Growing Flags With Fractal Proteins

Harry Booth

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science
of
University College London.

Department of Physics and Astronomy
University College London

August 26, 2021

I, Harry Booth, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This work develops an artificial model of development which permits a compact genetic representation. Specific model instances are evolved that are capable of achieving a targeted morphology through the development process. In order to achieve this, specific evolutionary strategies are designed and evaluated experimentally.

Acknowledgements

Many thanks to Dr. Peter J. Bentley for introducing me to the field of artificial development, and for providing inspiration and direction. Also to my wife, Liza, whose relentless support has made everything possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Outline

The phenotypes of organisms throughout the biological world are incredibly complex. As an example, a fully formed adult is made up of approximately 10^{13} cells, which form an all manner of specialised tissues and organs. The most important organ of this fully formed adult, the brain, has around 10^9 neurons and functions with a complexity we have only started to understand.

However, the complete instruction set for the development of a human being is found in every cell, and contains a relatively small amount of information compared to the complexity of the phenotype previously described. There are approximately 3^9 base pairs within the human genome [15], and since there are 4 types of nucleotide bases (adenine, cytosine, guanine and thymine) the identity of each base pair can be specified using only 2 bits. This means the entire genome contains around 750 megabytes of data required for development - for reference, this is slightly larger than a CD-ROM of your favourite 12 track album. Not bad for a piece of walking, thinking meat [12].

As a comparison, one of the largest artificial neural networks trained for human-like language generation, the Generative Pre-trained Transformer 3, has around 175^9 parameters [16] Assuming that each of these parameters is FP16 (16-bit), this puts the memory requirements of the model at around 165 gigabytes. This is a couple of orders of magnitude larger than the CD containing the human genome. Yet the function of the network imitates only a single one of the abilities demonstrated by developed humans.

These comparisons are slightly back of the envelope, and one could argue that equating analog representations with digital representations in terms of information capacity is misleading (and perhaps points to the critical difference). However the fact remains that human development achieves rather a lot with very little, and highlights the fact that the genotype-to-phenotype mapping found in biological or-

ganisms is indirect and many-to-one.

The genotype-phenotype link is a development process. Understanding this link is the primary concern of the field of evolutionary development biology. It is also an area of interest for researchers in the field of evolutionary computing, who seek the principles of compact genotype-phenotype mappings in order to solve increasingly complex problems.

1.2 Aims and Goals

The aim of this project is to investigate artificial models of development which permit compact genetic representations from both a conceptual and experimental standpoint. The project aims to identify desirable characteristics of biological development and implement these computationally within an integrated model, before investigating the viability of evolving specific model instances that are capable of achieving a targeted morphology through a development process. In order to achieve this, specific evolutionary strategies will be designed and evaluated experimentally.

It is hoped that the work presented here will guide future research, and provide inspiration for those designing their own artificial models of development.

1.3 Outline of Report

Various concepts important to the field of artificial development are introduced in Chapter 2. The relevant background literature is reviewed, and the conclusions presented in context of the aims and goals of the project. Chapter 3 contains the complete specification of the proposed development model and evolutionary strategies. Section 3.1 describes the motivations behind specific design choices and hypotheses about model behaviour, whilst the other sections within the chapter detail the specific model mechanisms. In Chapter 4 three experiments designed to test the capabilities of the model and evolutionary strategies are described. In Section 4.1 a set of desirable properties for the evolved organisms are described along with methods of measurement. These provide a set of criteria against which the organisms evolved in the experiments are assessed.

Chapter 2

Background

This chapter provides an overview of existing approaches to artificial development.

2.1 Embryology

The mapping from genotype to phenotype in natural evolution is a development growth process [14]. The initial stages of this developmental process is known as embryogeny. As stated by Bentley and Kumar in [35], Embryology is regarded to be the study of the controlled formation and development of plant and animal embryos and involves the study of three fundamental processes (adapted from [35]):

1. **morphogenesis** - the emergence and change of form
2. **regional specification (pattern formation)** - the compartmentalisation of the embryo into specific regions
3. **cellular differentiation** - the specialisation of cells for particular functions

Arising through a symphony of concerted cellular actions, these three processes are fundamental in the transformation of a single fertilized egg into an organism boasting spatially and functionally organized tissues and organs.

As the application domain of evolutionary computing becomes increasingly complex, it is clear that direct genotype-to-phenotype mappings will lead to intractable search spaces [54]. Inspired by nature, many researchers are therefore starting to make use of indirect mappings which incorporate a type of development process. These indirect mappings have the following properties (adapted from [10]):

1. **Indirect correspondence between alleles and phenotypic effects** - the genotype is regarded as a set of phenotype growing instructions
2. **Polygeny** - Phenotypic traits are produced by multiple genes acting in combination

Study of such mappings in the context of evolutionary computation has several names, including computational embryogeny [10] and artificial embryogeny [54].

2.1.1 Computational Embryogenies

In [54] computational embryogenies are divided into grammatical and cell chemistry approaches.

Grammatical approaches are similar to traditional computer programs; they include symbols, and rules to manipulate and interpret them. These grammatical programs provide an instruction set for developing the phenotype. The grammars can be evolved or hand designed - in [10], these are referred to as external and explicit embryogenies respectively. The first use of a grammar to model biological development was by Aristid Lindemayer, who introduced *L-Systems* to simulate plant growth. Using hand-designed grammars, Lindemayer was able to produce complex fractal like objects which looked remarkably similar to some biological structures.

Cell chemistry approaches attempt to mimic more closely how physical structures emerge in biology [54] and coincides with the 'implicit' embryogeny classification in [10]. Here, interacting systems inspired by nature generate complex and indirect chains which result in an 'emergent' phenotype. Often, this involves interacting chemicals produced by cellular assemblies which trigger cellular events such as mitosis, cell differentiation and apoptosis, for example in [24]. The addition of simulated physics to the cellular assemblies can produce complex 3D forms [13]. Chemicals and their rules of interaction can be defined by the genotype. An example of such a chemical system is Bentley's fractal proteins [8].

In [55], the mechanisms which construct various cell chemistry approaches are classified as either *phenotypic* mechanisms or *genetic control* mechanisms. The phenotypic mechanisms are those parts of the embryogeny which represent developing shape or behaviour, for example the implementation of cells and their behaviours (differentiation, mitosis, apoptosis, adhesion), simulated physics etc. Genetic control mechanisms are those components of the embryogeny which are responsible for coordinating these behaviours, and whose mechanics are usually defined by the genotype. Examples include gene regulatory networks and recurrent neural networks. The authors then state that

> All of these approaches have in common that they create a non-linear system where certain 'output nodes' are used to control development and 'input nodes' are carefully initialized to trigger dynamics or receive continuous environmental signals.

This implies that the design of cell chemistry approaches can be abstracted as the design of a non-linear dynamical system and its resultant phase space. The capability of the embryogeny to produce specific phenotypes relates directly to properties of the phase space, which are altered through evolution of the genetic control mechanisms. Therefore rather than evolve the controllers, they evolve the phase space directly using vector field editing [55].

Works such as [33] [29] investigate how properties of the phase space associated with gene regulatory networks corresponds to aspects of cell development. In [50], Seoane argues that rather than designing specific phase spaces, evolution has *exploited* good computational 'reservoirs' which are provided throughout nature by physical systems. This suggests a deep link between genetic control mechanisms and the physical media in which they are implemented. This underpins the formulation of fractal proteins in [8], in which computational mechanisms are extracted from a reservoir of spatial complexity provided by the Mandelbrot set.

2.2 Phenotypic Mechanisms

This section outlines some of the specific models found in the literature that are used to represent the phenotypic mechanisms forming a computational embryogeny.

2.2.1 Cellular Automata

Cellular Automata are one of the simplest and most studied artificial development systems. Perhaps the most well known cellular automata is Conway's Game Of Life [1]. The Game Of Life consisted of a grid, an initial seed state and a set of update rules. The initial seed state specifies which of the grid cells are unoccupied or occupied. The update rule then determines the state of a grid cell at each subsequent time step given the state of its immediate neighbours on the grid. Despite the apparent simplicity of the system, many interesting behaviours and patterns can be produced through repeated application of the update rule.

The question arises whether it is possible to design an update rule so that a particular form or behaviour is achieved. Suppose the update rule for a cell depends on the 8 immediate cells surrounding it on the 2D grid. Then an update rule is a mapping from each of these unique neighbourhood states to a binary outcome, live or die. Hence there are 2^{2^8} possible update rules. The direct search space is therefore extremely large. Whilst some work has focused on direct evolution of these

lookup-style updates [20], researchers have also opted to replace the lookup-style update with universal functions such as neural networks [25] [62]. Recently, grid-based cellular automata and their update rules have been recast as convolutional neural networks [27]. Using this framework additional grids have been 'stacked' on top of the singular state grid to represent the states of other environmental factors, such as morphogens. The update rule for living or dying then also depends on the state of these environmental factors. Update rules do not only determine the alive or dead status of cells - they also determine how the environmental factors are updated across the grid through time. This could be interpreted as representing a physical process such as diffusion. Within this framework, the full power of modern differentiable computing has been applied to finding update rules for target morphologies, with impressive results [42].

2.2.2 The Cellular Potts Model

Lacking from the cellular automata model is any real physical laws. Interactions between cells and their states is purely in the hands of the update rule, and as a result so is all emergent behaviour and patterns.

The Cellular Potts (CP) model (also known as the Glazier-Graner-Hogeweg model) is a computational model of cells and tissues. Introduced by Glazier et al. in [28], it was used to simulate the sorting of a mixture of two types of biological cells. This was motivated by the observation that embryonic cells of two different types, when dissociated, randomly mixed and re-aggregated, can spontaneously sort to re-establish homogeneous tissues. Experiments have demonstrated that differences in intercellular adhesivity (resulting in a differential surface energy between cells) drives the sorting, and this was the physical phenomena Glazier et al. successfully modelled through the CP abstraction. Through their computational model, they simulated the emergence of organised structure from a collection of cells with initially random positions, suspended in a substrate.

Spatial rearrangement of cell positions is important for the construction of specialised tissue structures [17]. The CP model shows that this type of self organisation can arise as a property of differentiated cells and their differential adhesivity. Since, the CP model and its extensions have been used to model a wide range of cellular behaviours such as tumour growth [4] and morphogenesis [57]. Popular extensions include the addition of cellular behaviours such as chemotaxis [49], or additional physical phenomena such as the distinction between fluids and solids [5].

Similar to cellular automata, the environment is modelled discretely as a lattice, with cells occupying various positions. In contrast to cellular automata, there is no

cellular update rule - only a global energy function which encodes the appropriate physics for the complete ensemble of cells. A Monte-Carlo method based on the metropolis algorithm [18] then determines the system evolution. For this reason, the original CP model can be seen as a pure phenotypic mechanism. However, researchers have added genetic control mechanisms to the original formulation to produce complete models of development [34].

2.2.3 Simulated Physics

Other physics-based environments have been used in the context of artificial development. In [52], Sims evolved virtual creatures with rigid articulated limbs in a physically simulated 3D world. In [13], the MathEngine physics-based simulation package [1] is used to study the relationship between symmetric morphology and efficient locomotion in evolved agents.

2.3 Genetic Control Mechanisms

2.3.1 Gene Regulatory Networks

Genes refer to specific regions of DNA that code for a specific protein [3]. Gene expression refers to the process in which specialist cell machinery interprets the DNA instruction set of a gene and produces the corresponding protein. In some instances the proteins produced by a cell directly contribute to the level of expression of other genes [38]. It is through this type of feedback loop that gene expression is regulated, and it is a collection of such mechanisms which forms a Gene Regulatory Network. These networks are responsible for specifying the sets of genes that must be expressed in specific temporal and spatial patterns to enable various developmental processes [37].

The modelling and implementation of artificial gene regulatory networks is an expanding field of research, and a large variety of models exist for a range of different purposes. Many employ systems of differential equations to model the binding of proteins to regulatory regions of genes and the subsequent reactions [45]. Some models can act as an inverse model for gene expression data, from which gene dependency relationships can then be inferred [21].

[1] MathEngine PLC, Oxford, UK, www.mathengine.com

Others aim to abstract the key properties of gene regulatory networks, whilst remaining connected to biological principles. Approaches taken by the theoretical biology community include discrete structural models such as boolean networks [51] or petri-nets [11], and aim to identify dynamical regimes which can then be experimentally investigated. Some approaches employ weight matrices to encode gene dependency structures [58].

In the computer science community, some gene regulatory networks have been made differentiable [59]. However, the prevailing strategy for finding networks with desirable behaviours is through genetic algorithms and evolution. This requires models with evolvable representations - in [6], bit strings encode a genome with specific sub-sequences indicating promoter regions. The gene coded after the promoter is represented by 160 bits, with a further 64 bits coding an enhancer and inhibitor site. The genome is capable of producing various activation dynamics. A review of similar encodings can be found in [23]. In [9], a protein chemistry was designed using fractal representations of proteins - each protein was coded using 3 real values. A model of gene regulation which exploited this fractal chemistry was then designed and evolved.

2.4 Novelty Search Algorithms

In [46], Pugh et al. suggest that natural evolution is a divergent search, which optimizes locally within each niche as it simultaneously diversifies. This diversification can be seen as the search for novelty - a mutation which allows an offspring to occupy a different environmental niche to its parents means that it no longer has to compete for the same resources. Rather than adapting to be fitter than its parents, it adapts to be different. Once it reproduces and produces offspring, the niche begins to be filled and competition starts again. This process continues, with evolution gradually discovering solutions of both increasing diversity and increasing quality. This view of natural evolution forms the inspiration for Quality-Diversity genetic algorithms, a class of algorithm which aims to return an archive of diverse, high-quality behaviours in a single run. Two such algorithms are MAP-Elites [43] and novelty search with local competition [36].

In many artificial development problems, specific objectives are often described by the fitness functions of the genetic algorithms in use. However, if the fitness function is too restrictive it can trap the system in a local optimum. Furthermore, it predetermines the types of morphology or patterns that might be expected to arise from the development system - in some ways negating the point of the experiment. For this reason, more natural fitness functions have been created. In [22],

the authors evaluated artificial creatures on the length of time they were able to survive within a hazardous environment. Using this natural measure of fitness, cells learnt defensive development strategies and adopted particular morphologies that enabled survival. In [32], a novelty search type genetic algorithm was used to map out the space of possible phenotypes given the chosen development system. Fitness was related to novelty, defined using a pixel based distance measure between the 3D phenotypes. By enforcing novelty, interesting shapes were discovered - the organisms had to produce unique morphologies to survive. In [40] a novelty measure was used in the evolution of soft robot morphologies.

