refer to a fixed point, or a specific set of behaviours. It is an equilibrium between the organism and environment.

3.4 Emergence Of Complex Behaviours *In Silico*

The aim of this thesis is to capture real world biological behaviours, specifically that of epigenetics in a computational representation. Representing a biological system *in silico* is a complicated process because the behaviours of complex biological systems (robustness, evovlability, self organisation, complexity) are abstracted from the behaviour of their constituent components. The important question is where do these behaviours originate, and how can they be transferred to a computational model?

It has been an underlying theme throughout the studies of complex systems that they are an emergent property of their underlying components and the interactions between these components. Banzhaf (2004) succinctly states that

> “The essence of this idea of bio-inspiration is emergence (of functionality) through (possibly unforeseen) interactions among components. Thus, instead of isolating the sub-parts of our systems in order to get ‘clean’ functionality, we should rather count on the interactions for securing the functionality”

This is a theme echoed by many scientists in the field (Clegg et al., 2007; Reil, 1999; Bull, 2012). It has also been shown that many features of real-life development, such as cyclic gene activity, differentiation into multiple cell types and robustness may be inherent properties of the system rather than necessarily specified in a top-down approach (Reil, 1999). From this it can be argued that it is perhaps more plausible to generate desired behaviours without explicitly coding for them, assuming that they arise through emergence. This theme is generally embodied in artificial gene regulatory networks, where their function is a property of smaller interacting components.

There are other aspects of gene regulation in biology that are somewhat difficult to translate into the computational domain. One key example is time, and how interconnected components interact over time. The reason this is difficult to model is that biological connections between elements of a system are temporal, being connected at certain instances, and unconnected at others (Holme & Saramäki, 2012; Hoffmann et al., 2002). In this sense, different variables are part of the network at certain times, and not others.

The main issue is that currently no model incorporates a perfect description of gene regulation in nature. There are limits to what can be achieved, how accurate models can be made, and what level of abstraction is most effective.

3.5 Complex Systems Analysis

Because the philosophy behind artificial gene regulatory networks is to not explicitly code for higher functions, but to let them arise as emergent properties, it is essential to understand its dynamic functionality. This is frequently not possible by looking at the architecture alone, as the network is more than the sum of its parts. Therefore there needs to be a methodology in which to determine network function. One of the most basic methods of network analysis is to look at the network's static structure. This is where all the individual components of the networks are plotted as nodes within a graph, and the directed edges of that graph correspond to connections. An example of this can be seen in Figure 3.3.

Static network analysis is prevalent in biology, as it allows the visualisation of interacting units. This generates an understanding of modularity within the network, which further provides information on how integral specific units are to the network. However, there is only limited information that can be attained this way. In order to generate more meaningful information about the networks, we have to look at their dynamical properties. A key method to achieve this is to look at the attractor space that a network's dynamics follow. The

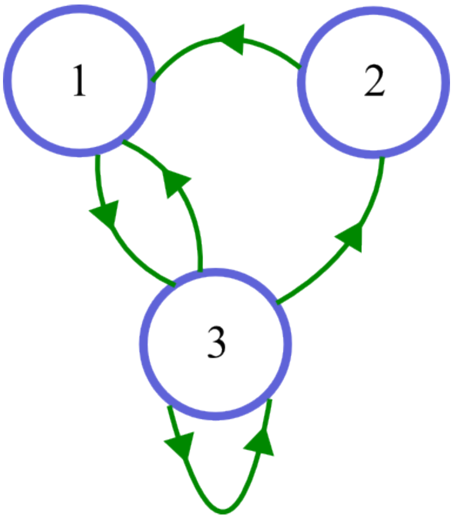


Figure 3.3: A static analysis of a simple network with 3 nodes. This provides a visual description of the connectivity within the network. As can be seen, node 1 is connected to node 3, node 2 is connected to node 1 and node 3 is connected to itself, node 1 and node 2.

attractor space is a set of states which a trajectory follows. The trajectory is the path within the attractor space corresponding to the behaviour of a system at a certain time. A system's current state is defined as a point within the attractor space. An example of attractor structure in a real world biological network can be seen in Figure 3.4, where a yeast model of regulatory functions has had its attractor space plotted. Each state within this model is shown as a dot, and each transition is modelled so that the next state of that system can be mapped. It can be seen that there are 7 attractor basins. A basin is a set of states which lead to an attractor. Some attractor basins may only be accessible if certain initial condition are met.

The attractor space shown in Figure 3.4 is a static model of a dynamically executed system showing all network states. A more dynamical example of network analysis can be found by looking at the Lorenz equations (Lorenz, 1963). The Lorenz equations have had significant impact in dynamical systems theory because they succinctly show the emergence of chaotic, complex dynamics (Figure 3.5). The Lorenz equation, along with sample parameters that produce chaotic behaviour are stated in equation 3.1.

$$\begin{aligned} \dot{x} &= \sigma(y - x) & \sigma &= 10 \\ \dot{y} &= \rho x - y - xz & \rho &= 28 \\ \dot{z} &= -\beta z + xy & \beta &= 8/3 \end{aligned} \tag{3.1}$$

The butterfly-like (Figure 3.5) attractor space achieved by plotting the Lorenz equation in

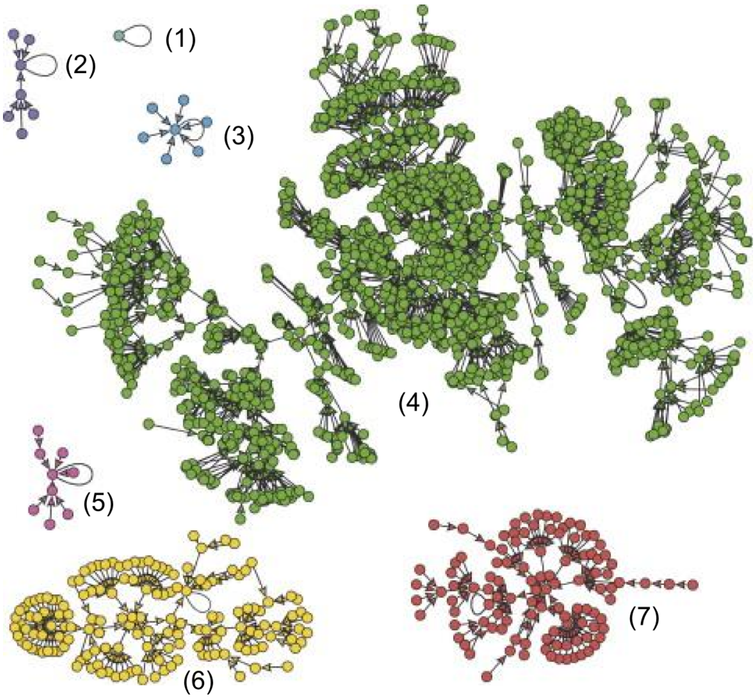


Figure 3.4: An illustration of the yeast model developed in (Li et al., 2004) and visualised in (Willadsen & Wiles, 2007). Each node in the graph represents a state of the system, and an edge between nodes represent a dynamic transition between states, with a loop showing a cyclic attractor.

three dimensional space consists of an infinite set of unstable orbits around two lobes. The trajectory through this space is chaotic, as the orbits are only followed for a set time before switching unpredictably (Viswanath, 2003; Lones et al., 2010). The attractor exists in three dimensions, and can therefore be plotted. This produces the graph in Figure 3.5. This is a complex emergent behaviour, which is only visible when analysing the network dynamics over time, and is not simply traceable to the three equations. The trajectory is also extremely sensitive to variances in initial starting conditions, a hallmark of chaotic systems.

In a mathematical model where all components of a dynamical system are accessible, analysis of the interaction of these variables and the overall system can be straightforward. This is more difficult when not all of the system data is available; how is it possible to produce an accurate model of the system dynamics? This is a commonplace problem when analysing real world dynamics. Takens' theorem (Takens, 1981) demonstrated that observation of a single variable can be used to reconstruct the qualitative properties of the attractor of the system (Huke, 2006). This is, however, dependent on the coupling of the components within a system.

A practical example of Takens' theorem applied to the Lorenz attractor is illustrated in Figure

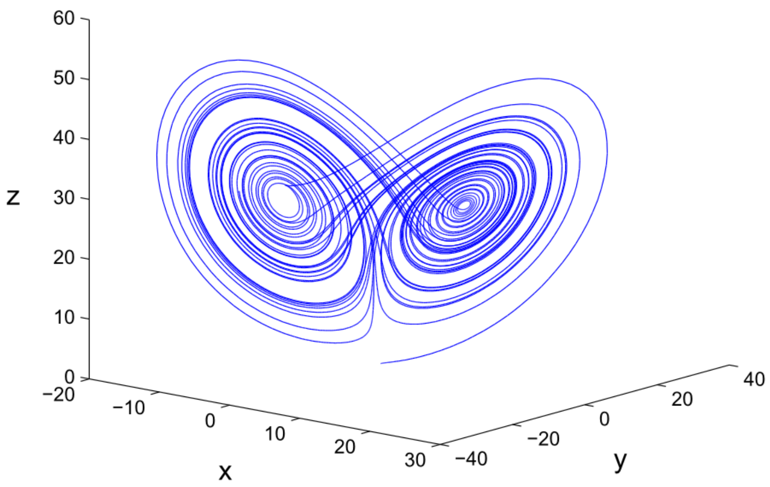


Figure 3.5: An illustration of the Lorenz attractor. The Lorenz attractor switches between the equilibrium points chaotically, and this behaviour cannot be deduced from the equations alone. Hence, plotting the trajectory allows a visualisation of the emergent behaviour.

3.5. Takens' theorem tells us that we can take a single observable variable from this system, and use it to reconstruct the attractor which preserves the mathematical properties of the Lorenz system. To show this, take the x coordinate from the Lorenz attractor in Figure 3.5 and use time delay embedding (Kantz & Schreiber, 2004), which is described in equation 3.2, to transform the data into three dimensions so it can be plotted. The delay embedding is created by taking a vector S$_n$ (an observable variable over time), and taking the embedding dimension m with a delay r.

$$S_n = (S_n - (m-1)r, S_n - (m-2)r, ..., S_n) \tag{3.2}$$

Time delay embedding is a method of transforming low dimensionality data into higher dimensionality data in order to reconstruct the phase space. This is done by taking a variable within a number of samples, and introducing a fixed delay to generate further dimensional points (the data represented in more dimensions). This means single dimensional data can be translated into 3 dimensional data, which can be plotted to observe the phase data in three dimensions. The reconstructed phase space using the x variable in Figure 3.5 can be seen in Figure 3.6.

The visualisation of the dynamical behaviour of systems is a useful tool because it facilitate the direct analysis of how perturbations can affect systems. This adds another layer of

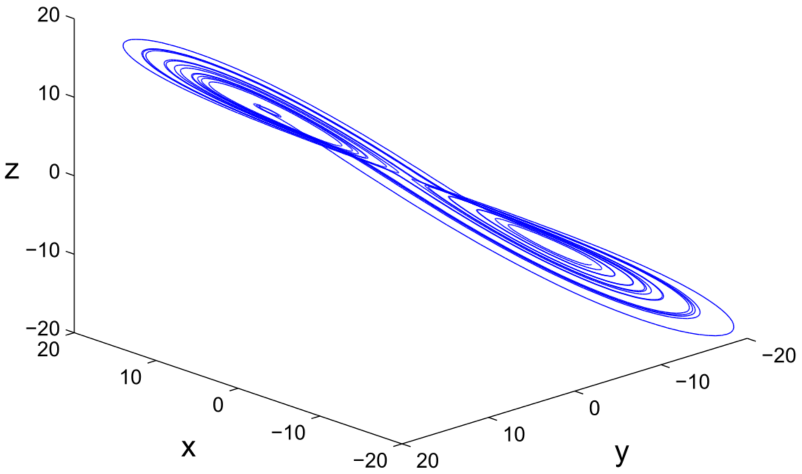


Figure 3.6: A reconstructed phase space using only the x time series values from the Lorenz attractor in Figure 3.5 using Taken's theorem. This time series data has been translated into three dimensions using time delay embedding. It is apparent that although 2/3 of the Lorenz attractor data is missing, the dynamical structure of the system is preserved.

dynamical analysis which is possible when observing the networks over time. The purpose of this section is in part to highlight an example of dynamical network analysis. However, there is not a fixed method of analysis which is suitable for all systems, and it is often complex to decide how to best describe the dynamics of a system.

3.6 Summary

This chapter provides the bridge between the biology and the computation of this thesis. It describes abstract concepts such as robustness, evolvability and the emergence of these properties, establishing a basic framework in which to base the computational models in the following chapters to allow for the highest chance of capturing these complex behaviours *in silico*. In addition the foundation of complex systems anaylis has been laid in order to best understand the functionality computational models which are created.

In the following chapter, the methods for artificially evolving the computational networks are discussed and evaluated in terms of the themes highlighted in this chapter.

Chapter 4

Evolutionary Algorithms

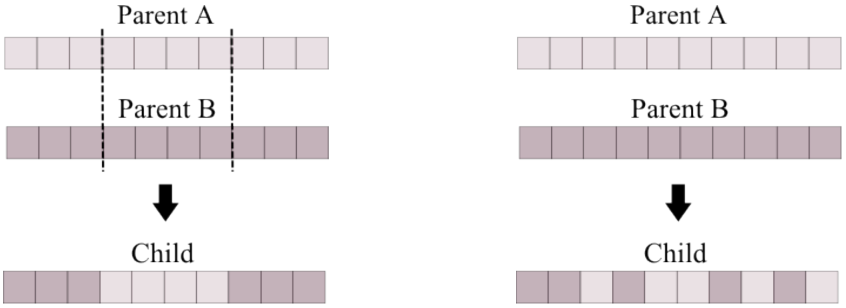
Contents

In this chapter, the focus is on evolutionary algorithms, which draw inspiration from natural evolution. This chapter outlines some of the better known evolutionary algorithms and describes their faithfulness to their biological underpinnings and how they represent evolvable entities. This is a key aspect of the work within this thesis, as evolutionary algorithm will be the tools used to evolve the gene regulatory networks and are therefore pivotal when it comes to capturing high level emergent properties.

4.1 Genetic Algorithms

Genetic algorithms were one of the later additions to evolutionary computation, originating in 1975 (Holland, 1975). Since then genetic algorithms have become one of the most widely adopted forms of evolutionary computation. The original purpose behind genetic algorithms was to observe evolvability and emergence, not to create an optimisation tool. In their original form, they were used to optimise binary data, but since then have been used on a range of data types. Genetic algorithms commonly have a distinct genotype and phenotype. The genotype is the data which is to be evolved, and the phenotype is its computational behaviour. An individual is evaluated according to its phenotype, but genetic operators are applied to its genotype.

Genetic algorithms use a recombination operator, as well as mutation and selection operators. The recombination operator, referred to as crossover, is a computational analogue of the vertical gene transfer model; however, certain genetic algorithms use a recombination operator inspired by horizontal gene transfer (Harvey, 2011). Generally, there are two main variants of recombination operators used in genetic algorithms, N- point crossover and uniform crossover. N-point crossover creates children based upon defined sections being used from each parent. In uniform crossover, crossover points are created with a certain probability of passing information on to the child (Figure 4.1).



(a) N-point crossover

(b) Uniform crossover

Figure 4.1: An illustration of n-point crossover (where N = 2) and uniform crossover, showing how children are created using the respective recombination strategies.

In terms of selection, there are three commonly used selection strategies: rank based, fitness proportional and tournament selection. Rank based selection scores all individuals in terms of their relative fitness within the population, which removes the absolute differences between each individual's fitness. Hence the difference between many very close fitness scores could be multiple ranks, and this also means that an individual with a fitness much higher than that of the rest of the population will always be just one rank higher than the next best solution, which could potentially punish that solution (Back et al., 2000). Fitness proportional selection maps the fitness of each individual using a scaling function. An example of this is roulette wheel selection, where each individual is assigned a section of a roulette wheel, the size of the section proportional to fitness. Therefore, the fitter individuals have a higher chance of being selected (Goldberg & Deb, 1991). Tournament selection selects a random number of individuals which compete in a tournament. The winner of this is selected as a parent. The evolutionary pressure placed on the population can be modified by changing the tournament size. Large tournaments make it hard for weaker individuals to be selected by increasing

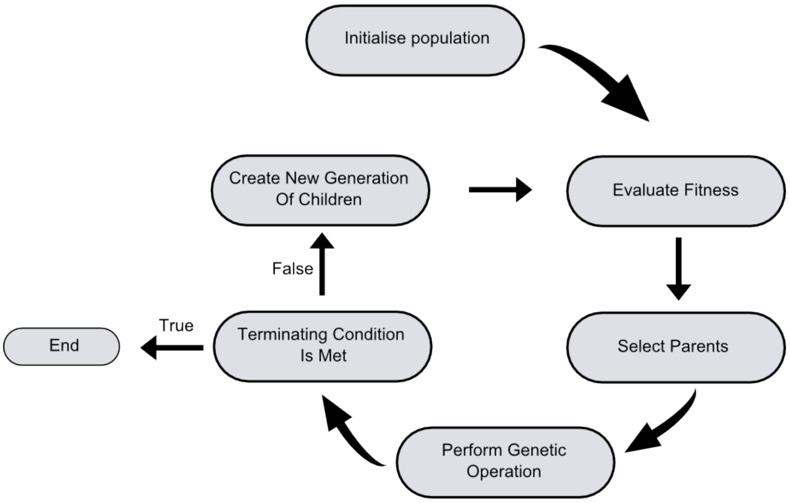

Initialise population
Evaluate Fitness
Select Parents
Perform Genetic Operation
Terminating Condition Is Met
True
End
False
Create New Generation Of Children

Figure 4.2: A generalised evolutionary algorithm.

selection pressure, whereas small tournaments make it easier by reducing selection pressure (Goldberg & Deb, 1991).

Mutation in genetic algorithms is dependent on the representation of the genome, and its corresponding data structures. If a genome uses binary numbers, the mutation operator will flip each of the bits with a probability according to the mutation rate. If the genome uses real numbers, a new number may be selected at random within a set range, or from a distribution centred around the original number. Mutation is usually applied to all individuals unless the selection strategy is elitist, then those elite individuals will be copied verbatim to the next generation (Back et al., 2000).

The basic execution of a genetic algorithm begins with a randomly initiated population of size N. After initiation the individuals within the population are then assigned a fitness which corresponds to how well a task has been completed. Then, the parents are selected according to a selection strategy, and the children created according to the recombination operator. This is repeated until there is a new population of size N. The new population is subjected to the mutation operator, and afterwards, the new population becomes the current population, and the steps after initiation are repeated for a set number of generations or until the stopping criteria are met (Figure 4.2).

In terms of creating the optimum environment in which to evolve gene regulatory networks to best capture high level emergent properties, there are several key facets of genetic algorithms which are well suited to this. Firstly, a genetic algorithm makes has no prerequisites about the individuals that it is evolving, all it requires is the data of the individual and a method for evaluating its fitness. This allows for there to be a decoupling between the data (genotype) and the function (phenotype) of the individuals which will potentially contribute to the emergent properties of the networks. Secondly, recombination is similar to that of real world biology, with mutation and crossover form other individuals occurring at each generation. Moreover, because its a population based algorithm, information can be gathered from the entire population, detailing the progression in reference to the task throughout the generations.

4.1.1 Non-Dominated Sorting Genetic Algorithm II

Non-dominated sorting genetic algorithm II (NSGA-II) is one of the principle multi-objective genetic algorithms (MOGA) (Deb et al., 2000, 2002; Coello et al., 2007). Within optimisation tasks, there are often multiple objectives which need to be optimised, and these objectives are often conflicting. For example, the balancing of risk and reward, where higher reward is more optimal, but is often twinned with more risk. In this sense, there is frequently no optimal answer, but a range of answers which could be considered optimal. Other such MOEAs exist, most notably strength Pareto evolutionary algorithm (SPEA), vector evaluated genetic algorithm and niched Pareto genetic algorithm.

NSGA-II allocates a rank to each member of the population based on dominance. An individual can be considered to dominate another if it is better in at least one objective, and not worse in all others. All instances of the population that achieve this will become part of the first non-dominated front (Figure 4.3). This process then repeats with the previous non-dominated front omitted, to produce the second non dominated front and so on.

A further operator within NSGA-II is that of crowding distance. Crowding distance is a measure of density of individuals within a non-dominated front. This measure is designed to create a uniform distribution of individuals across a non-dominated front. Within the population, each individual will have a non-domination rank and a crowding distance. The partial order is that individual i is greater then individual j if it has a better or equal rank, or has a better crowding distance (Deb et al., 2000, 2002; Coello et al., 2007).

NSGA-II provides a more realistic view of evolution in that fitness is not a single factor, but a multiple of factors. Individuals are better at some things than others, and NSGA-II represents this well. However, in terms of biological realism, it uses a forced elitism, ensuring that the fittest individuals are copied forward to successive generations. This philosophy in theory reduces diversity of the population, however, in terms of computational performance NSGA-II is extremely functional.

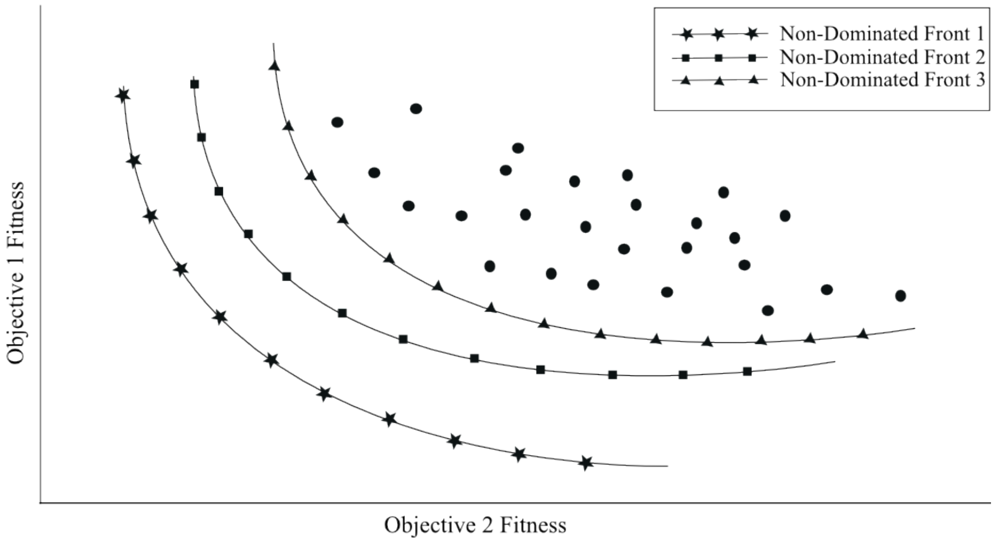


Figure 4.3: An illustration of three non dominated pareto fronts in an optimisation task where objectives 1 and 2 are minimised.

4.2 Genetic Programming

Genetic programming (GP) is an evolutionary evolutionary algorithm used to design programs (Koza, 1992). In conventional GP, a program is represented as a tree structure (Figure 4.4). Traditional GP requires a predetermined set of symbols (terminal and non-terminal set) which can be used to create the tree, as well as a fitness function to determine the fitness of a program. The initial population is created randomly by assembling members of the terminal and non-terminal sets into tree structures. The GP tree was a natural structure for the representation of programs within the programming language LISP, which was first used to implement genetic programming.

Variants of genetic programming represent their programs as structures other than trees. One example is that of linear GP, which uses a list of instructions to describe a program (Brameier & Banzhaf, 2007). Another, Cartesian genetic programming (CGP), represents a program as a graph structure, encoded as a set of integers (Miller & Thomson, 2000). CGP was designed to represent electronic circuits but has also been used to represent general programmable structures.

Genetic programming has similar genetic operators to those found in genetic algorithms. The recombination operator functions by interchanging sub-trees between two parents to generate two children (Figure 4.6). Similarly, mutation replaces a sub-tree with a randomly generated sub-tree.

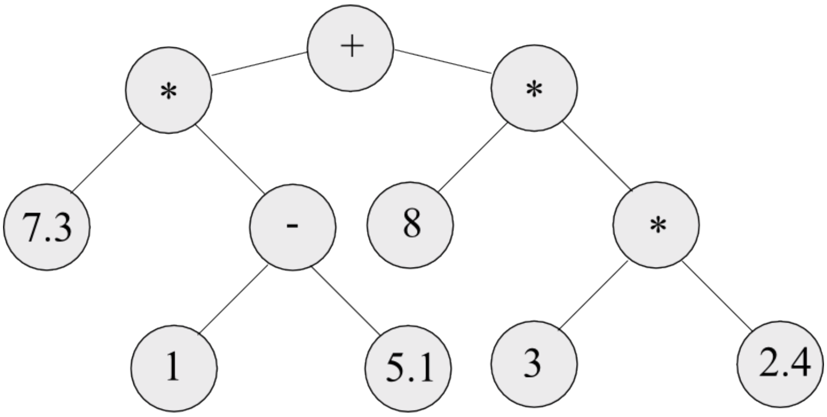


Figure 4.4: An example of a tree in genetic programming. The tree represents the equation (7.3 * (1-5.1)) + (8 * (3 * 2.4)). To determine the fitness of this program, a set of inputs can be provided and iterated through the program. The resulting outputs can be compared to the outputs required for the task.

Genetic programming has been applied to a wide range of problems including symbolic regression and the capture of behaviours found in metabolic pathways (Koza et al., 2000). However, despite the success of GP, there have been problems with its evolvability, which in part arises from sub-tree crossover not perform meaningful recombination (Lones, 2004). In addition, this may be due to the representation for the most parts being a tree structure, which limits the phenotype of the individuals. This also limits the ability to modify phenotypes to include new functionality which might not fit appropriately within tree structure.

4.3 Evolutionary Programming

Evolutionary programming was first defined in 1960 (Fogel et al., 1964) as a population based tool for optimisation. The individuals within the population are finite state machines with fixed structures. Each individual is treated as a fundamental component which is not structurally broken into sub-units; because of this, the recombination operator is not used. In order to ascertain the fitness for a given individual, that individual is placed into an environment and given a set of symbols as inputs, and the output is compared to the next input symbol. From this an error term can be produced which is accumulative over all input symbols. When the entire population has a fitness score, the best individuals are selected and mutated to produce children. Then the best parents and children are selected to become the next generation.

Evolutionary programming can be seen as a top down approach to optimisation, and takes inspiration from a more restricted view of evolution. In particular, the lack of a recombination

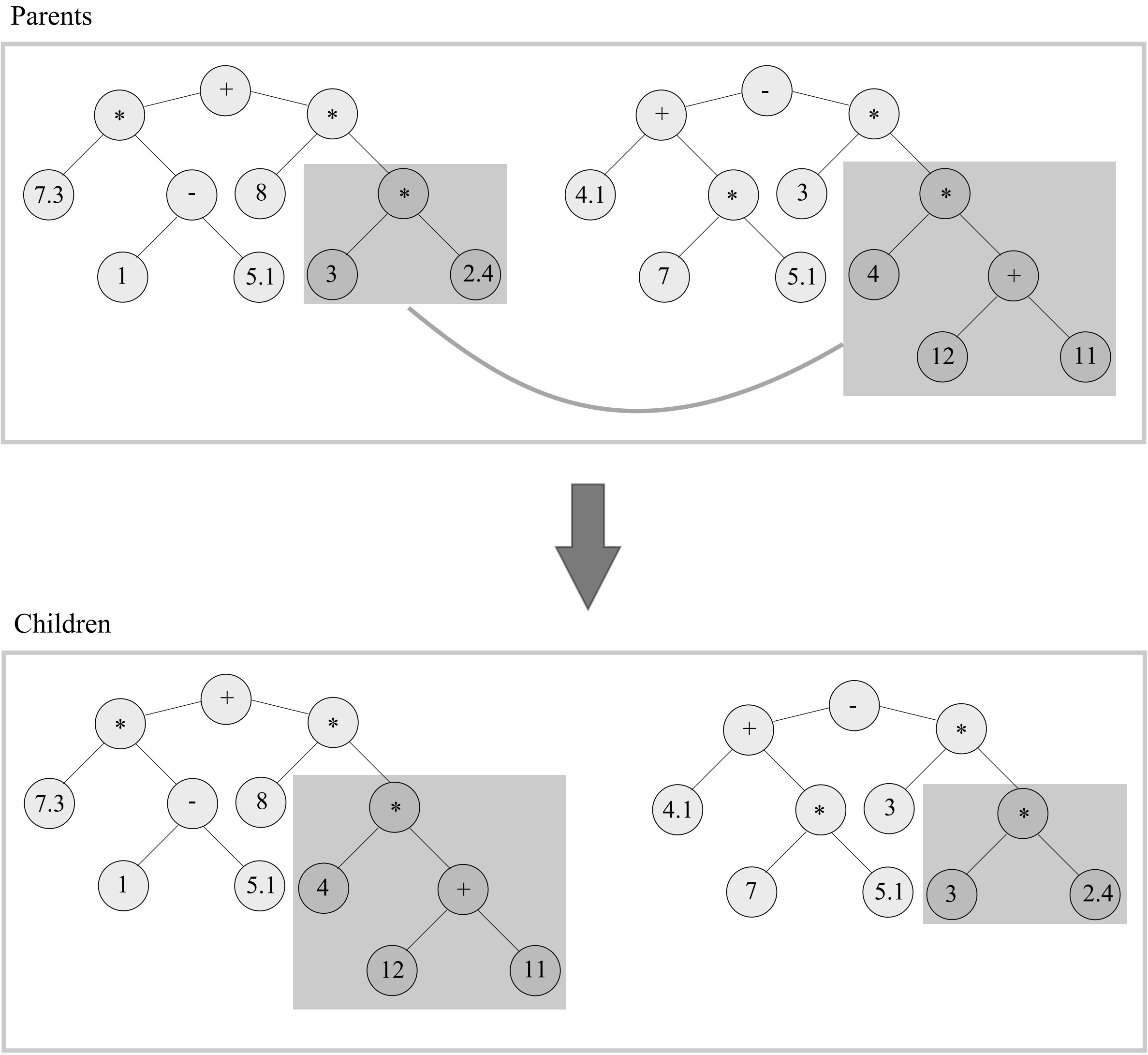


Figure 4.5: An example of the crossover operation with GP trees. Two sub-trees are selected and swapped between the parents to create the children.

operator means that individuals cannot take components and behaviours from other members of the population. In addition, evolutionary programming requires individuals to be finite state machines which can be limiting in terms of phenotype. These facets of the algorithm significantly limits the possibility of emerging evolvability using evolutionary programming. In terms of computation, evolutionary programming has faced criticism for slow convergence times. However, since its inception there have been advances, in particular the improvement of mutation strategies which have been shown to increase the rate of convergence (Yao & Liu, 1997).

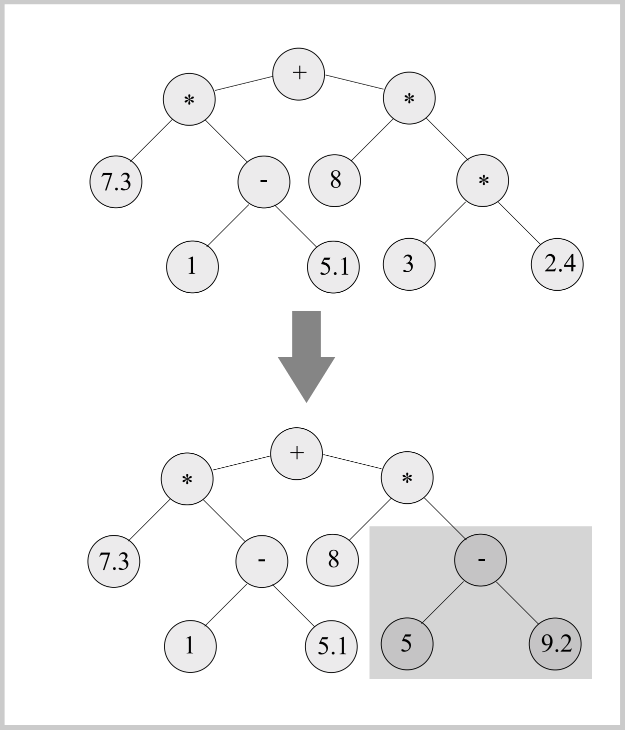


Figure 4.6: An example of mutation within a GP tree. A random sub-tree is replace with a sub tree from the GP tree.

4.4 Evolutionary Strategies

Evolutionary strategies originated in Germany at a similar time to Evolutionary programming (Back et al., 2000). They were created by Ingo Rechenberg, Hans-Paul Schwefel and Peter Bienert. Some of the earliest instances of evolutionary strategies had a single parent, and performed mutation to create a child. If the child was fitter then the parent, it became the parent. In this sense its origins were akin to a hill climbing algorithm. Since this time, two different selection strategies have become popular (Beyer & Schwefel, 2002). The first creates more than one child for a given parent, and to keep the population constant, the worst individuals are discarded. The second discards the parents regardless of their fitness, more closely approximating Darwinian evolution. Very soon after the inception of evolutionary strategies, a recombination operator was incorporated in most models which was able to create children with information from more than one parent Back et al. (2000).

Evolutionary strategies differ from genetic algorithms in two key respects. Firstly the selection and genetic recombination is usually done by selecting a parent, cloning them and using genetic recombination on the clones to create diversity. This is frequently done using an elitist strategy. Secondly, the population numbers are typically much smaller than that of a

Algorithm	Representation	Evolutionary operators
Genetic Algorithms	Real-values	Recombination, mutation, selection with optional elitism
Genetic Programming	Real-values or integers (typically tree based)	Recombination, mutation and deterministic selection with optional elitism
Evolutionary Programming	Real-values (finite state machine)	Recombination, mutation and elitist selection
Evolutionary Strategies	Real-values and strategy parameters	Recombination, mutation and deterministic or elitist selection

Table 4.1: A summary of the evolutionary algorithms detailed in this chapter. Adapted from (Hilder, 2010)

GA, limiting the diversity which can be held within any one generation.

4.5 Summary

In this chapter, four prominent families of evolutionary algorithms have been presented, each drawing inspiration from different aspects of biological evolution, combined with varying computational representations. A summary can be seen in Table 4.1. In terms of evolving gene regulatory networks some are more suited than others. Genetic programming and evolutionary programming are limited for this purpose because they require a fixed representation of either tree structure or finite state machine. This makes it more difficult to implement epigenetic like structures which may not necessarily fit with these prerequisites. Evolutionary strategies have no prerequisites in terms of computational phenotype, however, they generally take a limited perspective of population dynamics, frequently using a single individual to make multiple clones within a population. Taking these factors into account, genetic algorithms appear to create the best environment to allow complex properties to emerge from the evolutionary process. This is specifically down to their biologically realistic genetic recombination operators, that they have no prerequisites in terms of genotype (data) or phenotype (function) and that they have a more biologically faithful population based architecture. Therefore, genetic algorithms will be used in to evolve the artificial gene regulatory networks.

In the following chapter, the field of artificial gene regulatory networks is explored, highlighting their architecture and inspiration from biological systems.

Chapter 5

Artificial Gene Regulatory Networks

Contents

This chapter builds upon the themes within chapters 2 and 3, and moves the work into a computational domain. The purpose of this chapter is to highlight methods and techniques to create computational models of gene regulation, the properties these models have and the drive for doing this. This provides a basis on which to build upon these models in the following chapters.

Artificial gene regulatory networks (AGRNs) are computational models inspired by the interactions of genes within a cell. There are a range of computational architectures with varying levels of detail that can be described as AGRNs. AGRNs are usually designed for one of two reasons. Firstly, to model the interactions of genes in biological networks to better understand them (Karlebach & Shamir, 2008; Sturrock et al., 2013; Ribeiro et al., 2006). These

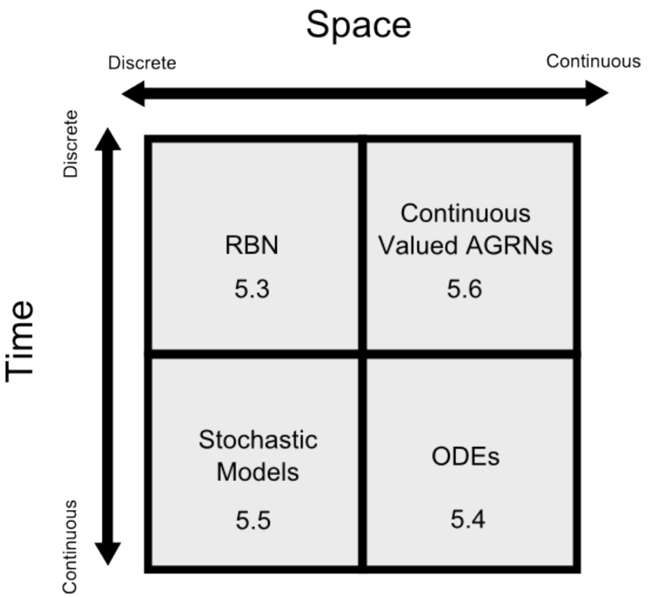


Figure 5.1: An illustration of how the models discussed in this chapter fit into the time-space domain and the corresponding sections in which they are explained within this chapter.

networks are typically high in detail in order to make the models as accurate as possible. Secondly, to build abstracted models which aim to acquire the beneficial properties of biological models *in silico*, without modelling them in precise detail (Kuyucu, 2010; Aldana et al., 2007; Lones et al., 2010, 2011). These models are typically used for solving complex computational problems and will be the primary focus of this chapter.

There are four types of artificial gene regulatory network that will be addressed in this chapter and can be seen in Figure 5.1. The reason for looking only at these four is that they represent each corner of the time and space map, and in this sense represent a vast majority of the possible AGRNs available. The time and space map is a representation of which type of data systems use, and how they process this data in terms of time. Time and space can be either discrete or continuous variables in simulation and depending on which variables a given model uses, can effect aspects of the networks properties. In the following sections we will describe a range of networks, and how their properties are suited to certain tasks.

5.1 Random Boolean Networks

The Random Boolean Network (RBN) (Kauffman, 1969) is one of the earliest computational models of gene regulation. It followed from the work on cellular automata (Von Neumann & Burks, 1966; Burks, 1969) and the idea that self organisation and stability can be found in randomly created networks. It is one of the most simplistic models of gene regulation,

Input Combination	Gene Expression
000	0
001	1
010	1
011	0
100	1
101	0
110	1
111	0

Table 5.1: Randomly initiated state transition table

Gene	Inputs	Gene	Inputs	Gene	Inputs
0	6,11,18	8	16,17,22	16	1,6,11
1	10,18,22	9	1,14,15	17	3,16,19
2	10,14,16	10	6,10,19	18	13,15,20
3	7,15,18	11	11,14,21	19	17,22,24
4	10,19,22	12	9,10,21	20	2,12,13
5	17,18,24	13	4,13,14	21	13,15,21
6	19,22,24	14	13,17,24	22	1,4,13
7	2,15,23	15	17,18,23	23	6,13,15

Table 5.2: The randomly initiated connections between the genes of size K (3 in this instance)

existing in the discrete space and time domains. An example of the execution of an RBN can be seen in Figure 5.2.

The RBN consists of a set of N Boolean states which represent a gene's activity level, where genes can be either active or inactive. The RBN has a global connectivity level K, which specifies how many inputs from other genes are required to update its own activity level. From this a state transition table can be randomly created (Table 5.1), specifying all possible combinations for a gene's next state based on its inputs. To execute a classic RBN, each node's state at $t + 1$ is calculated by taking each of the input values from connected nodes (Table 5.2) at time t and applying this to the Boolean updating rule associated with the node (Table 5.1). An example of the RBN execution can be seen in Figure 5.2.

During execution, RBNs can produce a wide range of complex dynamics ranging from highly ordered to chaotic (Stepney, 2009). Additionally, they express high levels of robustness to a range of perturbations including gene insertion and gene deletion (Aldana et al., 2007). These emergent properties show that it is possible to create robust, yet complex structures out of randomly ordered networks. This concept is pervasive throughout the study of biochemical networks; it is also a motivating idea behind connectionism, the idea that information pro-

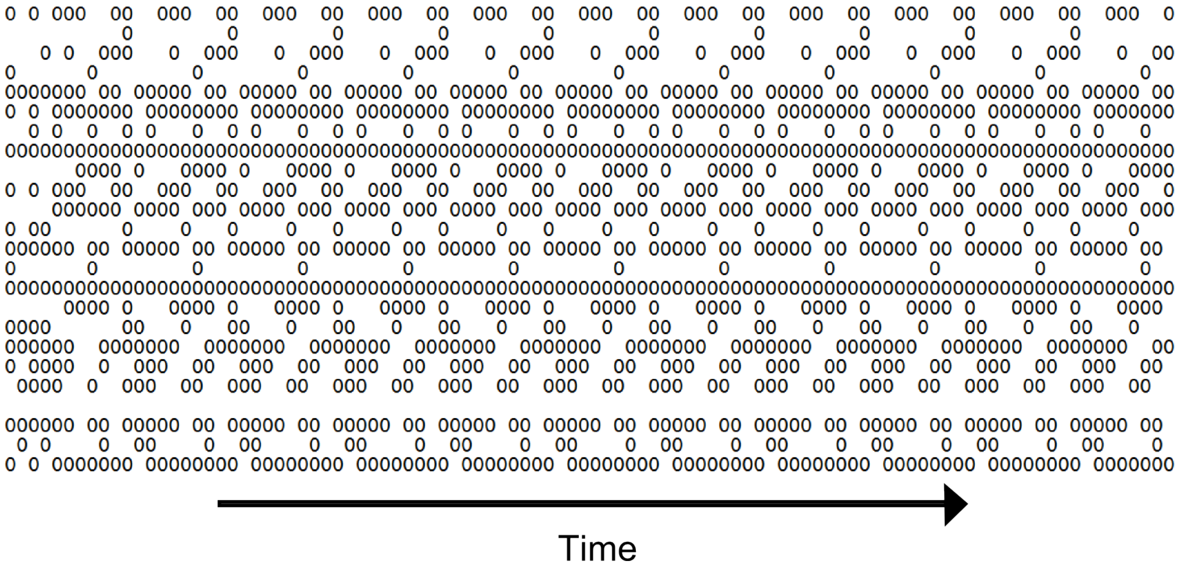


Figure 5.2: The execution of an RBN using the randomly initiated state transition table and connections from Table 5.1 and 5.2. A '0' represents a false Boolean value, and an ' ' (empty character) represents true. Moving from left to right in discrete time steps, it can be seen that initially there are low levels of order (steps 1-7) and thereafter, the network shows high levels of order.

cessing or intelligence emerges from the activity of a network of simple, non-linear elements (Lones et al., 2013).

RBNs are important in terms of network theory because they were one of the first examples of high level complex behaviours arising from the interactions between simple components. This made it possible to consider that interesting behaviours might not have to be specifically 'built' in a top down approach, but can arise out of the structures and interactions in a bottom up manner. This provides insight into how gene regulation in nature could have evolved to be so complex, but it may in fact be the property of more simplistic structures. Despite the RBN's simplicity, there has been a wealth of research which uses RBNs to not only model real world genetic circuits (Harris et al., 2002; Darabos et al., 2011; Bornholdt, 2008; Davidich & Bornholdt, 2008), but to infer knowledge about unidentified networks and how they might function (Akutsu et al., 1999; Gershenson et al., 2005). There has also been research investigating the use of RBNs as controllers for systems such as robots (Roli et al., 2011*b*, *a*) with some success. However, Boolean values are limited due to the need for discretisation of the real-valued data at a certain point during simulation, which in turn reduces the accuracy and usefulness of the model (Karlebach & Shamir, 2008).

5.1.1 RBN Variants

There are aspects of RBN behaviour which are very useful in certain research areas; however, although they exhibit emergent dynamics, they are simple models. This has lead to variants of RBNs which have been developed to include more biologically realistic elements. These additions are mostly variants to the updating schemes of the networks, as the classical RBN updated its expression values assuming all genes are connected at every time step - negating the temporal elements of real world biological networks. It is to be noted that, by definition, RBNs are discrete time and space models, and all variants still fit this definition.

An adaptation to the classic RBN model was to introduce multiple valued states, which removed these networks from the 'Boolean' characterisation (Solé et al., 1999). These networks were able to model complex systems, and added depth to the classical RBN. However, they were still limited in the fact that they used a narrow range of discrete values to model biological phenomena. The only significant benefit is that by increasing the number of states, there are more real world problems that can be more accurately mapped onto the network.

In (Harvey & Bossomaier, 1997) a novel RBN was created in an attempt to tackle the problem of non biologically realistic synchronous updating schemes. This model was known as the asynchronous RBN (ARBN). The asynchronous behaviour is created by randomly updating a gene at a given time step during execution. This had a profound effect on the dynamics of the networks, where the cyclical attractors of the classic RBNs no longer exist. There are however loose attractors, which keep the dynamics within a certain region of the attractor space, somewhat like a point attractor (Gershenson et al., 2005). The ABRN, taking inspiration from the asynchronous function of genes in nature, updates genes according to a given probability. Although gene updates are not synchronous, they are also not random. Hence, there is still the issue of ARBNs not being biologically consistent.

To deal with the issues of synchronisation in updating schemes, the Deterministic Asynchronous RBN (DARBN) was created (Gershenson, 2003; Gershenson et al., 2005). This model has a fixed probabilistic rule attached to each gene, consisting of two numbers (P,Q) generated randomly and remain fixed. An update to a given gene will occur when the modulus of the time step over P is equal to Q. The DARBN is an intermediary between the classic RBN and the ARBN, offering semi-synchronous updating schemes. The dynamics of the DARBN are more like that of the classic RBN than the ARBN, however, results from (Gershenson, 2003; Gershenson et al., 2005) imply that the networks are more capable of capturing complex dynamics and representing them effectively within the model.

5.1.2 RBN Analysis

The RBN model, despite being simplistic in structure, can become increasingly behaviourally complex as the number of nodes (N) and connectivity (K) increase. For any given network parameters (N,K), the number of possible networks is too large to exhaustively explore (Harvey & Bossomaier, 1997), as illustrated in equation 5.1.

$$\left(\frac{2^{2^k} N!}{(N-K)!} \right)^N \tag{5.1}$$

RBNs are the product of deterministic rules, and for a network of size N, there are 2^{N} possible states it can theoretically be in throughout execution. Therefore, eventually, the trajectory of the network will visit a previous state and will therefore fall into a cycle. These cycles are a form of attractors (Wuensche, 1998), known as cyclic attractors. If only a single point within an attractor is present, this is known as a point attractor or an attractor of length 1. The attractor space can be used to show how the network dynamics are changing over time, and the different phases the network operates within. The phases can be constructed analytically, or statistically. The phase space reconstruction show that the dynamics of RBN can either be ordered, critical, or chaotic, and these attractors are correlated with the connectivity within the network.

Statistical analysis has been conducted to ascertain how certain perturbations effect the dynamics of the network. In (Luque & Solé, 1997), an analytic determination of phase transitions was proposed which was able to monitor the effects of perturbations throughout execution. This allowed a much more dynamical analysis of network functionality than reconstructing the phase transformations alone.

There is a wide range of material on the analysis of RBNs; however ultimately, although they can capture complex dynamics, they are always limited by the discretisation of data to Boolean or multi state values. Because of this they are only applicable to certain types of problems.

5.2 Ordinary Differential Equations

Ordinary differential equations (ODEs) have been a principal modelling component of AGRNs, and are continuous in both time and space, making them more biologically realistic in terms of design philosophy when compared to the other networks in this chapter. Such models are most notably used to increase the understanding of biological gene regulation. An ODE refers to an equation which involves derivatives. Derivatives specify changes to a variable in terms of another variable. ODEs have been shown to be able to accurately model systems such as pendulum dynamics (Jordan & Smith, 1999), chaotic laser dynamics (Haken, 1975) and population dynamics (Kuang, 1993). ODEs as a gene regulatory network modelling tool exist on the continuous domain in both space and time.

ODEs have been applied widely within biological modelling due to their ability to accurately capture a wide range of complex dynamics within biological systems (Karlebach & Shamir, 2008). Typically, modelling gene regulation using ODEs assumes that the simulation exists as a 'mixed bag' where all components are free to interact at all times. The interactions between the components are modelled using rate equations. These rate equations specify, for example, how much protein concentration would change according to changes in mRNA concentration. An example of rate equations can be seen in equation 6.1 :

$$\tfrac{dx_i}{dt} = f_i(x), 1 \leq i \leq n \tag{5.2}$$

where x = $[x_i......x_n] \geq 0$ are the concentrations of molecules related to gene expression, such as transcription factors, and $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ a typically non-linear function such as a Michaelis Menten function (Karlebach & Shamir, 2008). The rate of concentration i is dependent on x. Such equations can also be extended to include time factors (De Jong, 2002).

To describe a system in terms of ODEs, multiple ODEs are coupled together, where they interact to replicate the properties of a genetic system. One of the issues with using ODEs to model biological systems is that they require high quality regulatory kinetic data in order to produce accurate models (Karlebach & Shamir, 2008). It is however possible to use generic functions to approximate causal relationships using methods such as bio-inspired evolutionary techniques, and although promising, the results produced commonly have an higher error rate than comparative methods and are computationally expensive (Moros et al., 1996; Katare et al., 2004).

Due to the structure of ODEs, it is very hard to study them analytically, and to ascertain if they are functioning appropriately are usually compared to the original data to see how well the regulatory kinetics have been updated. When there are many (usually > 5) interacting units within an ODE simulation, it becomes difficult to derive an picture of the overall function and how the interconnected parts are working together, even though methods such as Takens' theorem and time delay embedding can reduce this problem. However, it has been shown in a basic form that ODEs can reproduce a range of complex behaviour, but determining this when the data exists in high dimensions is difficult (Karlebach & Shamir, 2008).

As expressed earlier in the chapter, there are generally two reasons to model gene regulatory networks. Firstly to simulate biological gene regulation to further understanding, and secondly, to build abstracted models to capture the properties of gene regulation in a computational model. Using ODEs to model gene regulation is primarily for the former reason, but it has also been applied to the latter. In (Guo et al., 2009; Taylor, 2004), AGRNs which use ODEs have been designed to control a distributed set of robots for a specific task. Therefore, it has been shown that ODEs can be used both to model real world biology and also as computational models of controllers.

Despite their biological realism, modelling gene regulatory networks as ODEs is limited by various factors. Firstly, although they exist in a continuous time domain, all regulatory connections are permanent within the network, which is not akin to biology (Holme & Saramäki, 2012). Secondly, depending on the amount of variables that are being modelled, there has to be a trade off between computational expense and the number of structures that are being modelled. Thirdly, although it is simple to compare the outputs of the ODEs with biological data, it is difficult to ascertain why this behaviour occurs, and how true the simulations are to real world biology.

5.3 Stochastic Networks

Gene regulatory networks in nature are dynamic, stochastic structures which exhibit a range of complex feedback and feed forward control mechanisms over many different levels (El Samad et al., 2005). They are typically continuous time, discrete space models (Kierzek, 2002) which contain non-deterministic temporal structures. Due to their stochastic nature, they can take into account the idea that interactions between genes are probabilistic, and in this sense are biologically realistic. Stochastic principles have been used to model the elements within simulations of gene regulatory networks, ranging from simplistic models (probabilistic Boolean networks (Gershenson, 2004)) to complex models based upon simulations of molecular dynamics (Sturrock et al., 2013; Ribeiro et al., 2006; Karlebach & Shamir, 2008).

Stochastic models share a lot with ODE modelling, such as being continuous-time and having roots in real world regulatory kinetics. The difference, however, comes from the fact that stochastic models able to model probabilistic occurrences within the discrete space domain (Stoll et al., 2012). Stochastic networks can be used to model tens of different molecules with high specificity. The Gillespie algorithm is frequently used to limit computational expense, whilst maintaining accuracy (Gillespie, 1976).

The Gillespie algorithm is a form of Monte Carlo simulation, and can be described as follows (El Samad et al., 2005; Gillespie, 1976) :

1. Initialise the network's reaction constants and molecules.
2. Randomly determine the reactions to occur at the next time step (the probabilistic step, where the chance of reaction is proportional to the volume of substrate).
3. Update the simulation according to the data acquired from step 2.
4. Either move back to step 2 and repeat or end the simulation.

Stochastic models are an anomaly in terms of this chapter because there is very little research in using these as controllers for computational tasks outside of gene regulation. However, there are large similarities between stochastic networks and chemical reaction networks (Feinberg, 1995), which have been used as a model for swarm robotic assembly (Matthey et al., 2009). There are also many instances of stochastic Boolean models which have been described, although these share more of the attributes from RBNs that the stochastic equations described here (Gershenson, 2004).

Although there has been limited research conducted on stochastic networks, they have key facets such as a temporal nature that make them biologically realistic. They are however discrete space probabilistic models, which provide a direct line of cause and effect between chemicals. This may limit them as computational models because they are dependent on rules, which in itself can be debated in terms of biological reliability.

5.4 Continuous Valued Discrete Time Gene Regulatory Networks

Continuous valued, discrete time gene regulatory networks (CDGRNs) are networks whose functionality is based on the processing of continuous valued data over discrete time steps. They take inspiration from both RBNs and ODEs in the sense that they are discrete, rule based models (each gene updates its expression value according it's the expressions and weights of it connected genes), yet are able to model complex dynamics of a large range of systems as they operate in the continuous valued domain. These have been shown to be able to model complex biological regulatory networks (Kingsmore, 2006; Karlebach & Shamir, 2008), but more often than the other models described in this chapter, have been frequently used as computational controllers of complex dynamical systems.

CDGRNs comprise of genes, which are the fundamental units of their structure. Each gene consists of a set of inputs, which can be input from an external system, a regulatory function and an output. There are a range of regulatory functions that can be used in CDGRNs. In (Lones et al., 2010), three types of regulatory function were incorporated, these were the sigmoid function, the Michaelis-Menten equation and the logistic map. Each of these regulatory functions is parametrisable which allows specific processing for each gene. Generally, CDGRNs only contain a single type of regulatory function.

5.4.1 The Canonical Gene Regulatory Network Within This Thesis

This thesis follows on from earlier work (Lones et al., 2010) which utilised a particular form of continuous valued gene regulatory network. From this point on, this example of the continuous valued gene regulatory network will be referred to as *the* artificial gene regulatory network (AGRN). It is formally described as follows: $\langle G, L_G, I_G, O_G\rangle$ where :

G = Indexed genes $\{g_0, .., g_n : g_i = \langle \lambda_i, R_i, f_i \rangle\}$, where:

λ_i is the expression level of a gene

R_i is the set of regulatory inputs used by the genes

$f_i : R_i \rightarrow \lambda_i$ is a gene's regulatory function

L$_G$ is an set of randomly initiated initial expression levels, where,

|L$_G$| = |G|

I$_G$⊂G are the external inputs applied to the network

O$_G$⊂G are the outputs of the network

To execute the network, all variables are initialised according to L$_G$. The input variables from the task are then mapped on to certain genes (commonly specified by index). Then, for each gene, the connections to that gene are taken, along with their corresponding weights, and processed according to the regulatory function within that gene. The resulting value, updates the gene's current expression level. Once all genes have been updated, the outputs

Algorithm 1 Execute single iteration of network

```
Expression levels of enzymes in $\mathrm{I}_G$ are set by the external inputs

for $i = 1 \rightarrow NetworkSize$ do
    Each active gene $\mathrm{g}_i$ applies its regulatory
     function $f_i$ to the current expression levels of
     its active regulating genes $\mathrm{R}_i$ and updates $\lambda_i$ appropriately
end for

Expression levels of enzymes in $\mathrm{O}_G$ are copied to the external outputs
```

from the network (again, commonly specified via an index) can be taken and mapped back onto the task. This is specified in Algorithm 1.

Sigmoids are the most commonly used function in connectionist architectures, allowing meaningful comparisons with other models such as recurrent neural networks. In addition, (Lones et al., 2010; Mestl et al., 1995) have shown that they are generally effective for solving a range of problems. As such, sigmoid functions are used throughout the work described in this thesis. It is worth noting, however, that in (Lones et al., 2010) the authors found other nodal functions to be more useful for certain tasks. The sigmoid function in the context of a genetic function is as follows :

$$f(n) = (1 + e^{-sx-b})^{-1} \tag{5.3}$$

where x is the weighed sum of the expression values of all the connected genes, shown below.

$$x = \textstyle\sum_{j=0}^{n} \mathrm{i}_j \mathrm{w}_j \tag{5.4}$$

The execution of the AGRN is illustrated in Algorithm 1. This algorithm uses equations 5.3 and 5.4 to update the genes expression values, where s is the sigmoid slope, b the sigmoid bias and i and w are the corresponding expression values and weights from the connected genes.

There has been a range of work which uses AGRNs as computational tools for controlling complex dynamical systems. In (Lones et al., 2010), AGRNs were used to control the dynamics of both Chirikov's standard map and the Lorenz attractor, and following on from this, the same models were used to control legged robots (Lones et al., 2011). This is of particular interest because it has been shown that models such as the Lorenz attractor display dynamics similar to those found in the natural world (Haken, 1975), and moving this research directly onto physical systems such as robotics suggests that these models can be used to control a range of real world dynamical systems.

5.4.2 Variants Of Continuous Valued Discrete Time Artificial Gene Regulatory Networks

Other research has looked into variants of continuous valued gene regulatory networks. There have been some models which are similar in nature, such as the artificial metabolic network proposed in (Lones et al., 2010) which have been shown to be adept at controlling certain kinds of complex dynamics. Other such models, proposed in (Fuente et al., 2012) are based upon continuous valued gene regulatory networks, but used in conjunction with other artificial biochemical networks. Several of these considered the effect of coupling together networks in various ways. In one example, the networks were coupled by allowing cross talk between them, and were applied to the task of controlling trajectories of the Lorenz attractor and Chirikov's standard map (Fuente et al., 2012). What was prominent in this case is that coupling increased the effectiveness of these networks. Later work also showed how these coupled networks could be used to control the gaits of a hexapod robot (Fuente et al., 2013). It is clear that from the research outlined, there is interest in using continuous valued gene regulatory networks for the control of complex systems. Interestingly is that the research shows that networks can benefit from interacting with other networks and systems, some of which exist on different time-scales.

5.4.3 Similarities to other Models

The structure and execution of the AGRN model is similar to that of certain types of artificial neural network (ANN). These models were inspired by the biological functioning of networks of genes and neurons respectively. There are three significant differences between biochemical networks that make them distinct from neural networks. Firstly, the diverse set of complex nodal processes, secondly, the dynamical behaviours that result from higher-order self-modifying processes and thirdly, their emergent organisation (Lones et al., 2013). These principles also translate over to the model's artificial counterparts, and the differences between the two will be explained in a manner similar to that of the real world networks.

The nodal processes found in biology correspond to genes in AGRNs, and computational models of neurons in artificial neural networks. Both of these take inputs in some form and process them to produce an output; however, AGRNs have a range of regulatory functions, whereas computational models of neurons tend to use a single function.

Generally, ANNs are optimised via the modification of weights between perceptrons, using an algorithm such as error back-propagation (Anderson & Davis, 1995; Howarth et al., 2011). Gene regulatory networks are frequently optimised using a genetic algorithm. This has been shown to be effective (Lones et al., 2010, 2011; Fuente et al., 2012; Turner et al., 2012, 2013*a*). There are exceptions to this rule, where evolutionary algorithms have been used to optimise neural networks. Some of the earliest examples used evolutionary algorithms to evolve either the connection weights, architectures or learning rules, or combinations of the three (Yao, 1993; Yao & Liu, 1997; Yao, 1999). A further example of this is the neuroevolution of augmenting topologies (NEAT) architecture which uses genetic algorithms to evolve neural networks of varying topologies (Stanley & Miikkulainen, 2002, 1996). The NEAT framework has been shown to outperform fixed network topologies when applied to a challenging benchmark. This emphasises that the structure of the network may in part be responsive for its dynamics.

In summary, there are three significant differences between ANNs and AGRNs. Firstly, ANNs are generally optimised using training algorithms, and AGRNs are optimised using evolutionary algorithms. Secondly, AGRNs are able to evolve their architectures using evolutionary algorithms as part of the AGRN's optimisation. Thirdly, the AGRNs are able to evolve a range of parametrisable regulatory functions, along with weights during evolution. There have been clear exceptions noted to the rules above (Yao, 1993; Yao & Liu, 1997; Yao, 1999; Lindgren et al., 1993; Stanley & Miikkulainen, 1996, 2002), however, there is no research that breaks all three rules.

5.5 Summary

The work in this chapter serves to highlight that there are many implementations of models which are inspired by gene regulation. The purpose for building such models typically stems from attempting to model biological gene regulation, or attempting to capture the emergent properties of gene regulation in a computational model. There has been a larger body of work conducted on the former; the latter has been the focus of considerable research which has shown very promising results. All models detailed in this chapter capture interesting emergent dynamics which are not explicitly coded within the models. In this sense, they all, at least in part agree with the principals of emergence outlined in section 3.4. In this thesis,

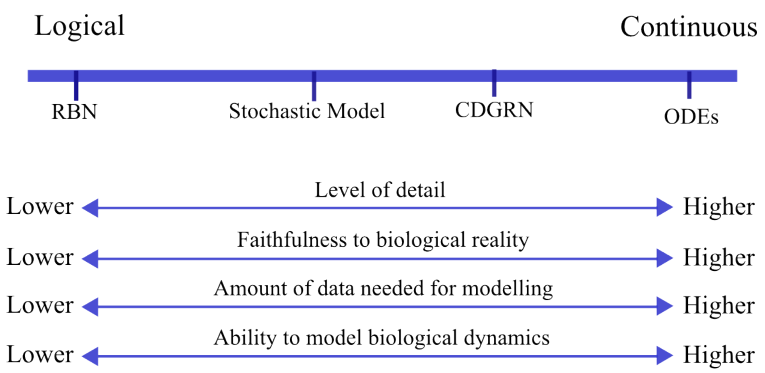

Logical
Continuous
RBN
Stochastic Model
CDGRN
ODEs
Lower
Level of detail
Higher
Lower
Faithfulness to biological reality
Higher
Lower
Amount of data needed for modelling
Higher
Lower
Ability to model biological dynamics
Higher

Figure 5.3: Models of gene regulation are listed on an arbitrary scale depicting certain characteristics. Adapted from (Karlebach & Shamir, 2008).

the body of work in concerned with utilising AGRNs as computational controllers of complex systems, and Figure 5.3 puts these models into perspective.

This chapter demonstrates that a lot of the models described can be utilised to do both biological modelling and carry out computation. The AGRNs operate well at both, functioning with no prerequisite regulatory kinetics needed, yet also able to model complex dynamics found in biology, and also control tasks which express such dynamics.

The following chapter outlines an addition to the AGRN, in which an epigenetic analogue is built of top of the ARGN, in a similar theme to that of the biology in chapter 2.

Chapter 6

The Artificial Epigenetic Network

Contents

The artificial epigenetic network (AEN) is a computational model whose form and function are inspired by epigenetics (Turner et al., 2013*a*). This chapter describes in detail the architecture of the model, how it was inspired by gene regulation and epigenetics, and the computational models that have preceded it.

6.1 Introduction

Previously in this thesis, it has been argued that there have been many successful computational models of gene regulation which have been inspired by their biological counterparts. These models capture properties found in nature such as self organisation, robustness and complex dynamics (chapter 5). However, research into epigenetics (chapter 2) has shown that, especially in eukaryotes, epigenetics plays a significant role in gene regulation, cellular differentiation and genetic packaging. Therefore, if epigenetics can produce these characteristics in nature, could epigenetics be translated into a computational model, and would this be beneficial in terms of computation?

The following sections discuss specific aspects of epigenetics, and how computational analogues of epigenetics could improve the functionality of the AGRN. Then, a computational analogue is described, which captures elements of both the structural and dynamical properties of epigenetics in nature.

6.2 Background Overview And Assertions

Within biology, genetics and epigenetics are structurally separate entities which are functionally linked together. From an engineering perspective, genetics can be seen as the biological instruction set, coding for proteins, the molecular machines of natural systems. Epigenetics can be seen as a structurally separate control for genes, specifying genetic activity over time. One of the most interesting facets of epigenetics is that of higher order gene regulation, that is, an epigenetic system which works functionally above the genetic system to control it. In nature this allows a wide range of benefits to species, the most significant being the ability to alter gene expression in a time frame which without epigenetic systems would be impossible (Veening et al., 2008). Referring to section 2.8.3, it can be seen that certain types of bacteria can produce epigenetically induced phenotypic changes to create a more optimum phenotype within a given environment. Considered from an engineering perspective, it can be seen that certain genes within the organism have evolved to do certain tasks. Depending on which task is required of the organism, epigenetic mechanisms can activate specific genes and deactivate others. Within a computational model, this would potentially be advantageous because it could allow the specialisation of set of genes for specific tasks.

To follow on from the above, if genes are to be specialised for a specific task within a computational model, and organised epigenetically so that they can be activated and deactivated efficiently, it means that it is possible that these epigenetic changes could provide additional information to allow a higher level of network analysis which would not otherwise be possible. This would be achieved by looking at the activation and deactivation of epigenetic circuits, which could provide an accurate method for ascertaining how the network is functioning. Moreover, if this were the case, it could be possible to modify these epigenetic circuits manually to control specific behaviours within the network.

It is therefore clear that if epigenetics could be incorporated into a model of gene regulation, there are a range of benefits which could emerge from the interactions of heterologous structures.

Taking the work in the previous chapters into account, it is asserted that :

- There is currently no artificial model of gene regulation designed for computation which explicitly models epigenetic mechanisms as well as genetic mechanisms.
- If epigenetics were introduced to a model of gene regulation, it is possible that some of the natural advantages of epigenetic structures may be transferred to the computational model

These are the assertions on which this work is based.

6.3 Representing Epigenetic Mechanisms *In Silico*

Epigenetic mechanisms in nature are complex structures that functionally are intertwined with many other processes. Epigenetics must be viewed on a specific level of abstraction as modelling these processes in perfect detail is not feasible. From the work in chapter 2, it is clear that epigenetic mechanisms effect gene regulation, and in the case of chromatin modifications, do this by physically inhibiting the cellular machinery responsible for gene regulation. These chromatin modifications can dynamically change their shape and position along the DNA strand to achieve this. Chromatin modifications in this sense can be seen as hard switches which operate dynamically, either allowing genes to be active or inhibiting them. It is reasonable to deduce that chromatin modification can be in part responsible for many of the biological advantages of epigenetic structures. Therefore chromatin modifications will be the specific inspiration for the artificial epigenetic analogue which will be part of the AEN.

One of the most important aspects to incorporate into the model is for the epigenetic analogue to change which genes are active *dynamically.* In order to achieve this, the epigenetic analogue has to do some computational processing to determine its activity. As it is designed to be built upon an existing AGRN, there are certain elements of the epigenetic analogue that are pre-determined. The epigenetic analogue must take into account expression values from genes within the AGRN to determine its own activity. This is because in nature, the activity

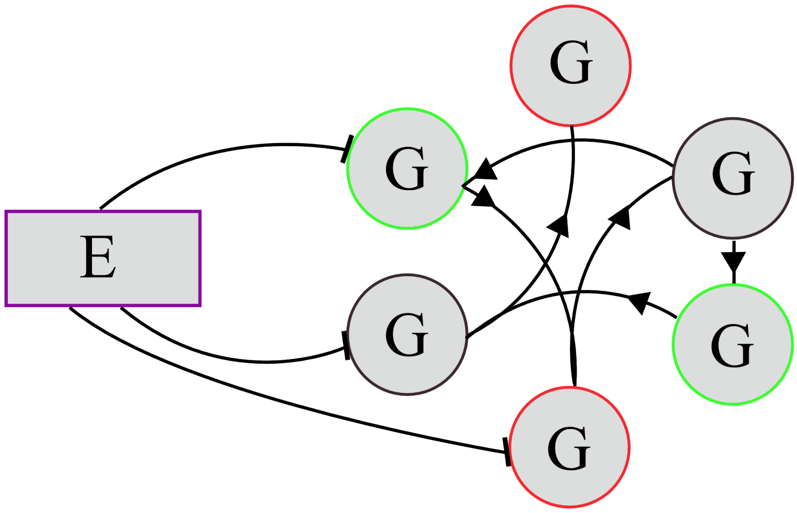


Figure 6.1: An illustration of how the epigenetic analogue interacts with an AGRN. The genes, (marked 'G') function within the network as normal, until their function is halted by the epigenetic molecule (marked 'E'). The epigenetic molecule takes inputs from the genes that it is connected to. In turn this allows inputs to be taken from the environment. If the inputs to the epigenetic molecule are above a certain threshold, the epigenetic molecule becomes active and prevents those genes from updating their expression value.

of genes has been shown to influence epigenetic structures, hence this should be reflected in this model. Again, in nature, chromatin can be either active or inactive. Therefore the activation value should act as a hard on/off switch to specify dynamically if the artificial epigenetic molecule is active. Hence, the activation function must convert continuous values into a discrete valued switch.

To best describe the epigenetic analogue in terms of how it interacts with the network, another concept must be described which details how connections are held within the network. This is known as the reference space.

6.3.1 The Reference Space

The reference space is a compact method of determining connections between genes. It is a form of indirect representation and template matching which draws inspiration from (Reil, 1999) who specified that :

> "many features of real-life development, such as cyclic gene activity, differentiation into multiple cell types, and robustness may be inherent properties of the system rather than necessarily designed from scratch by Natural Selection"

This suggests that indirect representation within a computational context may give rise to emergent behaviours, which ties in with the assertions in section 6.2. To embed this within the AEN, a from of indirect representation which utilises a reference space will be used to represent connections with the network. This reference space is an abstract, one-dimensional space in which genes and epigenetic analogues are located, and if they overlap they are considered to be connected (Figure 6.2).

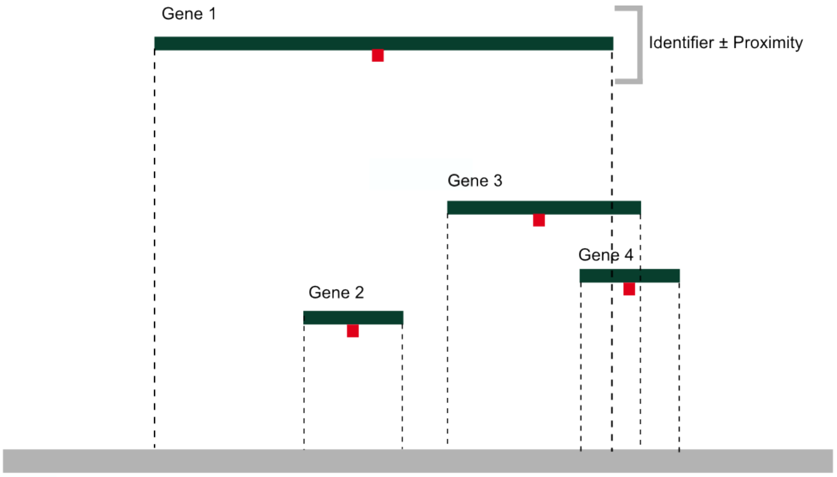


Figure 6.2: Illustration of the reference space of the genes. If any gene's identifier lies within the region of another gene's identifier ± its proximity, the former gene is used as a connection to update the expression of the latter gene. From this example, the connections for gene 1 are genes 2 and 3, gene 2 is connected to gene 1, gene 3 is connected to gene 4, and gene 4 has no connections.

As can be seen in Figure 6.2 each gene is represented by an identifier and a proximity. The identifier defines a location within the reference space where the gene in located. The proximity specifies a distance either side of the identifier, which is the space to derive connections to other genes. A gene can be considered to be connected to another if its own identifier lies within another gene's identifier ± its proximity. Using this technique, connections between genes can be described on a network wide level. Building upon this connections for the epigenetic analogues can be derived.

The data structures of the genes and epigenetic molecules can be seen in Tables 6.1 and 6.2 respectively. The epigenetic analogue can take information from a set of genes, process it, and depending on the value, prevent that set of genes from updating their expression. If we look at the example in Figure 6.3, it can be seen that although only two genes will be directly affected by the activity of the epigenetic molecule, as all the genes are connected to some manner, the genes that are not directly affected will be indirectly affected.

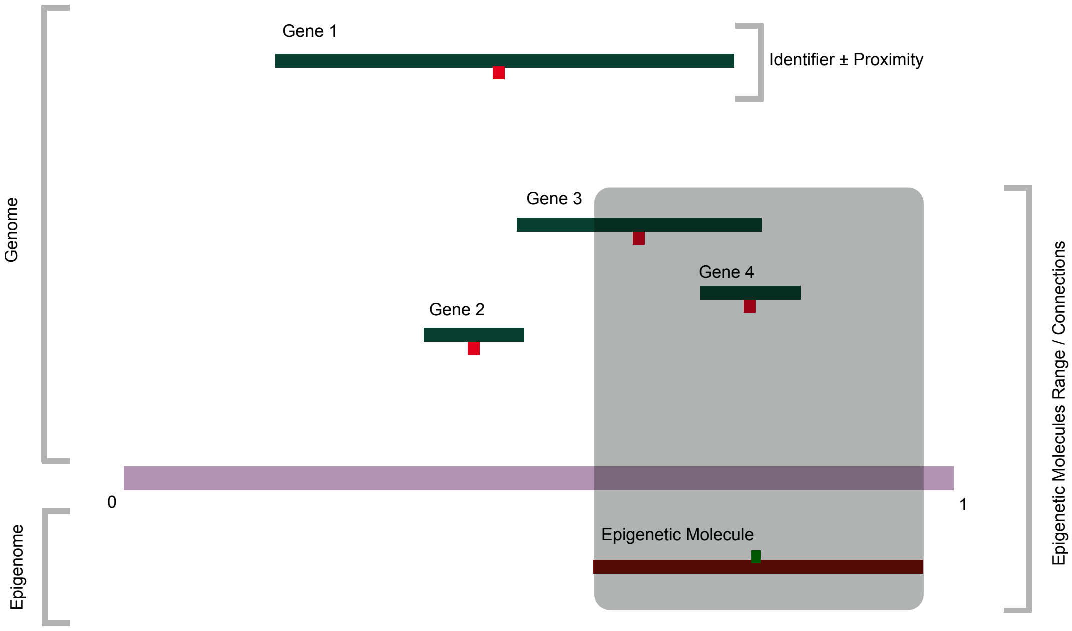


Figure 6.3: A visualisation showing how an epigenetic molecule interacts with the genes. The highlighted region shows that the epigenetic molecule is connected to genes 3 and 4. The epigenetic molecule can then set these genes expression levels to 0. As gene 3 is connected to gene 1, and gene 1 is connected to gene 2, hence this epigenetic change can effect all genes within the network.

Variable	Type	Range
Expression	Real	0;1
Weight	Real	-1;1
Identification	Real	0;1
Proximity	Real	0;0.15
Sigmoid Offset	Real	-1;1
Sigmoid Slope	Int	0;20

Table 6.1: Ranges of the variables within each gene.

Variable	Type	Range
Identification	Real	0;1
Proximity	Real	0;0.15
Sigmoid Offset	Real	-1;1
Sigmoid Slope	Int	0;20

Table 6.2: Ranges of the variables within each epigenetic molecule.

Aside from indirect representation, there are other reasons as to why using an indirect reference space is beneficial in this instance. Firstly, it allows positional independence where the location of the genes within the genome bears no effect either directly or indirectly on the phenotype. This has been shown to have benefits in biologically inspired algorithms (Lones, 2004). Secondly, it is a good analytical tool to visualise the interactions between gene and epigenetic molecules within the network.