2.5 The French Flag Problem

The French flag is a common morphology targeted by artificial development systems due to its spatially differentiated form. In [19], Chavoya and Duyen used an artificial gene regulatory network to control the expression of specific cellular automata update rules. Combined with pre-positioned morphogen gradients, the authors successfully grew scalable French flags. In [34], a continuous time gene regulatory network was added to the CP model as a cell controller, along with auxiliary lattices on which morphogens diffused. Protein concentrations had control over individual cell parameters like size, shape, adhesion, morphogen secretion and orientation. Evolved multicellular organisms were able to autonomously set up an asymmetric morphogen gradient and ultimately organize into a close match of the French flag pattern. In [31] the French flag problem was extended into 3D. The authors designed a development system consisting of spherical cells in a simulated physics environment. Through autonomously established morphogen gradients and exploitation of physical forces, an organism was evolved which successfully developed into a French flag like structure. Cartesian genetic programming was used in [41] to evolve a program which controlled the actions of cells on a grid. This organism was able to grow into a French flag pattern, and in addition the development program was robust to damage applied to the developing phenotype.

Chapter 3

Method

3.1 Design Choices and Hypotheses

As discussed in 2.1, a computational embryogeny has two primary components - a genetic control mechanism, and a phenotypic mechanism. These were therefore the main design choices required for the artificial development system.

3.1.1 Initial Explorations

Before describing the final choices made, a very brief examination of some early ideas will be presented along with reasons for their departure.

3.1.1.1 Formulation of Gene Regulatory Network as Recurrent ResNet

The first approach trialled implementing a differentiable gene regulatory network, which was based on the work in [59]. The output of the network was associated with an update rule in a cellular automata. This idea was quickly discarded, as the model showed no obvious capacity to be represented by a compact genome .

3.1.1.2 Formulation of Gene Regulatory Network as Boolean Network

The second approach trialled implementing a gene regulatory network based on the boolean network abstraction [51]. Each artificial cell was given a network with N nodes and a function $f : 2^N \rightarrow \{\text{behaviour}_1, \text{behaviour}_2, \cdots\}$ mapping network states to behaviours. Behaviours included cell duplication and death, as well as the release of morphogens. The release of specific morphogens connected specific output nodes of the releasing cells network to the input nodes of a neighbour. In this way, the behaviour of neighbouring cells was linked. This idea was discarded, as the development system was not deemed to be suitably expressive. In addition, the combined solution space of N node networks and state mapping functions f was too large to search using evolutionary methods.

3.1.2 Decisions

In the biological world, proteins are the metaphysical bridge between the static, non-spatial genome and the 3D world. Moreover, it is their shape and physical properties which allow them to interact in unique ways with the environment. In this sense computation takes place in a physical medium which is itself directly altered by the outcomes of the computation. In the context of a computational embryogeny, the previous distinction between the genetic control mechanism and the phenotypic mechanism blurs - they become one and the same. In neuroscience, these type of concepts are referred to as embodied cognition [60].

This work aims to integrate more tightly the genotypic and phenotypic mechanisms within the model of development. It is hoped that this will create a more expressive and compact computational embryogeny. For this reason, it was decided to pursue a computational formulation of genetic control which paid the appropriate level of homage to the inherently spatial nature of protein computation. Fractal proteins were first introduced by Bentley in [8], with further ideas relating to the formation of gene regulatory networks developed in [9] and [7]. These papers described how evolved genes can be expressed into fractal proteins, which can then form themselves into complex and useful gene regulatory networks capable of computation. The key idea was to find a way of generating spatially complex forms which could be easily computed and cheaply represented within computer memory. How this was achieved is explained in Section 3.2.1.

This establishes the first aim of the project. Bentley has demonstrated that evo-

lution can evolve the spatial characteristics of fractal proteins to achieve regulation of gene expression through exploitation of a shape-dependent protein chemistry. This work aims to connect the spatial nature of fractal proteins with biophysical properties of artificial cells through an appropriate model. In addition, it aims to derive further characteristics of the proteins themselves and their interaction with the environment, such as diffusion speed through a cellular substrate.

It is hypothesized that this could bring the following benefits to the development system:

1. The evolvability associated with fractal proteins should be translated into the development system

2. The addition of a phenotype mechanism does not explicitly introduce any more values into the genome to evolve than what already exists to specify the genetic control mechanism, as all the parameters relevant to the phenotype are determined by properties of the proteins themselves. The two mechanisms are therefore embodied within each other.

As discussed in Section 2.2.1, the CP model is a computational model of cells and tissues. When simulated with specific parameter choices, the model demonstrates the emergence of organised structure from a collection of cells with initially random positions and types. Importantly this is purely autonomous, in the sense that Physics is the only driver of the apparently complex self organisation task. Physics acts as an invisible hand, helpfully reorganising the cellular ensemble according to its own energetic preferences.

This suggests that the CP model is a good candidate to model a spatially differentiated organism such as a French flag. In addition, the energetic formulation exposes interpretable cellular parameters which could be integrated into the world of fractal proteins. In this work, both the concentration and shape of fractal proteins will determine various cellular parameters within the CP abstraction. Each cell within the model will be equipped with its own independent gene regulatory network, which will regulate the expression of fractal proteins and consequently the global behaviour of the cellular ensemble.

From an implementation perspective, the CP model is also computationally simple. This means that prototyping is fast, and integration with other code straightforward.

The remainder of this chapter is structured as follows. Section 3.2 introduces the computational components of the fractal protein formulation and their associated regulatory networks. It provides details on the adjustments made in this work to Bentley's original formulations found in [8] [7] [9]. Section 3.3 introduces the computational components of the CP model, including the metropolis-style update

used to simulate its dynamics. Section 3.4.1 describes the computational framework for combining the world of fractal proteins with the CP model, with a focus on the practical details. Section 3.4.2, Section 3.4.4, Section 3.4.5 and Section 3.4.3 all describe how fractal proteins are used to determine various cellular behaviours and biophysical parameters within the CP model.

3.2 Fractal Proteins and Networks

3.2.1 The Mandelbrot Set

The Mandelbrot set can be generated using the recursive formula $z_{n+1} = z_n^2 + c$ where z and c are imaginary numbers. For some values of c the numbers obtained through repeated recursion are unbounded whereas for others they are not. In Mandelbrot's experiments, a threshold of 2 was set. For $c = x + iy$, if after a number of iterations z_n was still bounded by 2, then a black dot was drawn at (x, y). Otherwise, the number of iterations taken to pass 2 was plotted as a colour. Completing this process over a grid of pixels produces the familiar fractal image shown in Figure 3.1.

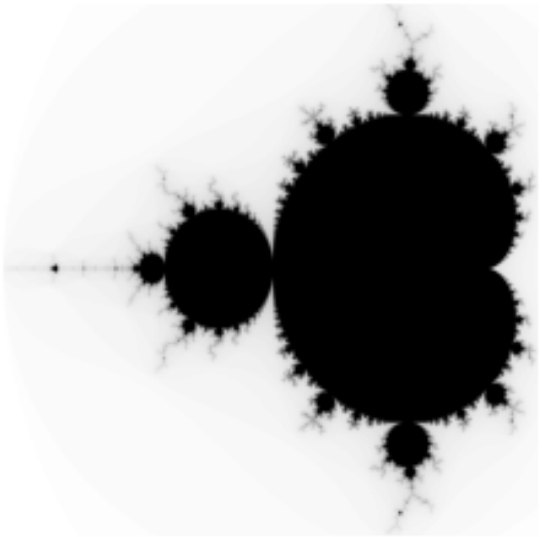

Figure 3.1: The Mandelbrot set

Due to its fractal nature, the Mandelbrot set is an excellent source of algorith-

mic spatial complexity. This motivates the idea of fractal proteins. Suppose we take a square of side length $w \in \mathbb{R}$ and place it's centre at a coordinate (x,y) on the Mandelbrot set, defining a finite region $R_{x,y,w} = [x-\frac{w}{2}, x+\frac{w}{2}] \times [y-\frac{w}{2}, y+\frac{w}{2}]$. By specifying the number of desired pixels $n_x, n_y \in \mathbb{N}$, a fractal image can be obtained by applying the recursion formula to each point $c_n = x_n + iy_n$ where $(x_n, y_n) \in [x-\frac{w}{2n_x}, x+\frac{w}{2n_x}] \times [y-\frac{w}{2n_y}, y+\frac{w}{2n_y}]$. For a fixed resolution (n_x, n_y) the width of the region $w \in \mathbb{R}$ therefore acts as an inverse magnification factor.

A maximum iteration is specified which determines the point at which z_n is considered to be bounded by 2 if it has not yet exceeded it. Assigning the number of iterations to the appropriate entry in a $n_x \times n_y$ array defines a bit array which represents the fractal protein for computational purposes. Assuming the maximum number of iterations is 255, the array can be stored as an unsigned 8-bit integer array. Adopting the typical convention of zero representing black and 255 representing white, a bitmap image such as Figure 3.2 can be produced.

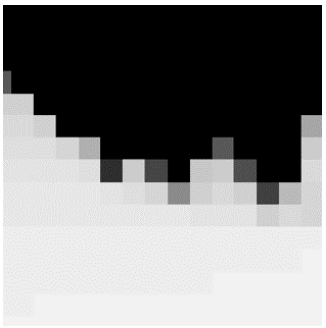

Figure 3.2: An example of a fractal protein bitmap

The resultant fractal protein has an extremely compact representation - it is completely defined by the triple (x,y,w). Furthermore, the geometry of the Mandelbrot set makes the "fractal genetic space" highly evolvable. As noted in [7], a small change to (x,y,w) corresponds to a small change to the fractal protein, and in addition the self-similarity of fractals ensures that any specific shape can be found in an infinite number of places.

3.2.2 Fractal Protein Interactions

One way biological proteins interact is through binding sites. A binding site is a region on a protein that binds to another molecule with specificity. This means that binding is targeted, in the sense that only proteins and molecules with complimentary shapes will successfully bind together. The binding molecule can be another protein - this results in a protein-protein interaction. Protein-protein interactions are

responsible for many biological mechanisms such as signal transduction, muscle contraction and cell metabolism [47].

There are two interactions of interest for fractal proteins and gene regulation. One is cell metabolism using fractal proteins - this is the creation of new fractal proteins using existing ones within a cell. The second is fractal protein binding - this is where a specifically shaped fractal protein is able to successfully attach to a binding site of another. This can then trigger some event of interest.

In [7], Bentley describes cell metabolism using fractal proteins as follows: First the set of reactant proteins have their respective fractal equations iterated through in parallel for a given pixel, stopping as soon as any become unbounded. This then defines a new fractal protein which is said to be the product of the reaction. The protein whose fractal equation became unbounded first can be thought of as the 'winning' reactant protein for a given pixel. The concentration of the reaction product is then calculated as the mean concentrations of all the winning reactant proteins.

In this work the fractal proteins are stored in computer memory as bit arrays, rather than calculated through recursion every time they are to be accessed. The fractal chemistry is therefore reformulated; the winning reactant proteins for a given pixel is simply the protein with the largest pixel value. Intuitively, pale regions win over dark regions. In other words; if $\hat{F} = \{F_1, F_2, \cdots, F_N\}$ are N fractal proteins considered to be reactants in the metabolism reaction, the product F_P is formed by determining each of its array entries as

$$F_P[i,j] = \max_{F \in \hat{F}} F[i,j] \tag{3.1}$$

The method of determining the product concentration remains unchanged from Bentley's method. Later, and in the context of gene regulation, an adjustment will be made to this metabolism process. The final process is made precise in Definition 3.2.2.

For protein binding, suppose there two fractal proteins F_1, F_2, one of which represents a binding site. The following defines a function which measures the similarity in shape between the two proteins:

Definition 3.2.1 (Fractal Protein Similarity Measure)**.** Let F_1, F_2 be two fractal proteins, represented by $n \times n$ 255-bit arrays, $F_1, F_2 \in [0, \cdots, 255]^{n \times n}$. Define the similarity measure to be a function $S : [0, \cdots, 255]^{n \times n} \to \mathbb{R}$ where

$$S(F_1, F_2) = 1 - \frac{1}{255 \times n \times n} \sum_{i,j \in n \times n} |F_1[i,j] - F_2[i,j]| \tag{3.2}$$

For any fractal proteins F_1, F_2 this measure is bound between zero and one, with smaller numbers representing fractal proteins of differing shape and larger numbers representing fractal proteins with similar shape. Whilst successful protein binding usually only occurs between proteins with complimentary shapes, it is still a stochastic process. Therefore, this similarity measure defines a probability of success for the binding event. In [8] Bentley uses a sigmoid activation-like function which consumes a similarity measure as well as two tuning parameters set by the user. In this work I have normalised the similarity measure, and hence its value can be directly associated with a probability. This has the advantage that there are fewer parameters to tune by the user, however it does make the relationship between probability change and shape adjustment linear - proportionate adjustments in the shapes results in proportionate adjustments to the binding probability. Later, and in the context of gene regulation, non-linearity is reintroduced and furthermore it is made evolvable.

3.2.3 Formulation of Fractal Gene Regulatory Networks

The basic dynamics of fractal gene regulatory networks can be described as follows: there are fractal proteins of varying concentration contained within the cytoplasm of an artificial cell. These proteins are either produced by the cell through gene transcription, or they can be received externally from the environment through the cell membrane. At any given moment, these proteins are undergoing a metabolism reaction. The result of this reaction is a protein product, which has the capability to either activate or inhibit further gene transcription, thereby changing the concentration and composition of fractal proteins within the cell. This process continues, causing a constant flux in the proteins found within and then outside the cell.

In order to bring some order to this dynamic protein soup, proteins are given particular functions. Proteins can act as receptors, trigger cellular behaviours, act as regulators, or travel into the cytoplasm from outside the cell. These are referred to as receptor proteins, behavioural proteins, regulatory proteins or environmental proteins respectively. A gene which codes for a regulatory protein is referred to as a regulatory gene, etc. Both receptor, regulatory and environmental proteins partake in the metabolism reaction producing protein products, whereas behavioural proteins do not - these proteins are used to instigate other cellular actions.