6.4 Artificial Epigenetic Network Model

This section formally defines the artificial epigenetic network (AEN) specifying in detail the epigenetic analogue, how it processes data, and how it ties in with the execution of the underlying AGRN.

6.4.1 The Epigenetic Analogue

The epigenetic analogue is a data structure that contains four variables (Table 6.2). The identifier and proximity define the region of the reference space which the epigenetic analogue occupies. The other two variables are the sigmoid slope and sigmoid offset which effect how it processes the expressions and weights from its connected genes. This is done with a discretised sigmoid function (equations 6.1 and 6.2). The epigenetic analogue is active when $f(n)$ of equation 6.1 is >0.5. If this is true, the genes covered by the epigenetic molecule will have their expression values set at 0 until $f(n)$ <0.5. If false, the genes operate unaffected.

$$f(n) = (1 + e^{-sx-b})^{-1} \tag{6.1}$$

where x is the weighed sum of the expression values of all the connected genes, shown below.

$$x = \textstyle\sum_{j=0}^{n} \mathrm{i}_j \mathrm{w}_j \tag{6.2}$$

6.4.2 Formal Description

The AEN is the combination of an AGRN and the epigenetic analogue. The AEN has two parameters that need to be set before execution. These are the number of genes and epigenetic molecules. In choosing the number of epigenetic molecules, it is optimal to give the epigenetic molecules the ability to be part of the network without forcing them to do so. Therefore, it is not suitable to have epigenetic molecules covering the entire reference space. Work in (Turner et al., 2013*a*) has shown that between three and five epigenetic molecules leads to good functionality.

The AEN can be formally described as: $\langle G,\ L_G,\ I_G,\ O_G,\ E\rangle$ where :

G = Indexed genes $\{g_0, .., g_n : g_i = \langle\lambda_i, \Re_i, f_i\rangle\}$, where:

λ_i is the expression level of a gene

$\Re_i$ is the set of regulatory inputs used by the genes

$f_i : \Re_i \rightarrow \lambda_i$ is a gene's regulatory function

L$_G$ is an indexed set of initial expression levels, where,

$|\text{L}_G| = |\text{G}|$

I$_G$⊂G are the external inputs applied to the network

O$_G$⊂G are the outputs of the network

E = Indexed epigenetic molecules $\{e_0, .., e_m : e_j = \langle\delta_j, T_j, Y_j\rangle\}$, where :

δ_j is the activity level of the epigenetic molecule

T_j is the set of regulatory inputs used by the epigenetic molecule

$Y_j : T_j \rightarrow \delta_j$ is an epigenetic molecule's regulatory function

The execution of the network (algorithm 2) begins by setting the numbers of genes and epigenetic molecules within the network. Then the genes and epigenetic molecules are initialised according to L_G (tables 6.1 and 6.2). Next, task variables are mapped onto the inputs of the genes. This means that the data from the task must be normalised between 0 and 1 so that it is within the bounds of the genes' expression values. The epigenetic molecules then take the expression values from the genes and ascertain if they are active. If the molecule is active, it prevents the updating of its connected genes and sets its own genes' expression value to 0, effectively removing them from the network. Next, the genes which are not inhibited by the epigenetic molecules take the expression values and weights from their connected genes (assuming they are not blocked by an epigenetic molecule) and update their expression value'.

Algorithm 2 Execute single iteration of AEN

```
if Starting then
    Set number of genes
    Set number of epigenetic molecules
end if

Map task variables onto input genes

for $x = 1 \rightarrow NumberOfEpigeneticMolecules$ do
    Derive if epigenetic molecule $x$ is active
end for
for $i = 1 \rightarrow NumberOfGenes$ do
    if Gene $i$ is not inhibited by epigenetics then
        Update gene $i$'s expression
    else
        Set gene $i$'s expression to 0
    end if
end for

Map network outputs back to the task
```

This is synchronous, hence, the network is not affected by which order the genes are updated. The outputs are then taken from the network and mapped back onto the task. This process can be iterated for however many iterations the task runs for.

6.5 Task Specificity

Given the reasoning behind the assertions made (section 6.2) regarding designing the epigenetically inspired networks, it is fair to assume that there are certain tasks that would not suit the functionality of the epigenetically inspired networks. If the tasks requires simple dynamics, that is, dynamics that can be *readily* achieved with gene networks alone, there would be no benefit in using epigenetic analogues. The AEN would most probably be best suited to tasks which require a range of complex dynamics. This is because it has been shown that AGRNs can produce complex dynamics, and that epigenetic analogues can in theory partition the AGRN into separate regions at specific times. If these separate parts can be applied to different tasks, it would be fair to assume that these partitions could each adopt a different set of complex dynamics, which could be beneficial from a computational perspective.

6.6 Optimisation Of The Networks For Computation

There are a large number of variables within the AEN which must be optimised for a specific task. For a network containing ten genes and three epigenetic molecules, there are 72 variables which must be optimised. This is done using a genetic algorithm (section 4.1) as research has previously shown that networks evolved using genetic algorithms have been able to express complex dynamics (Banzhaf, 2003; Nordin et al., 1995; Banzhaf et al., 2006; Turner et al., 2012, 2013*b*; Lones et al., 2010). To perform the crossover operation of the genetic algorithm, the genes and epigenetic molecules will be treated as the fundamental units of the network, and that they can only be crossed over as individual units to limit disruption. An illustration of this can be seen in Figure 6.4.

In this method of crossover, there are specific constraints for child networks. In this case it is assumed that each network has to contain between 6 and 10 genes, with between 3 and 5 epigenetic molecules. In Figure 6.4, for the two networks that are being crossed over, one has 9 genes and 3 epigenetic molecules, the other has 7 genes and 4 epigenetic molecules. A random number is picked between the ranges (6 and 10) and a new network is created of that size (called network A). Another network (network B) is created which is the size of the total number of genes between both networks minus the number of genes in network A. Using a similar method, network A is assigned a random number of epigenetic molecules within the range (3 and 5) and network B is assigned the number of epigenetic molecules remaining minus the number of epigenetic molecules in network A. The genes and epigenetic molecules are then pooled and randomly selected between the new networks until they are fully populated. The creates two new networks, typically with different amounts of genes and epigenetic molecules (however, this does not necessarily have to be the case, the randomly selected sizes could be the same as the original). These networks now go through a stage of mutation. This is achieved by mutating each element of each gene and epigenetic molecule according to the ranges in Tables 6.1 and 6.2, and according to a fixed mutation rate (probability).

6.7 Previous Model

The AEN builds upon an earlier model, termed the artificial epigenetic *regulatory* network (AERN) (Turner et al., 2012, 2013*b*). This network was built upon the same principles as the AEN; however its purpose was to determine if an artificial epigenetic analogue could be beneficial in a structurally and behaviourally more basic form. In particular it omitted the

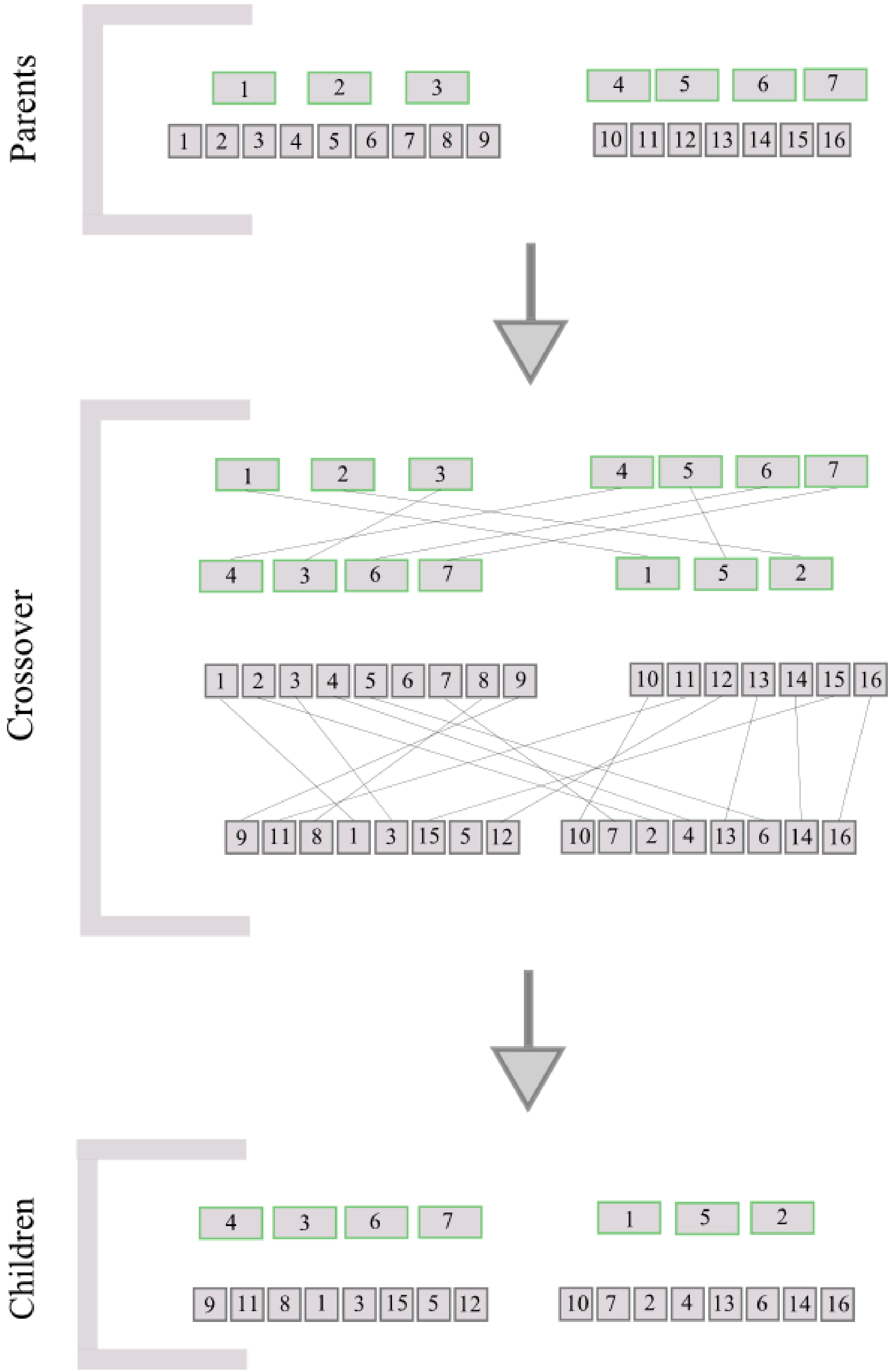


Figure 6.4: An illustration of the crossover operator in the genetic algorithm This operator takes two different sized AENs and crosses over the genes (squares) and epigenetic molecules (rectangles) into two new networks.

Variable	External Inputs ($\mathbf{I}_G$)		Genes					Outputs ($\mathbf{O}_G \subset$G)
Gene Expression Values ($\mathbf{L}_G$)	0.18	0.81	0.54	0.38	0.95	0.14	0.05	0.47
Weights	0.47	-0.27	0.24	0.99	-0.87	-0.02	-0.47	0.97
Sigmoid Offset	-0.18	0.24	0.14	-0.50	-0.21	0.57	0.31	0.38
Sigmoid Slope	1	10	5	19	2	14	3	7
Connections	5 7 5	2 4 2 4	1	5 5 5 4	7 2	3 7 3 1	2 1 2	3 1 3 7
Epigenetic Frame A ($\mathbf{E}_G \subset$G)	1	0	1	1	0	0	0	1
Epigenetic Frame B ($\mathbf{E}_G \subset$G)	0	1	1	0	1	1	1	1
Network Iterations	3							

Table 6.3: Example data attributes for an AERN containing 8 genes. The only difference between the AERNs and the AGNs is the introduction of epigenetic frames, which specify which genes will be active for each objective.

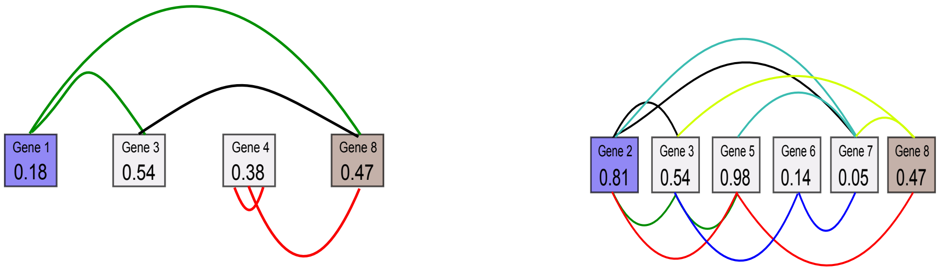
requirement to operate dynamically during execution and changes to the epigenetic state of the network were ‘pre-programmed’ to happen at a specific point during execution. From an engineering perspective, this model was much easier to design because it comprises a set of dynamically and structurally static analogues.

6.7.1 The Artificial Epigenetic Regulatory Network Structure

The epigenetic analogue proposed in the AERN operates by blocking out the operation of certain genes according to the indices of the genes. An example of this can be seen in Table 6.3. The AERN consists of a set of ‘frames’ each one being the length of the network and attaching a Boolean switch which can either activate or deactivate the genes. Only a single epigenetic frame can be active at any given moment. Switching between the frames changes which set of genes will be active at at given time. In the example outlined in Table 6.3, there are two frames, which can be programmed to change according to a predefined heuristic .

An example of the network which Table 6.3 describes can be seen in Figure 6.5. As can be seen, in this example, the epigenetic frames allow the partitioning of individual genes to certain tasks, which could allow for certain genes to become specialised towards a specific task. It is to be noted that the section of the epigenetic frame which covers the output is always active, so that each sub network is functional.

The main differences between the AEN and the AERN is that the epigenetic control structure of the AEN operates dynamically and uses a form of indirect representation to define its



(a) The network from Table 6.3 with epigenetic frame A active

(b) The network from Table 6.3 with epigenetic frame B active

Figure 6.5: An illustration of the network which results from the data held in Table 6.3. Purple genes are input genes, white genes are processing genes and brown genes are the outputs.

connection to genes. The epigenetic analogue within the AERN is a direct representation of a static structure, which will only change according to a pre-defined rule. This means that the AERN requires some knowledge about the task in order to define the most appropriate rules in which to change the epigenetic frames. A further difference is that the AERN iterates multiple times per call (algorithm 3). This was omitted from the AEN as exploratory work demonstrated it did not increase functionality and significantly impacted computational efficiency. During experimentation, the AERN will be compared to the the AGRN *with* multiple network iterations in place. The AEN will be compared to the AGRN using only a single network iteration. Hence, in each experiment, the epigenetic inspired networks will be identical to the network they are compared against, except that the latter contain no epigenetic information. Therefore, any difference in functionality or performance will be a direct result of the epigenetic structures.

6.7.2 Execution Of The Artificial Epigenetic Regulatory Network

The AERN is executed in a similar way to the AGRN (section 5.4.1), except the epigenetic frames dictate which genes are active at any given time. If we take the network from Table 6.3 and Figure 6.5 as an example, the first step is to map the task variables onto the inputs of the network. Then according to a predefined rule, set which epigenetic frame will be in use. Then iterate the network where only the active genes are able to update their expression values or be used as connections to other genes. Then, the outputs can be mapped back to the task.

Algorithm 3 Execute single iteration of the AERN

```
Expression levels of enzymes in $I_G$ are set by the external inputs

for $i = 1 \rightarrow NetworkSize$ do
    for $i = 1 \rightarrow NetworkIterations$ do
        if Epigenetics Layer specifies gene is active then
            Each *active* gene $g_i$ applies its regulatory
            function $f_i$ to the current expression levels of
            its *active* regulating genes $R_i$
        end if
    end for
end for

Expression levels of enzymes in $O_G$ are copied to the external outputs
```

6.8 Summary

This chapter has shown in detail the architecture and function of the epigenetically inspired artificial epigenetic network. This network incorporates an epigenetic analogue which can alter which genes are being expressed dynamically. This is a feature which has been inspired by gene regulation in nature. It is a novel contribution to the field of artificial gene regulatory networks, and a foundation in which to develop an understanding of how the artificial epigenetic network functions. In the following chapter, an overview of the experimentation with the artificial epigenetic network is described. This details the rationale behind the experimentation and explains why the particular tasks were chosen.

Chapter 7

Experimental Methods

Contents

This chapter serves as a preliminary guide to the experimental chapters (chapters 8, 9 and 10). It describes how the networks from the previous chapters are applied to tasks, and the specific reasons for choosing those tasks. In addition the choices behind how to optimise the networks are justified.

7.1 Chaos Targeting

Biological organisms are consistently trying to maintain an internal environment. This process is known as homoeostasis. There are a range of biochemical networks and processes which have a significant role in the maintenance of homoeostasis. Biochemical networks are particularly adept at controlling perturbations, hence, in this work the computation analogues of gene regulatory networks are applied to the control of systems, to stay faithful to their biological underpinnings. The systems which are most interesting in terms of control are those with chaotic or complex non-linear dynamics because it is typically difficult to control such dynamics.

Chaos can be defined as the qualitative study of unstable aperiodic behaviour in deterministic non-linear dynamical systems (Kellert, 1994). To put it more generally, a system can be described as chaotic when its current state expresses extreme sensitivity to its initial conditions. Chaos, despite its connotations in modern culture does not mean random, in fact it is quite the opposite. Chaotic systems are highly predictable when specific values capturing its current state are known, however, *approximating* chaotic systems does not lead to accurate predictions. Chaotic dynamics appear in a range of systems including wind turbulence (Hopf, 1948), weather (Fraedrich, 1986) and laser dynamics (Haken, 1975).

Chaos targeting, also known as state space targeting, is the process of controlling the dynamics of a chaotic system with the aim of preserving a specific state or set of behaviours. By way of example, Bollt & Meiss (1995) found that a chaotic transfer orbit between the earth and moon could be optimised to require 38% less velocity boost than a standard orbit.

7.1.1 Traditional Controller Design

The principle of controlling chaotic systems has been of interest for many years. There are several approaches to chaos targeting which exist, the most prominent being OGY (Romeiras et al., 1992). OGY functions by offering time dependent perturbations in the from of feedback to a system parameter, which in turn maintains the system at a fixed operating point (Tsai et al., 2002). A further method of chaos control has been shown in (Pyragas, 1992) where unstable orbits were stabilised via the use of delay feedback perturbations. A pervasive trait of the above methods is that to derive these analytical solutions a detailed understanding of the underlying state space is required (Lones et al., 2012). There has been a range of previous applications of both neural networks (Sanchez & Ricalde, 2003) and evolutionary algorithms (Richter, 2002) to perform chaos targeting.

7.2 Evolving Controllers

The method being used in this work approaches controller design from a different angle than those listed above, which are generally concerned with maintaining a trajectory within a system at a fixed point. The approach within this thesis is to manoeuvre a trajectory around the state space in order to control the dynamics of the system. In addition, the evolved controllers are given no information about the underlying dynamics of the system which they are controlling. Previous work has shown that artificial biochemical networks can be used to bridge the gap by controlling a chaotic system *without* having explicit knowledge of the underling state space (Lones et al., 2010, 2012). This section describes the application of artificial gene regulatory networks to the control of systems in which there is little information about the system which is being controlled.

7.2.1 Experimental Design

The objective of experimentation with the epigenetically inspired networks is to understand their emergent properties as well as ascertaining how they operate. To do this, the networks need to be optimised towards a specific task. In chapter 4, a range of evolutionary algorithms were outlined. Within this thesis, there is a focus on evolvability and emergence, and in turn staying faithful to biological principles because they are by definition, evolvable. Additionally, within this body of work, multiple representations of artificial gene regulatory networks are evolved, and the method of artificial evolution used needs to be flexible to accommodate different representations of executable structures. Genetic algorithms function well on a diverse set of computational representations (Mitchell, 1998), and implementations such as NSGA II (section 4.1.1) have been designed specifically for use with multi-objective tasks. Moreover, they have been previously used to evolve a range of artificial biochemical networks (Lones et al., 2010; Lindgren et al., 1993). It is because of these features that genetic algorithms will be used when evolving the networks.

Research has shown that using artificial biochemical networks to control chaotic systems is a difficult problem for them to solve, however, it is possible (Lones et al., 2010, 2012). Moreover, certain tasks can also be used to gauge the computational properties of the underlying networks. In this work we apply the epigenetically inspired networks to the control of both chaotic systems, and ordered systems which express complex non linear dynamics. It is to be noted that even though a system does not express chaotic dynamics, it is not necessarily easier to control than a chaotic system. Although a chaotic system is impossible to predict over a long time scale due to sensitivity to external perturbations, its behaviour is the product of a deterministic rule, therefore, it is controllable in the short term (Chen & Dong, 1993). Because of this, chaotic systems are innately controllable.

The application of a controller to a control task can be seen in algorithm 4. This method is applicable where the tasks are dynamical, but are also updated in discrete time. This method is referred to as a closed loop controller, and will be used for all experimentation throughout this thesis. Hence, at each discrete step the task will update at the same time as the controller.

Algorithm 4 Execute single iteration of a network when applied to a control task

```
if Starting then
    Set number of genes
    Set number of epigenetic molecules
end if

Initialise control task

for A set number of iterations do

    Map task variables onto input genes
    Execute the network
    Map network outputs back to the task
    Update the task

end for
```

7.2.2 Genetic Algorithms

Two different genetic algorithms will be used to evolve the network; the first is a standard genetic algorithm, and the second is NSGA II. Whether or not the task is multi-objective determines which algorithm will be used. However, most of the functionality of the different algorithms is identical. Firstly, the selection mechanism that will be used is rank based and additionally, tournament selection is used. The operators that will be used are crossover and mutation. The crossover operator that will be used is n-point crossover and can be seen in Figure 6.4. This type of crossover is able to deal with both same sized and different sized networks and is used for both genetic algorithms. The mutation operator changes a given value with the network structure to a random value within the possible range for the variable. When using NSGA II, elitism is ensured, however, elitism is not present within the standard genetic algorithm.

Aside from the experimentation with the AERN (section 8.2), all experiments have a population size of 500 and run over 200 generations (100,000 evaluations). The crossover rate is 0.5 and the mutation rate is 0.05. A tournament selection of size 4 is used. In the case of the earlier experimentation with the AERN, the parameters are identical apart from that the population size is 200 and the algorithm runs for 50 generations.

7.3 Control Tasks

It is important when evaluating a computational model that the range of tasks it is applied to are diverse enough to make general conclusions about its performance. As stated in section 6.5, it is expected that the AEN will produce the most interesting emergent behaviours when applied to tasks which require a range of complex dynamics to solve. With this in mind, one of the most appropriate control tasks is that of the Chirikov's standard map, which exhibits both ordered and chaotic dynamics. This has been previously used as a control task to test the performance of artificial biochemical networks (Lones et al., 2010). In addition, it is an approximate model of a range of physical systems, which means that conclusions can be drawn about the networks' ability to control real world dynamics. For this control task we look at both the AEN and the AERN showing how the former is able to automatically recognise subtasks. The application of the networks to Chirikov's standard map can be seen in Chapter 8.

Chirikov's standard map task is used in this instance as a general proof of concept model. This is because although it exhibits both complex and ordered dynamics, there is not enough evidence to define its computational complexity as a control task. However, from the surrounding research in (Lones et al., 2010) it is clear that solving the task is at least non trivial. Therefore, to generalise about the performance of the AEN, it will have to be applied to additional tasks to improve the validity of the findings.

In the second experiment the AEN is applied to a coupled inverted pendulum task (chapter 9). Although it does not express chaotic dynamics, state space targeting can still be applied to its complex non-linear dynamics. Because this task requires two distinct behaviours to solve optimally (swinging the pendulums, then balancing them in the upright position) it is well suited to the expected behaviour of the AEN. In addition, the coupled inverted pendulum tasks has been widely applied as a benchmark for decentralised robotic controllers (Hamann et al., 2011). Therefore, unlike Chirikov's standard map, direct conclusions can be made about the computational complexity required to solve the task. This is essential to validate the performance of the AEN and draw conclusions about its use as a computational tool.

The final experiment involves controlling transfer orbits in gravitational systems (Chapter 10). The objective is to control a rocket and navigate a path between two orbits whilst under the influence of a strong gravitational pull from another planet. In addition, the rocket must optimise its path in terms of fuel usage and a key way of doing this is to utilise the gravitational slingshot. As demonstrated previously, gravitational systems consisting of multiple bodies can exhibit chaotic dynamics (Bollt & Meiss, 1995). This ties in well with the experimentation with Chirikov's standard map, as controlling transfer orbits in gravitational systems is a real world analogue of the mixed conservative dynamics of Chirikov's standard map (Bollt & Meiss, 1995). The controlling transfer orbits in gravitational systems tasks is a rich environment in which to allow the emergence of complex behaviours within the networks. In terms of the complexity of the controller required to solve the task it is unclear what is required. However, because the tasks is a real world task, there are conclusions that can be drawn which point to a relative complexity that would be required to solve the task as optimally as possible.

Overall, the three experiments require the control of a wide range of dynamics. Chirikov's standard map is a model of mixed chaotic and ordered dynamics, the coupled inverted pendulum is a specific model of complex non-linear dynamics, and transfer orbits in gravitational systems is a specific model of complex, ordered, and chaotic dynamics. Each of these tasks has specific strengths and weaknesses as a control task. However, as a combination of tests, they can provide a solid test bench in on which to analyse the behaviour of the AENs by mitigating the weaknesses associated with individual tasks. This should not only provide reliable evidence of the emergent properties and behaviours of the AENs, but also provide reliable evidence as to their use as a computational tool.

7.4 Summary

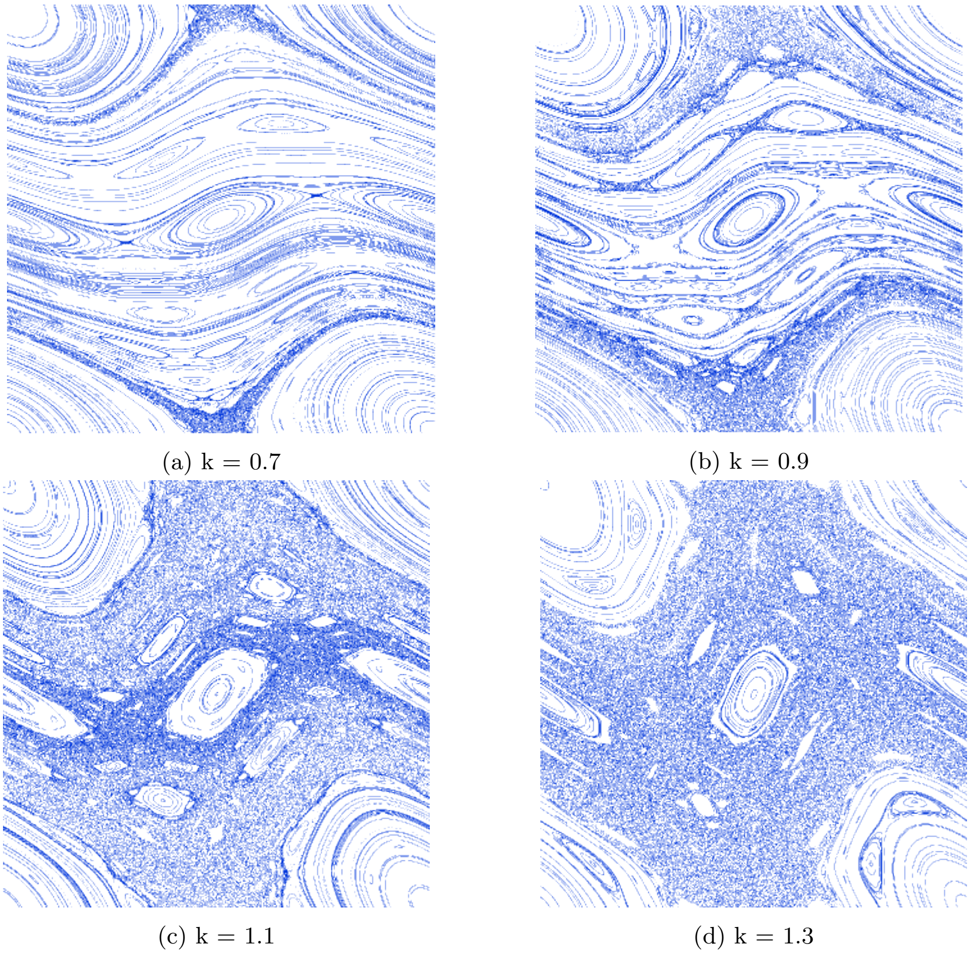
In this chapter, the overall design of the experiments used in this research is outlined. This provides a prior understanding of the experimental method which is for the most part identical within the following three chapters. In the following chapters, the artificial epigenetic network will be applied to a range of tasks, and from the analysis of their performance, network characteristics and network structure, a picture of the beneficial properties of the network can be created. From this it can be understood what benefits artificial epigenetics can contribute to the field of artificial gene regulatory networks.

Chapter 8

Chirikov's Standard Map

Contents

The work within this chapter describes the application of the epigenetically inspired networks to perform chaos targeting within Chirikov's standard map. In this chapter previous work from (Turner et al., 2013*b*, 2012) is covered which used the artificial epigenetic regulatory network (AERN). Later in the chapter, this is built upon by applying the artificial epigenetic network (AEN) to the control of the standard map. The AEN was inspired directly by the design and functionality of the AERN and this chapter describes how both networks evolve different emergent behaviours.


(a) k = 0.7

(b) k = 0.9

(c) k = 1.1

(d) k = 1.3

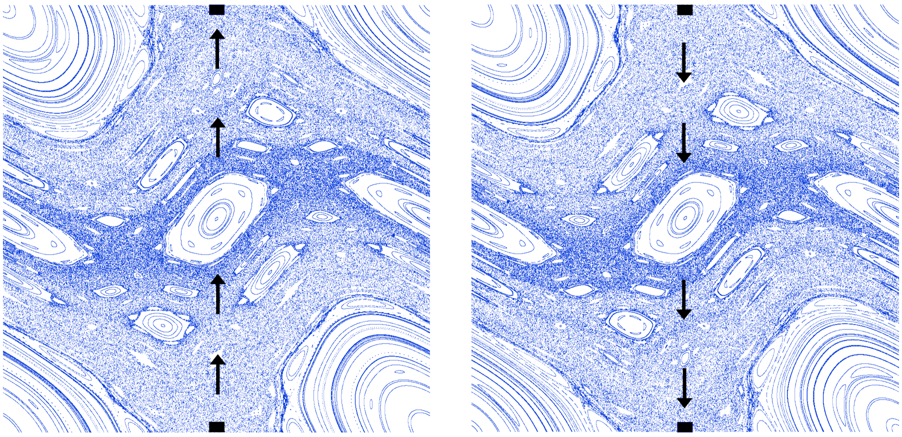
Figure 8.1: Chirikov's standard map for a range of k values.

8.1 Description of Chirikov's Standard Map

Chirikov's standard map is a two dimensional dynamical system that exhibits co-existing ordered and chaotic dynamics (Chirikov & Sanders, 1971; Turner et al., 2013*b*). The equations for the standard map are shown in equation (8.1).

$$\begin{aligned} x_{n+1} &= (x_n + y_{n+1}) \bmod 1 \\ y_{n+1} &= y_n - \frac{k}{2\pi}\sin(2\pi x_n) \end{aligned} \tag{8.1}$$

The k parameter within the equation controls the ratio between ordered and chaotic dynamics. For low values of k, the dynamics are more ordered (Figure 8.1a), and when k increases, chaotic dynamics become more prevalent (Figure 8.1d). At a k value of approximately 1.0 and above the natural dynamics of the map allow the traversal of trajectories from the top to the bottom of the map.



(a) Objective A - To move from the bottom of the map to the top (b) Objective B - To move from the top of the map to the bottom

Figure 8.2: The multi objective tasks where the trajectory has to traverse from the bottom of the map to the top, and then from the top of the map to the bottom. The black boxes mark the initial starting positions and the targets of the trajectory. For objective B, the initial position is the same as the target from objective A, and the target is the same as the initial position from objective A.

The standard map describes the motion of a simple mechanical system, the kicked rotator. The map is the Poincare's surface of section of the kicked rotator. Whilst the kicked rotator is a physical system consisting of a constrained element which is periodically perturbed by an external force, Chirikov's map can also be seen as a general model of conservative dynamical systems which contain co-existing ordered and chaotic behaviours (Chirikov, 1979). This makes the control of trajectories within Chirikov's standard map suitable for the epigenetically inspired networks, as it was hypothesised that they would be most functional when applied to a system with varying dynamics (section 6.5). A testament to the standard map's generality is that many systems are reducible to the standard map, including celestial body dynamics (specifically the map for Halley's comet (Chirikov & Vecheslavov, 1989)), particle dynamics (Izraelev, 1980) and microwave ionization of Rydberg atoms (Casati et al., 1988).

8.2 The Artificial Epigenetic Regulatory Network

8.2.1 Experimental Design And Parameters

The experimental design described herein has been adapted from previous work which used the standard map to analyse the performance of artificial gene regulatory network models (Turner et al., 2012, 2013*b*; Lones et al., 2010). There are two objectives within the task. First, to navigate a trajectory within the standard map from an area within the range

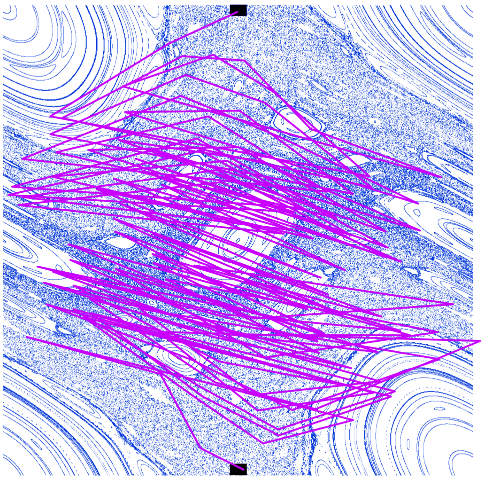


Figure 8.3: A representation of a typical path through the standard map which was achieved using the AEN.

(x[0.475,0.525], y[0,0.025]) to the target area within the range (x[0.475,0.525], y[0.975, 1]) (Figure 8.2a). Second, to reverse the direction of the trajectory and navigate to an area from the range (x[0.475,0.525], y[0.975, 1]) to (x[0.475,0.525], y[0,0.025]) (Figure 8.2b). An illustration of a typical path through the standard map can be seen in Figure 8.3. The starting points for the trajectory are randomly sampled from the ranges specified. A limit of 1000 steps is imposed on the completion of each objective within the task. Each objective has 10 attempts (20 in total for both directions). Assuming a successful path has been traversed in both directions, the score is the average of all 20 attempts in both directions. If a path is not traversed in both directions, a maximum score of 2000 is assigned.

The AERN and the AGRN are applied to control the trajectories through the standard map. At each step the networks are provided with three inputs. These are the x and y positions of the trajectory and the distance to the centre of the target. The trajectory is controlled via the modulation of the k parameter in equation 8.1 within the range [1,1.1]. Both the AERN and AGRN contain 10 genes, which is fixed throughout evolution.

The networks are evolved using a genetic algorithm with a population size of 200 over 50 generations. Tournament selection of size 4 is used, with a crossover rate of 0.5 and a mutation rate of 0.05. A total of 40 independent runs were completed for each network.

8.2.2 Results

The results can be seen in Figure 8.4 where it is apparent that the AERN out performs the AGRN. The range of results is smaller for the AERN and all instances of the AERN solved

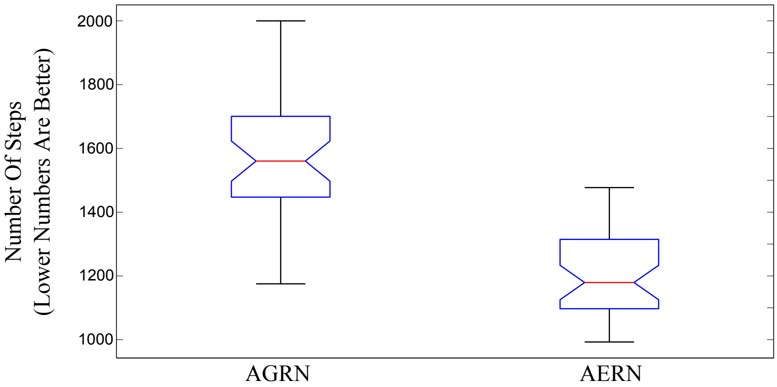


Figure 8.4: The best results of each run at 50 generations.

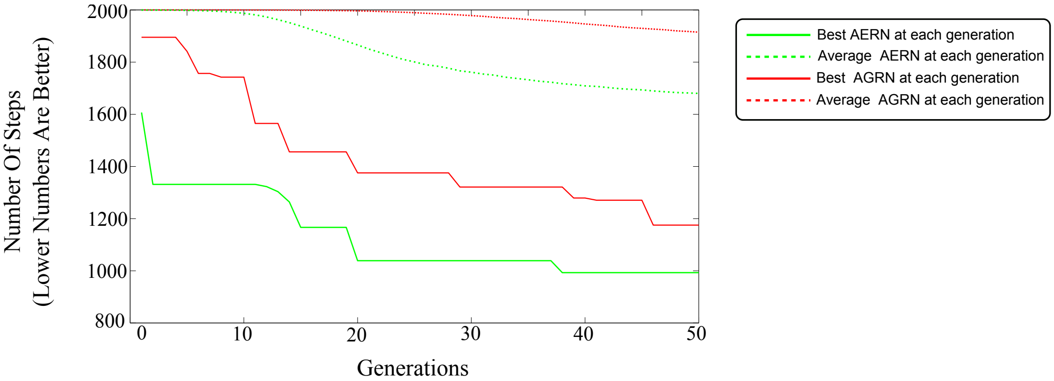


Figure 8.5: The best results and average results at each generation averaged over all runs.

the task, whereas only 36 out of the 40 runs with the AGRN did. The AERNs produced a statistically significant improvement over the use of the AGRNs in terms of mean number of steps with a significance value of p = 2.96 x 10^{-10} using the Wilcoxon rank-sum test, as the data did not fit a normal distribution according to the KolmogorovSmirnov test.