In this work an artificial gene contains 9 values - 1 string and 8 positive real values. The string indicates the type of protein the gene is coding for; regulatory, behavioural, receptor or environmental. The first three real values define which fractal protein shape(s) will trigger transcription. This is known as the promoter

protein. The next three values define the shape of the fractal protein translated by the gene - this is known as the coding protein. In both instances, the three values encode a fractal protein shape through the process described in Section 3.2.1. The remaining two values define meta properties of both the promoter and coding proteins.

In this work, the meta property of the coding protein is a minimum activation energy. This is a concentration threshold as in [7], however the mechanism is quite different. In this work, the threshold is a concentration level the coding protein must surpass in order for it to partake in a cell metabolism reaction. This was introduced to increase the variety of the protein products produced by the metabolism reaction. Recall that the protein product F_P is produced by taking the N fractal proteins found within the cell $F_1, F_2, \cdots, F_N$ and constructing the new array using

$$F_P[i,j] = \max_{F \in \hat{F}} F[i,j] \tag{3.3}$$

If the same fractal proteins $F_1, F_2, \cdots, F_N$ are found within the cell at each time step, the shape of the protein product does not change. Whilst Bentley does specify a minimum concentration below which a fractal protein is no longer considered present in the cell, it is not evolved as part of the genome and additionally the value is not specific to each protein. By introducing a minimum activation energy for each coding protein, the set of proteins found within the cytoplasm that are considered within the metabolism reaction can change suddenly depending on the concentration levels. It is anticipated that this might allow more complex behaviour to be evolved.

For the purpose of precision and clarity, the following combines the above discussion into a single definition for the fractal protein cell metabolism process:

Definition 3.2.2 (Fractal cell metabolism)**.** Suppose there are a set of proteins $\hat{F} = \{F_1, F_2, \cdots, F_N\}$ contained within a cells cytoplasm, where the proteins are represented by $n \times n$ 255-bit arrays, $F_a \in [0, \cdots, 255]^{n \times n}$. Suppose also that at a time t protein F_a has a concentration $C(F_a, t) \in \mathbb{R}$, as well as a minimum activation energy $E_a \in \mathbb{R}$. The set of reaction proteins $R_{\hat{F}}(t) \subset \hat{F}$ at time t is given by

$$R_{\hat{F}}(t) = \{F_a \in \hat{F} \text{ s.t. } C(F_a, t) \geq E_a\} \tag{3.4}$$

The reaction product at time t is a protein $F_P = P(t) \in [0, \cdots, 255]^{n \times n}$, whose $F_P[i,j]$ entry is given by

$$F_P[i,j] = \max_{F \in R_{\hat{F}}(t)} F[i,j] \tag{3.5}$$

The reaction product concentration $C(P,t) \in \mathbb{R}$ at time t is given by

$$C(P,t) = \frac{1}{n^2} \sum_{i,j \in n \times n} C(\text{argmax}_{F \in R_{\hat{F}}(t)} F[i,j],t) \tag{3.6}$$

The meta property of the promoter protein represents a minimum binding energy. Given the cellular protein product F_P and the promoter site protein F_{prom}, this is simply the minimum similarity $S(F_P, F_{\text{prom}})$ which must be achieved for a non-zero probability of transcription. The following definition makes this precise:

Definition 3.2.3 (Fractal protein transcription)**.** Let F_P, F_{prom} be two fractal proteins, represented by $n \times n$ 255-bit arrays, $F_P, F_{\text{prom}} \in [0, \cdots, 255]^{n \times n}$. Suppose F_{prom} represents the promoter protein of a particular gene, which if F_P successfully binds to, triggers a transcription event $T_{P \to \text{prom}}$. Suppose also that associated with F_{prom} is a minimum binding energy, $B_{\text{prom}} \in \mathbb{R}$. Then the probability of the transcription event $T_{P \to \text{prom}}$ occurring is given by

$$p_{P \to \text{prom}} = \mathbb{P}(T_{P \to \text{prom}}) = \max(0, S(F_P, F_{\text{prom}}) - B_{\text{prom}}) \tag{3.7}$$

3.2.4 Concentration Dynamics

Once a protein product has been produced through the fractal cell metabolism process defined in Definition 3.2.2, the product is compared against the promoter regions of all genes within the genome in order to determine the probability of a transcription event occurring, as defined in Definition 3.2.3.

If a transcription event does occur for a given gene, the coding protein is created and therefore its concentration within the cell will increase. A transcription event at time t for a given coding protein F_{cod} produces the following amount of the protein:

$$\Delta\text{Trans}(F_{\text{cod}}, t) = \frac{C(F_{\text{prod}}, t)}{C_i} * \tanh\left(\frac{C(F_{\text{prod}}, t) - C_q}{C_w}\right) \tag{3.8}$$

where $C_i, C_w, C_q \in \mathbb{R}$ are parameters set by the user and are problem specific.

If a transcription event does not occur for a given protein, its concentration (if non-zero) will decrease due to natural degradation. The amount of protein which degrades at time t is given by:

$$\Delta\text{Deg}(F_{\text{cod}}, t) = -\frac{C(F_{\text{cod}}, t)}{C_d} \tag{3.9}$$

Putting these together gives the following update rule: Given a protein product F_P, promoter protein F_{prom} with minimum binding energy B_{prom} the concentration of the associated coding protein F_{cod} at time step $t+1$ is given by

$$C(F_{\text{cod}},t+1) = C(F_{\text{cod}},t) + \mathbb{1}_{T_{\text{prod}\to\text{prom}}} * \Delta\text{Trans}(F_{\text{cod}},t) + \Delta\text{Deg}(F_{\text{cod}},t) \quad (3.10)$$

This is a stochastic update rule. One can make it deterministic by taking the expectation of both sides:

$$\begin{aligned}\mathbb{E}[C(F_{\text{cod}},t+1)] &= C(F_{\text{cod}},t) + \mathbb{E}[\mathbb{1}_{T_{\text{prod}\to\text{prom}}}] * \Delta\text{Trans}(F_{\text{cod}},t) + \Delta\text{Deg}(F_{\text{cod}},t) \\ &= C(F_{\text{cod}},t+1) + p_{P\to\text{prom}} * \Delta\text{Trans}(F_{\text{cod}},t) + \Delta\text{Deg}(F_{\text{cod}},t)\end{aligned} \quad (3.11)$$

where $p_{P\to\text{prom}} = \max(0, S(F_P,F_{\text{prom}}) - B_{\text{prom}})$.

At each time step, this update is applied to regulatory, behavioural and environmental genes and their associated coding proteins. The update of receptor protein concentrations within the cell is determined by external signals. This is explored in Section 3.4.6.

As previously mentioned, environmental proteins are permitted to diffuse throughout the environment, which adds a diffusive term to the update rule. This is made precise in 3.4.1, once the cellular environment has been defined.

To summarise, the following processes take place during one time step of development, and represent one cycle of gene regulation:

1. Given the concentrations of receptor, environmental and regulatory proteins, the set of reactant proteins $R_{\hat{F}}(t)$ is determined according to Definition 3.2.2

2. The cell metabolism process creates a new protein product F_P according to Definition 3.2.2

3. The protein product F_P is compared against the promoter region F_{prom} of all genes within the genome, determining a probability of transcription $p_{P\to\text{prom}}$ according to Definition 3.2.3

4. For environmental, regulatory and behavioural genes, an amount of protein F_{cod} specified by the genes coding region is produced according to Equation 3.11

5. For receptor genes, an amount of protein F_{cod} specified by the genes coding region is produced according to an external signal.

6. Receptor, environmental and regulatory proteins are added to the cell cytoplasm for the next round of metabolism.

7. Behavioural proteins trigger changes to the cell.

3.3 The Cellular Potts Model

Computationally, the CP model is a modified version of the Potts model [61] with differential adhesivity. The Potts model is a model from the field of statistical mechanics, and is itself a derivative of the Ising model, a model of ferromagnetism [39]. Common to all of these models is a hamiltonian which defines the dynamics of the system. The hamiltonian is a function which specifies the energy of a particular system state, or configuration.

The hamiltonian for the CP model is defined as follows

$$H = \sum_{\substack{(i,j),(i',j') \\ \text{neighbors}}} J\Big[\tau(\sigma(i,j)),\tau(\sigma(i',j'))\Big]\Big[1-\delta_{\sigma(i,j),\sigma(i',j')}\Big] + \lambda \sum_{\substack{\text{cells} \\ \sigma}} \Big[a(\sigma)-A_\sigma\Big]^2 \mathbb{1}(A_\sigma > 0) \tag{3.12}$$

Lets breakdown each component. The first summation is over all neighbouring sites on a finite 2-dimensional lattice $L \subset \mathbb{Z}^2$. Each lattice site $(i,j) \in L$ is given a cell identity - $\sigma(i,j)$. This indicates which larger cell the site belongs to, the lattice therefore representing a tissue or organism consisting of far fewer cells than the total number of lattice sites. The number of lattice sites making up a given cell σ is given by $a(\sigma)$, and represents the current area or volume of the cell. Each cell has a target volume - this is given by A_σ. In addition, each cell has an associated cell type, given by $\tau(\sigma)$. For two cell types τ, τ', the differential adhesivity between them is given by $J[\tau,\tau']$.

Given that the aim is to find a configuration which minimises this quantity, we can see that the requirements for cell sorting are captured through the hamiltonian. The first summation is effectively a sum over the boundary lines of all differing cell types. The contribution to the hamiltonian is the differential adhesivity given by J - cells of large differential adhesivity are therefore unlikely to be found next to each other. Furthermore, it is energetically advantageous for cells of the same type to clump together, so as to minimise the length of the boundary lines between differing cell types. Lastly, due to the second summation cells will attempt to maintain their target volumes. The degree to which they do this (the elasticity of the cell boundaries) is determined by λ.

The dynamics and evolution of a system governed by this hamiltonian can be simulated through a metropolis-style update, which is a Monte-Carlo algorithm. To simulate the system for one unit of time, a number of lattice sites are selected at random and their cell identities momentarily altered to those of a neighbouring lattice site. This change of state induces a corresponding change in energy, given by the hamiltonian. If the difference ΔE is negative, i.e. a configuration with a lower global energy has been found, then the new configuration is adopted as the system state at time $t+1$. Otherwise, the changes are discarded with a probability $1 - e^{-\frac{\Delta E}{T}}$. The temperature T adjusts the likelihood that the system state will move to a configuration of higher energy, and hence can be tuned so that the dynamics avoid local minima in the energy landscape. Further details relating to the implementation of the basic CP model and parameter choices can be found in [28].

It is possible to add other terms to the hamiltonian in order to incorporate other cell behaviours into the CPM dynamics [5]. This often includes the addition of further 'sub-lattices' to represent other types of spatial information within the growth environment, such as chemical gradients. An extension used in this work is the addition of chemotaxis - the direction of movements by an organism in response to certain chemicals within the environment. This behaviour is incorporated into the CP model by introducing a lattice on which a chemokine diffuses, and increasing the likelihood that a lattice site $(i, j) \in L$ will be changed to its neighbour $(i', j') \in L$ if the chemokine concentration is higher at (i, j). This is done by modifying the change in energy ΔE computed in the metropolis-style update as follows [49]

$$\Delta E' = \Delta E - \mu(C_{i',j'} - C_{i,j}) \tag{3.13}$$

where $C_{i,j} \in \mathbb{R}$ is the chemokine concentration at lattice site $(i, j) \in L$.

3.4 Unifying Fractal Proteins and the Cellular Potts Model

The artificial development model in this work aims to integrate fractal proteins and gene regulatory networks into the CP framework. To begin, it must be decided how the proteins and their corresponding concentrations are associated with a cell within the CP model.

3.4.1 The Nuts and Bolts

Recall that the fundamental components of the basic CP model are a lattice $L \subset \mathbb{Z}^2$ representing a spatial environment and two functions, the cell identity function $\sigma : L \to \{1, \cdots, N_{\text{cells}}\}$ and the cell type function $\tau : \{1, \cdots, N_{\text{cells}}\} \to \{1, \cdots, N_{\text{types}}\}$. In addition, there is a matrix containing the differential adhesivity values between cell types, $J \in \mathbb{R}^{N_{\text{types}} \times N_{\text{types}}}$, and a hamiltonian $H : L \to \mathbb{R}$ which describes the dynamics of the model.

Consider a hypothetical cell $k \in \{1, \cdots, N_{\text{cells}}\}$ which exists within the environment $L \subset \mathbb{Z}^2$. The subset of the lattice which is considered to 'belong' to cell k is the set $\{(i, j) \in L \text{ s.t. } \sigma(i, j) = k\}$. Biologically this could be considered the cells interior, and for simplicity we refer to this as the cells cytoplasm.

Definition 3.4.1. Let (L, σ, τ, J, H) define a basic CPM. The cytoplasm of cell $k \in \{1, \cdots, N_{\text{cells}}\}$ is the subset

$$\text{cyto}_k = \{(i, j) \in L \text{ s.t. } \sigma(i, j) = k\} \tag{3.14}$$

Now we are in a position to integrate fractal proteins into the CP model. Suppose $\hat{F} = \{F_1, F_2, \cdots, F_N\}$ is a set of fractal proteins which are found in cells and the environment L. We assume that at each point $(i, j) \in L$ in the environment there is a concentration of each fractal protein (which can of course be zero). The concentration of protein F_a at site $(i, j) \in L$ at a time t is denoted $C_{i,j}(F_a, t)$. This then motivates the following definition:

Definition 3.4.2. Let (L, σ, τ, J, H) define a basic CP model, and $\hat{F} = \{F_1, F_2, \cdots, F_N\}$ be a set of fractal proteins which are found in the environment L. Suppose cyto_k is the cytoplasm of cell $k \in \{1, \cdots, N_{\text{cells}}\}$. The total concentration of fractal protein F_a found within cyto_k at time t is

$$C_k(F_a, t) = \sum_{(i,j) \in \text{cyto}_k} C_{i,j}(F_a, t) \tag{3.15}$$

Each cell within the CP model completes its own independent gene regulation process, based on the proteins found within its cytoplasm. Updates to protein concentrations are computed for each cell synchronously - once the concentrations of the fractal proteins associated with a cell have been aggregated according to Definition 3.4.2 and the resultant protein product from the metabolism process has been determined, the concentrations are updated according to Equation 3.11.

One aspect that must be defined is how the production of a protein within a cell is redistributed across the lattice sites of the CPM. The concentration update rule at

time t for a given fractal protein F_a with promoter F_{prom} and metabolism product F_{prod} is the following:

$$C(F_{\text{a}},t+1) = C(F_{\text{a}},t) + \Delta C(F_{\text{a}},t) \tag{3.16}$$

where $\Delta_{\text{cell}} C(F_{\text{a}},t) = \mathbb{1}_{T_{\text{prod}\to\text{prom}}} * \Delta\text{Trans}(F_{\text{a}},t) + \Delta\text{Deg}(F_{\text{a}},t)$. Using a slight abuse of notation one can therefore write

$$C_k(F_{\text{cod}},t+1) = C_k(F_{\text{cod}},t) + \Delta C_k(F_{\text{cod}},t) \tag{3.17}$$

to represent the update to cell k within our CP model. This update is then distributed across the lattice sites L as follows:

For each $(i,j) \in \text{cyto}_k \subset L$,

$$C_{i,j}(F_a,t) = C_{i,j}(F_a,t) + \frac{\Delta C_k(F_{\text{cod}},t)}{|\text{cyto}_k|} \tag{3.18}$$

This is very simple; concentration updates due to gene regulatory processes within a cell are distributed across its cytoplasm area uniformly and instantly.

3.4.1.1 Diffusion of environmental proteins

Environmental fractal proteins have the ability to travel across cellular boundaries. Therefore, environmental proteins can be absorbed into one cells cytoplasm after being produced within the cytoplasm of a neighbour. Practically, this is achieved by introducing a diffusion process on the lattice L. This diffusion is modelled using a finite difference method, where the lattice size used in the CP model defines the discretization scheme.

The finite difference method used is the forward Euler scheme. Assuming the cellular environment was continuous, the diffusion of a quantity $C(x,y,t)$ could be expressed through the famous heat equation:

$$\frac{\partial C}{\partial t} = \alpha\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}\right) \tag{3.19}$$

where $\alpha \in \mathbb{R}$ is the diffusion coefficient. This reduces to the following discrete update given the forward Euler scheme:

$$\frac{C_{i,j}(t+1)-C_{i,j}(t)}{\Delta t}=\alpha\left[\frac{C_{i+1,j}(t)-2C_{i,j}(t)+C_{i-1,j}(t)}{(\Delta x)^2}+\frac{C_{i,j+1}(t)-2C_{i,j}(t)+C_{i,j-1}(t)}{(\Delta y)^2}\right] \tag{3.20}$$

and hence the state of the system at time step $t+1$ can be calculated as

$$C_{i,j}(t+1)=C_{i,j}(t)+\alpha_e\Delta t\left[\frac{C_{i+1,j}(t)-2C_{i,j}(t)+C_{i-1,j}(t)}{(\Delta x)^2}+\frac{C_{i,j+1}(t)-2C_{i,j}(t)+C_{i,j-1}(t)}{(\Delta y)^2}\right] \tag{3.21}$$

For our purposes, it is enough to let $(i,j)\in L$ be the CP lattice sites, and set $\Delta x=\Delta y=\Delta t=1$. Then, so long that $\alpha\leq\frac{1}{4}$ the numerical scheme is stable. To make things more concise, let $D_x\cdot C_{i,j}(t)=C_{i+1,j}(t)-2C_{i,j}(t)+C_{i-1,j}(t)$ and $D_y\cdot C_{i,j}(t)=C_{i,j+1}(t)-2C_{i,j}(t)+C_{i,j-1}(t)$. Then our update can be written

$$C_{i,j}(t+1)=C_{i,j}(t)+\alpha\left[D_x\cdot C_{i,j}(t)+D_y\cdot C_{i,j}(t)\right] \tag{3.22}$$

Given an environmental fractal protein F_e we add this diffusion term to the usual update rule, which then gives us the following:

For each $(i,j)\in \text{cyto}_k\subset L$,

$$C_{i,j}(F_e,t)=C_{i,j}(F_e,t)+\frac{\Delta C_k(F_e,t)}{|\text{cyto}_k|}+\alpha_e\left[D_x\cdot C_{i,j}(F_e,t)+D_y\cdot C_{i,j}(F_e,t)\right] \tag{3.23}$$

The complete dynamics of environmental proteins can therefore be considered as a reaction-diffusion system, much like morphogens responsible for pattern formation [56].

In addition, the reaction and diffusion processes operate at two distinct spatial and temporal scales; the reaction (update due to transcription) results in an instant and uniform update to all lattice sites of the relevant cell. On the other hand, the diffusion process acts at the scale of the individual lattice site, and is agnostic to cell identity.

3.4.2 Fractal Induced Intercellular Communication

The ability for environmental proteins to move across cell boundaries allows the actions of one cell to be felt by another. More precisely, transcription events in one cell which lead to an increase in the concentration of a given environmental protein

will, after some time, increase the concentration of the same environmental protein in a neighbouring cell's cytoplasm. This in turn could alter the protein product resulting from the metabolism reaction, which could lead to alternate transcription events and subsequent cell behaviour. This type of diffusion therefore constitutes a form of inter-cellular communication.

An important quantity which dictates the speed of diffusion is the constant $\alpha_e \in \mathbb{R}$. This will determine how fast or slow the environmental protein F_e can move across the lattice.

In general, diffusive constants are a function of both the substance and the medium through which it diffuses. In this work, it is imagined that the medium through which proteins diffuse is a type of 'cellular substrate'. This medium contains a specific fractal protein F_{sub} in which all entries are zero, i.e. $F_{\text{sub}}[i, j] = 0$.

The speed at which a environmental protein F_e diffuses through the cellular substrate is then assumed to be its similarity in shape to F_{sub}. As a result, the diffusion constant for F_e is defined as

$$\alpha_e = \frac{1}{4} \cdot S(F_e, F_{\text{sub}}) \tag{3.24}$$

where $S(\cdot,\cdot)$ is the similarity measure defined in Definition 3.2.1. Since the similarity measure is bounded between zero and one for any two proteins, scaling by $\frac{1}{4}$ ensures that the numerical stability condition for the forward Euler method is satisfied.

This formulation introduces pleiotropy into the genome; the gene encoding the shape of the environmental protein determines both its role within cell metabolism and the speed at which it can diffuse through the cellular environment.

3.4.3 Fractal Induced Mitosis, Apoptosis and Differentiation

3.4.3.1 Mitosis and Apoptosis

In biological development, the interplay between the cell cycle and cell differentiation is incredibly complex. Different relationships between cell cycle and differentiation can be combined in a succession of events during development, leading to complex developmental programs [30].

At its simplest level, the cell cycle refers to the process of cell division (mitosis) and ultimately cell death (apoptosis). In this work, mitosis occurs once a cell has achieved a certain size. Growth of a cell is determined by the concentration of a

specific behavioural protein within its cytoplasm - this behavioural protein therefore controls mitosis events. First, recall the hamiltonian of the basic CP model:

$$H = \sum_{\substack{(i,j),(i',j') \\ \text{neighbors}}} J\Big[\tau(\sigma(i,j)),\tau(\sigma(i',j'))\Big]\Big[1-\delta_{\sigma(i,j),\sigma(i',j')}\Big] + \lambda \sum_{\substack{\text{cells} \\ k}} \Big[a(k)-A_k\Big]^2 \mathbb{1}(A_k>0) \tag{3.25}$$

Here, each cell k has a target area $A_k \in \mathbb{R}$. This is compared against $a(k) \in \mathbb{R}$, the actual size of the cell on the lattice, and large deviations are energetically penalised according to the magnitude of the elasticity parameter $\lambda \in \mathbb{R}$. This results in cells growing towards their target size A_k as the simulation progresses. In this work, the target size A_k is determined by the concentration of a specific behavioural protein, and therefore changes through time. This also makes the Hamiltonian time dependent, and is used in the natural way in the metropolis-style update - the energy of the system at update time t is evaluated using the hamiltonian $H(t)$, where

$$H(t) = \sum_{\substack{(i,j),(i',j') \\ \text{neighbors}}} J\Big[\tau(\sigma(i,j)),\tau(\sigma(i',j'))\Big]\Big[1-\delta_{\sigma(i,j),\sigma(i',j')}\Big] + \lambda \sum_{\substack{\text{cells} \\ k}} \Big[a(k)-A_k(t)\Big]^2 \mathbb{1}(A_k(t)>0) \tag{3.26}$$

How is $A_k(t)$ set practically? A user-defined parameter referred to as the 'cell size limit' is first set. This establishes the maximum number of lattice sites a cell can occupy on the lattice, and is chosen depending on the lattice size and computational restrictions (a smaller number will result in more cells). At time t of the organisms development, the instantaneous concentration of the behavioural protein scales this value up or down to create the growth target.

Suppose F_g is the behavioural protein responsible for determining cell size and $C_k(F_g,t)$ its concentration in cell k. Also let $\text{size}_{\max} \in \mathbb{R}$ be the user specified cell size limit, and $\beta \in \mathbb{R}$ another user set parameter. Then

$$A_k(t) = \beta * C_k(F_g,t) * \text{size}_{\max} \tag{3.27}$$

This sets the scene for the next mechanism - cellular division. Once a cell has grown to a size greater than the cell size limit $\text{size}_{\max}$, it undergoes mitosis. At this point the cell splits into two new cells, with the corresponding cellular proteins split between them. It is clear now that the parameter β introduces an upward pressure on the propensity for a cell to grow and ultimately multiply. This parameter is tuned according to the details of the experiment, and in this work was set equal to two.

Cell division in the context of the CP model is a matter of reassigning the cell identities of lattice sites. As a result, the proteins are split very naturally - the concentrations at each lattice site remains the same, however the cell cytoplasm to which they are associated with changes. In order to split a cell on the lattice an axis is determined - lattice sites on one side of this axis are then reassigned a new cell identity.

The cell is split along the axis of greatest variation associated with the shape of the cell. The axis of greatest variation can be thought of as the principle components of the 2D cell shape. Computationally it is determined using the following method:

1. Calculate the centre of mass of all the lattice points $(i, j) \in L$ belonging to the cell k, i.e. cyto_k
2. Subtract the centre of mass (i_m, j_m) component wise from the lattice points $(i, j) \in \text{cyto}_k$.
3. Calculate the covariance matrix associated with the matrix of mean subtracted lattice points.
4. Determine the eigenvector associated with the largest eigenvalue. This is the axis of greatest variation.

This follows closely the calculation of principle components in data analysis using the covariance method [53]. Code associated with its implementation can be found in Appendix A.2.1.

Supposing that the cell has a method of "elongating" along a particular axis, this mechanism provides the cell with a method of multiplying in a particular direction - it will be in the direction normal to the elongation axis.

In addition to cell mitosis, there is apoptosis - cellular death. In this work, cell death is simply programmed to occur after a certain number of mitosis events. The number of mitosis events is specified by the user - in these experiments it was set at two. Its value controls the proliferation of cells, and was set according to both the maximum cell size and the size of the lattice.

This section has so far explained how the concentration of a single behavioural protein F_g controls a cells growth, subsequent mitosis events and eventual death. Finally, we establish how a cell undergoes type differentiation.

3.4.3.2 Type Differentiation

The type of a cell is determined by the concentration of two specific behavioural proteins F_{b1} and F_{b2} found within the cell cytoplasm. Since behavioural proteins

do not partake in the cell metabolism reaction, the minimum activation energies E_{b1} and E_{b1} are not used to establish membership to the set of metabolism reaction proteins $R_{\hat{F}(t)}$ as is the case with regulatory, environmental and receptor proteins. Instead they are given a new interpretation, which is inspired by the colouration of flowers.

The colouration of flowers depends on the production of the appropriate anthocyanin pigments. Anthocyanin pigments are biosynthesized through specific chemical reactions catalysed by structural enzymes - the genes encoding structural enzymes can hence be considered to be the effector genes for this biochemical pathway [26]. For the purpose of defining a cell differentiation mechanism, our behavioural proteins are therefore thought of as enzymes, and the minimum activation energies specify the minimum amount of enzyme needed to catalyse the appropriate reaction in the cell. Once this threshold is reached, a certain anthocyanin pigment is successfully biosynthesized. It is then imagined that each different combination of the two pigments produces a different type of coloured cell. Table 3.1 enumerates the possibilities and the associated cell types.

State	$C(F_{b1},t) > E_{b1}$: Pigment A	$C(F_{b1},t) \leq E_{b1}$: Pigment B
$C(F_{b2},t) > E_{b2}$: Pigment C	green	blue
$C(F_{b2},t) \leq E_{b2}$: Pigment D	white	red

Table 3.1: Specification of cell type through behavioural protein concentrations

Since the minimum activation energies E_{b1} and E_{b1} are encoded by the gene, evolution has control over how much behavioural protein is required before a pigment is synthesized.

3.4.4 Fractal Induced Adhesion

In the CP model, different levels of adhesion exist between different cell types. Within the hamiltonian

$$H(t) = \sum_{\substack{(i,j),(i',j') \\ \text{neighbors}}} J\Big[\tau(\sigma(i,j)),\tau(\sigma(i',j'))\Big]\Big[1-\delta_{\sigma(i,j),\sigma(i',j')}\Big] + \lambda \sum_{\substack{\text{cells} \\ k}} \Big[a(k)-A_k(t)\Big]^2 \mathbb{1}(A_k(t)>0), \tag{3.28}$$

the matrix $J[\cdot,\cdot]$ encodes surface energies and consequently the level of adhesion between cell types.

Similar cell types are expected to adhere strongly to each other. The aim of fractal induced adhesion is to determine the surface energy between two cell types using a type of fractal chemistry.

As described in Section 3.4.3.2, the type of a cell is determined by the concentration of two specific behavioural proteins F_{b1} and F_{b2} that are found within the cell cytoplasm. The differentiation mechanism is unaffected by the shape of the behavioural protein - it is simply the concentration and the associated minimum activation energy. Therefore, the shape of the behavioural proteins F_{b1} and F_{b2} are used to compute surface energies between cells of differing type.

This is done as follows - first, imagine that the two behavioural proteins react in a way similar to the cell metabolism reaction, and produce a product as defined in 3.2.2. This gives us 4 four possible combinations of reactant proteins $R_{\hat{F}}$ and four possible protein products $P_{(\dots)}$. These are enumerated in 3.2.