The averages and the best fitness over all runs at each generation can be seen in Figure 8.5. This highlights that the best randomly initiated example of the AERN is more adept at traversing the map compared to the AGRN. Moreover, the best examples of the AERN are consistently better than that of the AGRN over all 50 generations. The average results produced by both networks are similar up until the 10^{th} generation, where the average fitness of the AERNs begins to improve, a trend which only occurs at the 20^{th} generation for the AGRNs. Both networks are improving in fitness over all generations. Overall, the AERNs produce consistently better controllers than the AGRNs and, additionally, the AERNs evolved faster than the AGRNs.

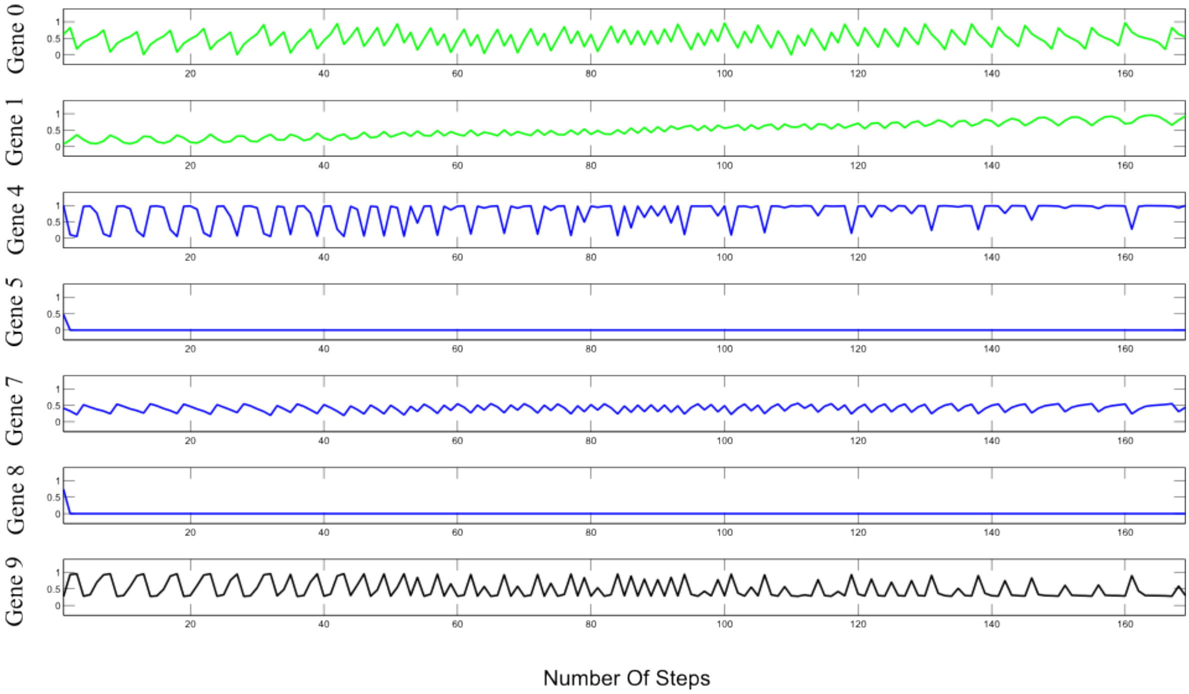


Figure 8.6: The expression values of the genes used by the AERN during a successful traversal of the standard map in one direction. Gene 0 represents the current x-coordinate of the trajectory, and gene 1 the y-coordinate. Gene 9 is the output.

8.2.3 Analysis

The analysis of the static structures of the AERNs provides insight into the functionality of the epigenetic control layer. It can be seen that the epigenetic frames play a significant part in the functionality of the AERNs by selecting certain genes to be active during each of the objectives within the task. By analysing the data held within the epigenetic frames it is apparent that they also reduce the amount of genes which are being used during the tasks. On average the AERN used 7 genes, whereas the AGRN used all 10 of its genes during the tasks. However, generally different genes were being used depending on the current objective. In all instances, the AERN used either the current y position of the trajectory, or the distance to the target (gene 1 or gene 2) to complete either objective. Additionally, less than half of the networks used the input for the x position of the trajectory (mapped to gene x) during execution, highlighting that the AERN did not require it to solve the task. An example of the expression values of the genes used for the AERN when completing a single objective within the task can be seen in Figure 8.6. Although the graph shows the AERN, it is a good representation of how both the AERNs and AGRNs go about solving the task. The output (gene 9 in this instance) is very similar for both networks during both objectives within the task, highlighting that the networks may be carrying out a similar control strategy.

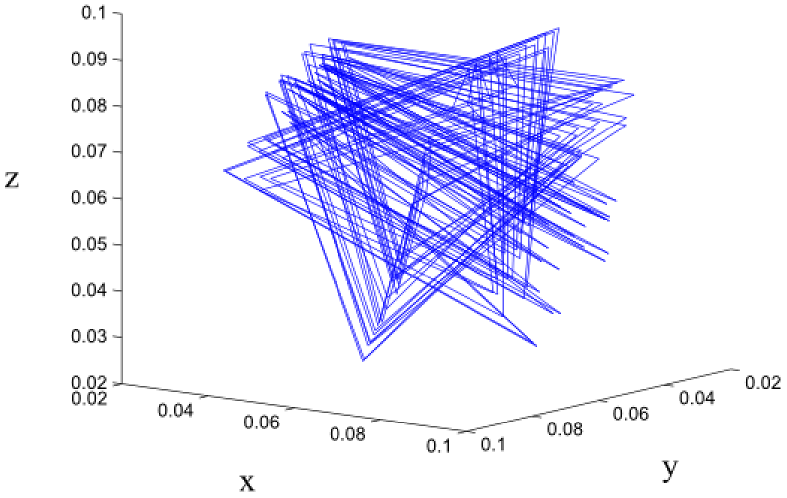


Figure 8.7: The phase portrait of an AERN over a single successful traversal of the standard map.

To generate a better understanding of the functionality of the networks, dynamical systems analysis is performed. In this instance Takens' theorem is used to reconstruct the attractor space of the networks (Section 3.5). To do this, the output variable from each network is taken at each step during a single successfully traversed path through the standard map. Time delay embedding is used to move this single dimensional data into three dimensions. This process reconstructs the attractor and improves visualisation of the attractor as it can be plotted in three dimensions. The resulting phase portraits for representative examples of the AERN and AGRN can be seen in Figures 8.7 and 8.8 respectively. These figures illustrate a similar set of interlocking triangular orbits, although there are slight variations in the overall structure. This demonstrates that the networks' functional dynamics are somewhat similar. This suggests that the implementation of the epigenetic analogue in the AERN does not necessarily change the network dynamics, but does allow certain genes to be discarded from the network and that other genes can become more specialised towards a certain sub task.

8.2.4 Reduced Dimensionality Controllers

An emergent property of the AERNs is that the structure of the epigenetic analogue provides a level of external control over the dynamics of the networks and, in turn, the current trajectory through the standard map. This is because the AERN in this instance uses the epigenetic control layer (frames) to specify that certain genes are used for a given direction of travel within the map. Hence, changing which frame is currently in use changes the direction

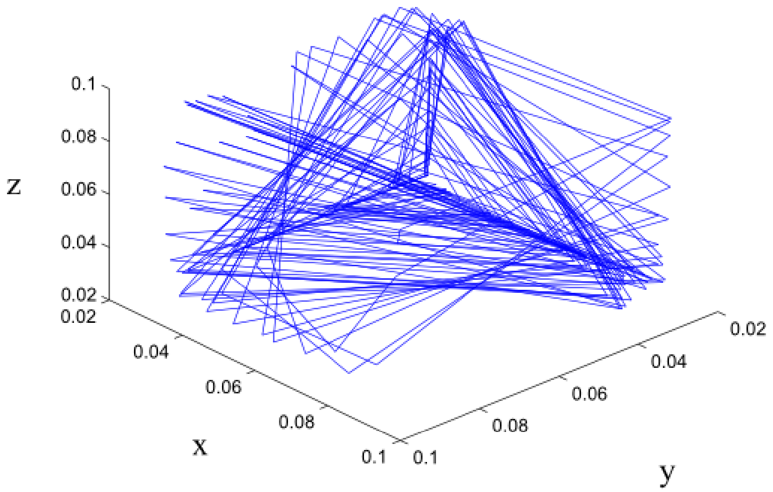


Figure 8.8: The phase portrait of an AGRN over a single successful traversal of the standard map.

of the trajectory. This gives the AERN the potential to be used as an interface to complex dynamical systems such as the standard map, which reduces the complexity of their dynamics to the extent where a user can have specific control of useful functionality of both systems via a simple interface. In effect, this amounts to a reduction in dimensionality from a user's perspective.

8.3 The Artificial Epigenetic Network

The AEN was inspired by the functionality of the AERN in the previous experiment. In this section the AEN is again applied to the control of Chirikov's standard map with some slight modifications. The AGRN that will be used as the measuring stick for the performance of the AEN differs from the AGRN in the previous section as it uses only a single network iteration per call (see section 6.7 for more detail), and is functionally identical to the AEN except for the epigenetic molecules.

8.3.1 Experimental Design And Parameters

The experimentation with the AEN has a similar basis to that of experimentation with the AERN. A key difference is the use of NSGA II. The advantage of NSGA II is that it avoids condensing objectives into a weighted sum, which generally improves evolvability, transparency and the diversity of solutions. There are three objectives: Objectives A (Figure

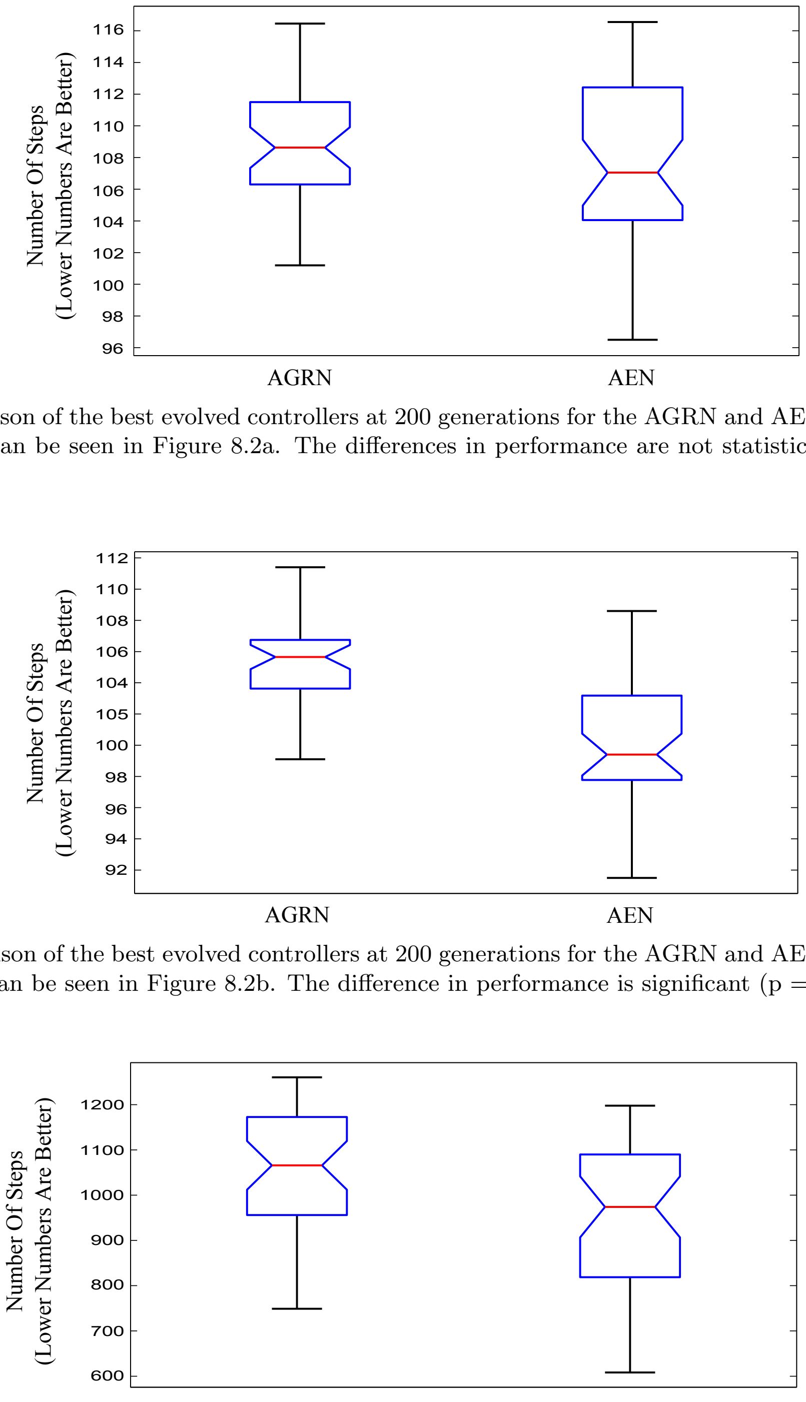
8.2a) and B (Figure 8.2b) are identical to that of the previous experiments, apart from that each objective is scored independently. Objective C is the average of the first two scores. For the first two objectives, there are 10 attempts at each. The average over these 10 attempts is taken as the score. Objective C has a valid score only if 1 of the 10 attempts in both objectives successfully manages to traverse the path. If this is the case, the third objective will return a score which is the average from the first two objectives summed together (The number of steps required to traverse the map in both directions). Otherwise, a score of 2000 is returned. The purpose of objective C is to encourage the networks to be able to traverse the map in both directions. Again, the networks will be provided with three inputs, the x and y positions of the trajectory and the distance to the centre of the target. The networks provide one output which modulates the k parameter of the map within the range of [1,1.1]. There are 40 runs for both the AEN and the AGRN.

The networks contain between 10 and 20 genes; in addition, the AEN contains between 3 and 5 epigenetic molecules. NSGA II has a population size of 500 and runs over 200 generations. The crossover rate is 0.5 and the mutation rate is 0.05. Tournament selection of size 4 is used.

8.3.2 Results

Results for the three objectives are compared in Figure 8.9. It is apparent that the AEN outperforms the AGRN over objectives B and C. These performance differences are statistically significant. Although the difference between the two networks for objective A are not statistically significant, the AEN still outperforms the AGRN in terms of the median results and additionally, the best results produced for objective A are better than that of the AGRNs. The difference in objective C is particularly important as it indicates that the AENs are more likely to solve the bidirectional control task. The AEN and the AGRN are identical except for the use of the epigenetic molecules in the AEN. This indicates that this performance increase is a direct result of the functionality of the epigenetic molecules.

The best instance at 200 generations over all objectives from both networks is represented in 3 dimensions in Figure 8.10. There is a clear distinction between the points of the AEN and the AGRN which demonstrates a performance difference in favour of the AEN.



(a) A comparison of the best evolved controllers at 200 generations for the AGRN and AEN for objective A. The task can be seen in Figure 8.2a. The differences in performance are not statistically significant (p = 0.5285).

(b) A comparison of the best evolved controllers at 200 generations for the AGRN and AEN for objective B. The task can be seen in Figure 8.2b. The difference in performance is significant (p = 0.037).

(c) A comparison of the best evolved controllers at 200 generations for the AGRN and AEN for objective C. The difference in performance is significant (p = 0.048).

Figure 8.9: A comparison of the best result from each run over the three objectives. The AEN shows a statistically significant improvement in objectives B and C. The Wilcoxon rank-sum test was used as the data did not fit a normal distribution according to the Kolmogorov-Smirnov test.

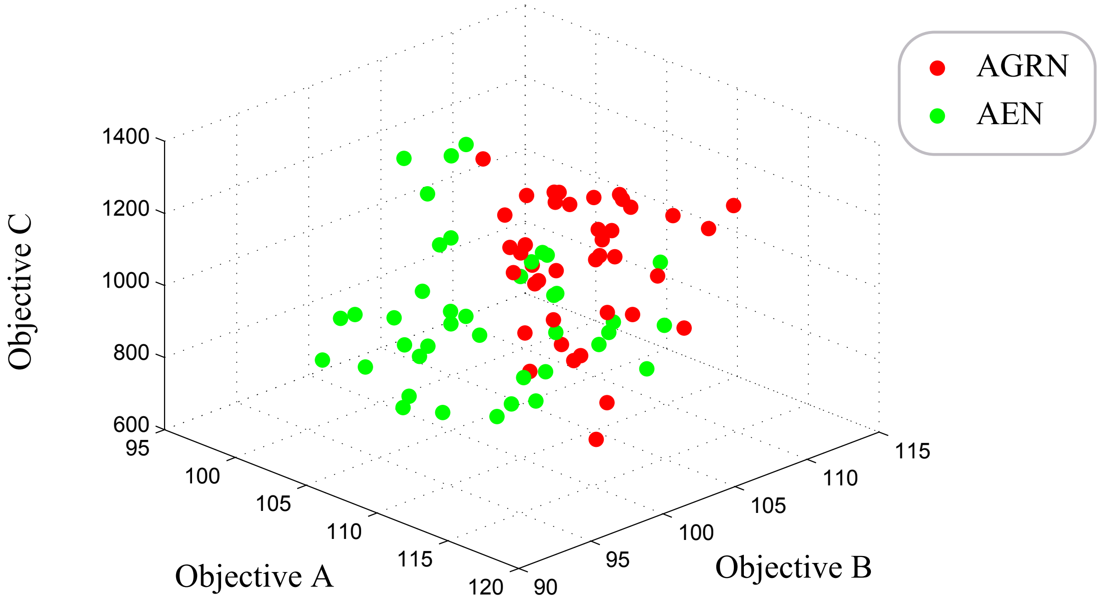


Figure 8.10: The best score for each objective achieved at the end of each run (effectively plotting the data from Figures 8.9a, 8.9b and 8.9c in three dimensions). It can be seen that there is clear distinction between the performance of the two networks, with scores from the AEN occupying the lower regions of the graph.

8.3.3 Analysis

Networks consisting of non-linear elements are intrinsically hard to understand, especially when combined with a dynamic topology. To reduce this issue, the networks that have been analysed have been reduced to their minimum working example (MWE). To achieve this, a gene is removed from the network, then the network is re-evaluated against the fitness function. If the network still maintains its behaviour, the removed gene is omitted. If not, the removed gene is placed back in the network. This is continued until all the genes have been classified as either functionally insignificant, or functionally important. In the case of the AEN, this process is continued with the epigenetic molecules. Through the creation of MWEs, it became clear that there are large numbers of genes within the network which are not used by the network in their overall functionality. The average size of the AGRN with non-functional genes omitted was 5, and the average size of the AEN was 3 genes and a single epigenetic molecule (i.e. 4 functional units). This is surprising as some of the networks were up to 20 genes in size, yet contained only 3 functional genes. A pervasive trait over the majority of networks is that they only required input 2 (distance to the target) to navigate the standard map in both directions. When creating the MWEs of the AENs, 34 out of the 40 networks used the functionality of the epigenetic molecules to dynamically alter gene expression. The remaining networks used the epigenetic molecules as static structures which prevented certain genes from executing. This suggests that a benefit of AENs is that they mask out interference from irrelevant parts of the network. This may provide a means for evolution to explore a larger network space, whilst mitigating against the likelihood of genetic interference in larger networks.

Dynamical Network Analysis

To generate a clearer understanding of how the epigenetic molecules affect the performance of the network, dynamical network analysis was conducted. To do this, the MWE of both the AEN and AGRN are taken, and the expression values of each active gene are plotted as a time series (Figures 8.11 and 8.12). The values were plotted for the objectives A and B; that is, to traverse the map from top to bottom, and then do the reverse. The point at which the target is reached in objective A in the case of both networks can be distinctly seen as the point approximately in the middle of the graphs where the spike on gene 2 occurs.

One of the first things this highlights is that the outputs from the network closely follow the information provided from gene 2 (distance to target). In the case of the AGRN, the output produced almost exactly follows the values from gene 2, except with a higher oscillatory range. Figure 8.12 shows the behaviour of the AEN. It is apparent that the outputs almost exclusively consists of two values, close to 0 and 1. However, the frequency of this output changes in accordance with the values from gene 2. This was verified via analysis of the static structure of the network, as the epigenetic molecule is only connected to gene 2. This suggests that the AEN and the AGRN achieve a solution to the problem of traversing the map differently, where the AGRN produces an output where the values are modulated to control the dynamics of the standard map and the AEN produces almost exclusively 2 values, but modulates the frequency between these values to control the behaviour of the trajectory. This is an interesting aspect of the of the AEN, as it has evolved a different way to solve that task which is better than most instances of the AGRN.

Dynamical Systems Analysis

The previous section suggested that the AGRN and AEN achieve solutions to the task in different ways. To ascertain if this is the case, phase portraits are constructed, visualising the difference between the networks' underlying dynamics. For continuity, the phase portraits are from the networks in figures 8.11 and 8.12. The phase portraits can be seen in figures 8.13 and 8.14. The phase portrait of the AGRN in Figure 8.13 shows a generally unstructured

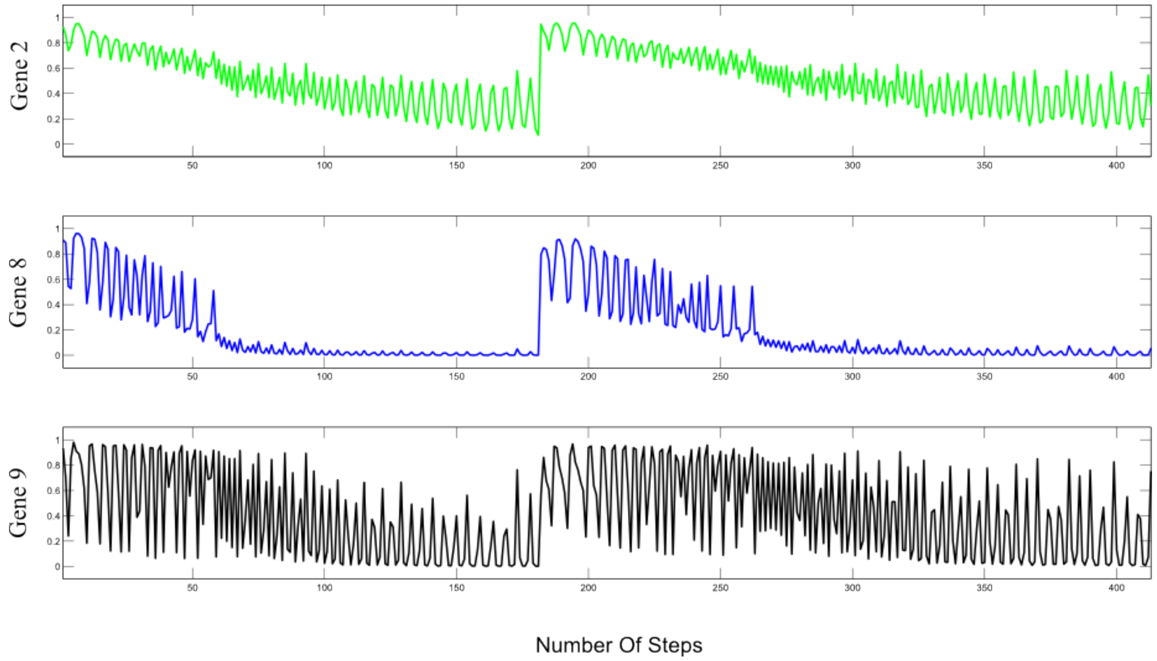


Figure 8.11: The expression values from the three genes in the smallest minimum working example of the AGRN when completing objectives A and B. This behaviour is representative of the majority of AGRNs.

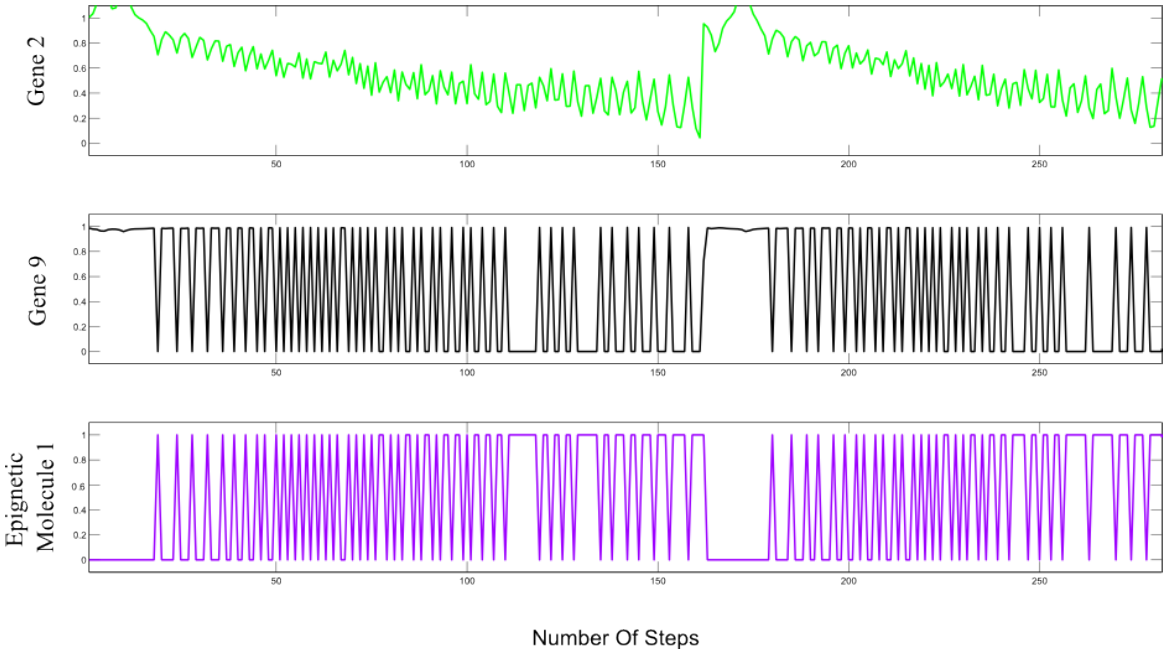


Figure 8.12: The expression values from the three genes in the smallest minimum working example of the AEN when completing objectives A and B. This behaviour is representative of the majority of AENs.

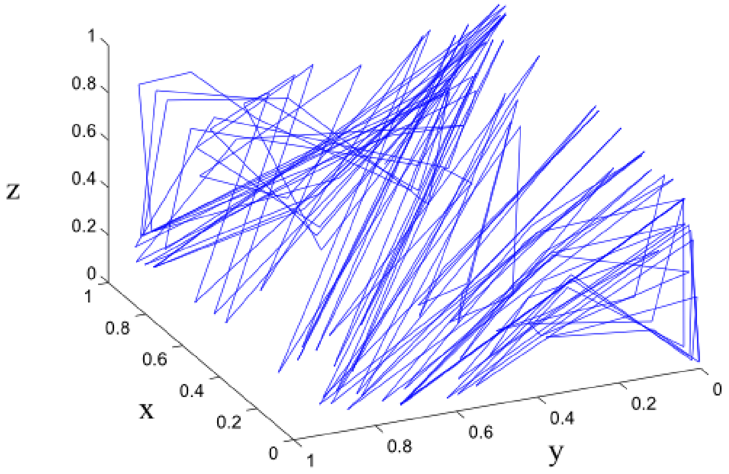


Figure 8.13: The phase portrait describing the dynamical properties of the AGRN from Figure 8.11.

state space, however, there are regions of the space which demonstrate different behaviours. In the centre of the portrait, there is somewhat of an oscillatory behaviour where the points within the portrait span the entire y axis. Also, at the higher and lower values of the y axis, a more cyclical behaviour can be observed.

The phase portrait of the AEN (Figure 8.14) shows a highly structured portrait consisting of three cyclic triangular structures. The lower triangle which starts at 0 on the x axis shows the behaviour when the trajectory of the map is near the target, whereas the other two triangles show the behaviour when the trajectory is traversing the centre of the map. This highlights that the AEN's solution has a more defined structure than that of the AGRN. This different structure reflects the discrete values outputs shown in Figure 8.12. It is likely that the ability to change the output of the network abruptly gives the networks the ability to more quickly traverse the regions of the map with different dynamics.

8.3.4 Reduced Dimensionality Controllers

An emergent property of the evolved AENs is the way in which the epigenetic molecules tend to organise themselves in terms of the dynamic regimes within the standard map. Furthermore, the activity of the epigenetic molecules changes at very sort intervals, as can be seen in Figure 8.12. Given the discretised output of the AEN caused by the epigenetic molecule, it is only possible to either deactivate the molecule, which will in turn set the output at 0 or activate the molecule which will set the output to either 0 or 1. However, by manually

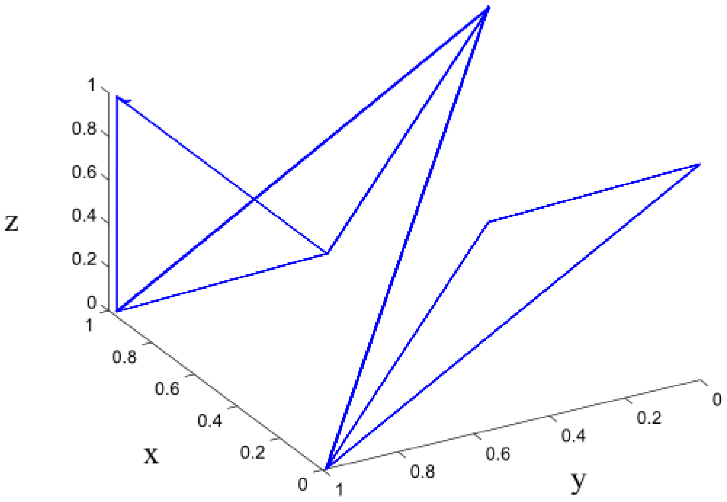


Figure 8.14: The phase portrait describing the dynamical properties of the AGRN from Figure 8.12.

controlling the activity of the epigenetic molecule it is possible to control which region of the standard map the trajectory resides within. The control is more subtle than that seen in the AERN, although somewhat less precise. The behaviour of modifying the activity of the epigenetic molecules can be described in terms of the highlighted regions in Figure 8.15. The regions where the trajectory can be most easily stabilised are either side of the green strip through the middle. Essentially, if the epigenetic molecule is deactivated on either sided of the green strip (which correspond to a defined region iof mixed ordered and chaotic dynamics in the centre of the map), the trajectory will remain in that space. However, if the epigenetic molecules is activated it will begin the process of traversing the centre of the map, and over to the other side. Again, the trajectory will remain in that region almost indefinitely whilst the epigenetic molecule is deactivated. The direction the trajectory travels is dependent on the distance from the target (mapped on gene 2). Hence, the epigenetic structures provide a level of control over the dynamics of the network, and in turn the standard map. This means that the AEN can act as a reduced controller for a trajectory within the standard map via modification of the epigenetic molecules.

8.4 Summary

The results from the experiments described above have highlighted that the epigenetically inspired gene regulatory networks generally out perform their counterparts which contained no

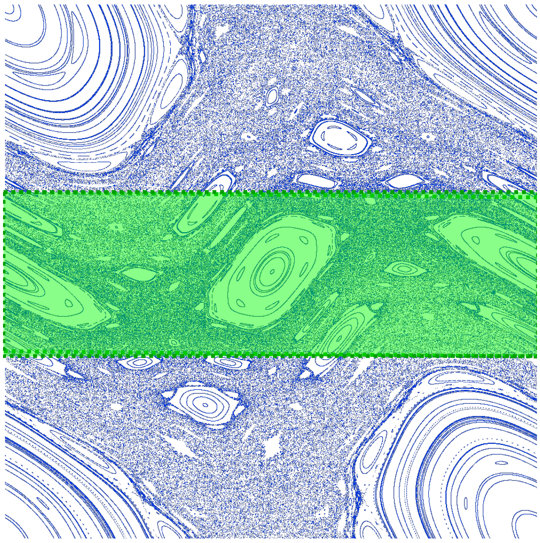


Figure 8.15: An illustration of the regions in which the trajectory will stay if the epigenetic molecule is deactivated.

epigenetic analogue when applied to the control of dynamics within Chirikov's standard map. The AERN served as an initial model of an epigenetically inspired network architecture, highlighting that partitioning the network according to specific objectives improved performance and can create an emergent reduced dimensionality controller. This provided inspiration for the creation of the AEN, which uses a dynamic epigenetic structure. Similarly, the AEN outperformed its counterpart which contained no epigenetic analogue. However, the varying architectures of the epigenetic structures had a profound effect of their emergent behaviour. Because the AERN requires a predefined rule describing when to switch its epigenetic frame, its behaviour is constrained by this. The AEN, however, evolves an epigenetic structure autonomously during evolution, and in turn evolves the functions which define the epigenetic structure's activity. This is a key difference, because it is reasonable *a priori* to assume the partitioning of the networks in terms of objective is an optimal decision, but the AEN evolves an alternative partitioning which modifies its activity based on dynamical regimes occurring in the task. It is because of the static nature of the AERN and the required user input to define the rules which control the epigenetic analogue which makes the AEN a more appropriate model for the control of systems with varying and poorly understood dynamics.

The AEN developed an interesting behaviour in which the epigenetic molecule instead of explicitly regulating the expression of the genes within the network, predominately regulated the output genes' expression between two contrasting values and regulated the frequency of this switching behaviour. This evolved characteristic gives the network a more temporal structure, which allows the the AENs to operate in the frequency modulation domain (FM) and the more typical amplitude modulation (AM) domain. This is a fundamental characteristic of the networks because it demonstrates that the epigenetic functionality can produce a novel regulatory behaviour which is also highly functional.

The standard map has been utilised in the experimentation as a proof of concept model. As hypothesised earlier in section 7.3, there are only certain conclusions about the functionality of the AENs that can be drawn from experimentation from the standard map. However, critically, the AEN's showed their ability to generate emergent behaviours which successfully take advantage of the dynamic functionality of the epigenetic molecules. In addition, from the evidence gained from looking at the dynamic functionality of the networks, it is clear that the tasks requires some form of computational complexity solve well. On balance, at present, it is difficult to quantify this level of computational complexity. From the results shown in this chapter, it is clear that the standard map task has served as a suitable proof of concept tasks in which to primarily analyse the networks.

In the following chapter the AEN is applied to the coupled inverted pendulums task to provide a better understanding of their computational complexity. In addition, this will provide evidence as to whether the results acquired using the standard map are generalisable to more real word physical systems.

Chapter 9

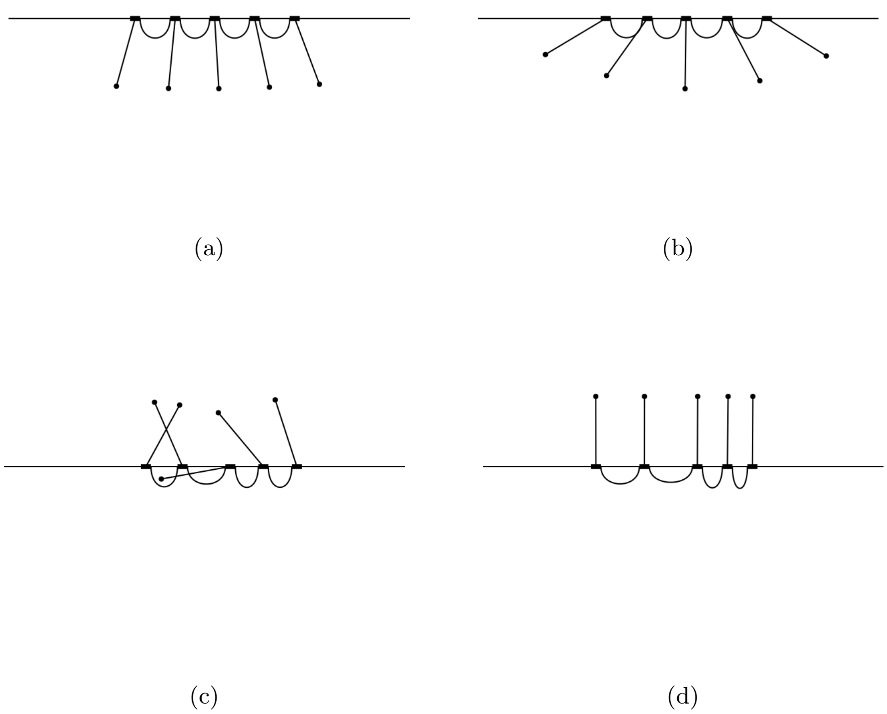
Coupled Inverted Pendulums

Contents

In the previous chapter the artificial epigenetic network (AEN) was applied to the task of controlling a trajectory through Chirikov's standard map. The standard map is an approximate model of a range of physical systems. In this chapter, to investigate the AENs ability to control different tasks, it is applied to a specific mechanical model, the coupled inverted pendulums task. The coupled inverted pendulums task is a benchmark which was developed as a proxy for evolving decentralised robotic controllers (Hamann et al., 2011; Cazenille et al., 2012). The networks are compared against AGRNs (The same as the AENs, but with their epigenetic elements removed). Their performance can be analysed, and network analysis conducted to understand the dynamical properties of the network and how it solves the task.

9.1 Description Of The Coupled Inverted Pendulum Task

The coupled inverted pendulums task consists of a set of pendulums (between 1 and 5) which are mounted to carts (1 per cart) on a 1-dimensional track (Figure 9.1). Assuming there is



(a) (b)

(c) (d)

Figure 9.1: An illustration of the 5 pendulum task being solved, with an optimum behaviour achieved. The carts initially start to swing, generating momentum in the pendulums (figures 9.1a and 9.1b). Once the pendulums are in the upper equilibrium state, the are maintained in that state (Figure 9.1d).

more than one pendulum, these carts are coupled together by a tether which restricts each cart's movement. The carts exist within a finite space, and must avoid the edges of this finite space. The objective of the task is to move the pendulums from the lower equilibrium position (Figure 9.1a), and balance them in the upper unstable equilibrium position (Figure 9.1d) via the movements of the carts to which they are attached, without exceeding the length of the tether. To do this requires a periodic swinging movement to generate momentum in the pendulums; when the upper unstable equilibrium point is reached, the carts have to adapt their periodic behaviour to maintain it in that position.