State	$C(F_{t1},t) > E_{t1}$	$C(F_{t1},t) \leq E_{t1}$
$C(F_{t2},t) > E_{t2}$	$P_{\text{green}} : R_{\hat{F}} = \{F_{t1}, F_{t2}\}$	$P_{\text{blue}} : R_{\hat{F}} = \{F_{t2}\}$
$C(F_{t2},t) \leq E_{t2}$	$P_{\text{white}} : R_{\hat{F}} = \{F_{t1}\}$	$P_{\text{red}} : R_{\hat{F}} = \emptyset$

Table 3.2: Protein metabolism products associated with type-defining behavioural proteins

Note that the product P_{red} associated with the empty set of reactant proteins $R_{\hat{F}} = \emptyset$ is associated with the empty protein product i.e. $P_{\text{red}}[i,j] = 0$. The others are computed according to the fractal chemistry. Once these four proteins have been determined, the surface energy between two cell types is defined to be the similarity between their associated products, and is calculated using the fractal similarity measure $S(\cdot,\cdot)$. With a slight abuse of notation one can therefore write

$$J\Big[\text{Red}, \text{White}\Big] = 1 - S(P_{\text{Red}}, P_{\text{White}}) \tag{3.29}$$

The similarity is subtracted from one so that cells of the same type have an energy of zero. This means that cell types with very different associated protein products have an energy close to one. In the CP model it is necessary to define the surface energy of each cell type with the surrounding 'empty' medium. This is set equal to one for all cell types.

3.4.5 Fractal Induced Chemotaxis

Chemotaxis can be incorporated into the CP model by introducing a lattice on which a chemokine diffuses, and increasing the likelihood that a lattice site $(i,j) \in L$ will

be changed to its neighbour $(i', j') \in L$ if the chemokine concentration is higher at (i, j). This is done by modifying the change in energy ΔE computed in the metropolis-style update according to:

$$\Delta E' = \Delta E - \mu(C_{i',j'} - C_{i,j}) \tag{3.30}$$

In this work environmental proteins diffuse across the lattice. By specifying one such protein to act as a chemokine, chemotaxis can be added to the list of possible cell behaviours. The environmental protein representing the chemokine is denoted F_c. Therefore, the modification to the hamiltonian energy at time t is

$$\Delta E' = \Delta E - \mu\Big(C_{i',j'}(F_c,t) - C_{i,j}(F_c,t)\Big) \tag{3.31}$$

The chemotaxis strength μ is then determined as follows. First, a receptor protein is selected to act as the chemotaxis controller. The promoter site of the receptor protein R_{prom} binds specifically with the chemokine - currently transcription events for receptor proteins are not triggered through comparison of the receptor promoter site with the metabolism product, as they are with regulatory, behavioural and environmental proteins. They are therefore free to be used for another purpose. μ is then defined as the similarity between the chemokine and the promoter site,

$$\mu = \beta_{\text{chemo}} * S(R_{\text{prom}}, F_c) \tag{3.32}$$

where β_{chemo} is a user-specified parameter. Through this formulation, evolution can adjust the level of chemotaxis the organism exhibits.

3.4.6 Knowing Ones Place

In this model of artificial development, it is imagined that type differentiation gives cells the ability to interact with the environment in a unique way. In other words, a cells specialisation provides it with a specific and unique external signal. The signal manifests itself through the concentration of a receptor protein, which is therefore made specific to the cell type.

For the four possible cell types (blue, white, red and green) there are four associated receptor proteins. If a cell is red, the red receptor protein has its concentration raised to a value proportional to the similarity between the receptors promoter site and the substrate protein F_{sub}. If in the next time step the cell type changes to blue,

the cell no longer contains any of the red receptor protein - instead the concentration of the blue receptor protein would increase, and so on.

In this work, two different types of signal are made available to the cell as it develops. Both are specific to cell type in the sense described above, however one is also made specific to the cells location and is designed to make development of a French flag form easier.

Given a CP lattice $L \subset \mathbb{Z}^2$, a perfect French flag would involve the leftmost third of L having lattice sites belonging to cells with a blue cell type, the middle third having having lattice sites belonging to cells with a white cell type and the rightmost third having lattice sites belonging to cells with a red cell type. Such a lattice is shown in Figure 3.3.

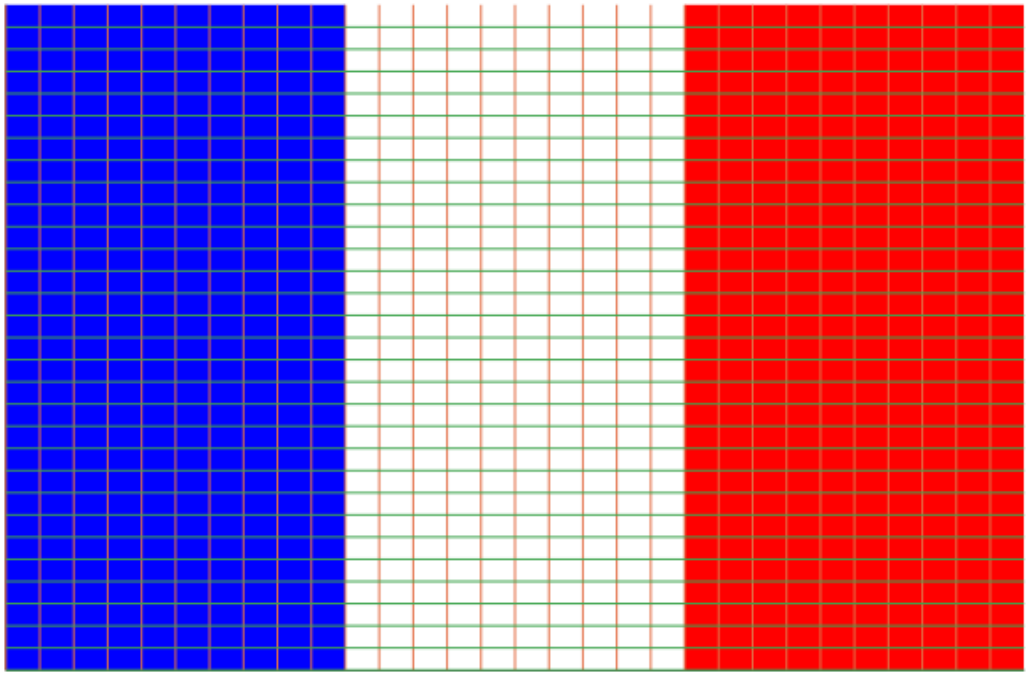

Figure 3.3: A French flag lattice

In the following definitions, we denote the lattice sites belonging to the blue, white and red thirds as $L_{\text{blue}}, L_{\text{white}}, L_{\text{red}} \subset L$ respectively.

The basic idea behind the location-specific signal is to give cells an indication that they are in the correct position and of the correct cell type for the French flag morphology. It should be stressed that the signal does not tell the cell *where* it is on the lattice, or *what* cell type it is expected to be if it was in that position. Rather, if the cell has a blue cell type and its location is in L_{blue}, a blue-specific receptor protein within its cytoplasm has its concentration increased. A red cell within L_{blue} would not receive any signal, and similarly neither would a blue cell within L_{red}.

Therefore, a cell must have already *achieved* the correct position and type in order to receive any signal that indicates it has done so.

Let k be a cell, and let $\tau(k) \in \{\text{blue}, \text{white}, \text{red}, \text{green}\}$ be its cell type. Let $F_{\tau(k)}$ be the protein coded by the appropriate receptor's coding region, and let $R_{\tau(k)}$ be the protein coded by the receptor promoter region. Let F_{sub} be the empty substrate protein and $S(\cdot,\cdot)$ the protein similarity measure. Then the **location specific signal** at time t for cell k is as follows:

For $(i,j) \in \text{cyto}_k$,

$$C_{i,j}(F_{\tau(k)},t) = \frac{S(R_{\tau(k)},F_{\text{sub}})}{|\text{cyto}_k|} \cdot \mathbb{1}((i,j) \in L_{\tau(k)}) \tag{3.33}$$

In contrast, the cell-type only signal does not require the cell to be in the correct position for its cell type - it simply increases the concentration of the relevant receptor protein regardless of position. Therefore the **cell-type only signal** at time t for cell k is as follows:

For $(i,j) \in \text{cyto}_k$,

$$C_{i,j}(F_{\tau(k)},t) = \frac{S(R_{\tau(k)},F_{\text{sub}})}{|\text{cyto}_k|} \tag{3.34}$$

To summarise, the cell-type only signal provides the cell with implicit information about its own cell type, whereas the location-specific signal provides the cell with information about both its own cell type and its position within the cellular assembly, conditional on its cell type and position being aligned with the French flag morphology. Examples of the location-specific signal at development times $t = 400, 800$ are shown in Figure 3.4 and Figure 3.5.

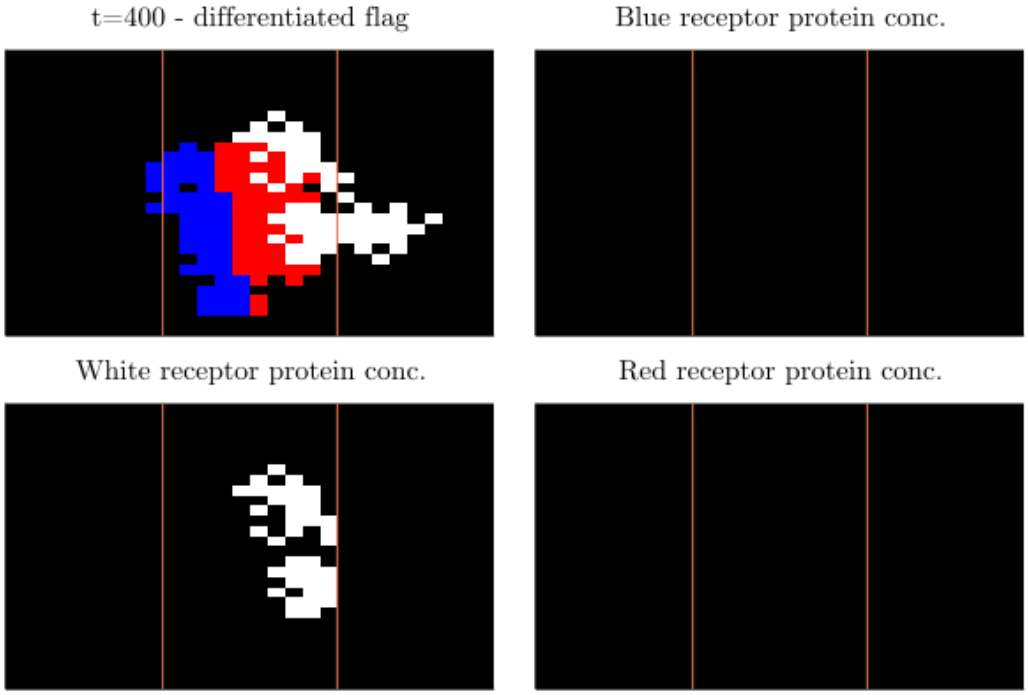

Figure 3.4: Evolution of French Flag receptor proteins : t=400

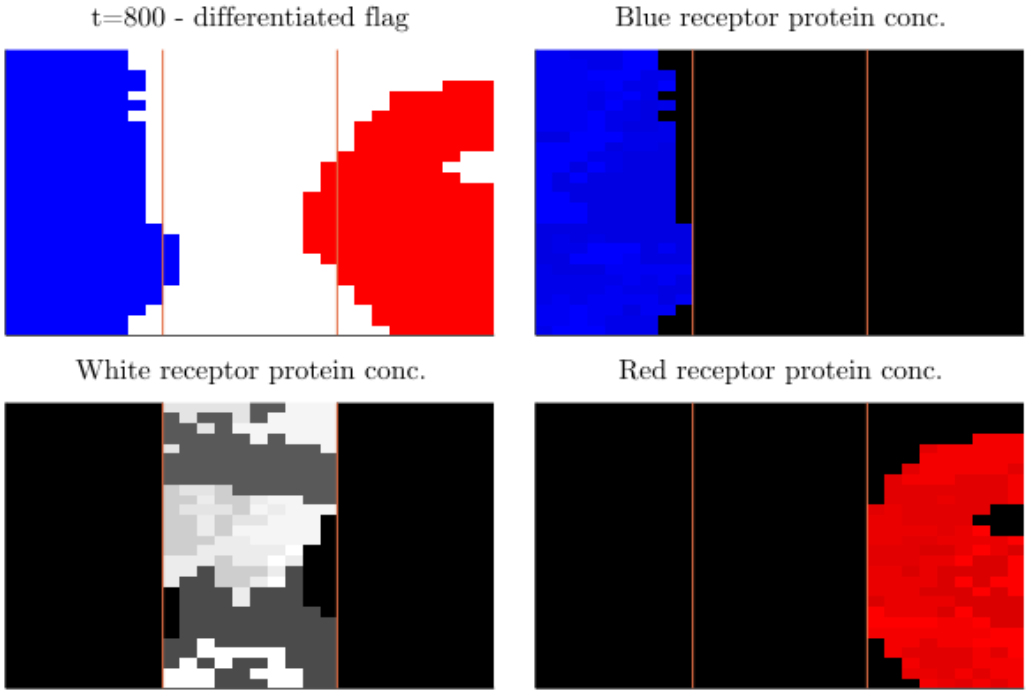

Figure 3.5: Evolution of French Flag receptor proteins : t=800

3.5 Evolving Fractal Gene Regulatory Networks

Table 3.3 represents a single gene with each of its components.

Protein Type	Promoter Region	Coding Region	Minimum activation energy	Minimum binding energy
$\text{type}_a \in \{\text{reg}, \text{rec}, \text{beh}, \text{env}\}$	$(\text{xp}, \text{yp}, \text{wp})_a \in \mathbb{R}^3$	$(\text{xc}, \text{yc}, \text{wc})_a \in \mathbb{R}^3$	$E_a \in \mathbb{R}$	$B_a \in \mathbb{R}$

Table 3.3: Gene components

In evolutionary experiments the number of genes within a genome is set at an initial value, however the genomes are variable length - evolution is free to increase or decrease the number of genes. In the following section, various genetic operators are defined which will aid evolutionary search.