The coupled inverted pendulums benchmark is a proxy for distributed robotic controllers; because of this, the sensor values do not directly deliver all the information within the simulation. Rather, this information is shared between different sensors, as shown in Figure 9.2 and Table 9.1. This creates an environment which is similar to that of distributed robotic controllers, where only certain information is valid at a certain time.

ID	Sensor Name	System to sensor mapping
S_0	Pendulum Angle 0	ø ∈ [0,0.5π] → [127, 0], 0 else
S_1	Pendulum Angle 1	ø ∈ [1.5π,2π] → [0,127], 0 else
S_2	Pendulum Angle 2	ø ∈ [0.5π,π] → [127, 0], 0 else
S_3	Pendulum Angle 3	ø ∈ [π,1.5π] → [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 ∈ [-2,0] → [127,0], 0 else
S_7	Cart Velocity 1	v ∈ [0,2] → [0,127], 0 else
S_8	Angular Velocity 0	w ∈ [-5π,0] → [127,0], 0 else
S_9	Angular Velocity 1	w ∈ [0,5π] → [0,127], 0 else
A_i	Actuators 0	A_i ∈[0,127], for i ∈ 0,1
u	Motor Control 0	2(A_0/127 - A_1/127) → [-2,2]

Table 9.1: Sensor outputs describing the state of each cart. These values are mapped onto the input genes of the networks within the ranges of [-1,1].

In order to increase the difficulty of the task, more carts can be placed on the same track. This limits the possible movements for each cart, requiring a more cooperative approach. The width of the track stays the same, hence each cart will have proportionally less space in which to operate. The carts are each controlled by two actuator values (A_0 left actuator and A_1 right actuator). The difference between the two actuator values determines the acceleration of the cart. Each cart is controlled independently of the others.

The fitness function is an aggregate of the number of time steps which the pendulums spend in the upper equilibrium ($\sum_{t=0}^{t_{max}} \sum_{j=0}^{P} \frac{|ø_j(t)-\pi|}{t_{max}P\pi}$) where P is the number of pendulums, t is the time steps and ø is the pendulum angle). A fitness value of 1 indicates that the pendulum(s) spend all time steps in the upper equilibrium position, and a fitness value of 0.5 indicates

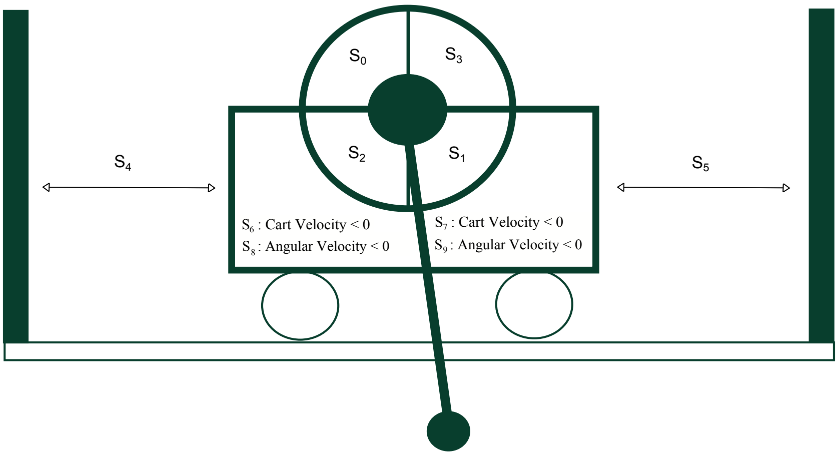


Figure 9.2: Illustration of how the variables from Table 9.1 map onto a cart.

Parameter	Value
gravitational acceleration	9.81 ms^{-2}
pendulum length	0.5m
max. positive acceleration	7.0 $\frac{m}{s^2}$
min. positive acceleration	8.5 $\frac{m}{s^2}$
world width	2m
tether length	0.35m
proximity sensor range	1.0m
cart width	0.1m
time steps (t)	4000

Table 9.2: The parameter values which are used for the coupled inverted pendulums task.

that the pendulum(s) spend half their time in the upper equilibrium position. A fitness of 0 means that the pendulum(s) spent all their time in the lower equilibrium. The dynamics of the pendulum are calculated using the Runge-Kutta third order method of integration with a time step of δ t = 0.01 (Hamann et al. (2011) [1]). To make the simulation more realistic noise is added to the sensor values in the form of a random number in the range [-3,3].

9.2 Experimental Design and Parameters

The artificial epigenetic networks (AEN) are applied to the task of controlling the coupled inverted pendulums. The controllers are evolved using a genetic algorithm. The results from these experiments are compared with the results from the AGRN. The parameters of the genetic algorithm are a population size of 500, a tournament selection of size 4, crossover rate of 0.5 and a mutation rate of 0.05. The genetic algorithm is run over 200 generations which equates to 100,000 evaluations per experiment. The networks (both AEN and AGRN) have between 12 and 25 genes. The AEN has between 3 and 5 epigenetic molecules per network.

The sensor values are mapped onto the network in order of gene index. Hence, gene 0 will take the value of sensor 0, gene 1 will take the value of sensor 1 and so forth. The sensor values are provided between the range of [0,127], and these will be linearly normalised to the values [-1,1] before being mapped to the genes. The networks use the reference space to derive their connections (section 6.3.1), which allows them to evolve their size and structure. The networks are evolved to control 1, 3, and 5 pendulums, in separate experiments. To do this, for the experiments containing more than a single cart, a single controller is evolved and mapped onto each cart.

[1] The code for the coupled inverted pendulums task is available online (Hamann & Schmickl, n.d.)

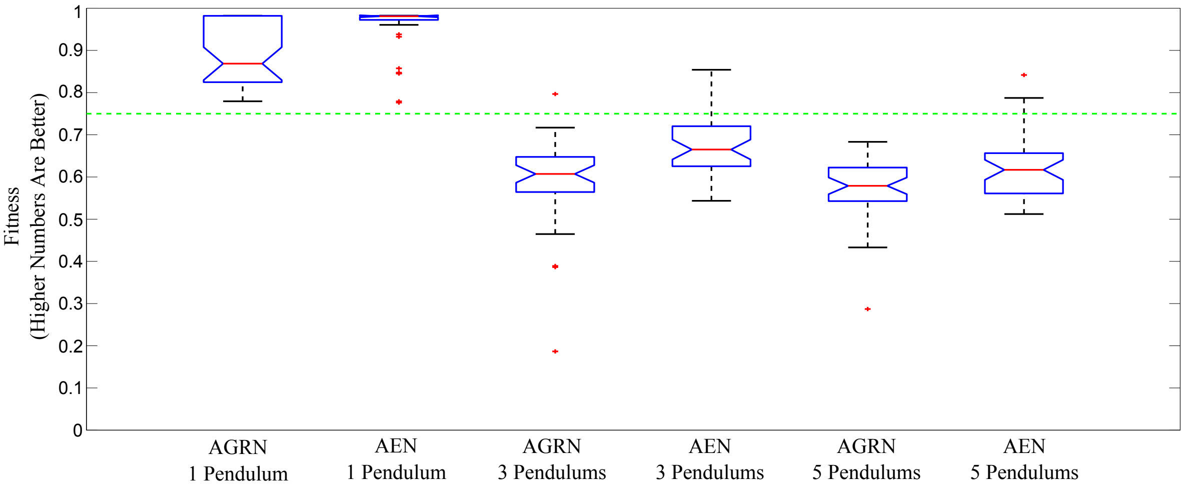


Figure 9.3: A summary of the performance of the AEN against the AGRN with 1, 3 and 5 pendulums over 40 runs. The dotted green line at 0.75 denotes the fitness where the controller can maintain the pendulum in the upright equilibrium position. Controllers with fitness above this threshold are able to produce the optimum balancing behaviour (Figure 9.1d).

9.3 Results

The results of the experiments can be seen in Figure 9.3. The AEN evolved an optimum behaviour for 1, 3 and 5 pendulums, whereas the AGRN only evolved an optimum behaviour for 1 and 3 pendulums. Furthermore, for 3 pendulums the AGRN only evolved 1 controller out of the 40 with an optimum behaviour, an outlier of the distribution. The AEN produces a significantly better fitness over 200 generations for 1 (p = 0.029), 3 (p = 7.5 x 10 $^{-5}$) and 5 (p = 0.010) pendulums (using the Mann Whitney Wilcoxon test as the data did not fit a normal distribution).

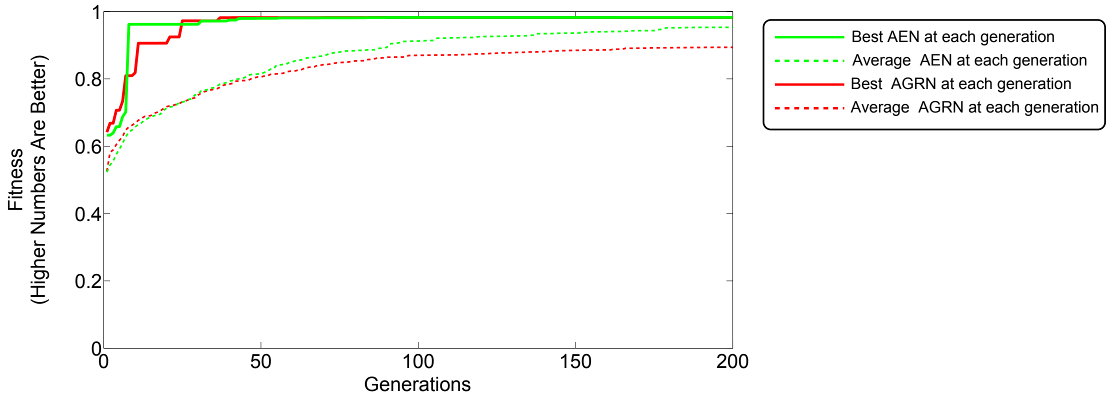
The results show that the AEN is producing significantly better final solutions than the AEN. To better understand how the AENs achieve this compared to the AGRNs this, the best and average fitness of the networks over all generations need to be analysed. These are plotted in Figure 9.4. When looking at the performance over time for the single pendulum task (Figure 9.4a), it can be seen that the best instance of the AEN has a very large fitness jump from 0.75 to 0.97, whereas the rise in fitness of the AGRN consists of many smaller jumps. Below 5 generations, there are two instances where the best AGRN is better then the best AEN. This is shown also when looking at the averages for the networks, where up until 40 generation, the AGRN is at points outperforming the AEN. Post 50 generations, it can be seen that the average fitness of the AEN remains higher than that of the AGRN.

For the 3 pendulum task (Figure 9.4b), the graphs show a different evolutionary path. In terms of the best instances in the population, the optimum behaviour is found at $\sim$ 125 generations for the AEN and $\sim$ 150 generations for the AGRN. Again, similar to that of the single pendulum task, there is a point at $\sim$ 35 generations where the best fitness of the AGRN eclipses that of the AEN. However, shortly afterwards, the AEN begins outperforming the AGRN. In terms of the average, the AEN can be seen to evolve significantly faster than that of the AGRN, with the best average fitness of the AGRN (0.59) being eclipsed by generation 71 of the AEN.

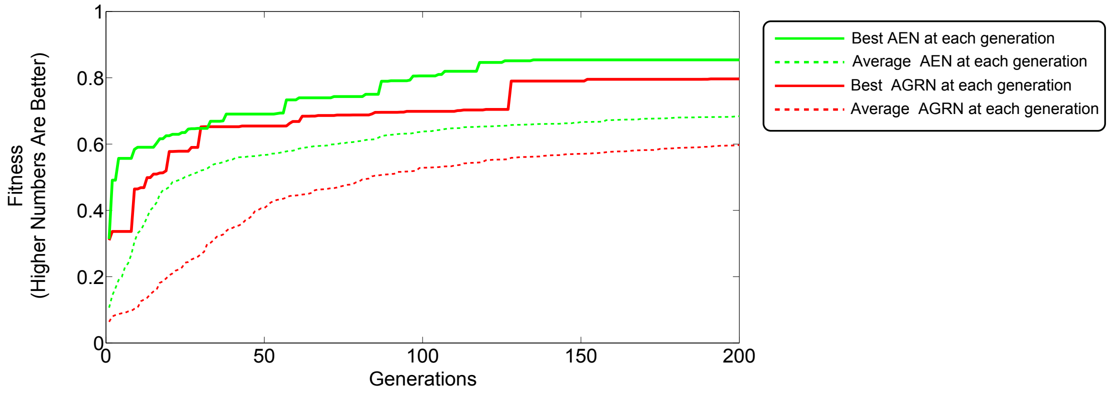
The AGRN fails to produce an optimum behaviour for the 5 pendulum task (figures 9.3 and 9.4c). The evolutionary curve for the best average examples of the AEN is much flatter that that of the AGRN. This could be explained by the AENs producing a significantly better randomly initiated example that that of the AGRN (0.45 for the AEN, 0.08 for the AGRN). However, this is uncharacteristic, as the 5 pendulum task is more difficult than that of the 3 pendulum task, yet the randomly initiated examples for the AEN are significantly better. This reflects when looking at how the best instances evolve, with the AGRN having a steeper curve compared to that of the AEN. The averages for the 5 pendulum tasks look very similar to that of the three pendulum task, with the AEN evolving faster than the AGRN, however, at 200 generations, the averages are very similar (0.57 for the AGRN and 0.61 for the AEN), a trend which is mirrored in the final distributions of the best results for 5 pendulums (Figure 9.3).

The data shown in Figure 9.4 demonstrates that the coupled inverted pendulums tasks has many local optima. In the 5 pendulum task, the AGRN spends the last 100 generations in a local optima. From looking at how the networks learn over time, it is apparent that the AEN is more adept as escaping these local optima, and thus more successful at reaching the global optimum.

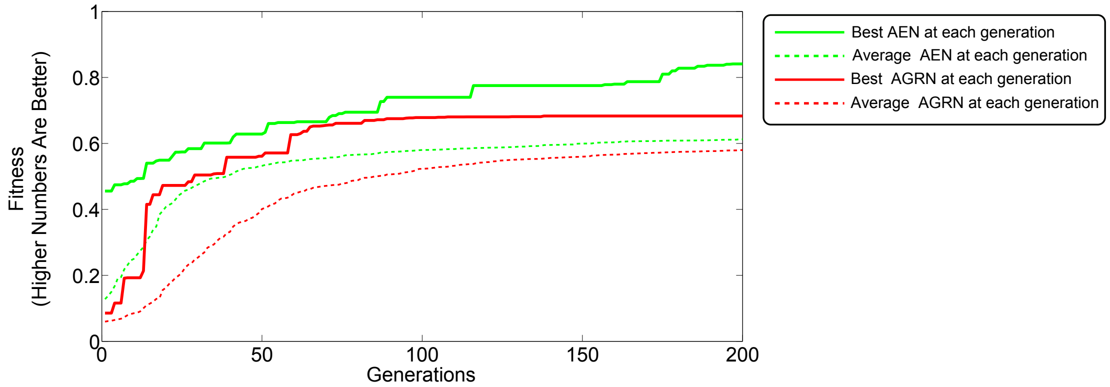
It is apparent from looking at these results that the AEN outperforms the AGRN when applied to the coupled inverted pendulums task. The only difference between the AEN and the AGRN is the use of epigenetic molecules, therefore this increase in performance is an emergent property of the interactions between the genes and epigenetic molecules. In order to ascertain exactly how the epigenetic molecules are beneficial, an analysis of their behaviour is conducted.



(a) A graph showing the evolution of both the AEN and AGRN over time when applied to the single pendulum task.



(b) A graph showing the evolution of both the AEN and AGRN over time when applied to the 3 pendulum task.



(c) A graph showing the evolution of both the AEN and AGRN over time when applied to the 5 pendulum task.

Figure 9.4: The set of graphs depicting the evolution of both the AEN and AGRN over time for the 1,3 and 5 pendulum tasks. The solid lines show the best controller at that generation over all runs, and the dotted lines show the average over all runs at that generation.

9.3.1 Network Analysis

The evolved networks were a minimum of 12 genes in size. To make analysis as straightforward as possible, the minimum working examples (MWE) from each network were analysed (MWEs are described in section 8.3.3). From this, the networks are reduced in size by ≈ 50% of their genes and epigenetic molecules. Because the sensor values from the pendulum are mapped onto 10 genes, it emphasises that not all of the sensor values are required to solve the task. Specifically, inputs 4,5 and 6 (Table 9.1) were frequently found to not contribute to the optimal behaviour. The networks that will be analysed were evolved to solve the 3 pendulum task. This is because it is computationally more difficult for the networks to solve optimally than the single pendulum task (having to take into account the proximity of other carts on the track), and both the AGRN and AEN can produce the optimal behaviour required to balance the carts in the upper equilibrium position.

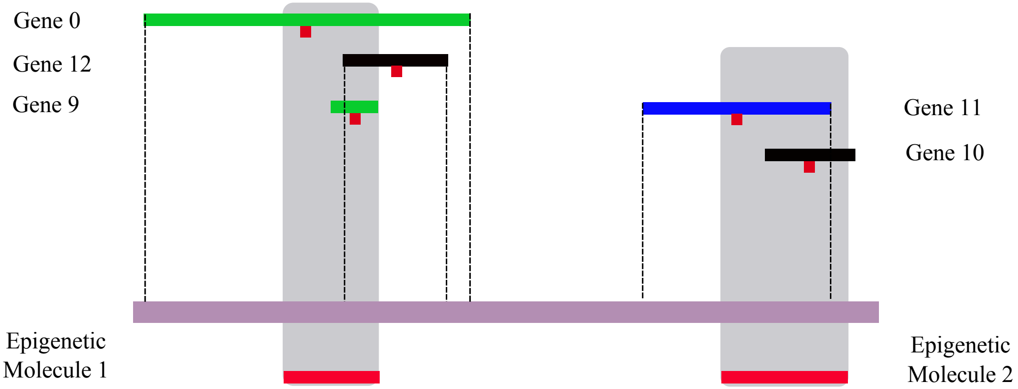


Figure 9.5: An example of the reference space (section 6.3.1) for a typical minimum working example AEN evolved for the 3 pendulum task. Only the genes which are required to generate the optimal behaviour are shown. The green genes are input genes, which take the tasks variables according to Table 9.1 (i.e. Gene 0 is mapped to sensor 0, etc). Hence all genes with an index less than 10 are input genes. The blue genes perform regulatory functions (gene 11), and the black genes are the output genes, which are mapped to each cart. The epigenetic molecules can be seen to take certain genes as inputs, and from this, they can determine those genes' activity.

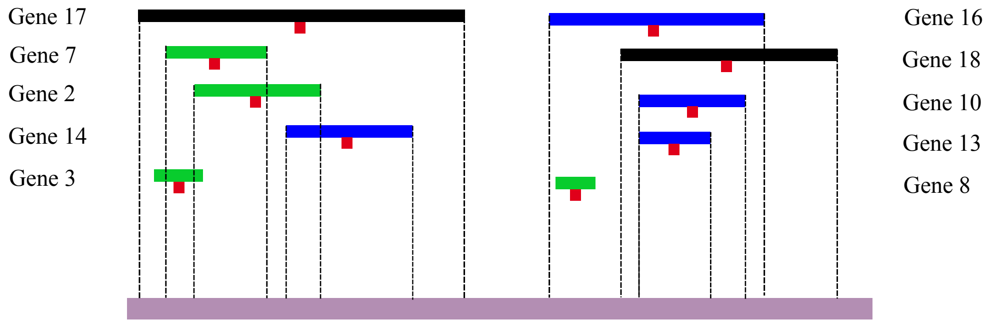


Figure 9.6: The reference space of the only AGRN which evolved to optimum behaviour on the 3 pendulum task. This is the visualisation of the minimum working example, hence only the genes which are required to generate the optimal behaviour are shown. The green genes are input genes, which take the task's variables according to Table 9.1 (i.e. Gene 0 is mapped to sensor 0, etc). Hence all genes with an index less than 10 are input genes. The blue genes perform regulatory functions (genes 10, 13, 14, and 16), and the black genes are the output genes, which are mapped to each cart.

Static Network Analysis

To better understand the networks we begin with static network analysis. This plots the genes in the reference space to view their connectivity within the network, and how it is structured (the reference space is explained in section 6.3.1). Two examples can be seen plotted in figures 9.5 and 9.6. It should be noted that Figure 9.5 represents the majority of instances of the AEN, in terms of both network inputs and structure. However, there was only a single AGRN which evolved optimal behaviour, hence it was the only viable option to analyse.

It is apparent that the networks partition themselves into two completely separate regulatory circuits. Each partition always contains an output (the network has 2 outputs, one for each wheel of the cart, the acceleration of the cart being the difference between the two). This trend was seen over every network which was analysed. More interestingly with the AENs, a significant portion of each regulatory circuit is controlled by an epigenetic molecule. This was found with all the AENs analysed. Moreover, in all instances, the epigenetic molecules operated dynamically, changing the network structure during execution.

It can be seen that the AGRN and AEN use different sensor values (i.e, inputs) to produce their optimum behaviour. The AGRN typically used variables 2, 3, 7 and 8 (Table 9.1) which corresponds to the angle of the pendulum in the upper equilibrium, the velocity of the cart and the angular velocity when in the lower equilibrium. The AEN typically only uses

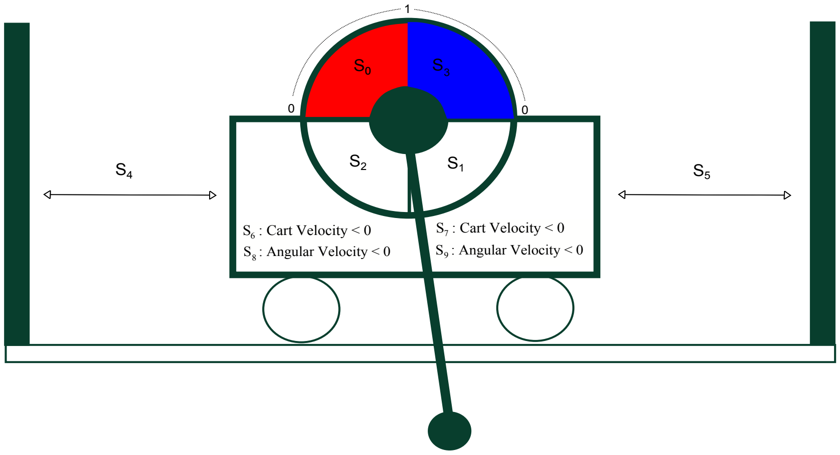

1
0
0
S_0
S_3
S_4
S_2
S_1
S_5
S_6 : Cart Velocity < 0
S_8 : Angular Velocity < 0
S_7 : Cart Velocity < 0
S_9 : Angular Velocity < 0

Figure 9.7: An illustration of the sensor maps from Table 9.1 on a cart. Sensor 0 (red section) is typically used by the AEN to denote when its in the upper equilibrium. Sensor 3 (blue section) is typically used by the AGRN to determine when it is in the upper equilibrium position. Both sensors produce a higher value when the pendulum approaches the vertical position in the upper equilibrium.

task inputs 0 and 9 which correspond to angular position and velocity when in the upper equilibrium. This suggests they are using different logical approaches to solve the tasks. A surprising trait is that very few networks used any kind of proximity detection to avoid other carts. Instead, they tended to evolve mechanisms that reduced the likelihood of collision.

Dynamical Network Analysis

To understand the networks' operation dynamically, firstly the MWE of the networks from figures 9.5 and 9.6 have the expression values of each of their active genes plotted. In order to generate the most useful information, the plots are not over the 4000 time steps, but over the transitional period between the swinging of the pendulums (Figure 9.1b and 9.1c) and their balancing in the upper equilibrium (Figure 9.1d). The plot of the AEN demonstrates that the epigenetic molecules play a key part in developing the behaviour required to balance the pendulums. The most obvious trend is that the epigenetic molecules are directly controlling the two regulatory circuits. One epigenetic molecule creates an oscillatory circuit (Genes 10 and 11, and epigenetic molecules 2, Figure 9.9) throughout execution. The other regulatory circuit is controlled dynamically, and is responsible for the change in dynamics between the swinging behaviour and the balancing behaviour.

Specifically, during the AEN's execution, gene 0 is providing a value according to the pendulum angle sensor 0, which only provides a reading if the pendulum is in the upper equilibrium position (Figure 9.7). This sensor provides a value of 1 when the pendulum is in the vertical position and -1 when in the horizontal position. Sensor 9, mapped to gene 9, provides a reading of the speed of the pendulum in the counter-clockwise direction. Both these values are used by epigenetic molecule 1 to determine its activity. There are two rules epigenetic molecule 1 adheres to. Firstly, if angular velocity is high, become active. Secondly, if the angular velocity is low, but the pendulum position is high, become active. With the AEN, the pendulum is always entering the upper equilibrium via moving in the counter clockwise direction; hence, as soon as the pendulum enters the field highlighted in Figure 9.7, the sensor registers a high value, and epigenetic molecule 1 becomes active, blocking out genes 0 and 9. This then removes any possible inputs to output gene 12, which due to the parametrisation of its sigmoid function, produces a value close to 1. This moves the cart from right to left sharply until the pendulum leaves the field highlighted in Figure 9.7. This then produces a sensor value close to -1, which then deactivates the epigenetic molecule, and gene 12 can then become active, producing a value close to 0, which pushes the cart from right to left. This behaviour is what controls the pendulum in the upper equilibrium, and can be seen in part in Figure 9.9. The oscillating effect of the other regulatory circuit works as a dampener to control the sharp moments created by epigenetic molecule 1 causing large fluctuations in gene 12's expression. This is probably the reason why the networks do not use the proximity inputs, as they move slowly in unison by default, meaning it is rare for the carts to either go too far apart or collide.

The AGRN, although using different genes, solved the coupled inverted pendulums tasks in a similar manner to that of the AEN. A key difference is that more genes are required to solve the task than the genes and epigenetic molecules combined. This is also the case when considering the MWEs of both networks. As can be seen from Figure 9.6, genes 2, 3, 7, 14 and 17 form a regulatory circuit, and the other circuit comprises of genes 8, 10, 13, 16, and 18. This is reflected when looking at the dynamics shown in Figure 9.8. The output from gene 17 matches the peaks and troughs of the input of gene 3. Gene 3 registers a high sensor value when the pendulum is close to the upper equilibrium point, however, if the pendulum moves slightly into the section of sensor value 0, gene 3 will produce a value close to -1. Hence, when gene 3 starts rapidly fluctuating at around 1440 time steps, the pendulum is within the upper equilibrium. This acts as a type of soft switch for the output gene 17. Gene 18 produces an oscillatory behaviour, which is controlled by the angular velocity sensor mapped to gene 8. When the velocity is low, gene 18 starts to oscillate, and this combined with the behaviour of gene 17 holds the pendulum in the upper equilibrium position.

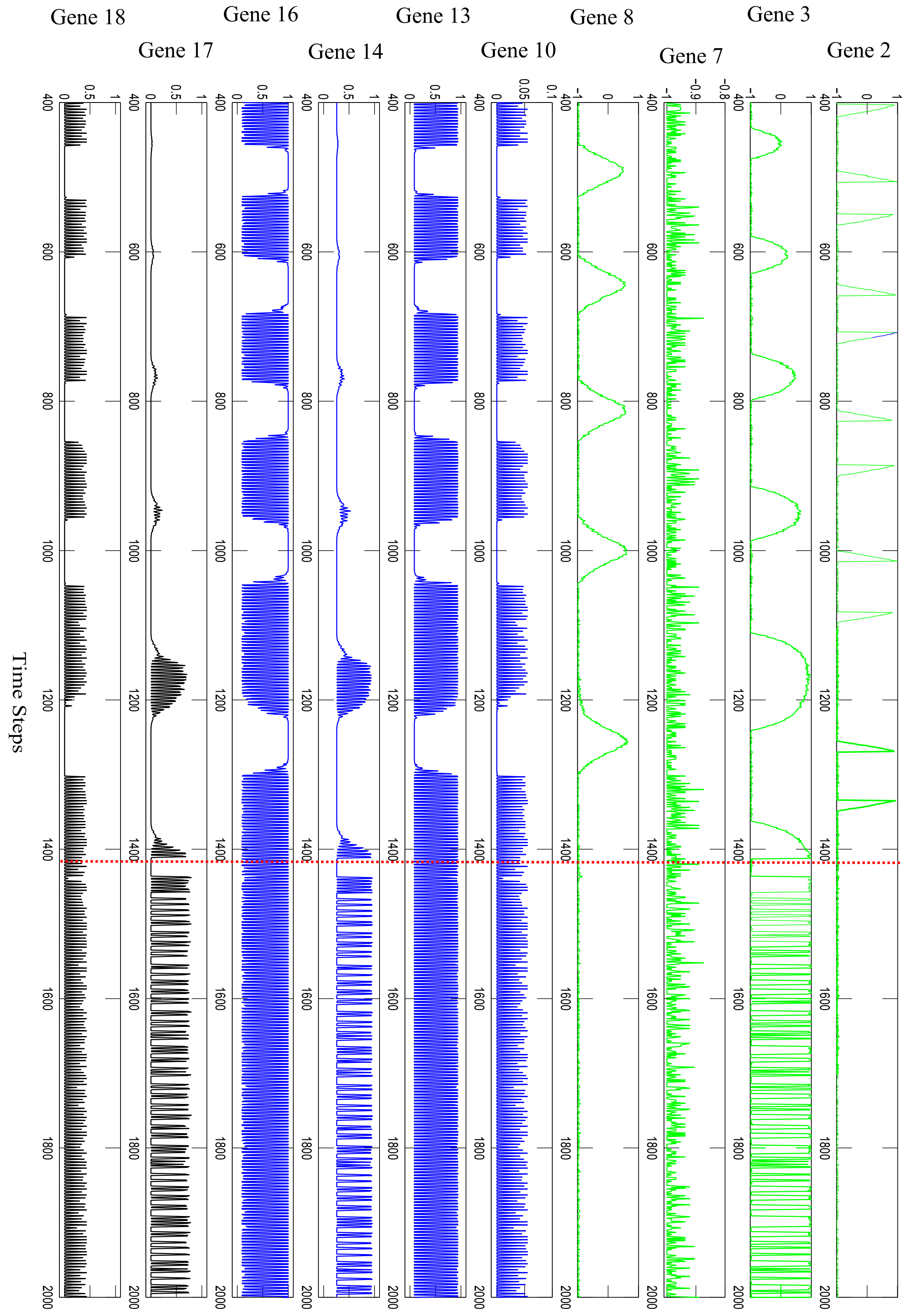


Figure 9.8: An illustration the network shown in Figure 9.6, showing the gene values produced throughout execution. The window shows time steps 200-2000, which capture the behaviour of the pendulums swinging, and then the change of expressions when the pendulums are in the upper equilibrium. This change is marked with the red dotted line at approximately 1420 time steps. The input genes (sensor values) are marked as green, regulatory genes as blue, and output genes as black. It is apparent that output gene 17 mimics the activation of input gene 3, which marks the pendulum position (Figure 9.7). Output gene 18 is active when the angular velocity (gene 8) is low, which is true during the swinging phase, and when the pendulums are in the upper equilibrium.

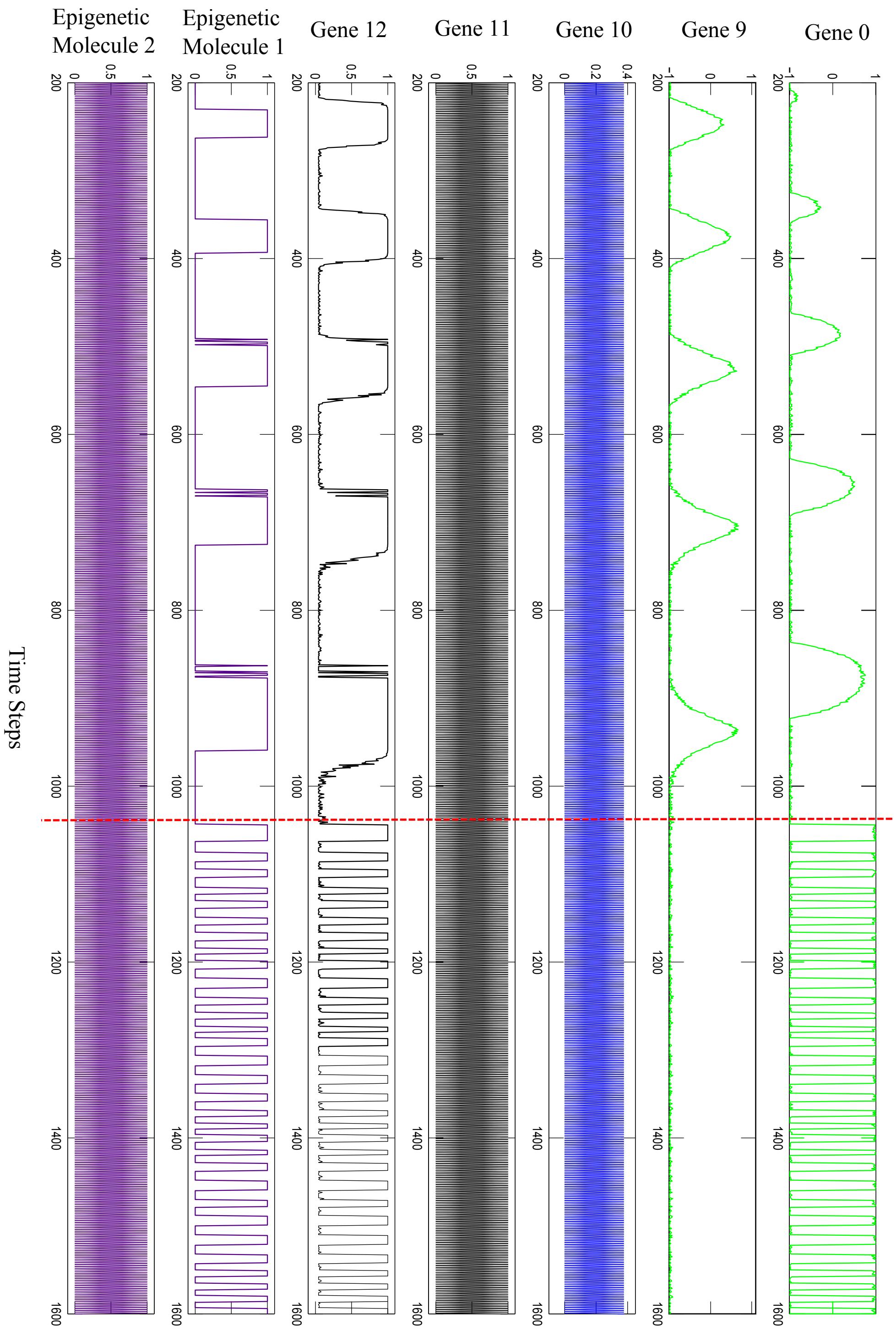
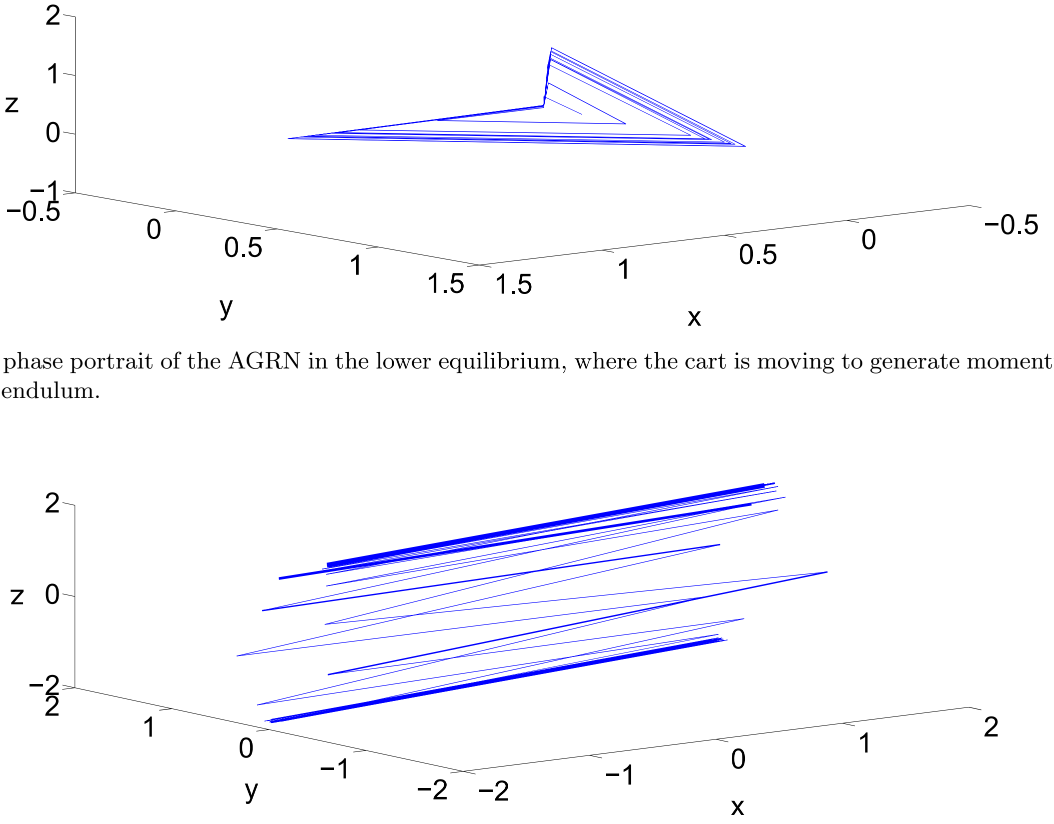


Figure 9.9: An illustration the network shown in Figure 9.5, showing the gene values produced throughout execution. The window shows time steps 200-1800 steps, which capture the behaviour of the pendulums swinging, and then the change of expressions when the pendulums are in the upper equilibrium. This change is marked with the red dotted line at approximately 1010 time steps. The input genes (sensor values) are marked as green, regulatory genes as blue, and output genes as black. The epigenetic activation is marked as purple. It is apparent that epigenetic molecule 2, and genes 10 and 11 are consistently oscillating throughout execution. Epigenetic molecule 1 become active when gene 0 (sensor 0 in Figure 9.7) produces a high value when in the upper equilibrium. Epigenetic molecule 1 also becomes active when gene 9 (angular velocity) produces a high value. Gene 12s expression is indirectly controlled by epigenetic molecule 1 (as can be seen in Figure 9.5).