3.5.1 Genetic Operators

Given two parent genomes, the crossover operator simply performs uniform crossover at the level of the genetic components, i.e. protein type, promoter region, coding region, minimum activation energy and minimum binding energy. This follows Bentley in [8]. Since genomes are variable length, it is possible that two parent genomes will have different lengths. In these cases uniform crossover is applied at genes common to both genomes, and then the genes which exist only in the longer genome are simply copied into the new child genome.

Mutation is more interesting. In this work, the following mutation operators are used:

1. **Gene duplication** - a random gene is selected, and is duplicated within the genome. This creates redundancy, and furthermore allows the number of genes within a genome to grow.

2. **Gene deletion** - a random gene is selected and deleted. This allows shorter and more compact genomes to be evolved.

3. **Gene creep** - a random gene is selected. Next, one of the 8 real values is selected and adjusted using a normally distributed number.

4. **Gene copy** - Two genes are selected at random. Subsequently, the promoter site of one protein is copied into the coding site of the other. This mutation 'links' genes within the regulatory network, in the sense that the promoter

region of one gene now has perfect similarity to the protein coded by another gene.

5. **Gene copy (environmental)** - A gene is selected at random, and then an environmental gene is also randomly selected. The coding site of the environmental gene is then copied into the promoter site of the other gene. Similar to above this links genes within the regulatory network, however it explicitly increases the sensitivity of gene transcription to environmental proteins which are diffusing through the environment.

6. **Gene copy (receptor-environmental)** - Both a receptor gene and an environmental gene are randomly selected. The coding site of the receptor gene is then copied into the promoter site of the environmental gene. This explicitly increases the sensitivity of environmental protein transcription to the concentration of receptor proteins.

7. **Gene type mutation** - A gene is selected at random and its protein type is changed.

3.6 A Flag Taxonomy

Fitness functions in nature are often relatively abstract from the biological details of the solution. For example, flowers with bright and contrasting colours help to attract pollinators and hence contribute to the reproductive success of the species. The 'fitness function' for a given flower pattern is therefore 'the extent to which pollinators are attracted'. This is disconnected from the precise levels of anthocyanin pigments found within the petals, in the sense that two flower species may be equally attractive to the pollinator and yet their patterns and colours could result from two completely different types of pigment. This suggests that the features of flowers deemed attractive are perhaps better understood through the eyes of a bee - in evolution beauty is definitely in the eye of the beholder. In addition, aspects of the phenotype can offer a reproductive advantage through multiple mechanisms. In [48], Rudall suggests the following:

> The red colours present in the seed-cone bracts of some living conifers result from accumulation of anthocyanin pigments; their likely primary function is to help protect the growing plant tissues under particular environmental conditions. Thus, the visual cue provided by colour in flower petals could have first evolved as a secondary effect, probably post-dating the evolution of bee colour vision but occurring

before the subsequent functional accumulation of a range of different flower pigments.

This suggests that not only is beauty in the eye of the beholder, there are also multiple beholders! Returning to the French flag problem, this discussion suggests the following - it might be fruitful to firstly try and abstract the fundamental features which define the French flag as a form, and then design a flexible fitness function which can recognise them. In addition, we might want to design a function which encourages evolution to explore multiple objectives, the result of which might be the discovery of a French flag pattern as a 'secondary effect'. These ideas motivate the formulation of a flexible fitness function and new genetic algorithm for the search of flag-like organisms.

The new fitness function works as follows: It firstly counts the number of unique cell types there are in the organism - this consequently defines the number of distinct colours, n_{color}. This then defines an objective - produce n_{color} equal size vertical stripes. Any ordering has equal fitness - in the case of the French flag colours, blue-white-red has equal fitness to white-red-blue, etc. This is achieved practically by evaluating the fitness considering all $n_{\text{color}}!$ orderings and returning the maximum. Furthermore, it is colour agnostic - blue-white-red has the same fitness as green-white-red. Therefore, we could also find Italian flags (and we wouldn't favour them over the French)! Implementation code for this fitness function can be found in Appendix A.2.2.

Cells in the development system have four possible cell types, and hence an organism can have up to four distinct colours in its final form. This suggests the following 'families':

1. 1-colour organisms
2. 2-colour organisms
3. 3-colour organisms
4. 4-colour organisms

Within each of these families, there are species defined by the exact colours in use. For example, the $\binom{4}{3}$ possible species within the 3-colour family are

1. Blue,Red,White (BRW)
2. Green,White,Red (GWR)
3. Green,Blue,White (GBW)
4. Green,Red,Blue (GRB)

Considering all four families, there are $2^4 - 1 = 15$ possible species within the class of organisms under study. This defines a natural 'flag taxonomy'.

The fitness function can evaluate all species. Arguably however, the fitnesses between families should not be compared - a maximum fitness organism from the 1-colour family has accomplished a considerably less impressive feat than a maximum fitness organism from the 4-colour family. However, the fitnesses of species within a single family can be compared - from our point of view, the development of an Italian Flag would be equally as impressive as the development of a French Flag. Therefore from an evolutionary perspective, organisms from the French flag species should not compete with individuals from the Italian flag species - as a result of their phenotype, they occupy different environmental niches.

In the following section, a novelty-search algorithm is developed which exploits this flag taxonomy and flexible fitness evaluation.

3.7 MAP-Flags

The MAP-Elites genetic algorithm is designed to deliver a large set of diverse, high-performing individuals, embedded in a map that describes where they are located in the feature space [43]. This map is a collection of 'cells' - regions of interest predetermined within the feature space. This map is decided by the user through choosing N dimensions of interest within the solution feature space, and then discretizing each of these dimensions into bins. The example given in [43] is robot morphologies - suppose the dimensions of interest to the user are robot height, weight and energy consumption per metre moved. These dimensions could be discretized into bins - a simple approach would be Height $\in \mathbb{R} \rightarrow$ [Tall, Short], Weight $\in \mathbb{R} \rightarrow$ [Heavy, Lightweight], Energy Consumption per m $\in \mathbb{R} \rightarrow$ [Efficient, Inefficient]. The complete map of cells is then the cross product of these bins - e.g. one cell would consist of robot morphologies that are tall, lightweight and efficient.

In this way, any genome and its resultant phenotype can be identified with one of the cells. Like all genetic algorithms, there is also a method of evaluating the fitness of the phenotype with respect to some objective. To initialize the algorithm a set of randomly generated genomes are evaluated and placed into their respective cells. If there is more than one genome mapped to a cell then the one with the highest fitness is retained. The following steps are then repeated until a termination condition is reached: (1) A cell in the map is randomly chosen and the genome in that cell produces an offspring via some genetic operator (2) The features and performance of that offspring are determined, and the offspring is placed in the cell

if the cell is empty or if the offspring is higher-performing than the current occupant of the cell, in which case that occupant is discarded.

An interesting observation is that the elite genome found in a given cell may have being produced as an offspring via the elite of a different cell. Hence, the search for a solution in any single cell might be aided by the simultaneous search in other cells. There is evidence to support this claim; a specific experiment conducted in [44] found that MAP-Elites does produce higher performing solutions in each cell than separately searching for a high-performing solution in each of those cells. The idea of running several evolutionary processes in parallel is found in many genetic algorithms, and the benefits of doing so in some problem domains are well documented [2]. However MAP-Elites differs in that the parallel processes explicitly maintain diversity, since they are themselves associated with the exploration of different regions in the phenotype space.

This work adapts the original MAP-Elites algorithm described above for the purpose of flag discovery and development. The two main methodological modifications made were suggested by the authors in [43] - the storage of more than one genome per feature cell to promote diversity, and the use of crossover between organisms nearby in the feature space as a method of producing offspring.

Firstly, the dimensions of interest within the feature space are the colours found in the final organism. As discussed in 3.6, the kingdom of n_{colour} organisms can be grouped into a natural taxonomy, ultimately consisting of $2^{n_{\text{colour}}} - 1$ distinct species. Each of these flag species can then be associated with a cell as required in the MAP-Elites algorithm, producing a complete partition of the phenotype feature space. This is our map.

Unlike the original MAP-Elites, there will be more than one genome maintained per cell. This produces competition within a species, and is expected to help the search for fitter organisms. A variable to be defined therefore is the population size of each cell, and furthermore the number of offspring which are produced by the cell population per generation.

For the purpose of crossover, two flag species are considered nearby in the feature space if they belong to the same family, i.e. contain the same number of colours. This form of crossover, between different species but within the same family, will be referred to as inter-species crossover. Crossover can also be defined in the usual sense, between members of the same species. This form of crossover, between members of the same species and hence only between individuals within a cell, will be referred to as intra-species crossover. The set of mutation operators described in 3.5 are also used to produce offspring from a cell population.

The algorithm proceeds as follows: a set of randomly generated genomes are evaluated. The species to which the resultant flags belong to are then identified and

the genomes are correspondingly placed into their respective cells. Depending on the size of the initial set of random genomes, there may be more candidate genomes for a particular species cell than the maximum population size P_{max} of the cell. In this instance, the top P_{max} fittest solutions are assigned to initialize the cell. When there is less than P_{max} genomes for a given species they are all placed into the cell, which will then continue to fill up to the maximum population size as the appropriate offspring are created in future generations. Competition does not therefore start within a species until the cell is full.

Once initialization is completed, evolution can begin. Firstly, each cell provides a number of offspring $N_{\text{offspring}}$. These are produced through either inter-species crossover, intra-species crossover or mutation - the associated probabilities per offspring are p_{inter}, p_{intra}, p_{mutation}. Once this is complete, there are $(2^{n_{\text{colour}}} - 1) * N_{\text{offspring}}$ total offspring - these are placed in a list and distributed to parallel workers to be evaluated. Evaluation determines both an offspring's fitness and its species type. Once evaluated, the offspring are subsequently assigned to the cell associated with their species, and if they have a fitness greater than the weakest then they replace that individual within the cell. This cycle continues until a termination condition is met.

Chapter 4

Experiments and Results

This chapter describes the experiments which were completed to test the capabilities of the artificial development system and evolutionary strategies described in Chapter 3. The following section outlines a set of desirable properties for the evolved organisms, along with methods for assessing them. All assessments were made on the fittest individual from the evolutionary runs.

4.1 Organism Properties of Interest

4.1.1 Temporal Stability of Forms

During evolution the development process of an organism was simulated until $t = 800$, at which point the fitness of its phenotype was evaluated. Temporal stability of forms aims to investigate the effects of longer development times, in particular whether the phenotype can maintain the level of fitness achieved at $t = 800$. In order to assess the stability of the phenotype at $t = 800$, development is simulated until $t = 1500$ and a fitness comparison between $t = 800$ and $t = 1000, 1250, 1500$ is then made. Given the stochastic nature of the development system, this is repeated 100 times. For each of the samples a stability metric is computed - the stability at time $t > 800$ is defined as $\text{Stab}(t) = 100 * \frac{\text{Fitness}_t}{\text{Fitness}_{800}}$, and hence represents the percentage increase or decrease in fitness observed in that trial.

4.1.2 Response to Phenotype Damage

Response to phenotype damage aims to evaluate how robust the developed phenotype is to damage, and whether the organism has the capability to regenerate its

phenotype if damage is to occur. This ability is measured as follows: At $t = 801$, damage is applied to the organism. This is achieved by 'killing' a certain proportion of random cells, and additionally removing all of the fractal proteins contained within the cell from the environment. Since multiple lattice sites are associated with a single cell in the CP model, visually this amounts to blocks of image pixels been erased as shown in Figure 4.1.

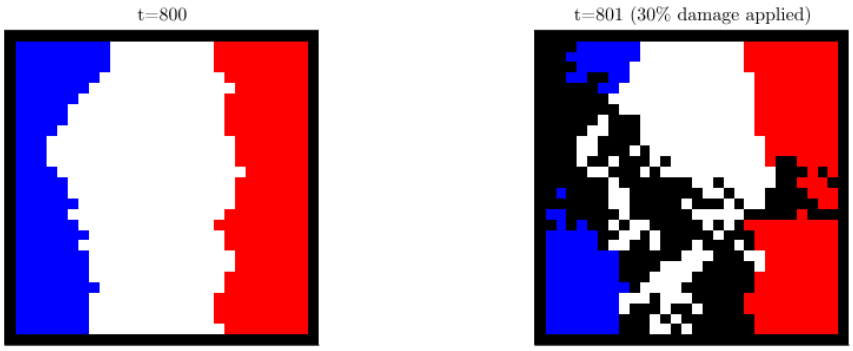

Figure 4.1: Example of phenotype damage in a French flag organism

Development is then simulated until $t = 850$, representing a recovery period. The fitness is then evaluated and compared to the fitness of the organism pre-damage at $t = 800$ in order to assess the regeneration achieved. The regeneration for damage level d is calculated as $\text{PhenoDamage}(d) = \frac{\text{Fitness}_{850}}{\text{Fitness}_{800}}$. This is repeated 100 times, and is completed for varying degrees of damage ranging from limited (10% of cells killed) to severe (90% of cells killed).

4.1.3 Response to Genotype Damage

Response to genotype damage aims to evaluate how robust the coordination of an organisms development process is to gene deletions. In each trial a number of genes are randomly deleted before the development process is then simulated. At $t = 800$ the fitness of the genetically damaged organism is compared to the fitness of the undamaged organism - the robustness to d% of gene deletions is calculated as $\text{GenoDamage}(d) = \frac{\text{DamagedFitness}_{800}}{\text{NoDamageFitness}_{800}}$. The number of trials was set at 100 per damage level, so that different combinations of gene deletion could be evaluated.

4.1.4 Fractal Induced Cellular Properties

In addition to the above tests, the cellular properties induced by fractal proteins are determined and recorded to allow comparison between individuals. The following parameters are determined:

1. The level of chemotaxis exhibited μ, as defined in Equation 3.32
2. The differential adhesion matrix J as defined in Equation 3.29, representing the adhesion between cell types
3. The dependency on the external cell signal as defined in either Equation 3.34 for the type-only signal or Equation 3.33 for the location-specific signal. Dependency is characterised as the similarity $S(R_{\tau(k)}, F_{\text{sub}})$.

The diffusion constants for each environmental protein as defined in Equation 3.24 were deemed out of scope.