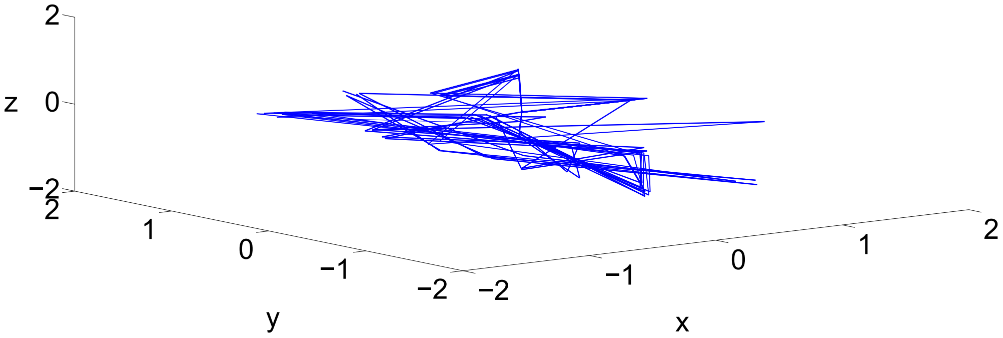
(a) The phase portrait of the AGRN in the lower equilibrium, where the cart is moving to generate momentum in the pendulum.

(b) The phase portrait of the AEN in the lower equilibrium, where the cart is moving to generate momentum in the pendulum. It is to be noted that there are two key regions in which the attractor is predominantly located (The darker blue regions).

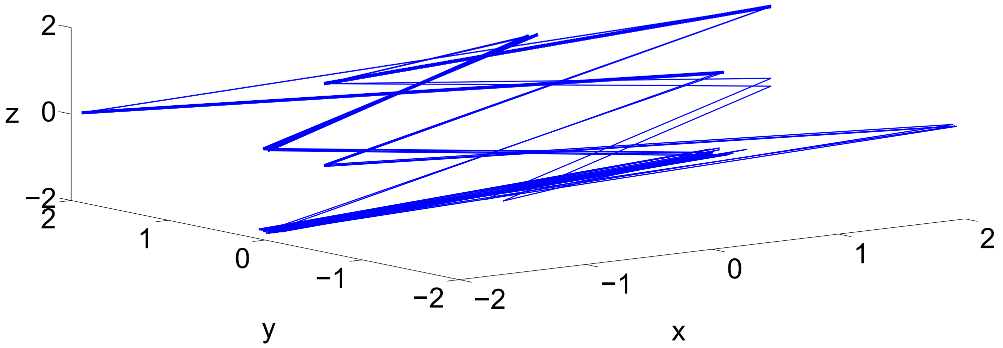
Figure 9.10: The top image (a) is depicting the phase space of the AGRN when swinging the carts in the lower equilibrium position, and (b) showing the phase space of the AEN when swinging the pendulums in the lower equilibrium position. The phase portrait was created using the difference between the outputs of the network, and time delay embedding was used to transform the data into 3 dimensions.

Dynamical Systems Analysis

Takens' theorem (section 3.5) specifies that if the elements within a dynamical system are coupled, the network's dynamical properties can be captured in a phase portrait via the time delay embedding of a single variable. In many cases the portrait can be depicted in 3-dimensions via time delay embedding. To accomplish this, the difference between the two output genes can be taken at each time step (the difference between the two outputs result in the movement of the cart). First, the behaviour of the AGRN and the AEN are compared when the pendulums are swinging, yet to reach the upper equilibrium (Figure 9.10). It is apparent from looking at the phase portraits whilst swinging that the dynamical behaviour is different between the AEN and AGRN. The structure of the AEN phase portrait shows a clear transition between two fixed regions within the attractor space. These two regions relate to the movements of the carts, with each specifying a direction for the cart to move. The transitions between refer to the ‘rocking’ motion used to generate momentum to move the pendulum into the upright position. The phase portrait for the AGRN is highly structured, consisting of a set of repeated structures slowly moving through the z-dimensional plane. The values within the respective phase space is smaller for the ARGN. This suggests that the AEN is more adept at using the full accelerative force of the carts to generate momentum within the pendulum.



(a) The phase portrait of the AGRN in the upper equilibrium. The phase portrait was created using the difference between the outputs of the network, and time delay embedding was used to transform the data into 3 dimensions.



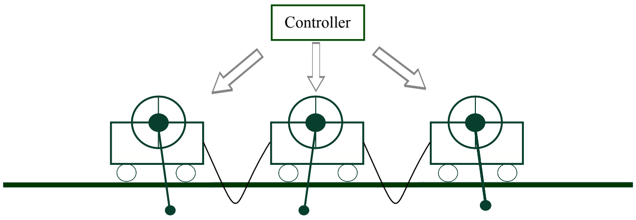
(b) Two illustrations, the top depicting the phase space of the AGRN when cart is balanced in the upper equilibrium, and the lower image showing the phase space of the AEN the cart is balanced in the upper equilibrium position. The phase portrait was created using the difference between the outputs of the network, and time delay embedding was used to transform the data into 3 dimensions

Figure 9.11: The phase spaces of the AEN and the AGRN when the pendulums are being balanced in the upper equilibrium position.

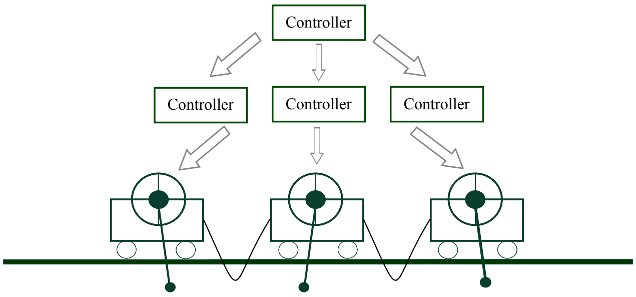
Looking at the phase portraits of the AEN and AGRN when the pendulums are balanced in the upper equilibrium, there is a distinct difference between the two networks (Figure 9.11). The AEN shows a highly ordered structure, with the trajectory moving between two general partitions in the space. The regions correspond to movements of the cart, the trajectory switching between the two depends on the pendulum(s) positions in the upper equilibrium. Conversely, the AGRN has a phase space which is comparatively disordered, with no specific structures or characteristics within the phase space. The difference between the two could result from the epigenetic molecules being able to remove genes from the network dynamically, creating distinct regulatory circuits with specific genes with specific behaviours, being able to control these regulatory circuits via the epigenetic molecules at short intervals. It can be deduced that the epigenetic molecules promote the formation of distinct attractor structures that correlate with the distinct behaviours within the task.

9.4 Reduced Dimensionality Controllers

A key emergent property of the AEN is that the network evolves a dynamic way of switching between behaviours. The points at which the epigenetic molecules switch is an evolved trait. In this instance, there is a key switch between when the pendulums are in the lower equilibrium and when they are in the upper equilibrium. This is a logical point to change because the dynamics needed to maintain the pendulums in the upper equilibrium are dissimilar to the dynamics when the pendulum is in the lower position. This is demonstrated in figures 9.10 and 9.11. Hence, certain genes can become specialised at certain tasks during the evolution of the networks. Of interest in this instance is that evolution of the AENs gives rise to a switch which can be altered manually to control the behaviour of the network, and via proxy the behaviour of the pendulum. Assuming the pendulums have reached the upper equilibrium, the networks can be pushed into the lower equilibrium by deactivating epigenetic molecule 1. Then, at a further point, if epigenetic molecule 1 is reactivated, the pendulums will begin swinging, and the pendulum will then re-enter the upper equilibrium where it will remain balanced. This, in effect, creates a controller with reduced dimensionality, where the behaviour of the networks and the tasks can be controlled by a hard switch, a trait which can be somewhat visualised by looking at the attractors for the AEN in Figures 9.10 and 9.11. In this sense, the epigenetic molecules draw behavioural traits from their biological counterparts, which are able to control large regulatory regions with relatively smaller epigenetic changes.



(a) Experimentation from Chapter 9 using a duplicated controller for each cart



(b) Experimentation shown here, in the appendix where a single controller is used to control all carts

Figure 9.12: A comparison between the experimental setup in the previous chapters, and the experiment shown here

9.5 Further Experimentation

The results have shown that the AEN was successfully able to produce an optimal behaviour for the 5 pendulum problem, where the AGRN was incapable of doing so. However, this line of experimentation was done using a single controller which was mapped onto each cart which can be seen in Figure 9.12a. To test the functional limitations of the AEN, the the experiments were repeated, but a single controller controls all carts within the task. An illustration of this can be seen in Figure 9.12b.

The results, shown in figure 9.13 show that the AEN is capable of producing an optimal behaviour when a single controller is applied to 3 carts, whereas the AGRN does not produce the optimum behaviour throughout experimentation. The difference between the results for the 3 pendulum tasks was significant according to the Wilcoxon rank-sum test ($p = 7.1879$ x 10^{-4}) (The Wilcoxon rank-sum test was used because the data was shown not to be normally distributed according to the Kolmogorov-Smirnov test). For 5 pendulums, neither network could produce the optimal behaviour, and the difference between the AEN and AGRN was not significant ($p = 0.1733$, using the same statistical tests outlined previously), highlighting that there was no performance increase whilst using the AEN.

Figure 9.14 highlights that the AEN is more adept at controlling the three pendulum tasks than the ARGN. For the 5 pendulum task, the performance of the two networks is similar,

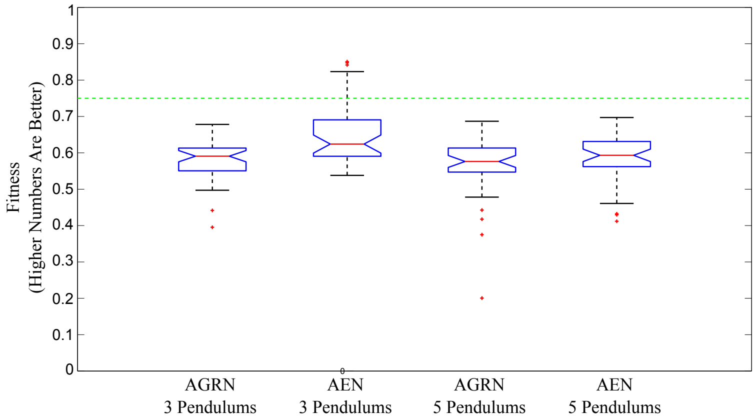
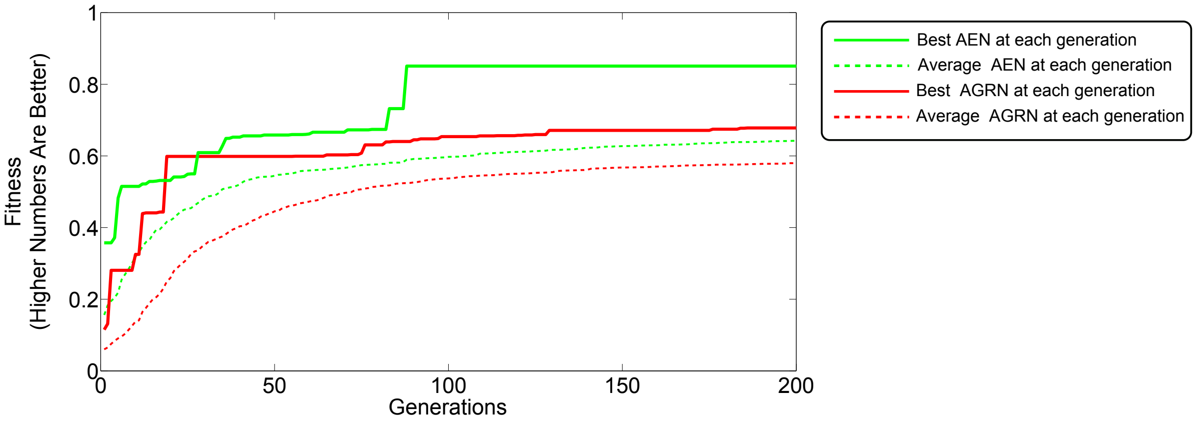
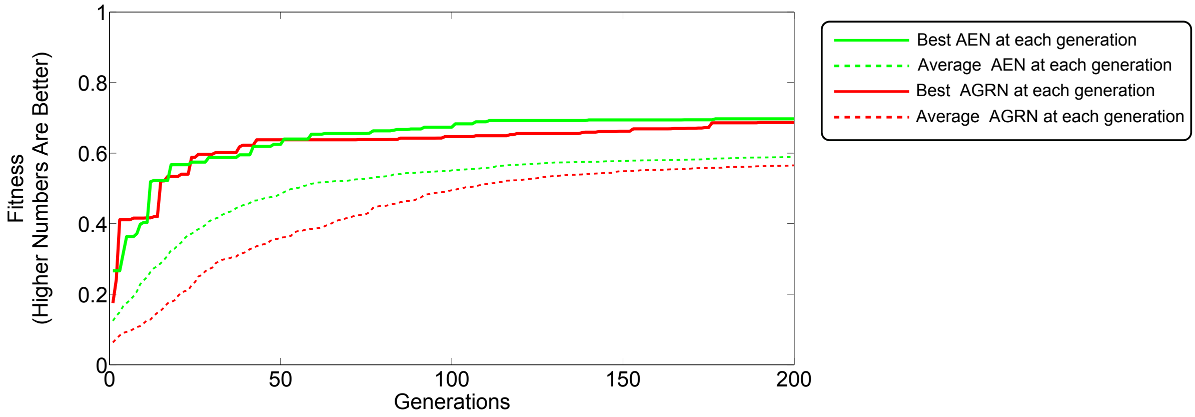


Figure 9.13: Application of the AEN to the coupled inverted pendulums task where a single network controls all carts.



(a) A graph showing the evolution of both the AEN and AGRN over time when applied to the three pendulum task where a single network controls all carts.



(b) A graph showing the evolution of both the AEN and AGRN over time when applied to the five pendulum task where a single network controls all carts.

Figure 9.14: Graphs showing the evolution of the networks at each generation

although it can be seen that the average results for the AEN rise faster than the AGRN but plateau earlier. This suggests that the AEN can evolve faster, but both networks are not able to escape the local optima.

9.6 Summary

In this chapter the artificial epigenetic networks (AEN) are applied to the control of a complex dynamical system, the coupled inverted pendulums task. These AENs are compared against identical networks with epigenetic structures omitted (artificial gene regulatory networks (AGRN)). The results from this comparison show that the AENs outperform the AGRNs when applied to the 1,3 and 5 pendulums task, with the AENs being able to evolve an optimum behaviour for all configurations of pendulums, and the AGRNs only being able to optimally solve the 1 and 3 pendulum tasks. Significantly, the AENs evolve to partition the networks dynamically during execution so that certain genes are active at certain times under the control of artificial epigenetic molecules. This allows certain genes to become specialised towards certain aspects of the task. An emergent property of this behaviour is increased performance when solving the coupled inverted pendulums task. A further emergent property of the AENs is the reduced dimensionality element (section 9.4), which allows a user to simply interface with the AENs dynamics and through the epigenetic molecules, the coupled inverted pendulum dynamics. This facet allows the AEN to be less of a black box controller than the AGRN, providing insight into the functionality and control of the network.

The results of further experimentation whereby a single controller was used to control all the carts within the task further highlights that the AEN is more capable of solving complex tasks than the AGRN. However, when applied to the control of 5 carts with a single controller, it can be seen that both the AEN and AGRN fail to produce an optimum behaviour. This demonstrates that the AEN has limits in terms of its behaviour within the task. However, it is to be noted that the controllers are evolved with set population, crossover and mutation parameters and changing these may produce different results. In terms of highlighting the potential performance increases and varied behaviour of the AEN, the coupled inverted pendulums task has provided a rich environment in which to achieve this.

In the following chapter, the AEN is applied to the control of transfer orbits in gravitational systems. This will help generate a further understanding of the AENs functionality when applied to higher dimensionality tasks.

Chapter 10

Controlling Transfer Orbits In Gravitational Systems

Contents

Controlling orbits within gravitational systems is a control task where a moving body (such as a rocket) is guided to a target whilst having to negotiate the gravitational fields of multiple bodies. The AENs are applied to this multi-objective task and are then analysed to ascertain how the epigenetic molecules affect performance. Their performance is evaluated against AGRNs.

10.1 Description Of The Controlling Transfer Orbits In Gravitational Systems Task

Controlling transfer orbits in gravitational systems is a 3-dimensional control task where a given body (rocket) is required to traverse a path around neighbouring celestial bodies. In addition to this, the rocket has to maximise its efficiency by achieving this whilst expending

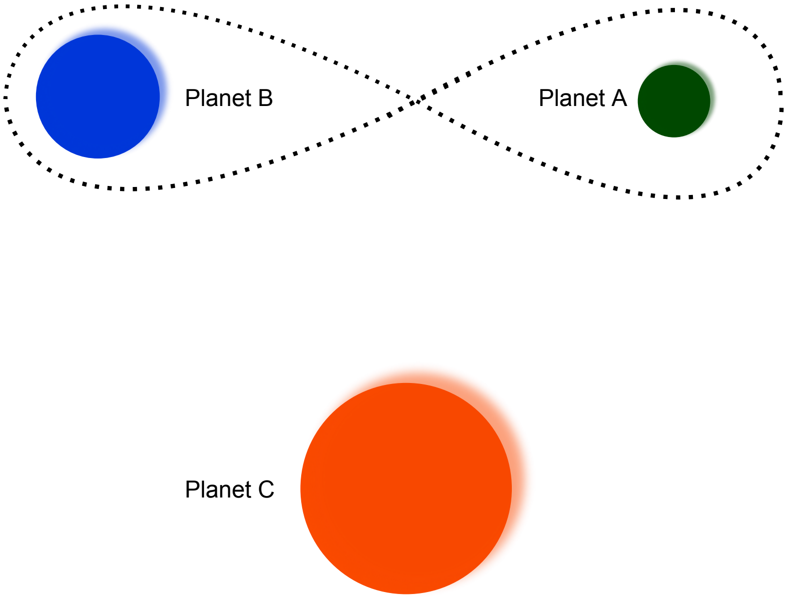


Figure 10.1: A 2-dimensional representation of the navigation of a celestial body task. The dotted line indicates an optimal path between planet A and planet B using a gravitational slingshot around both. Planet C provides a strong gravitational pull which gives the rocket the ability to use a further gravitational slingshot.

as little fuel as possible. In this instance, there are 4 bodies within the task: 3 planets (A,B,C) and the rocket (Figure 10.1). The rocket has to navigate between planets A and B (a planetary hop) as many times as possible within a set amount of time, whilst avoiding a collision with any planet. The number of planetary hops completed will be the primary objective. Planet C exists to create a strong gravitational pull away from planets A and B. This was done for two reasons. First, to increase the difficulty of the task by making it harder for the rocket to take the optimal path between planets. Second, to give the rocket the ability to use a gravitational slingshot to achieve planetary hops whilst using less fuel. The planets are arranged so that the rocket must navigate in 3 dimensions in order to navigate between planets A and B and use the gravitational slingshot around planet C.

During the simulation, all the planets are static and are not subjected to gravitational forces. The rocket, however, is affected by the gravitational pull from the planets. The force exerted on the rocket is calculated using the mathematical formula in equation 10.1, where m is the mass of a body and q is a 3-dimensional vector (j specifies an instance of a body, and k represents an instance which is not equal to the first, i.e., force i is the sum of all other forces (k) which are not force i). From this, the acceleration of the rocket due to the gravitational forces of the other planets can be calculated using Newton's second law of motion. In this instance, equations are integrated using leapfrog integration as it is well suited to the problems of orbital mechanics due to its symplectic nature and that it is time reversible, which is beneficial for analysis (New et al., 1998; Mikkola, 1999).

$$m_j q_j = G \sum_{k \neq j} \frac{m_j m_k (q_k - q_j)}{|q_k - q_j|^3} \tag{10.1}$$

At the beginning of the task the rocket has to navigate from a close orbit of planet A to planet B. This is successful when the rocket is less than 2 x $10^5 m$ from the planet center and at least 1 x $10^5 m$ from the centre. As soon as this objective is achieved the current objective switches the target to the orbit of planet A. If this is achieved the target becomes planet B and this is repeated for the number of time steps within the task. If at any point the rocket comes within 2 x $10^4 m$ of the centre of any planet, that instance of the task returns the worst fitness score. Similarly, if within 8000 time steps, the rocket has not reached planet B, that instance of the task returns the worst fitness score.

10.2 Experimental Design And Parameters

The AEN is applied to the navigation of celestial body tasks, where the controllers are evolved using non-dominated sorting genetic algorithm (NSGA II - Section 4.1.1). There are three objectives that NSGA II will use to optimise the networks. First, the number of times the rocket moves into orbit of its target planet. Second, the amount of fuel used. This is calculated by taking the acceleration values passed to the rocket in each plane and squaring them (to remove negatives). Third, a multiplication of the values from the first two objectives, assuming at least one planetary hop is achieved. The purpose behind the third objective is to prevent rockets achieving a high score from remaining relatively static and conserving fuel (If only the two objectives existed, a rocket which did not move would always produce a perfect sore for a single objective which would negatively effect the evolutionary process). The performance of the AENs on this task will be compared to the AGRNs (AENs without the epigenetic analogue).

At each time step the network is provided with 10 inputs (Table 10.1). These are mapped to the range [0,1]. The network provides three outputs (orthogonal accelerative values) which are mapped onto the rocket within the range [-25,25]. The planets' and rockets' positions and masses are initiated according to the values in Table 10.2. The simulation runs over

Input	Variable	Range
0	Distance To Target (m)	0, 2 x 10^6
1	Target (x position)	-1.5 x 10^6, 1.5 x 10^6
2	Target (y position)	-1.5 x 10^6, 1.5 x 10^6
3	Target (z position)	-1.5 x 10^6, 1.5 x 10^6
4	Rocket Acceleration (x plane)	-50, 50
5	Rocket Acceleration (y plane)	-50, 50
6	Rocket Acceleration (z plane)	-50, 50
7	Rocket Position (x plane)	-1.5 x 10^6, 1.5 x 10^6
8	Rocket Position (y plane)	-1.5 x 10^6, 1.5 x 10^6
9	Rocket Position (z plane)	-1.5 x 10^6, 1.5 x 10^6

Table 10.1: The input values which are given to the networks. Each of these values is mapped onto the network within the range [0,1].

Body	Position (x,y,z)(m)	Mass (kg)
Planet A	1 x 10^6, 0, -1 x 10^5	5.972 x 10^{22}
Planet B	1 x 10^5, 0, -1 x 10^6	5.972 x 10^{23}
Planet C	0, -2 x 10^6, 0	5.972 x 10^{24}
Rocket	0, 0, 0	2000

Table 10.2: The positions and masses for the bodies within the task.

50,000 time steps with an integration step of 0.02 (Table 10.3). The networks are evolved with between 12 and 25 genes, and the AEN has between 3 and 5 epigenetic molecules. The population is 500 and NSGA IIl runs over 200 generations resulting in 100,000 evaluations per run. The crossover rate is 0.5 and the mutation rate is 0.05. A total of 40 runs each are conducted for the the AENs and the AGRNs.

10.3 Results

The results showing the number of planetary orbits reached against the amount of fuel used can be seen in Figure 10.2. Both the AGRN and AEN were able to navigate between the

Parameter	Variable
Gravitational Constant	6.67384 x 10^{-11} N $(m/kg)^2$
Time Steps	50000
Integration step	0.02
Rocket Max. Acceleration	± 25 m/s^2

Table 10.3: The parameter values which are used within the task.

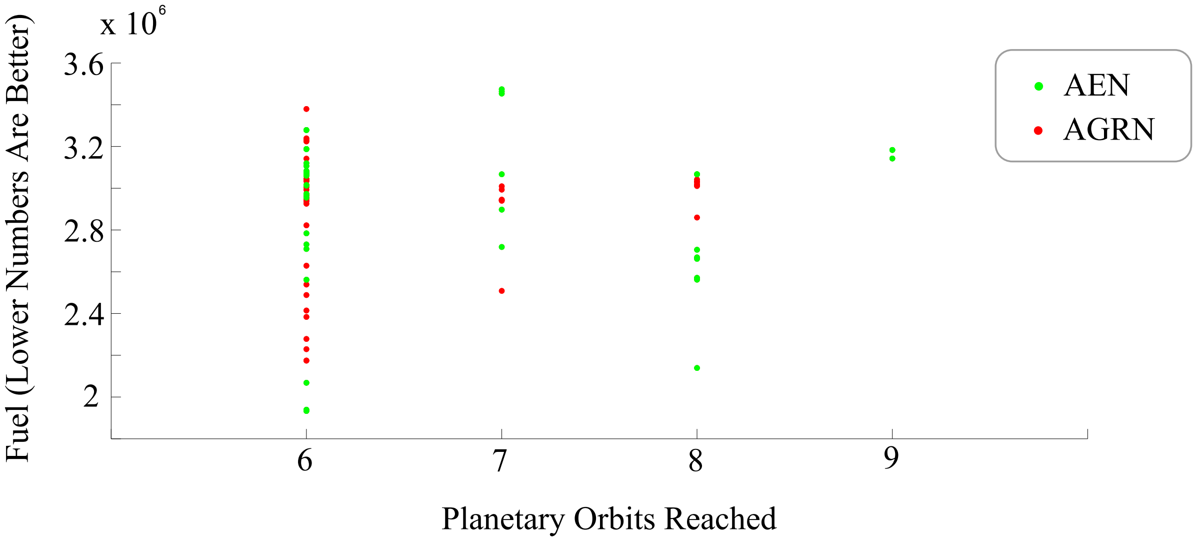


Figure 10.2: The best number of planetary hops achieved, alongside the amount of fuel used for each run.

two planets at least 6 times. The best instance of the AEN was able to achieve 9 planetary hops (Figure 10.3) whereas the best instance of the AGRN was only able to achieve 8 hops. This orbital behaviour can be seen in Figure 10.3. A pervasive trait over all networks, which is shown in Figure 10.3, is the extensive use of planet C to produce a gravitational slingshot. This creates planetary hops which are longer then the optimal path, but not necessarily slower (due to the gravitational slingshot changing the speed of the rocket) and which often use less fuel. All networks produce an unstructured behaviour, in that they do not follow a repeating pattern of planetary hops, combined with gravitational slingshots. They appear to use multiple gravitational slingshots between certain hops, and none between others. The results also show that both networks spend a considerable amount of time using the gravitational slingshot around planet C which does not necessarily translate into increased planetary hopping. The behaviour of gravitational systems with multiple bodies have varying regimes of dynamics ranging from highly ordered to chaotic (Contopoulos & Voglis, 1997). The behaviour seen here could be a reflection of the underlying non linear dynamics of the system.

The results shown in Figure 10.2 are two dimensional because the objective fitness function contained two key objectives (planetary hops and fuel used). The third objective was an aggregate of the first two objective and is not shown here. These results are transformed into data specifying the fuel per planetary hop (Figure 10.4). These results show that the distributions between the AGRN and the AEN are very similar, and the medians are not significantly different. However, it can be seen that the AEN produces controllers which use less fuel per planetary hop than the AGRN, and that the worst results of the AEN are better than the worst results of the AGRN. Furthermore, if the amount of planetary hops and the fuel used are treated as independent statistical tests, their distribution is not statistically different which indicates that the average performance of the AGRNs and AENs is very similar.

10.3.1 Network Analysis

Upon the creation of the minimum working examples (MWE) (Section 8.3.3) of the networks, a key trend that was discovered is that the AGRNs used more inputs in their regulatory circuits, whereas the AENs took fewer inputs but used more regulatory genes. The AEN's epigenetic elements functioned dynamically in 90% of the runs (36 runs). The remaining 4 runs, the epigenetic molecules prevented certain genes from executing over every time step, and this was their only function.

Static Network Analysis

The analysis of the static structures of the networks highlighted that all networks contain 3 separate, commonly independently functioning regulatory circuits. It is apparent when looking at the genes that were omitted from the networks to create the MWEs (Figures 10.5 and 10.6) that the AGRNs generally used more input genes than those of the AEN. This causes the regulatory circuits of the AGRN to be generally larger than those of the AEN. The structure of the networks varied little in terms of their form; however, one evolved AGRN has a particular trait which is highlighted in Figure 10.5. This is that an input is directly mapped onto an output of the network. In this case, gene 9 (rocket position on the z plane) is mapped to gene 18 (rocket's z dimensional thruster). Although it was only seen once, it was an interesting characteristic for the network to adopt. A possible explanation of this would be that the input from gene 9 could provide stability if used as an output, whereby if gene 9 specifies the rocket is close to the bounds of the model on the z plane, accelerating on the x plane may prevent the rocket from drifting too far away from the planets. Aside from that, the reference space showing this particular network (Figure 10.5) is typical of the evolved AGRNs.

The reference space of the AEN in Figure 10.6 shows that a single epigenetic molecule interacts with a single regulatory circuit containing two genes. In 36 of the 40 runs, the epigenetic molecules were dynamically modifying gene expression values over time. Of these epigenetic

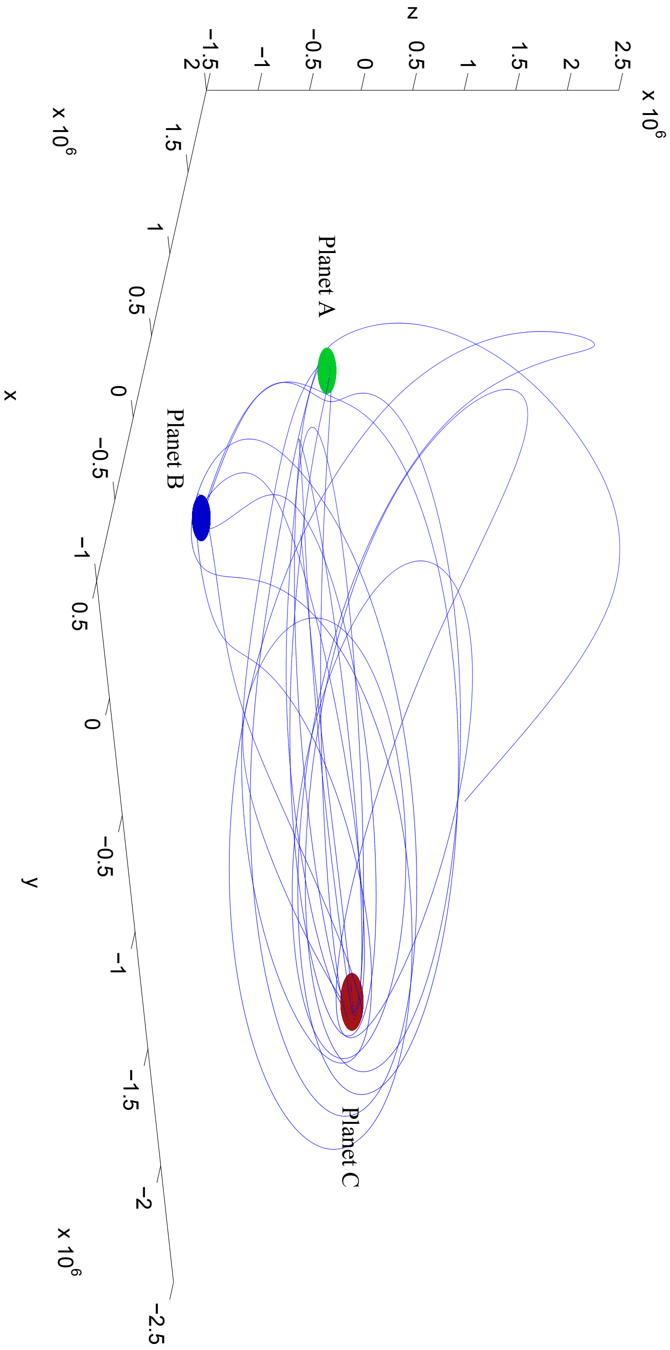


Figure 10.3: An illustration of the orbit of the AEN controller which achieves 9 planetary hops (the highest number achieved throughout experimentation). It can be seen that the controller utilises the gravitational slingshot effect, a key theme throughout all networks. The orbital path is relatively unstructured, which could possibly be caused by chaotic dynamics within the model.

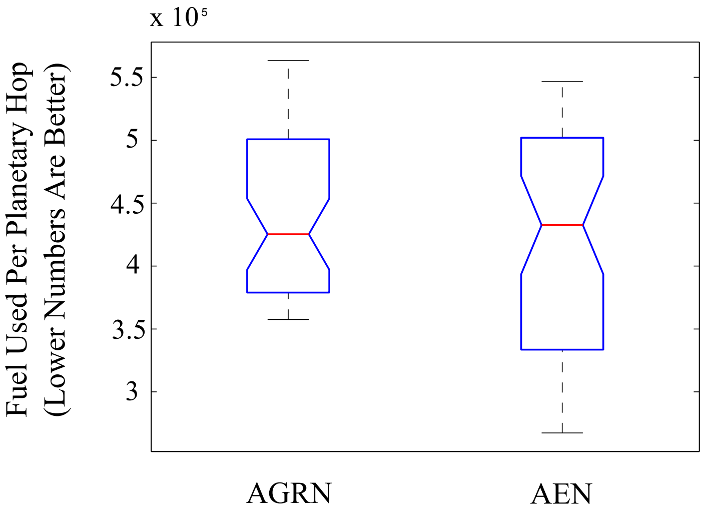


Figure 10.4: The average amount of fuel used per planetary hop. The difference between the networks performance is not statistically significant (using the MannWhitneyWilcoxon test as the data did not fit a normal distribution) (p = 0.6985).

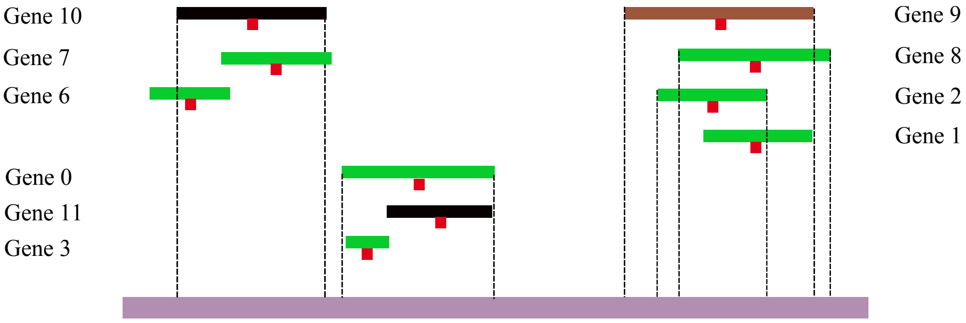


Figure 10.5: The reference space for an evolved AGRN controller. The green genes are the inputs (10.1) and the black genes are the outputs. The brown gene is an input which directly maps onto an output. Hence, one of the rocket thrusters is controlled directly by an input to the network. This network was able to complete 7 planetary hops.

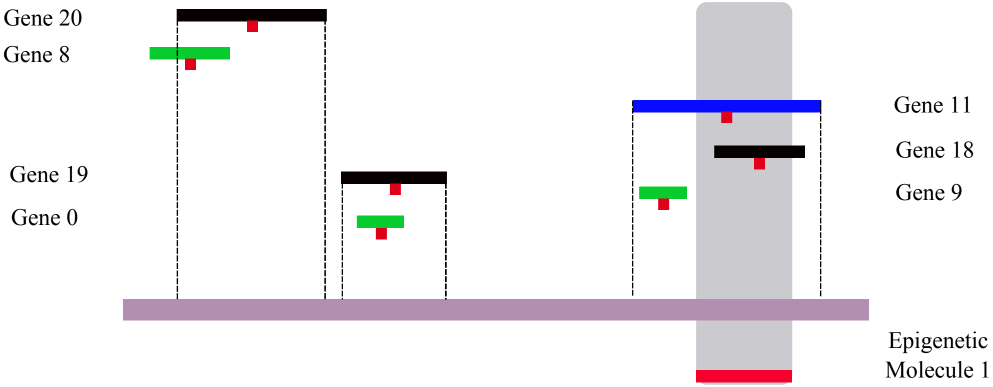


Figure 10.6: The reference space for an evolved AEN which was able to achieve 9 planetary hops. This was the best evolved controller.

modifications, all modified the expression of a particular output gene. This provided the AENs with a type of computational processing not available to the AGRNs, which could explain why the AGRNs generally had smaller regulatory circuits. The AENs contained 3 epigenetic molecules, and in 76% of the evolved controllers only one molecule was active. 14% had 2 molecules dynamically active, and the remaining 10% did not utilise the dynamical behaviour of the epigenetic molecules. This demonstrates that although epigenetic molecules on the whole were utilised by the AENs, the majority of molecules were inactive throughout execution.

Dynamical Network Analysis

In order to ascertain how the networks function over time and how much influence the epigenetic molecules had over network functionality the networks inputs and expression values are plotted over time. Because the task functioned over 50000 time steps, the amount of data produced is difficult to analyse. To reduce the amount of data, the variables of each gene and epigenetic molecule were sampled at every 10 steps. Hence, the data used will only contain 5000 time steps.