4.2 Finding the French Flag Using a Location-Specific Signal

The first experiment aimed to provide a proof-of-concept for the artificial development system described in Chapter 3. The goal was to evolve an organism which developed into a French flag. Since the experiment related to an earlier iteration of the final system, a number of cellular parameters were not induced by properties of fractal proteins and were rather set as a constant by the author. These were:

1. The level of chemotaxis exhibited μ, as defined in Equation 3.32. In this experiment, $\mu = 1000$.
2. The dependency on the external cell signal as defined in Equation 3.33 (this experiment used the location-specific signal). In this experiment $S(R_{\tau(k)}, F_{\text{sub}}) = 1$ for all receptor types τ.
3. The diffusion constants for each environmental protein as defined in Equation 3.24. In this experiment $\alpha_e = \frac{1}{4}$ for all environmental proteins F_e.

Consequently, the cellular properties that were determined by fractal proteins were as follows:

1. The area target of each cell $A(k)$, as defined in Equation 3.27.

2. The differential adhesion matrix J as defined in Equation 3.29, representing the adhesion between cell types.

The evolutionary strategy was also simplified from the MAP-Flags algorithm described in Section 3.7. It was decided to search specifically for the French flag, and therefore an explicit fitness function was used. The state of the organism was simply compared with the French flag lattice, and the number of matches summed to produce an integer score. In addition, a colour penalty was introduced - the number of matches was weighted by $\exp^{-(n_{\text{colour}}-3)^2}$ where n_{colour} was the number of final colours in the organism. This was to encourage the full use of the three colours. The fitness function can be found in Appendix A.2.2. The fitness of an organism was determined as the average of n_{eval} independent evaluations - in this experiment, $n_{\text{eval}} = 20$.

The genetic algorithm maintained 20 independent pools of organisms undergoing tournament selection. Each pool had both an adult and a child population, as in [8]. The child population size was set at 80% of the adult population size. The top 40% fitness of adults were used to produce offspring for the child population, using the genetic operators described in Section 3.5.1. Each organism pool was evolved in parallel, with fit solutions sporadically exchanged between pools at a rate specified by an 'emigration' and 'immigration' rate. The full set of user-set parameter values needed to replicate the experiment can be found in Appendix B.1.

4.2.1 Evolutionary Results

The genetic algorithm ran for 50 generations. This resulted in a population of highly fit individuals, as can be seen in Figure 4.2. Individuals in the initial population had 20 genes. The fittest individual at the end of the evolutionary run had 22 genes. This genome coded for 3 receptor, 5 behavioural, 9 regulatory and 5 environmental proteins. An example of the development and final form obtained by the fittest individual is in Figure 4.3.

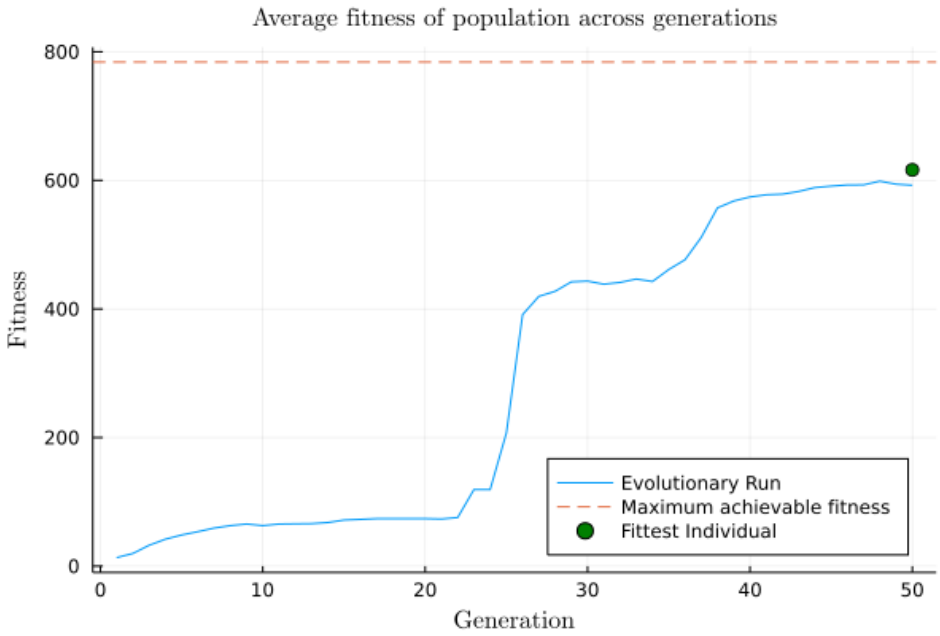

Figure 4.2: Experiment 1: Average fitness of population across generations

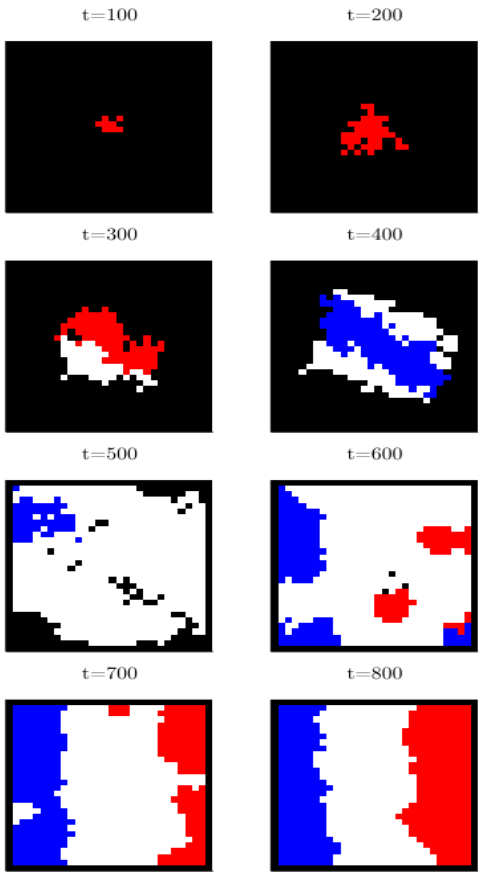

Figure 4.3: Experiment 1: Example development of French flag organism

4.2.2 Organism Properties

4.2.2.1 Temporal Stability of Forms

The experimental results associated with the temporal stability test are summarised in Figure 4.4. Interestingly, it appears as though the organism had a propensity to increase its fitness as development was allowed to run for longer - for $t = 1000, 1250, 1500$, the median response of the organism was to obtain a fitness greater than 100% of the fitness measured at $t = 800$.

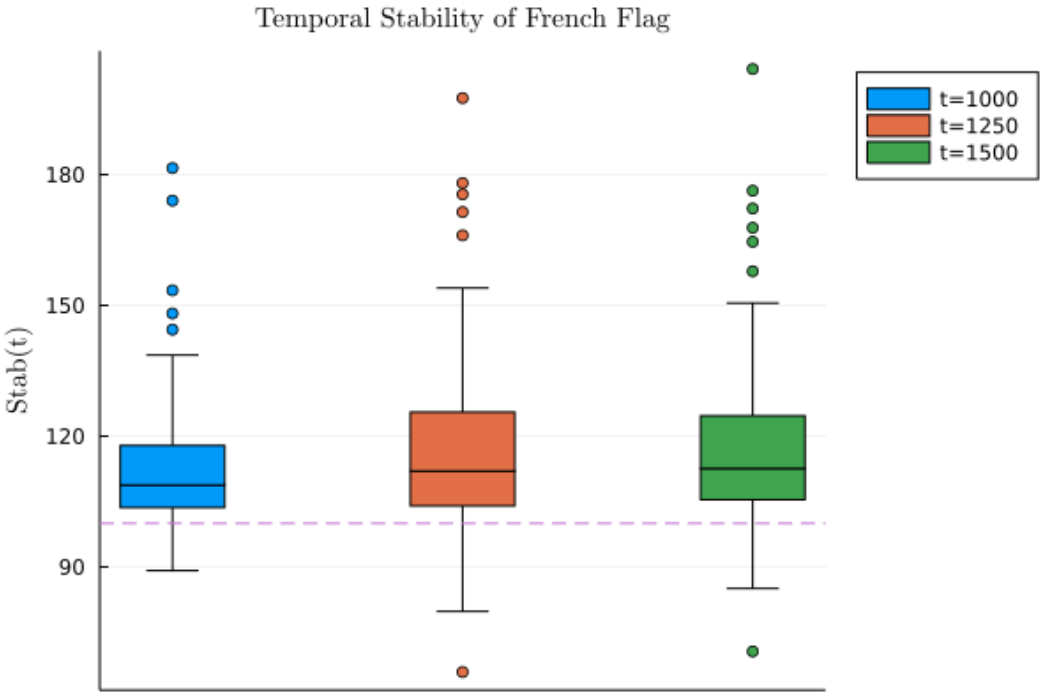

Figure 4.4: Experiment 1: Temporal stability of French flag

4.2.2.2 Response to Phenotype Damage

The regeneration capability of the French flag organism is summarised in Figure 4.5. The organism appears to have good regeneration capabilities - for damage severities up to 50%, the organism was able to regenerate to the fitness obtained pre-damage for the median of responses. Examination of the outliers indicates that phenotype damage can result in both significant increases and decreases in fitness. Examples of regeneration are shown in Figure 4.6.

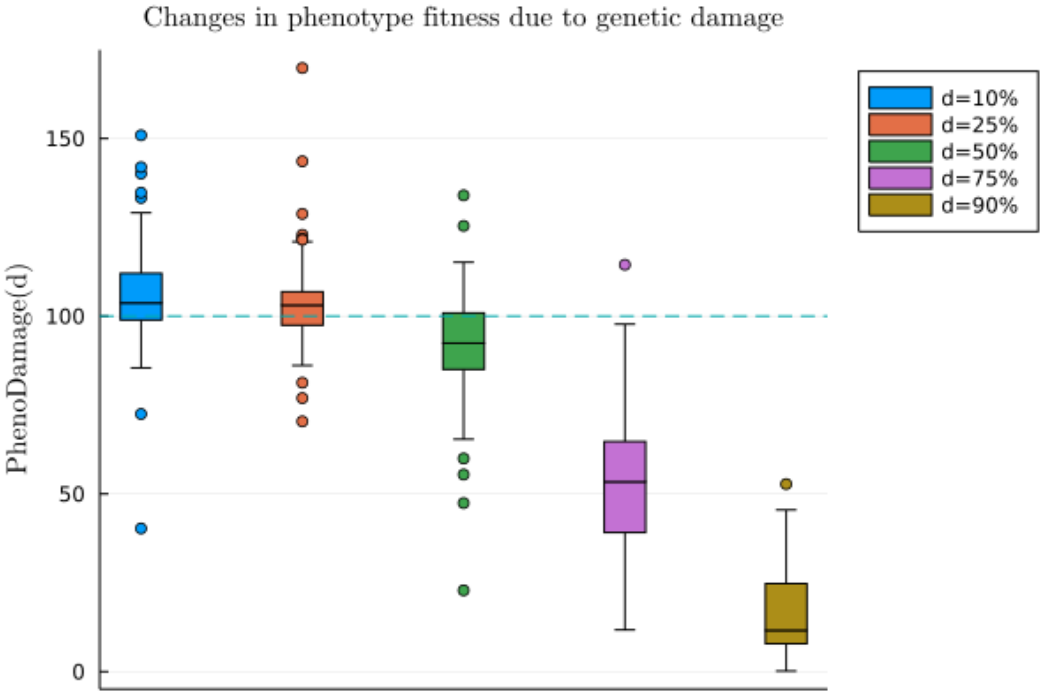

Figure 4.5: Experiment 1: Regeneration capability of the French flag organism

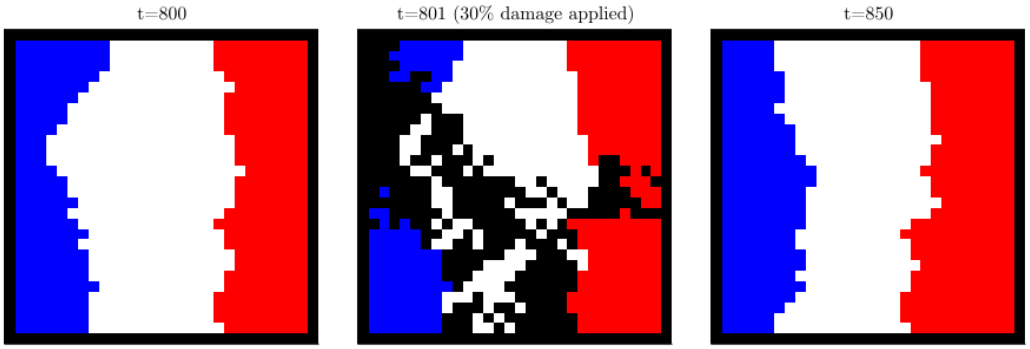

Figure 4.6: Experiment 1: Example of regeneration for the French flag organism

Since an organisms ability to regenerate did not form an explicit part of the fitness function, an interesting question is how this ability is obtained. One hypothesis explored in this work is that regeneration is enabled through a combination of cellular adhesion and growth mechanisms. Since similar cell types have low differential adhesion this results in a tendency for similar cells to clump together and form homogenous regions, repairing any irregularities. In the context of the CP model, it is hypothesized that a stable organism phenotype is represented by a low energy configuration on the lattice. Damage to this configuration then results in a configuration of higher energy. Subsequently, the search for a lower energy configuration manifests in the regeneration of the phenotype. This hypothesis is supported

by Figure 4.7. It shows the energy associated with the system hamiltonian as defined in Equation 3.26 over the course of the organisms development, as well as the organism fitness. In addition, the dashed line indicates the point at which 30% of the organisms cells are destroyed. The chart was produced as the average trajectories of 100 such organism evaluations. Immediately one can see that an inverse relationship exists between system fitness and system energy. In addition, damage to the phenotype results in a higher energy configuration, which is then 'repaired' by transitioning the system to a lower energy state. This idea is examined further in later experiments.