In Figure 10.7, the gene expression values during one run of the evolved AGRN from Figure 10.5 are shown. It is apparent from observing this plot that over the majority of time steps, output gene 10 is providing a constant value. This means that the controller specifies that the rocket accelerates at full thrust for the majority of the simulation. Gene 18, although directly taking the input from gene 9, processes that value which results in minor dynamical changes; however, the underlying trend is very similar to that of gene 9. Output gene 11 is directly connected to gene 0, which is in turn connected to gene 3. From the visualisation it can be seen that gene 11 is functioning as an inverter for the values produced from gene 0. Gene 0 provides the distance to the target. Loosely translating the above statements into a logical structure, it would appear that, when the rocket is close to the target, it up-regulates output gene 11's expression increasing its acceleration. The overall trend that can be deduced from looking at the data within the graph is that the rocket changes its acceleration patterns gradually over time without many large peaks or troughs. Although outputs genes 9 and 11 are altering thrust dynamically to the engines, gene 10 is providing almost constant thrust.

The gene expression values from the AEN (Figure 10.8) show a different trend to that of the AGRN. It can be seen that the epigenetic molecule is controlling the expression of genes 11 and 18 throughout execution. There are 3 key points when this happens, which correspond with high values from gene 9 (gene 9 species the rocket position on the z plane, and through visualising the behaviour, the epigenetic molecule becomes active when the rocket is reaching the bound of the simulation on the z plane). Output gene 20 takes the input gene 8 and processes it; however, it is almost a direct mapping. Output gene 19 provides something close to a constant output; nonetheless, it can be seen that there are very small undulations throughout the task. These undulations are produced when gene 0 has a low expression value. The undulations are small; however by conducting exploratory experimentation and replacing this gene 19 with a gene which produces a constant output within the range of gene 19, was found to cause a significant loss of functionality. This suggests that gene 19 has a specific function integral to the functionality of the network. Output gene 18 can be seen to produce a constant value, except when the epigenetic molecule becomes active and prevents its execution.

Dynamical Systems Analysis

To better understand the overall behaviour of the networks, phase portraits are created using the x,y and z values from the outputs of the network. Similar to the dynamical network analysis, the gene outputs are taken at every 10$^{\text{th}}$ time step to reduce the volume of data to display. One of the most distinct trends, which can be seen in figures 10.9 and 10.10, which existed over all the networks analysed is that the phase portrait is generally 2-dimensional. This ties in with the dynamical network analysis, where one output was producing a near constant value in both the AEN and AGRN. This behaviour functionally flattens the 3-dimensional task into a more 2 dimensional task. Outlined in the description of the task, the planetary positions were devised to make the task 3 dimensional, where in order to move between the

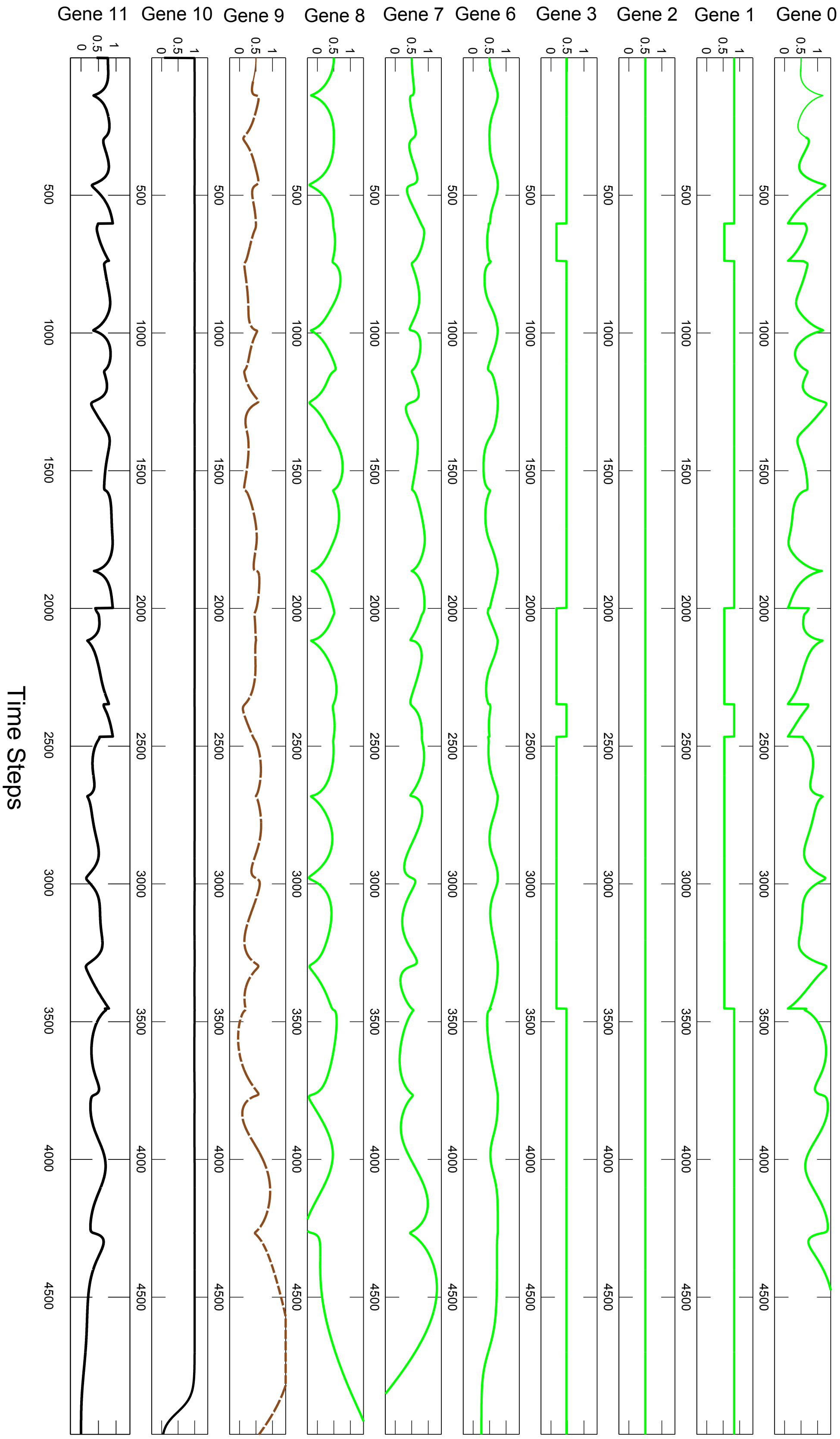


Figure 10.7: The gene expression values of the AGRN from Figure 10.5. Gene 9 is both an input and an output of the network. The inputs of the network are coloured green, and the outputs are black. The brown gene is both an input and an output.

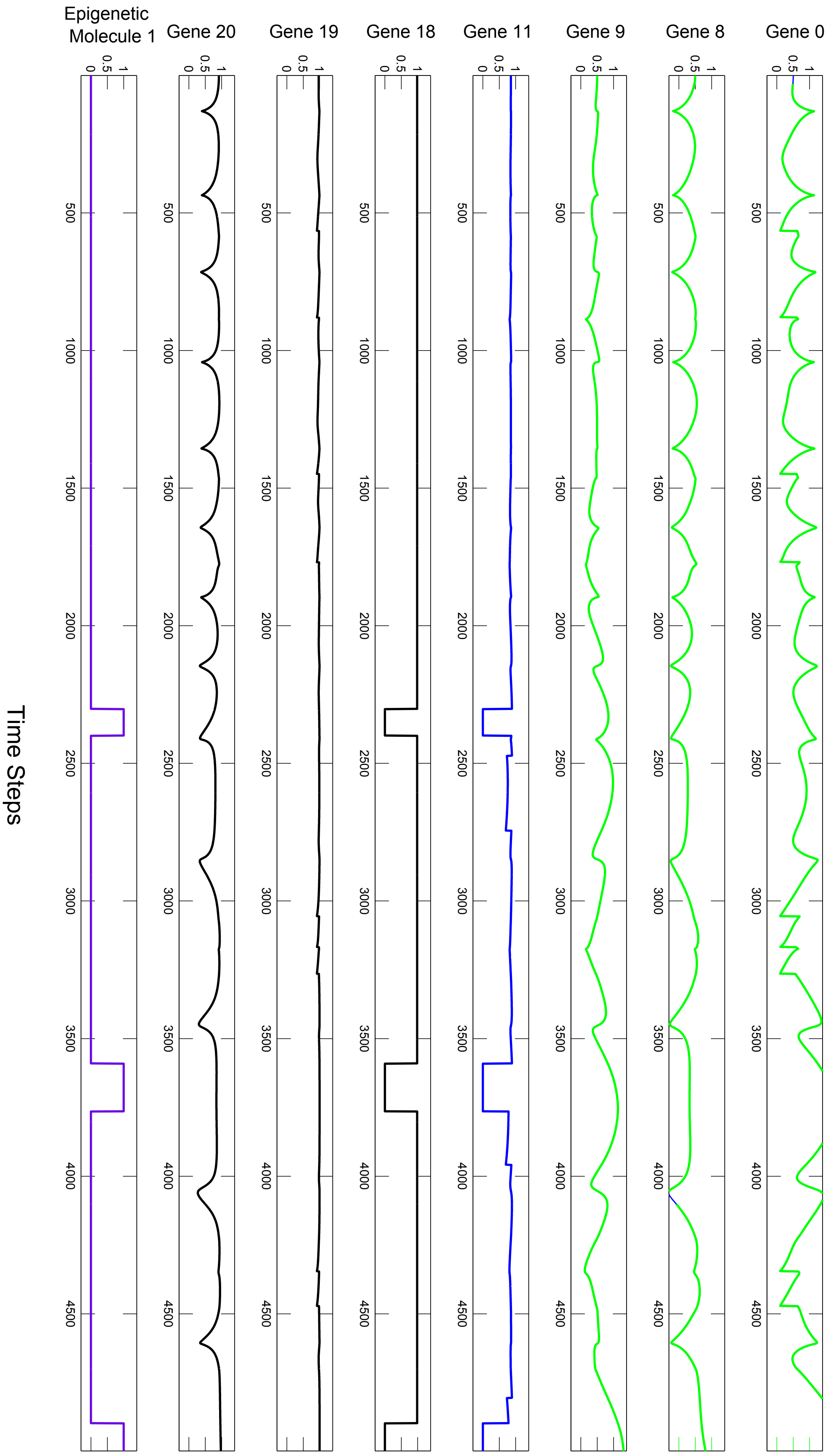


Figure 10.8: The gene expression values of the genes and epigenetic molecule of the AEN shown in Figure 10.6. The green genes are input genes, the blue genes are regulatory genes, the black genes are output genes and the purple output is that of the epigenetic molecule.

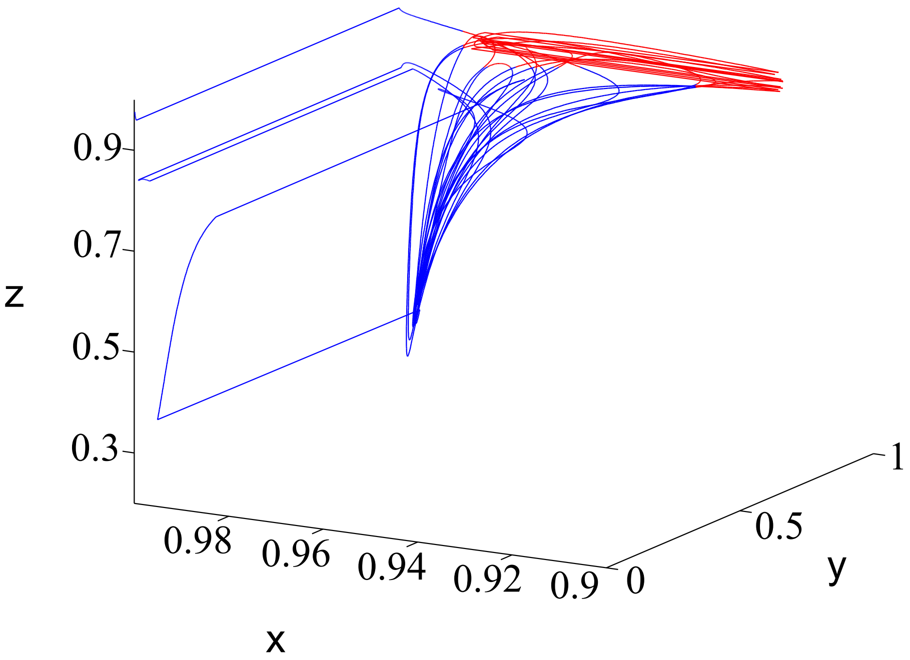


Figure 10.9: Phase portrait of the AEN from Figures 10.6 and 10.8. The behaviour shown is typical of that for the evolved AENs. The red region denotes the orbits around planet C which do not reach planet A or B.

three planets requires propulsion on all three planes. Reviewing the expression data from the outputs of all evolved controllers, commonly genes which produce a near constant output are in fact producing dynamic behaviour, but the range of the expression values is too small to visualise and changes between the genes' expression occur very infrequently. Hence, it would seem that the evolved controllers utilise all 3 dimensions, yet commonly only produce large variable changes on certain outputs.

The phase portrait of the AEN in Figures 10.6 and 10.8 can be seen in Figure 10.9. It is apparent that the phase portrait exists predominately in 2 dimensions. The one region which is most densely visited is that at the top of the graph (z ≈ 1) which occurs when the rocket is orbiting planet C (the red section of Figure 10.9). The other region of the phase space (z ≈ 0.4) occurs when the rocket is using the gravitational slingshot around planet C to reach either planet A or B. The sharp spikes in the graph occur when the epigenetic molecules become active.

The phase portrait of the AGRN from figures 10.5 and 10.7 is shown in Figure 10.10. The portrait shows a similar flattening effect, where the majority of the portrait exists in 2-dimensions. The phase portrait is larger than that of the AEN, emphasising that the AGRN in this instance generates a greater range of gene expressions and thus, accelerative values. The region of the portrait where x ≈ 1 and z ≈ 0 marks the region where the gravitation

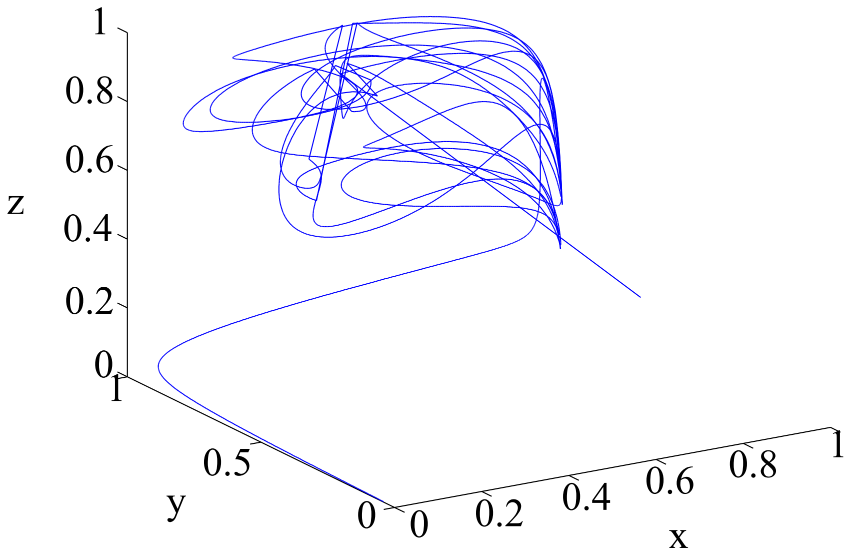


Figure 10.10: Phase portrait of the AGRN from figures 10.5 and 10.7. This shows a typical evolved behaviour.

slingshot occurs around planet C. The other region of the portrait where x ≈ 0.4 marks where the rocket traverses between planets A and B.

10.4 Summary

In this section, both the AEN and the AGRN were applied to the task of controlling a rocket whilst optimising the amount of fuel used when traversing gravitational orbits. The AEN showed certain performance benefits, finding better solutions compared to that of the AGRN. In addition, of the solutions which achieved 8 planetary hops (the maximum of the AGRN) the AEN used less fuel. Of these solutions, all instances of the AEN which achieved 8 or 9 planetary hops used their epigenetic analogue to dynamically modify gene expression, suggesting it had a key part in the functionality of the best solutions. However, the overall frequency in which the epigenetic molecules were dynamically active was lower than that of previous tasks. This emphasises that if epigenetic molecules may not be fully utilised in certain situations, they do not inhibit the evolution of the network as a whole. This is important, because it was never the intention to force functionality on the network, but to let it emerge as naturally as possible.

The application of the networks to the control of transfer orbits in gravitational systems is essential to provide a wrapper of the themes and tasks within this thesis. This is because primarily, the task is based on real world physics, is theoretically capable of exhibiting chaotic dynamics, and is a complex dynamical system. This is a very positive characteristic, as it amalgamates the behaviours of the previous tasks within a single model and evaluates the networks in terms of this. However, unlike previous tasks, the task of controlling transfer orbits was very computationally expensive. This was a limiting factor and contributed to the somewhat restricted amount of exploratory experimentation that was possible. Moreover, although the simulation was based on real world dynamics the planetary positions, sizes and masses are set to optimise computational efficiency, whilst maintaining the complex behaviours within the task. Optimally, the model would be based upon real world orbits to ascertain the functionality of the network controllers in a relatively more applicable domain.

On balance, within the limitations of the computational complexity, the task contained a fair representation of orbital dynamics, and the conclusions drawn on top of the previous chapters results, describe a positive set of behaviours regards to the AENs ability to control potentially complex chaotic dynamics.

Chapter 11

Summary And Conclusions

Contents

This chapter provides a summary of the work reported in this thesis and the rationale behind it. The conclusions of the work are presented as well as the contributions. Thereafter, the experimental limitations are described and potential avenues for further work are discussed.

11.1 Work Conducted and Rationale

Biological sciences are consistently increasing the understanding of the natural world, whether it be by discovering new biological processes, being able to control or modify organisms or redefining older work in the light of new evidence. The field is constantly updating and evolving. It has been understood that biological systems have been faced with, and overcome, many of the problems faced in computer science and engineering, such as robustness, autonomy, fault tolerance and optimisation. It is this principal that biological organisms and systems hold many solutions to issues of computational interest which inspired the work within this thesis.

There are a wealth of systems which have biologically inspired roots such as neural networks, evolutionary algorithms and artificial immune systems. However since the inception of such models, they often remain bound by their initial biological underpinnings, regardless of advances in the biological theory which inspired them. Moreover, these models generally draw on a limited view of their biological counterparts. However, it is clear that these biologically inspired models work. They capture useful properties that exist in such abundance in the natural world and make them available within a computational system. Therefore, the philosophies behind these models are at least in part logically sound. However, modelling a biological system in perfect detail is not possible, and frequently any level of detail close to that is infeasible. There is a balance to achieve between biological realism and functionality. This thesis stands on the idea that there is an opportunity to shift this balance, creating more biologically faithful models, which in turn capture a wider range of sought after biological traits.

Specifically, this thesis focuses on the idea of artificial biochemical networks, and how, for the most part, they are underpinned by homogeneous interpretations. That is, a network is the product of multiple interacting units, of which all units share the same structure. It has been widely accepted even since the inception of these models that biological neural networks, gene regulatory networks, metabolic networks, cell signalling networks and immune systems are not homogeneous. They consist of many different interacting sub units. In this thesis, the argument is presented that there are instances where artificial biochemical networks could benefit from an increase of biological faithfulness via the introduction of heterogeneous elements which are inspired by epigenetic mechanisms.

Artificial gene regulatory networks are a particular kind of artificial biochemical networks which draw inspiration from the functionality of gene regulatory circuits in nature. Genes are one of the most pervasive biological structures, a functional unit of hereditary information which generally specifies the primary structure of a protein. However, since the 1940's there has been growing evidence that gene regulatory networks are not comprised of genes alone, but a mixture of genetic and epigenetic structures. Research specified that pervasive epigenetic structures such as chromatin provide an additional control layer which exists on a different level of organisation to genes alone. This is interesting in a biological sense because the epigenetic structures can govern which genes are active at specific times meaning that specific genes can be applied to particular tasks. At present computational models of gene regulation designed for computation do not contain epigenetic analogues. [1]

The objective of this thesis was to create a dynamically functioning epigenetic analogue and to experiment with this to ascertain if any of the biological functionality of epigenetics could be incorporated in an artificial gene regulatory network. Structurally, it can be seen that epigenetic and genetic structures, for the most part, are separate systems. Logically, it can be deduced that complex genetic structures evolved before epigenetic because epigenetics serves no function without genetics. Moreover, there is an abundance of bacteria and single celled organisms which do not contain complex epigenetic elements such as chromatin. Using this evolutionary lineage, and not wanting to reinvent the wheel *per se*, it was decided that the epigenetic analogue should be a structure which co-exists and cooperates with a pre-existing artificial gene regulatory network. Generally, epigenetic structures such as chromatin have the ability to prevent gene expression via physically inhibiting cellular processes which facilitate gene regulation.

The combination of the artificial gene regulatory network and the epigenetic analogue is referred to as the artificial epigenetic network. In keeping with biological faithfulness the epigenetic analogue was designed to dynamically prevent the execution of certain genes within a gene regulatory network according to internal and external environmental cues. The epigenetic elements are similar to genes, in that they take inputs, process them and produce an output. The inputs for the epigenetic molecule are in the form of regulatory values from the artificial gene regulatory network or from the environment. The epigenetic molecule then processes these values, and depending on the output can either remain inactive, or prevent the genes it used as connections from executing. This is achieved dynamically throughout execution of the network. This provides the artificial network with the ability to designate genes towards specific tasks. In order to understand the functionality of the artificial epigenetic network, it was applied to a range of control tasks, which were chosen to encourage development of a range of dynamical properties within the network.

11.2 Conclusions

The objective of this research was to ascertain if the introduction of and epigenetic control layer to an artificial gene regulatory network would improve functionality. The artificial

[1] However, elements of the work described within this thesis which has been previously published (Turner et al., 2013*b*) have been acknowledged as inspiration for new epigenetically inspired networks (Bull, 2013)

epigenetic network developed throughout this thesis has been applied to a range of control tasks. The performance of the networks and the analysis of their structure and dynamical properties have resulted in the following conclusions about their functionality

Improved Computational Performance

Throughout experimentation the artificial epigenetic network consistently outperformed its counterpart, the artificial gene regulatory network. Where the performance increase was not significantly better, the results were not significantly hampered by the presence of the epigenetic analogue. This performance increase was a product of the epigenetic analogue's effect on the dynamics of the network.

Reduced Dimensionality Controllers

One of the most interesting emergent functionalities is that of reduced dimensionality controllers. This is most apparent when the networks were applied to the control of Chirikov's standard map and coupled inverted pendulums (chapters 8 and 9). In the majority of instances, the artificial epigenetic network partitioned certain genes to be active at a given time within the task. It was apparent that these partitions each produced a useful behaviour, which could then be controlled externally by modifications to the epigenetic switching pattern. Hence, manual modification of the epigenetic switching can alter the dynamics of the network, and in turn the system it is controlling in useful ways. An illustration of this behaviour can be seen in Figure 11.1.

Application Specificity Of Genes

During execution, the networks were able to dynamically alter gene expression via the modification of their epigenetic molecules. This allowed the networks to apply certain genes to certain tasks depending on the dynamics within the network, allowing the genes to become optimised for a specific function within a task. This is a fundamental characteristic of chromatin modification which the epigenetic structure of the artificial epigenetic molecule was based upon.

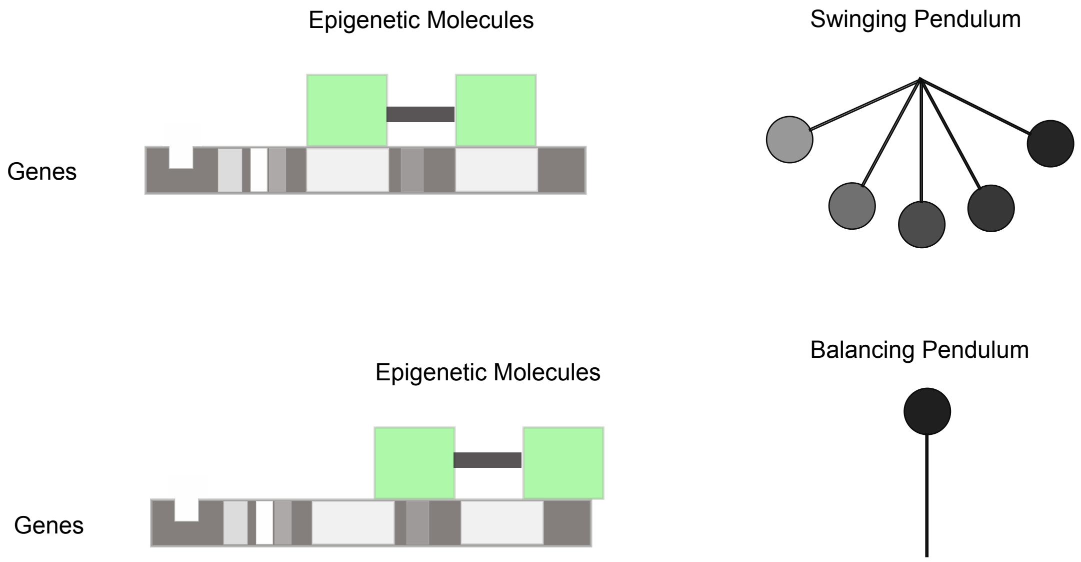


Figure 11.1: An illustration of how the reduced dimensionality controller within the artificial epigenetic network works. Manually modifying the position of the epigenetic molecules changes the network dynamics, which in turn changes the state of the pendulums. Once the pendulum has built up momentum, the balancing functionality can be controlled as a simple binary switch.

Temporal Functionality

An emergent property of the networks was an ability to develop temporal functionality, where the outputs of the network would not be continuous, but they would alternate between two values, and the period of alteration was rigidly tied to the performance and dynamics of the network. More importantly, this behaviour was frequently beneficial in terms of network performance.

Evolutionary Selection Of Epigenetic Behaviours

A key trait that was noticed is that not all instances of the artificial epigenetic network used their epigenetic analogue. The epigenetic structures could be out-bred from the networks. This is important, as it means that the structures did not force functionality onto the networks, they were only present when they were found to be useful.

In addition, the epigenetic structures can permanently prevent certain genes from executing throughout execution. This allows the artificial epigenetic network to mask out interference from irrelevant parts of the network. This provides a means for evolution to explore a large network space whilst limiting the likelihood of interference within larger genetic networks.

The epigenetic structures also provide a method of augmenting network topologies both as a static structure, permanently preventing the execution of certain genes, and as a dynamic structure, preventing the execution of genes dynamically.

11.3 Hypothesis Revisited

The hypothesis of this thesis stated that :

> An artificial epigenetic analogue can be added to a pre-existing artificial gene regulatory network, capturing certain beneficial properties of epigenetic structures *in silico*, and in turn improving functionality.

From the work conducted throughout this thesis, firstly, it is apparent that we have created a versatile, dynamic epigenetic analogue which cooperates with a pre-existing artificial gene regulatory network (the artificial epigenetic network). Secondly, the artificial epigenetic network was able to capture useful aspects of biological epigenetics *in silico*, which in turn improved functionality and computational performance. Taking the conclusions into account, it is fair to say that the hypothesis outlined can be accepted.

11.4 Contributions

Given that the hypothesis has been accepted, and that the artificial epigenetic network captures benefits of epigenetic structures in nature, providing benefits in functionality outlined in the conclusions, this thesis has contributed the following :

- An artificial epigenetic network has been proposed, which frequently improves computational performance when compared to the artificial gene regulatory network alone. In addition, the artificial epigenetic network is a general model which can easily be applied to a range of tasks with no reprogramming of the underlying algorithms required. This allows the model to be used by a wide range of people to further utilise and evaluate its functionality.

- The understanding that epigenetic structures can be modelled *in silico*, and that the benefits of epigenetics in nature can be captured within a computational model.

- The ability of the artificial epigenetic network to automatically partition the network into useful, functional circuits which can be simply controlled by an external user (a reduced dimensionality controller) and in turn, provide information about the underlying task.

- Support for the idea that improving biological realism can improve the behaviour and performance of computational models

11.5 Discussion

The work within this thesis is centred on being a proof of concept. Because of this, the focus was not to conduct exhaustive testing but rather to emphasise the potential of an epigenetically inspired computational network. This also required that the research was not particularly explorative of the surrounding implementations, representations and parameters. The epigenetic molecules within the AEN have been reasonably static throughout development. This is specifically case when it comes to the internal functionality of the epigenetic molecules. It was of key importance to derive a model which captured the intrinsic nature of epigenetic molecules (i.e. the inactivation of genetic circuits) and once this was captured, the drive was to best understand this functionality, rather than to fine tune the representation. This has the benefit of being able to explore the representation in an in depth manner, yet the drawback of only analysing a specific subset of all possible epigenetic molecules.

The core functionality of the epigenetic molecule is based upon the sigmoid function. Work in (Lones et al., 2010) highlighted that there are many different genetic regulatory mappings, each of which has varying properties. It would be beneficial to understand how these regulatory mappings effect the behaviour of the epigenetic molecules when used as their regulatory function. Moreover, within the current implementation of the epigenetic molecule, the threshold of activation was 0.5. This could potentially limit the evolvability of the individual epigenetic molecule, and more significantly, the evolvability of the networks as a whole. The ideal solution to this would be to have a function which chose its biological mapping, along with the threshold it used for activation.

The numbers of epigenetic molecules (between 3 and 5) were kept static throughout experimentation as they were found to work well, covering a reasonable amount of the reference space in which to derive corrections but not enforce their functionality. However, there was little testing was done with other combinations of molecules, and this would be an important experiment to ascertain the optimum combination of genes and epigenetic molecules.

The parametrisation of the genetic algorithm used to evolve the networks was a product of trying to balance two separate and conflicting objectives. The first being outright optimisation, the ability to evolve towards the most optimum point in the least amount of time. This frequently involves a larger population sizes at the expense of computational time. The second was the ability to keep the parameters the same over each experiment in order to best draw accurate conclusions about the model. Hence, the parameters were chosen from a combination of small amounts of exploratory testing and the trade off of computational time. In terms of the genetic recombination operators, the crossover and mutation rates were held at a static value of 0.5 and 0.05 respectively. This was found to be a good balance through previous experimentation. However, one key characteristic which had to be chosen was that of the crossover operation. A full crossover operator was used which gave each gene and epigenetic molecule a set probability of being crossed over which potentially allowed the entirety of the network to be crossed over. This is an aggressive operator, but previous testing highlighted that this method was generally very good at escaping local optima. In addition, when combined with high population sizes and the elitism of NSGA-II, some of the potential problems of an aggressive genetic operator were mitigated. This also allowed the transfer of entire regulatory circuits throughout evolution in an aim to allow the emergence of genetic redundancy *in silico*. If more computing power was available, it would be highly beneficial to do a parameter sweep within the genetic algorithm to empirically understand what is the optimal environment in which to evolve the networks.

The choices of tasks used in which to test the AEN were derived from two essential criteria. Firstly, the tasks had to be complicated enough to be able to justify the computational properties of the networks. Secondly, the tasks had to contain multiple *sub*-objectives to give the AEN the best opportunity to develop emergent characteristics. This is to create the best environment is which to gauge the validity of the hypothesis that epigenetic characteristics may be captured *in silico*.

The experimentation with Chirikov's standard map was scalable to an extent in terms of difficulty however, the task could only really be scaled within very limited confides such as increasing objectives or modifying path trajectory. In each of these cases, such scales are somewhat arbitrary and difficult to quantify. The coupled inverted pendulums task was very scalable in terms of difficulty. This has been highlighted by showing the limits of the controllers when it comes to having a single controller controlling multiple pendulums simultaneously, where the AENs could achieve optimum behaviours with 3 pendulums, but not 5. The AGRN's were incapable of producing the optimum behaviour for both 3 and 5 pendulums. Additionally, there were many other facets which could scale both the task and its difficulty. For example, multiple pendulums could be mounted to multiple carts, as well as having a non linear track in which to operate. This addition could also introduce multi-dimensional control of the pendulums. More subtle modifiable factors which effect difficulty are the length and weight of the pendulum, gravitational constant and the limits (such as force and momentum) placed upon the carts during the simulation.

The control of transfer orbits task was the most scalable of all the tasks. In its current form, only 4 gravitational bodies are used, however, there is an almost unlimited amount of bodies that can be introduced to the tasks, and well as introducing more objectives, such as landing the rocket on a certain body. This scaling also translates into the scalability difficulty within the task. Within these two scales, the task can be made as big, or as complicated as required within certain theoretical bounds. The biggest problem with scalability in this context is the increase in computational complexity as additional bodies are introduced. For each body in the simulation, its acceleration is determined by the forces of every other body acting upon it. This acceleration directly effects its position which effects every other body in the simulation. The solution in this instance was to have static bodies which only exert force, but do not receive it. With greater computing resources more realistic and explorative experimentation could have been conducted.

With all 3 tasks, there is the dilemma of quantifying their complexity, and in turn, justifying them as suitable benchmarks. Previous work (Lones et al., 2010; Fuente et al., 2013) has shown the standard map to be an effective benchmark but in these instances, only a small subsection of computational structures have been applied. This is why the coupled inverted pendulums task played an essential role in this thesis, as it was originally designed as a computational benchmark. It has therefore been used as a test for a wide range of computational structures, and because of this, the AENs can be justified in their computational merits, as well as gaining some comparison to other models. The controlling transfer orbits in gravitational systems task was bespoke, but based upon previous work which has shown some level of complexity (Mikkola, 1999). A further benefit of this was that, unlike Chirikovs standard map, it is possible to see the task being solved in a real world frame. For example, getting into the orbit of a planet is more useful as an analytical observation than traversing steps within a standard map. On balance, the complexity of the tasks is varied, but as a collective, it is fair to say that they exhibit a level of complexity which is suitable to draw conclusions about the AEN's computational characteristics. It is however true that with more time, a significantly wider range of tasks could be used in which to better understand and analyse the networks emergent properties.

The philosophy behind the networks and their resulting behaviour begs a very important question. What are the limits of the AENs? The epigenetic molecules allow the partitioning of the networks therefore theoretically it is possible for the network to complete as many objectives simultaneously as there are partitions (Assuming an AGRN is capable of individually completing all the tasks individually). So why was this not the case? There are two key reasons which potentially contribute to this. The first being the lack of computational processing power in which to evolve the networks. It is likely that AENs which perform many tasks well would require exponentially more time to evolve. The second, and maybe more fundamental reason is that the AENs in their present state might not be capable of completing an extreme multi-objective task due to the reference space. This is because the connections between the genes and epigenetic molecules are based on a proximity model (because of this, the genes and their products can not interact directly if they are not within each others proximity) and this is not akin to real word gene regulation and could be potentially limiting. Therefore, a more advanced method of deriving gene interaction may be required to achieve this.

On the wider scale of the work conducted in this thesis, it has been shown that an epigenetic analogue can produce beneficial behaviours within a computational network. From a connectionist perspective, there are a key similarities between gene regulatory networks and other computational structures namely, neural networks. In recent times, the relationship between neuronal functionality and epigenetics has become clearer, and it is now widely acknowledged that epigenetics plays a key role in memory formation within the brain (Levenson & Sweatt, 2005). This raises the issue of where exactly artificial epigenetics falls within the range of connectionist architectures. The current scientific trends would suggest that artificial epigenetics could provide benefits in the world of neural networks.

Overall, there are significant benefits of using the AEN over other models specifically in terms of understanding and controlling a task, for which there are currently no competition. The key example of this, is within the coupled inverted pendulums task, where the AENs partition the genes so that certain genes are active when swinging, and certain genes are active when the pendulum is balancing. These partitions are controlled generally by a single epigenetic molecule, and modification of this molecule can switch between the swinging and balancing of the pendulums. The additional benefit of this is it helps generate core understanding about the task that is being controlled. At present, there are no other methods which are able to do with within a single network.

A large amount of the conclusions which were drawn from this thesis were achieved using dynamical systems analysis. In particular there were three techniques used to achieve this. Firstly, by plotting the outputs of the network either as a time-delayed series, or in the case of the control of transfer orbits task, all outputs were plotted over time. This provided an understanding of the overall function of the networks, highlighting that in certain circumstances, the behaviours of the networks would abruptly change (which was down to the epigenetic molecules). The second method was to plot all gene expressions and epigenetic activity over time to ascertain which regions of the network were responsible for certain behaviours. Thirdly, the process of making minimum working examples of the networks whilst maintaining their range of behaviours. This removed a large amount of surplus material from the network which was not part of its functioning. These combinations of techniques create generally small well understood networks which drill down into the constituent behaviours of the networks.

One key aspect of the networks functionality which was not investigated is the how the networks acquire behaviours over their evolutionary life span? This is a key question when it came to creating the minimum working networks, which were often smaller than what was expected. Were the genes that were not essential to the function of the networks *always* surplus to requirements, or were they required during the earlier stages of evolution, but not the later stages? And why were certain networks highly robust to gene deletion, and others so sensitive? One way to better understand potential answers to these questions would be to perform detailed analysis on the networks during their evolutionary life span, rather than after their termination criteria has been met. In addition this would provide a better understanding of exactly how the complex behaviours of the networks emerged. However, to achieve this would require a vastly larger amount of time. A solution would be to create a computational framework which can automate dynamical systems analysis. This is also met with problems such as getting the framework to understand novel behaviours which are previously unknown. Within the work of this thesis, the most effective way to understand the networks was to manually examine the data from the three methods of dynamical systems listed above.

Ultimately, the work done within this thesis has highlighted that the balance between biological faithfulness and computational functionality may need to be readdressed. This is the first attempt in the scientific community of an epigenetically inspired gene regulatory network which functions as a computational controller. Although there are large amounts of further research required to ascertain the true form and functionality of epigenetically inspired networks, the AEN has served as a proof of concept which has specifically demonstrated that elements of epigenetic functionality can be incorporated and captured within a computational model.

Further Work

To summarise the information within the discussion, the best avenues for further work and experimentation are as follows :

- To investigate a range regulatory functions within the epigenetic analogue and to evolve their activation threshold.
- Investigate a wide range of the parameters associated with both the AEN and the genetic algorithms used to evolve them.
- Apply the networks to a wider range of tasks to better understand their functionality.
- Explore the possibility of incorporating epigenetic information in a wider range of computational models.
- Analyse how networks evolve over their evolutionary time span and how specific behaviours and characteristics develop.

Bibliography

Ahnert, S. E., Fink, T. & Zinovyev, A. (2008), 'How much non-coding DNA do eukaryotes require?', *Journal of Theoretical Biology* **252**(4), 587–592.

Akutsu, T., Miyano, S., Kuhara, S. et al. (1999), Identification of genetic networks from a small number of gene expression patterns under the Boolean network model, *in* 'Pacific Symposium on Biocomputing', Vol. 4, World Scientific Maui, Hawaii, pp. 17–28.

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K. & Watson, J. (1994), *Molecular Biology of the Cell, 3rd edition*, Oxford Univ Press.

Aldana, M., Balleza, E., Kauffman, S. & Resendiz, O. (2007), 'Robustness and evolvability in genetic regulatory networks', *Journal of theoretical biology* **245**(3), 433–448.

Allis, C., Jenuwein, T. & Reinberg, D. (2007), *Epigenetics*, Cold Spring Harbor Laboratory Press.

Ancel, L. W. & Fontana, W. (2000), 'Plasticity, evolvability, and modularity in rna', *Journal of Experimental Zoology* **288**(3), 242–283.

Anderson, J. A. & Davis, J. (1995), *An introduction to neural networks*, Vol. 1, MIT Press.

Avrameas, S., Ternynck, T. & Guesdon, J.-L. (2009), 'Coupling of enzymes to antibodies and antigens', *Scandinavian Journal of Immunology* **8**(s7), 7–23.

Back, T., Fogel, D. B. & Michalewicz, Z. (2000), *Evolutionary computation 2: advanced algorithms and operators*, Vol. 2, CRC Press.

Banzhaf, W. (2003), 'Artificial regulatory networks and genetic programming', *Genetic Programming Theory and Practice* pp. 43–62.

Banzhaf, W. (2004), On evolutionary design, embodiment, and artificial regulatory networks, *in* 'Embodied Artificial Intelligence', Springer, pp. 284–292.

Banzhaf, W., Beslon, G., Christensen, S., Foster, J. A., Képès, F., Lefort, V., Miller, J. F., Radman, M. & Ramsden, J. J. (2006), 'Guidelines: From artificial evolution to computational evolution: a research agenda', *Nature Reviews Genetics* **7**(9), 729–735.

Bao, N., Lye, K.-W. & Barton, M. K. (2004), 'MicroRNA binding sites in arabidopsis class iii hd-zip mrnas are required for methylation of the template chromosome', *Developmental cell* **7**(5), 653–662.

Berg, J., Tymoczko, J. & Stryer, L. (2002), *Biochemistry, Fifth Edition.*

Berger, S. L., Kouzarides, T., Shiekhattar, R. & Shilatifard, A. (2009), 'An operational definition of epigenetics', *Genes & development* **23**(7), 781–783.

Beyer, H.-G. & Schwefel, H.-P. (2002), 'Evolution strategies–a comprehensive introduction', *Natural computing* **1**(1), 3–52.

Bhalla, U. S. & Iyengar, R. (1999), 'Emergent properties of networks of biological signaling pathways', *Science* **283**(5400), 381–387.

Blackburn, G. (2006), *Nucleic Acids in Chemistry And Biology*, RSC Pub.

Blom, N., Sicheritz-Pontén, T., Gupta, R., Gammeltoft, S. & Brunak, S. (2004), 'Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence', *Proteomics* **4**(6), 1633–1649.

Blount, Z. D., Borland, C. Z. & Lenski, R. E. (2008), 'Historical contingency and the evolution of a key innovation in an experimental population of escherichia coli', *Proceedings of the National Academy of Sciences* **105**(23), 7899–7906.

Bollt, E. M. & Meiss, J. D. (1995), 'Targeting chaotic orbits to the moon through recurrence', *Physics Letters A* **204**(5), 373–378.

Bonasio, R., Tu, S. & Reinberg, D. (2010), 'Molecular signals of epigenetic states', *Science* **330**(6004), 612–616.

Bornholdt, S. (2008), 'Boolean network models of cellular regulation: prospects and limitations', *Journal of the Royal Society Interface* **5**(Suppl 1), S85–S94.

Brameier, M. & Banzhaf, W. (2007), *Linear Genetic Programming*, Genetic and evolutionary computation series, Springer.

Bray, D. (1995), 'Protein molecules as computational elements in living cells', *Nature* **376**(6538), 307–312.

Bull, L. (2012), 'A simple computational cell: Coupling Boolean gene and protein networks', *Artificial Life* **18**(2), 223–236.

Bull, L. (2013), 'Evolving Boolean regulatory networks with epigenetic control', *arXiv preprint arXiv:1306.4793* .

Burks, A. (1969), *Von Neumann's self-reproducing automata*, University of Michigan.

Bushman, F. (2002), *Lateral DNA Transfer: Mechanisms and Consequences*, Cold Spring Harbor Laboratory Press.

Casati, G., Guarneri, I. & Shepelyansky, D. L. (1988), 'Hydrogen atom in monochromatic field: chaos and dynamical photonic localization', *Quantum Electronics, IEEE Journal of* **24**(7), 1420–1444.

Cazenille, L., Bredeche, N., Hamann, H. & Stradner, J. (2012), Impact of neuron models and network structure on evolving modular robot neural network controllers, *in* 'Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference', ACM, pp. 89–96.

Chen, G. & Dong, X. (1993), 'From chaos to order, perspectives and methodologies in controlling chaotic nonlinear dynamical systems', *International Journal of Bifurcation and Chaos* **3**(06), 1363–1409.

Chen, L. S., Emmert-Streib, F. & Storey, J. D. (2007), 'Harnessing naturally randomized transcription to infer regulatory relationships among genes', *Genome biology* **8**(10), R219.

Chia, N., Cann, I. & Olsen, G. J. (2010), 'Evolution of dna replication protein complexes in eukaryotes and archaea', *PLoS One* **5**(6), e10866.

Chirikov, B. & Sanders, A. (1971), *Research concerning the theory of non-linear resonance and stochasticity*, Nuclear Physics Institute of the Siberian Section of the USSR Academy of Sciences.

Chirikov, B. V. (1979), 'A universal instability of many-dimensional oscillator systems', *Physics reports* **52**(5), 263–379.

Chirikov, R. & Vecheslavov, V. (1989), 'Chaotic dynamics of comet Halley', *Astronomy and Astrophysics* **221**, 146–154.

Chou, K.-C. (2009), 'Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology', *Current Proteomics* **6**(4), 262.

Chuang, J. C. & Jones, P. A. (2007), 'Epigenetics and microRNAs', *Pediatric Research* **61**, 24R–29R.

Clegg, K., Stepney, S. & Clarke, T. (2007), Using feedback to regulate gene expression in a developmental control architecture, *in* 'Proceedings of the 9th annual conference on Genetic and evolutionary computation', ACM, pp. 966–973.

Coello, C. A. C., Lamont, G. B. & Van Veldhuisen, D. A. (2007), *Evolutionary algorithms for solving multi-objective problems*, Springer.

Coffee, C. (2004), *Quick Look: Metabolism*, Quick look medicine, Hayes Barton Press.

Contopoulos, G. & Voglis, N. (1997), 'A fast method for distinguishing between ordered and chaotic orbits.', *Astronomy and Astrophysics* **317**, 73–81.

Cooper, T. F., Rozen, D. E. & Lenski, R. E. (2003), 'Parallel changes in gene expression after 20,000 generations of evolution in escherichia coli', *Proceedings of the National Academy of Sciences* **100**(3), 1072–1077.

Crick, F. & Watson, J. (1953), 'Molecular structure of nucleic acids', *Nature* **171**(4356), 737–738.

Critchley, D. R. (2009), 'Biochemical and structural properties of the integrin-associated cytoskeletal protein talin', *Annual review of biophysics* **38**, 235–254.

Darabos, C., Di Cunto, F., Tomassini, M., Moore, J., Provero, P. & Giacobini, M. (2011), 'Additive functions in Boolean models of gene regulatory network modules', *PloS one* **6**(11), e25110.

Darwin, C. (1859), *On the Origin of the Species by Means of Natural Selection: Or, The Preservation of Favoured Races in the Struggle for Life*, John Murray.

Davidich, M. & Bornholdt, S. (2008), 'Boolean network model predicts cell cycle sequence of fission yeast', *PLoS One* **3**(2), e1672.

Dawkins, R. (2006), *The selfish gene*, Oxford university press.

De Berardinis, P. & Haigwood, N. L. (2004), 'New recombinant vaccines based on the use of prokaryotic antigen-display systems', *Expert review of vaccines* **3**(6), 673–679.

De Jong, H. (2002), 'Modeling and simulation of genetic regulatory systems: a literature review', *Journal of computational biology* **9**(1), 67–103.

Deb, K., Agrawal, S., Pratap, A. & Meyarivan, T. (2000), 'A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II', *Lecture notes in computer science* **1917**, 849–858.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (2002), 'A fast and elitist multiobjective genetic algorithm: NSGA-II', *Evolutionary Computation, IEEE Transactions on* **6**(2), 182–197.

Dworkin, M., Falkow, S., Rosenberg, E., Schleifer, K. & Stackebrandt, E. (2006), *The Prokaryotes: A Handbook on the Biology of Bacteria:Vol. 5: Proteobacteria: Alpha and Beta Subclass*, The Prokaryotes: A Handbook on the Biology of Bacteria. Proteobacteria : Alpha and Beta Subclasses, Springer London, Limited.

Egger, G., Liang, G., Aparicio, A. & Jones, P. A. (2004), 'Epigenetics in human disease and prospects for epigenetic therapy', *Nature* **429**(6990), 457–463.

El Samad, H., Khammash, M., Petzold, L. & Gillespie, D. (2005), 'Stochastic modelling of gene regulatory networks', *International Journal of Robust and Nonlinear Control* **15**(15), 691–711.

Feinberg, A. P., Ohlsson, R. & Henikoff, S. (2006), 'The epigenetic progenitor origin of human cancer', *Nature reviews genetics* **7**(1), 21–33.

Feinberg, M. (1995), 'The existence and uniqueness of steady states for a class of chemical reaction networks', *Archive for Rational Mechanics and Analysis* **132**(4), 311–370.

Fogel, L. J., Owens, A. & Walsh, M. (1964), On the evolution of artificial intelligence(artificial intelligence generated by natural evolution process), *in* 'National Symposium On Human Factors In Electronics, 5 Th, San Diego, California', pp. 63–76.

Fraedrich, K. (1986), 'Estimating the dimensions of weather and climate attractors', *Journal of the atmospheric sciences* **43**(5), 419–432.

Fuente, L. A., Lones, M. A., Turner, A. P., Stepney, S., Caves, L. S. & Tyrrell, A. M. (2012), Evolved artificial signalling networks for the control of a conservative complex dynamical system, *in* 'Information Processing in Cells and Tissues', Springer LNCS 7223, pp. 38–49.

Fuente, L. A., Lones, M. A., Turner, A. P., Stepney, S., Caves, L. S. & Tyrrell, A. M. (2013), 'Computational models of signalling networks for non-linear control', *BioSystems* **112**(2), 122–130.

Gershenson, C. (2003), 'Classification of random Boolean networks', *Artificial Life* **8**, 1–8.

Gershenson, C. (2004), 'Introduction to random Boolean networks', *arXiv preprint nlin/0408006* .

Gershenson, C., Kauffman, S. & Shmulevich, I. (2005), 'The role of redunbdancy in the robustness of random Boolean networks', *arXiv preprint nlin/0511018* .

Gillespie, D. T. (1976), 'A general method for numerically simulating the stochastic time evolution of coupled chemical reactions', *Journal of computational physics* **22**(4), 403–434.

Goldberg, A. D., Allis, C. D. & Bernstein, E. (2007), 'Epigenetics: a landscape takes shape', *Cell* **128**(4), 635–638.

Goldberg, D. E. & Deb, K. (1991), 'A comparative analysis of selection schemes used in genetic algorithms', *Urbana* **51**, 61801–2996.

Gottschling, D. (2004), Summary: epigenetics from phenomenon to field, *in* 'Cold Spring Harbor symposia on quantitative biology', Vol. 69, Cold Spring Harbor Laboratory Press, pp. 507–520.

Griswold, A. (2008), 'Genome packaging in prokaryotes: the circular chromosome of e. coli', *Nature Education* **1**(1).

Guo, H., Meng, Y. & Jin, Y. (2009), 'A cellular mechanism for multi-robot construction via evolutionary multi-objective optimization of a gene regulatory network', *BioSystems* **98**(3), 193–203.

Haken, H. (1975), 'Analogy between higher instabilities in fluids and lasers', *Physics Letters A* **53**(1), 77–78.

Halfmann, R., Alberti, S. & Lindquist, S. (2010), 'Prions, protein homeostasis, and phenotypic diversity', *Trends in cell biology* **20**(3), 125–133.

Hamann, H. & Schmickl, T. (n.d.), 'Coupled inverted pendulums benchmark'.
URL: *http://heikohamann.de/coupledInvertedPendulums/*

Hamann, H., Schmickl, T. & Crailsheim, K. (2011), Coupled inverted pendulums: a benchmark for evolving decentral controllers in modular robotics., *in* 'GECCO', pp. 195–202.

Hannon, G. J. (2002), 'RNA interference', *nature* **418**(6894), 244–251.

Harris, S., Sawhill, B., Wuensche, A. & Kauffman, S. (2002), 'A model of transcriptional regulatory networks based on biases in the observed regulation rules', *Complexity* **7**(4), 23–40.

Harvey, I. (2011), The microbial genetic algorithm, *in* 'Advances in Artificial Life. Darwin Meets von Neumann', Springer, pp. 126–133.

Harvey, I. & Bossomaier, T. (1997), Time out of joint: Attractors in asynchronous random Boolean networks, *in* 'Proceedings of the Fourth European Conference on Artificial Life', MIT Press, Cambridge, pp. 67–75.

Hickey, G., Fletcher, H. & Winter, P. (2007), *Genetics*, Bios Instant Notes, Taylor & Francis Group.

Hilder, J. A. (2010), 'Evolving variability tolerant logic'.

Hintze, A. & Adami, C. (2008), 'Evolution of complex modular biological networks', *PLoS computational biology* **4**(2), e23.

Hoffmann, A., Levchenko, A., Scott, M. L. & Baltimore, D. (2002), 'The iκb-nf-κb signaling module: temporal control and selective gene activation', *Science* **298**(5596), 1241–1245.

Holland, J. H. (1975), *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.*, U Michigan Press.

Holliday, R. (2006), 'Epigenetics: a historical overview', *Epigenetics* **1**(2), 76–80.

Holme, P. & Saramäki, J. (2012), 'Temporal networks', *Physics reports* **519**(3), 97–125.

Hopf, E. (1948), 'A mathematical example displaying features of turbulence', *Communications on Pure and Applied Mathematics* **1**(4), 303–322.

Howarth, L. M., Wood, H. L., Turner, A. P. & Beukers-Stewart, B. D. (2011), 'Complex habitat boosts scallop recruitment in a fully protected marine reserve', *Marine biology* **158**(8), 1767–1780.

Huang, K. & Fan, G. (2010), 'Dna methylation in cell differentiation and reprogramming: an emerging systematic view', *Regenerative medicine* **5**(4), 531–544.

Huke, J. (2006), 'Embedding nonlinear dynamical systems: A guide to Takens' theorem'.

Hwang, K.-K., Eissenberg, J. C. & Worman, H. J. (2001), 'Transcriptional repression of euchromatic genes by drosophila heterochromatin protein 1 and histone modifiers', *Proceedings of the National Academy of Sciences* **98**(20), 11423–11427.

Izraelev, F. (1980), 'Nearly linear mappings and their applications', *Physica D: Nonlinear Phenomena* **1**(3), 243–266.

Jacob, F. & Monod, J. (1961), 'Genetic regulatory mechanisms in the synthesis of proteins', *Journal of molecular biology* **3**(3), 318–356.

Jaenisch, R. & Bird, A. (2003), 'Epigenetic regulation of gene expression: how the genome integrates intrinsic and environmental signals', *Nature genetics* **33**, 245–254.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N. & Barabási, A.-L. (2000), 'The large-scale organization of metabolic networks', *Nature* **407**(6804), 651–654.

Jones, J. (2002), *Amino Acid and Peptide Synthesis*, Oxford Chemistry Primers, 7, Oxford University Press.

Jones, P. A. & Takai, D. (2001), 'The role of dna methylation in mammalian epigenetics', *Science* **293**(5532), 1068–1070.

Jordan, D. & Smith, P. (1999), *Nonlinear Ordinary Differential Equations: An Introduction to Dynamical Systems*, Oxford Applied and Engineering Mathematics, Oxford University Press.

Kantz, H. & Schreiber, T. (2004), *Nonlinear Time Series Analysis*, Cambridge nonlinear science series, Cambridge University Press.

Kaplan, N., Moore, I. K., Fondufe-Mittendorf, Y., Gossett, A. J., Tillo, D., Field, Y., LeProust, E. M., Hughes, T. R., Lieb, J. D., Widom, J. et al. (2008), 'The DNA-encoded nucleosome organization of a eukaryotic genome', *Nature* **458**(7236), 362–366.

Karlebach, G. & Shamir, R. (2008), 'Modelling and analysis of gene regulatory networks', *Nature Reviews Molecular Cell Biology* **9**(10), 770–780.

Katare, S., Bhan, A., Caruthers, J. M., Delgass, W. N. & Venkatasubramanian, V. (2004), 'A hybrid genetic algorithm for efficient parameter estimation of large kinetic models', *Computers & chemical engineering* **28**(12), 2569–2581.

Kauffman, S. (1969), 'Metabolic stability and epigenesis in randomly constructed genetic nets', *Journal of theoretical biology* **22**(3), 437–467.

Kellert, S. H. (1994), *In the wake of chaos: Unpredictable order in dynamical systems*, University of Chicago Press.

Khavari, D. A., Sen, G. L. & Rinn, J. L. (2010), 'DNA methylation and epigenetic control of cellular differentiation', *Cell Cycle* **9**(19), 3910–3913.

Kholodenko, B. N. (2006), 'Cell-signalling dynamics in time and space', *Nature reviews Molecular cell biology* **7**(3), 165–176.

Kierzek, A. M. (2002), 'Stocks: Stochastic kinetic simulations of biochemical systems with gillespie algorithm', *Bioinformatics* **18**(3), 470–481.

Kim, J., Samaranayake, M. & Pradhan, S. (2009), 'Epigenetic mechanisms in mammals', *Cellular and molecular life sciences* **66**(4), 596–612.

Kingsmore, S. F. (2006), 'Multiplexed protein measurement: technologies and applications of protein and antibody arrays', *Nature reviews Drug discovery* **5**(4), 310–321.

Kirschner, M. & Gerhart, J. (1998), 'Evolvability', *Proceedings of the National Academy of Sciences* **95**(15), 8420–8427.

Kitano, H. (2004), 'Biological robustness', *Nature Reviews Genetics* **5**(11), 826–837.

Kitano, H. (2007), 'Towards a theory of biological robustness', *Molecular systems biology* **3**(1).

Kouzarides, T. & Berger, S. (2007), *Epigenetics*, Cold Spring Harbor Laboratory Press, chapter Chromatin Modifications And Their Mechanism Of Action.

Koza, J. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, A Bradford book, Bradford.

Koza, J. R., Mydlowec, W., Lanza, G., Yu, J. & Keane, M. A. (2000), Reverse engineering and automatic synthesis of metabolic pathways from observed data using genetic programming, *in* 'Pacific Symp Biocomput. v6', pp. 434–445.

Krude, T. (2004), *DNA: Changing Science and Society*, Darwin College Lectures, Cambridge University Press.

Kuang, Y. (1993), *Delay differential equations: with applications in population dynamics*, Academic Press.

Kutschera, U. & Niklas, K. J. (2004), 'The modern theory of biological evolution: an expanded synthesis', *Naturwissenschaften* **91**(6), 255–276.

Kuyucu, T. (2010), Evolution of circuits in hardware and the evolvability of artificial development, PhD thesis, University of York.

Lai, L., Park, K.-W., Cheong, H.-T., Kühholzer, B., Samuel, M., Bonk, A., Im, G.-S., Rieke, A., Day, B. N., Murphy, C. N. et al. (2002), 'Transgenic pig expressing the enhanced green fluorescent protein produced by nuclear transfer using colchicine-treated fibroblasts as donor cells', *Molecular reproduction and development* **62**(3), 300–306.

Lee, J.-H., Hart, S. R. & Skalnik, D. G. (2004), 'Histone deacetylase activity is required for embryonic stem cell differentiation', *Genesis* **38**(1), 32–38.

Lenski, R. E. (2010), 'Bacterial evolution and the cost of antibiotic resistance', *International Microbiology* **1**(4), 265–270.

Levenson, J. M. & Sweatt, J. D. (2005), 'Epigenetic mechanisms in memory formation', *Nature Reviews Neuroscience* **6**(2), 108–118.

Li, F., Long, T., Lu, Y., Ouyang, Q. & Tang, C. (2004), 'The yeast cell-cycle network is robustly designed', *Proceedings of the National Academy of Sciences of the United States of America* **101**(14), 4781–4786.

Li, J. B., Levanon, E. Y., Yoon, J.-K., Aach, J., Xie, B., LeProust, E., Zhang, K., Gao, Y. & Church, G. M. (2009), 'Genome-wide identification of human rna editing sites by parallel dna capturing and sequencing', *Science* **324**(5931), 1210–1213.

Lindgren, K., Nilsson, A., Nordahl, M. G. & Råde, I. (1993), Evolving recurrent neural networks, *in* 'Artificial Neural Nets and Genetic Algorithms', Springer, pp. 55–62.

Lister, R., Pelizzola, M., Kida, Y. S., Hawkins, R. D., Nery, J. R., Hon, G., Antosiewicz-Bourget, J., OMalley, R., Castanon, R., Klugman, S. et al. (2011), 'Hotspots of aberrant epigenomic reprogramming in human induced pluripotent stem cells', *Nature* **471**(7336), 68–73.

Lones, M. A. (2004), 'Enzyme genetic programming: Modelling biological evolvability in genetic programming', *PhD Thesis, University of York* .

Lones, M. A., Turner, A., Fuente, L., Caves, L., Stepney, S. & Tyrrell, A. (2013), 'Biochemical connectionism', *Natural Computing - In press* .

Lones, M. A., Tyrrell, A. M., Stepney, S. & Caves, L. (2011), Controlling legged robots with coupled artificial biochemical networks, *in* 'Proc. 11th European Conference on the Synthesis and Simulation of Living Systems, Advances in Artificial Life, ECAL', pp. 465–472.

Lones, M., Turner, A., Caves, L., Stepney, S., Smith, S., Tyrrell, A. et al. (2012), 'Artificial biochemical networks: Evolving dynamical systems to control dynamical systems'.

Lones, M., Tyrrell, A., Stepney, S. & Caves, L. (2010), 'Controlling complex dynamics with artificial biochemical networks', *Genetic Programming* pp. 159–170.

Lorenz, E. N. (1963), 'Deterministic nonperiodic flow', *Journal of the atmospheric sciences* **20**(2), 130–141.

Lu, H., Isralewitz, B., Krammer, A., Vogel, V. & Schulten, K. (1998), 'Unfolding of titin immunoglobulin domains by steered molecular dynamics simulation', *Biophysical Journal* **75**(2), 662–671.

Luger, K., Mäder, A. W., Richmond, R. K., Sargent, D. F. & Richmond, T. J. (1997), 'Crystal structure of the nucleosome core particle at 2.8 å resolution', *Nature* **389**(6648), 251–260.

Lunyak, V. V. & Rosenfeld, M. G. (2008), 'Epigenetic regulation of stem cell fate', *Human molecular genetics* **17**(R1), R28–R36.

Luque, B. & Solé, R. (1997), 'Controlling chaos in random Boolean networks', *EPL (Europhysics Letters)* **37**(9), 597.

Maison, C., Bailly, D., Peters, A. H., Quivy, J.-P., Roche, D., Taddei, A., Lachner, M., Jenuwein, T. & Almouzni, G. (2002), 'Higher-order structure in pericentric heterochromatin involves a distinct pattern of histone modification and an rna component', *Nature genetics* **30**(3), 329–334.

Matthey, L., Berman, S. & Kumar, V. (2009), Stochastic strategies for a swarm robotic assembly system, *in* 'Robotics and Automation, 2009. ICRA'09. IEEE International Conference on', IEEE, pp. 1953–1958.

Mattick, J. S. (2001), 'Non-coding rnas: the architects of eukaryotic complexity', *EMBO reports* **2**(11), 986–991.

McCabe, L., McCabe, E. & McKusick, V. (2008), *Dna: Promise and Peril*, University of California Press.

Meissner, A., Mikkelsen, T. S., Gu, H., Wernig, M., Hanna, J., Sivachenko, A., Zhang, X., Bernstein, B. E., Nusbaum, C., Jaffe, D. B. et al. (2008), 'Genome-scale dna methylation maps of pluripotent and differentiated cells', *Nature* **454**(7205), 766–770.

Mestl, T., Plahte, E. & Omholt, S. W. (1995), 'A mathematical framework for describing and analysing gene regulatory networks', *Journal of Theoretical Biology* **176**(2), 291–300.

Mikkola, S. (1999), 'Efficient symplectic integration of satellite orbits', *Celestial Mechanics and Dynamical Astronomy* **74**(4), 275–285.

Miller, J. F. & Thomson, P. (2000), Cartesian genetic programming, *in* 'Genetic Programming', Springer, pp. 121–132.

Miller, J. & Reznikoff, W. (1978), *The Operon*, Cold Spring Harbor monograph series, Cold Spring Harbor Laboratory.

Millevoi, S. & Vagner, S. (2010), 'Molecular mechanisms of eukaryotic pre-mrna 3 end processing regulation', *Nucleic acids research* **38**(9), 2757–2774.

Mitchell, M. (1998), 'An introduction to genetic algorithms (complex adaptive systems)', *A Bradford Book* .

Moros, R., Kalies, H., Rex, H. & Schaffarczyk, S. (1996), 'A genetic algorithm for generating initial parameter estimations for kinetic models of catalytic processes', *Computers & chemical engineering* **20**(10), 1257–1270.

Muehlenbein, M. (2010), *Human Evolutionary Biology*, Cambridge University Press.

Nagai, T., Unno, H., Janczak, M. W., Yoshimura, T., Poulter, C. D. & Hemmi, H. (2011), 'Covalent modification of reduced flavin mononucleotide in type-2 isopentenyl diphosphate isomerase by active-site-directed inhibitors', *Proceedings of the National Academy of Sciences* **108**(51), 20461–20466.

New, K. C., Watt, K., Misner, C. W. & Centrella, J. M. (1998), 'Stable 3-level leapfrog integration in numerical relativity', *Physical Review D* **58**(6), 064022.

Newman, M. E. (2006), 'Modularity and community structure in networks', *Proceedings of the National Academy of Sciences* **103**(23), 8577–8582.

Nishikura, K. (2010), 'Functions and regulation of RNA editing by adar deaminases', *Annual review of biochemistry* **79**, 321.

Nordin, P., Banzhaf, W. et al. (1995), Complexity compression and evolution., *in* 'ICGA', pp. 310–317.

Online Etymology Dictionary (2013), *http://www.etymonline.com/* .

Petruk, S., Sedkov, Y., Johnston, D. M., Hodgson, J. W., Black, K. L., Kovermann, S. K., Beck, S., Canaani, E., Brock, H. W. & Mazo, A. (2012), 'Trxg and pcg proteins but not methylated histones remain associated with DNA through replication', *Cell* .

Petsko, G. & Ringe, D. (2004), *Protein Structure and Function*, Primers in Biology, New Science Press.

Phelps, K. & Hassed, C. (2012), *Immune System: General Practice: The Integrative Approach*, Elsevier Health Sciences APAC.

Phillips, T. (2008), 'The role of methylation in gene expression', *Nature Education* **1**(1).

Pitera, J. W. & Swope, W. (2003), 'Understanding folding and design: Replica-exchange simulations of"trp-cage"miniproteins', *Proceedings of the National Academy of Sciences* **100**(13), 7587–7592.

Popovych, N., Tzeng, S.-R., Tonelli, M., Ebright, R. H. & Kalodimos, C. G. (2009), 'Structural basis for camp-mediated allosteric control of the catabolite activator protein', *Proceedings of the National Academy of Sciences* **106**(17), 6927–6932.

Pyragas, K. (1992), 'Continuous control of chaos by self-controlling feedback', *Physics Letters A* **170**(6), 421–428.

Radzicka, A., Wolfenden, R. et al. (1995), 'A proficient enzyme', *Science* **267**(5194), 90–93.

Reil, T. (1999), Dynamics of gene expression in an artificial genomeimplications for biological and artificial ontogeny, *in* 'Advances in Artificial Life', Springer, pp. 457–466.

Ribeiro, A., Zhu, R. & Kauffman, S. A. (2006), 'A general modeling strategy for gene regulatory networks with stochastic dynamics', *Journal of Computational Biology* **13**(9), 1630–1639.

Richmond, T. J. & Davey, C. A. (2003), 'The structure of DNA in the nucleosome core', *Nature* **423**(6936), 145–150.

Richter, H. (2002), An evolutionary algorithm for controlling chaos: The use of multiobjective fitness functions, *in* 'Parallel Problem Solving from NaturePPSN VII', Springer, pp. 308–317.

Riddihough, G. & Zahn, L. M. (2010), 'What is epigenetics?', *Science* **330**(6004), 611–611.

Robinson, C. V., Sali, A. & Baumeister, W. (2007), 'The molecular sociology of the cell', *Nature* **450**(7172), 973–982.

Robinson, P. J. & Rhodes, D. (2006), 'Structure of the 30nm chromatin fibre: a key role for the linker histone', *Current opinion in structural biology* **16**(3), 336–343.

Robinson, S., de Reuille, P. B., Chan, J., Bergmann, D., Prusinkiewicz, P. & Coen, E. (2011), 'Generation of spatial patterns through cell polarity switching', *Science* **333**(6048), 1436–1440.

Roli, A., Manfroni, M., Pinciroli, C. & Birattari, M. (2011*a*), 'Boolean network robotics: a proof of concept', *arXiv preprint arXiv:1101.6001* .

Roli, A., Manfroni, M., Pinciroli, C. & Birattari, M. (2011*b*), On the design of Boolean network robots, *in* 'Applications of Evolutionary Computation', Springer, pp. 43–52.

Romeiras, F. J., Grebogi, C., Ott, E. & Dayawansa, W. (1992), 'Controlling chaotic dynamical systems', *Physica D: Nonlinear Phenomena* **58**(1), 165–192.

Ros, V. I. & Hurst, G. D. (2009), 'Lateral gene transfer between prokaryotes and multicellular eukaryotes: ongoing and significant?', *BMC biology* **7**(1), 20.

Rudolph, J. L. & Stewart, J. (1998), 'Evolution and the nature of science: On the historical discord and its implications for education', *Journal of Research in Science Teaching* **35**(10), 1069–1089.

Saito, Y., Liang, G., Egger, G., Friedman, J. M., Chuang, J. C., Coetzee, G. A. & Jones, P. A. (2006), 'Specific activation of microrna-127 with downregulation of the proto-oncogene bcl6 by chromatin-modifying drugs in human cancer cells', *Cancer cell* **9**(6), 435–443.

Sanchez, E. N. & Ricalde, L. J. (2003), 'Chaos control and synchronization, with input saturation, via recurrent neural networks', *Neural Networks* **16**(5), 711–717.

Sarkar, S. & Plutynski, A. (2008), *A Companion to the Philosophy of Biology*, Blackwell Companions to Philosophy, Wiley.

Schalch, T., Duda, S., Sargent, D. F. & Richmond, T. J. (2005), 'X-ray structure of a tetranucleosome and its implications for the chromatin fibre', *Nature* **436**(7047), 138–141.

Schroeder Jr, H. W. & Cavacini, L. (2010), 'Structure and function of immunoglobulins', *Journal of Allergy and Clinical Immunology* **125**(2), S41–S52.

Siomi, H. & Siomi, M. C. (2009), 'On the road to reading the RNA interference code', *Nature* **457**(7228), 396–404.

Slesarev, A. I., Belova, G. I., Kozyavkin, S. A. & Lake, J. A. (1998), 'Evidence for an early prokaryotic origin of histones h2a and h4 prior to the emergence of eukaryotes', *Nucleic acids research* **26**(2), 427–430.

Solé, R., Luque, B. & Kauffman, S. (1999), 'Phase transition in random networks with multiple states', *arXiv preprint adap-org/9907011* .

Speijer, D. (2011), 'Does constructive neutral evolution play an important role in the origin of cellular complexity?', *Bioessays* **33**(5), 344–349.

Stanley, K. O. & Miikkulainen, R. (1996), 'Efficient reinforcement learning through evolving neural network topologies', *Network (Phenotype)* **1**(2), 3.

Stanley, K. O. & Miikkulainen, R. (2002), 'Evolving neural networks through augmenting topologies', *Evolutionary computation* **10**(2), 99–127.

Stepney, S. (2009), Visualising random Boolean network dynamics, *in* 'Proceedings of the 11th Annual conference on Genetic and evolutionary computation', ACM, pp. 1781–1782.

Stoker, H. (2011), *Organic and Biological Chemistry, 6th ed.*, Brooks Cole Publishing Company.

Stoll, G., Viara, E., Barillot, E. & Calzone, L. (2012), 'Continuous time Boolean modeling for biological signaling: application of gillespie algorithm', *BMC systems biology* **6**(1), 116.

Strachan, T. & Read, A. (2004), *Human Molecular Genetics 3*, Garland Press.

Sturrock, M., Hellander, A., Matzavinos, A. & Chaplain, M. A. (2013), 'Spatial stochastic modelling of the hes1 gene regulatory network: intrinsic noise can explain heterogeneity in embryonic stem cell differentiation', *Journal of The Royal Society Interface* **10**(80).

Takens, F. (1981), Detecting strange attractors in turbulence, *in* 'Dynamical systems and turbulence, Warwick 1980', Springer, pp. 366–381.

Taylor, T. (2004), A genetic regulatory network-inspired real-time controller for a group of underwater robots, *in* 'Intelligent Autonomous Systems', Vol. 8, pp. 403–412.

Tollefsbol, T. (2010), *Handbook of Epigenetics: The New Molecular and Medical Genetics*, Elsevier Science.

Tora, L. & Timmers, H. (2010), 'The tata box regulates tata-binding protein (tbp) dynamics in vivo', *Trends in biochemical sciences* **35**(6), 309–314.

Tsai, C. (2007), *Biomacromolecules: Introduction to Structure, Function and Informatics*, Wiley.

Tsai, H.-H., Fuh, C.-C. & Chang, C.-N. (2002), 'A robust controller for chaotic systems under external excitation', *Chaos, Solitons & Fractals* **14**(4), 627–632.

Turner, A., Lones, M., Fuente, L., Stepney, S., Caves, L. & Tyrrell, A. (2012), 'Using artificial epigenetic regulatory networks to control complex tasks within chaotic systems', *Information Processing in Cells and Tissues, LNCS 7223* pp. 1–11.

Turner, A. P., Lones, M. A., Fuente, L. A., Caves, L. S., Stepney, S. & Tyrrell, A. M. (2013*a*), 'The artificial epigenetic network', *SSCI 2013, Singapore, April 2013, IEEE Press* .

Turner, A. P., Lones, M. A., Fuente, L. A., Caves, L. S., Stepney, S. & Tyrrell, A. M. (2013*b*), 'The incorporation of epigenetics in artificial gene regulatory networks', *Biosystems 112(2):65-62* .

Vargas, A. O. (2009), 'Did paul kammerer discover epigenetic inheritance? a modern look at the controversial midwife toad experiments', *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution* **312**(7), 667–678.

Veening, J.-W., Smits, W. K. & Kuipers, O. P. (2008), 'Bistability, epigenetics, and bet-hedging in bacteria', *Annu. Rev. Microbiol.* **62**, 193–210.

Viswanath, D. (2003), 'Symbolic dynamics and periodic orbits of the lorenz attractor', *Nonlinearity* **16**(3), 1035.

Von Neumann, J. & Burks, A. (1966), 'Theory of self-reproducing automata', *University of Illinois, Urbana* .

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

Walsh, C. (2006), *Posttranslational Modifications of Proteins: Expanding Nature's Inventory*, Roberts and Company Publishers.

Willadsen, K. & Wiles, J. (2007), 'Robustness and state-space structure of Boolean gene regulatory models', *Journal of theoretical biology* **249**(4), 749–765.

Wilson, B. A., Salyers, A. A., Whitt, D. D., Winkler, M. E. et al. (2011), *Bacterial pathogenesis: A molecular approach.*, number Ed. 3, American Society for Microbiology (ASM).

Wolfenden, R. & Snider, M. J. (2001), 'The depth of chemical time and the power of enzymes as catalysts', *Accounts of chemical research* **34**(12), 938–945.

Wuensche, A. (1998), 'Discrete dynamical networks and their attractor basins'.

Yao, X. (1993), 'A review of evolutionary artificial neural networks', *International journal of intelligent systems* **8**(4), 539–567.

Yao, X. (1999), 'Evolving artificial neural networks', *Proceedings of the IEEE* **87**(9), 1423–1447.

Yao, X. & Liu, Y. (1997), 'A new evolutionary system for evolving artificial neural networks', *Neural Networks, IEEE Transactions on* **8**(3), 694–713.