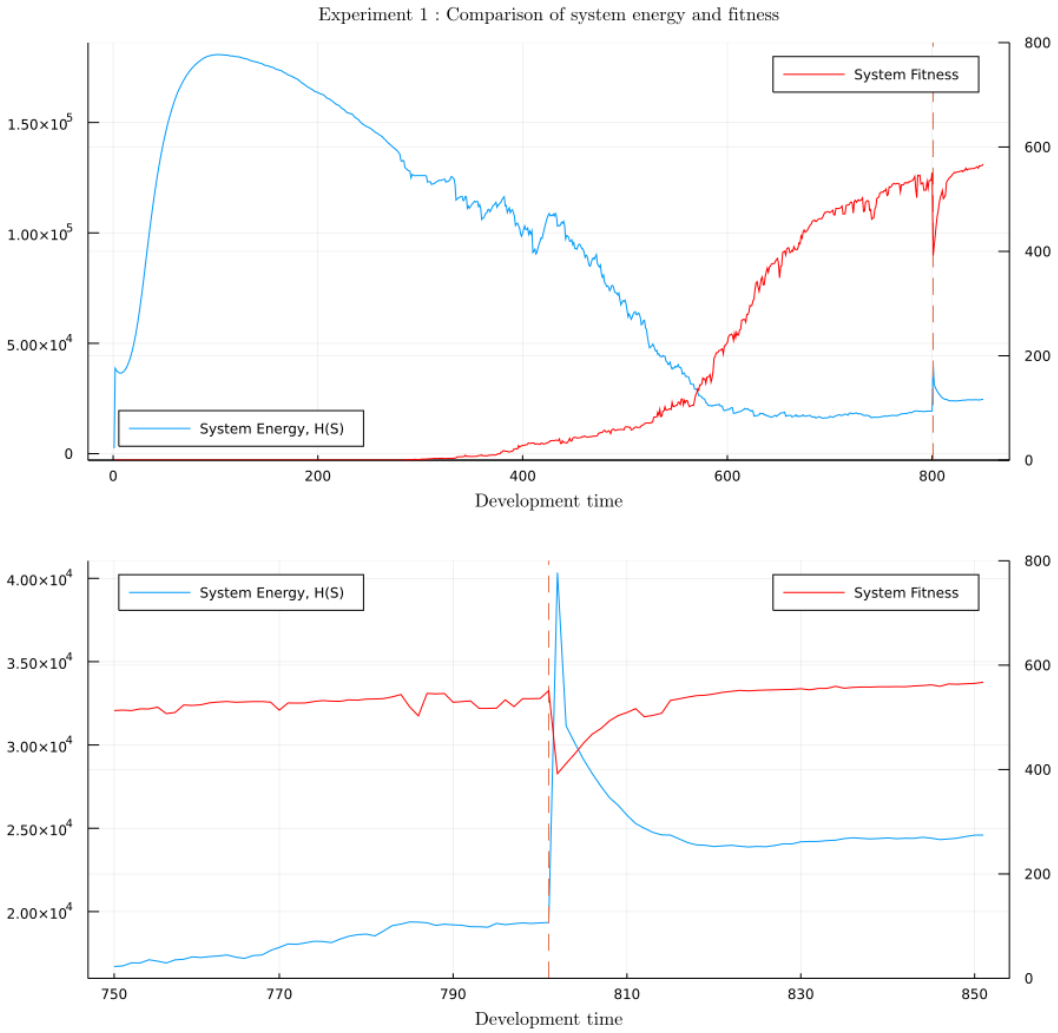

Figure 4.7: Experiment 1: Comparison of system energy and fitness with 30% damage applied

4.2.2.3 Response to Genotype Damage

The experimental results associated with the genetic damage test are summarised in Figure 4.8. The organism demonstrates a good level of robustness at the 10% deletion level, with both the median and inter-quartile range indicating minimum deviation from the fitness of the phenotype obtained with an undamaged genome. At the 20% deletion level its median response was reasonably robust, at over 60% of the original fitness. The results do not suggest that the evolved genome is confidently robust to the damage incurred at the 30% deletion level.

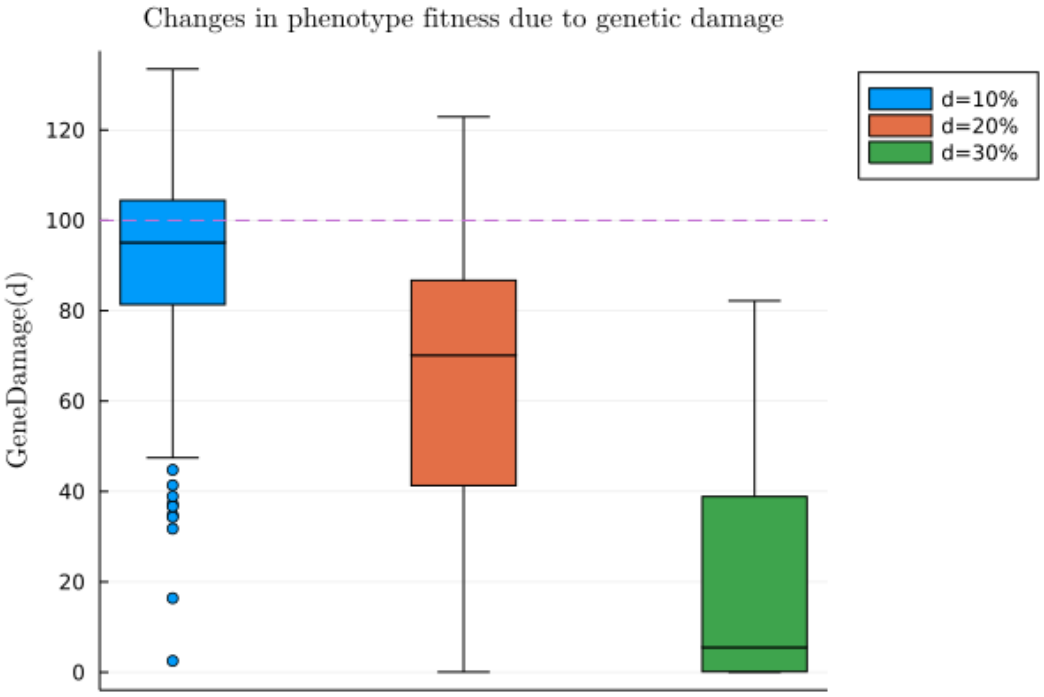

Figure 4.8: Experiment 1: Phenotype damage associated with various degrees of genetic deletion for the French flag organism

4.2.2.4 Fractal Induced Properties

The differential adhesion value between cell types is presented in Table 4.1. Whilst a full investigation into the values and their significance for the CP dynamics was out of the scope of this report, an attempt at an interpretation is made. Firstly, an obvious observation is that the two cell types belonging on the right and left sides of the French flag (blue and red) have a low differential adhesion, whereas both have a comparatively higher differential adhesion with the cell type in the middle, white.

One interpretation is that white energetically 'pins' the size of the blue and red cell regions - it is energetically expensive for blue to grow into the white region, since this increases the boundary length between the two. This could explain the apparent stability of the forms.

Species	Blue	White	Red	Green
Blue	0	0.678	0.046	0.678
White	0.678	0	0.632	0
Red	0.046	0.632	0	0.632
Green	0.678	0	0.632	0

Table 4.1: Experiment 1: Evolved differential adhesion values for the French flag organism

4.3 Finding Three Colour Flags Using a Location-Specific Signal

In the second experiment, the artificial development system in its complete specification was used. Therefore, the following cellular properties were induced by properties of fractal proteins and hence evolved:

1. The level of chemotaxis exhibited μ, as defined in Equation 3.32
2. The differential adhesion matrix J as defined in Equation 3.29, representing the adhesion between cell types
3. The dependency on the external cell signal as defined in Equation 3.33 (this experiment used the location-specific signal).
4. The diffusion constants for each environmental protein as defined in Equation 3.24.
5. The area target of each cell $A(k)$, as defined in Equation 3.27.

This was therefore the first true test of the artificial development system. Organisms continued to receive the location specific external signal, so that the experiment results could be interpreted in terms of the changes listed above. Receiving the signal is therefore conditional on cell type and location. In addition, the signal was kept as the French flag, and therefore the cell types which allowed a signal to be received corresponded to the French flag colours.

The genetic algorithm used was the MAP-Flags algorithm as described in Section 3.7. This experiment therefore aimed to test another hypothesis - that a novelty search algorithm could be used to discover many different species of flag in a single evolutionary run. The search was focused on the 3-colour family, of which there are 4 possible species:

1. Species 3-1 : Blue, Red and Green

2. Species 3-2 : Blue, White and Green

3. Species 3-3 : Blue, Red and White

4. Species 3-4 : White, Red and Green

Since the search was focused on 3-colour flags, the fitness function assigned zero fitness to any offspring which had more or less than 3 total colours in its phenotype.

4.3.1 Evolutionary Results

The genetic algorithm ran for 360 generations. In this run, three out of four of the possible species were discovered. As can be seen in Figure 4.9, the species which obtained the highest average population fitness was species 3-3. This is the same species to which the original French flag organism found in the first experiment belongs - interestingly the fittest organism found in this experiment demonstrates an alternative spatial arrangement of colours, which can be seen in Figure 4.10.

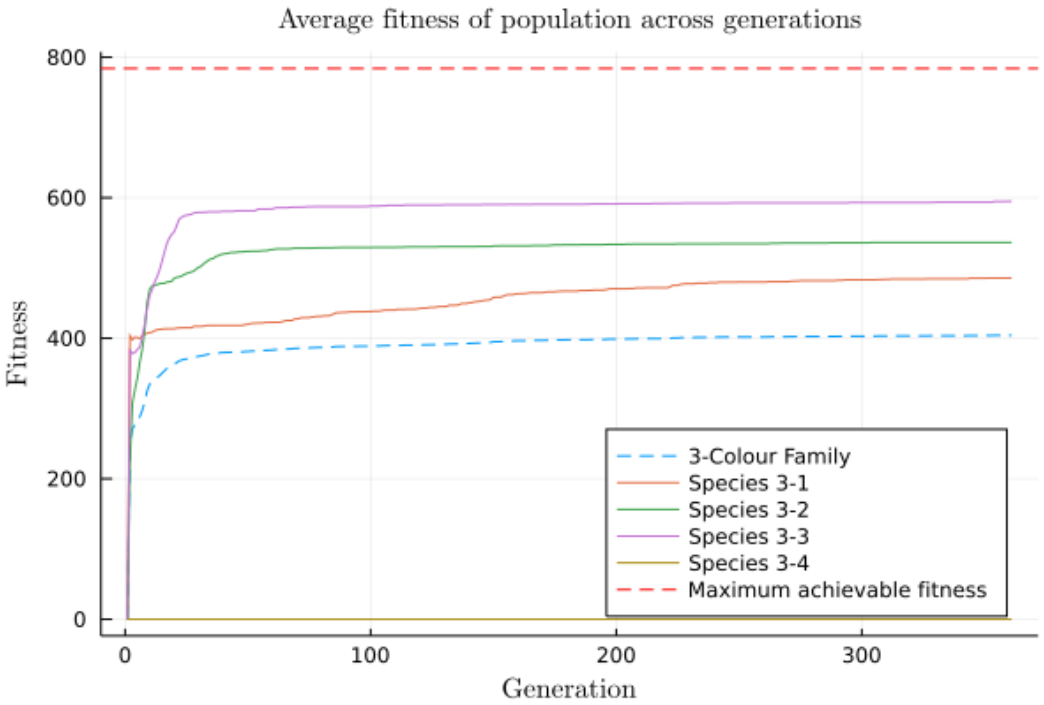

Figure 4.9: Experiment 2: Average fitness of population across generations

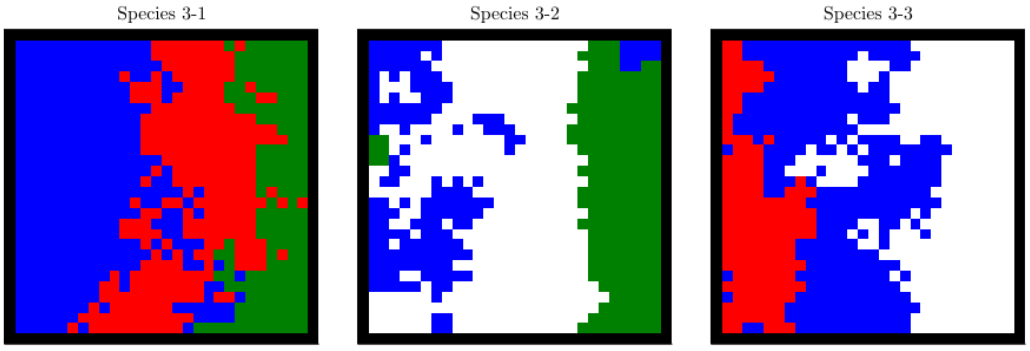

Figure 4.10: Experiment 2: All discovered species belonging to the 3-Colour family

4.3.1.1 Lineage

As described in Section 3.7, the MAP-Flags algorithm allows inter-species crossover to occur. Table 4.2 examines the origin of parents across the entire population of each species present at the end of the evolutionary run, and Table 4.3 examines the fittest organisms found.

Species	2 parents within species	1 parent outside species	2 parents outside species
Species 3-1	100%	0%	0%
Species 3-2	100%	0%	0%
Species 3-3	100%	0%	0%

Table 4.2: Experiment 2: Origin of parents within the final population

Species	Highest Fitness	Generation Discovered	Parent Origin
Species 3-1	496.64	14	Both parents from within species
Species 3-2	543.29	24	Both parents from within species
Species 3-3	609.06	124	Both parents from within species

Table 4.3: Experiment 2: Origin of parents of fittest individuals

Whilst in the final population organisms were ultimately bred from within their own species, an examination of Figure 4.11 indicates a significant amount of genetic material is shared amongst the fittest organisms of the three discovered species. In Figure 4.11, dark blue represents the presence of the gene indicated on the y-axis. The y-axis labels are string encodings [1] of all the unique genes found across the three organisms. There were 52 unique genes - each organisms genome contained 26 genes, and so the maximum number of unique genes possible was 78. In addition, there were 100 unique proteins defined by the combination of coding and promoter regions, out of a possible maximum of 156. Given that genes in the initial population were randomly initialized, the presence of shared genes between different species indicates a common ancestor. This lends evidence to the hypothesis that inter-species crossover is beneficial in the search for fit organisms.

[1] The gene strings can be decoded as follows - the first character indicates the gene type, "v","b","r","e" representing receptor, behavioural, regulatory and environmental respectively. There are then 4 additional characters which separate two digit numbers. The numbers after "c" and "p" each encode one of the 100 unique proteins, and represent the coding and promoter proteins. The numbers after "t" and "a" encode the minimum activation energy and minimum affinity values respectively.

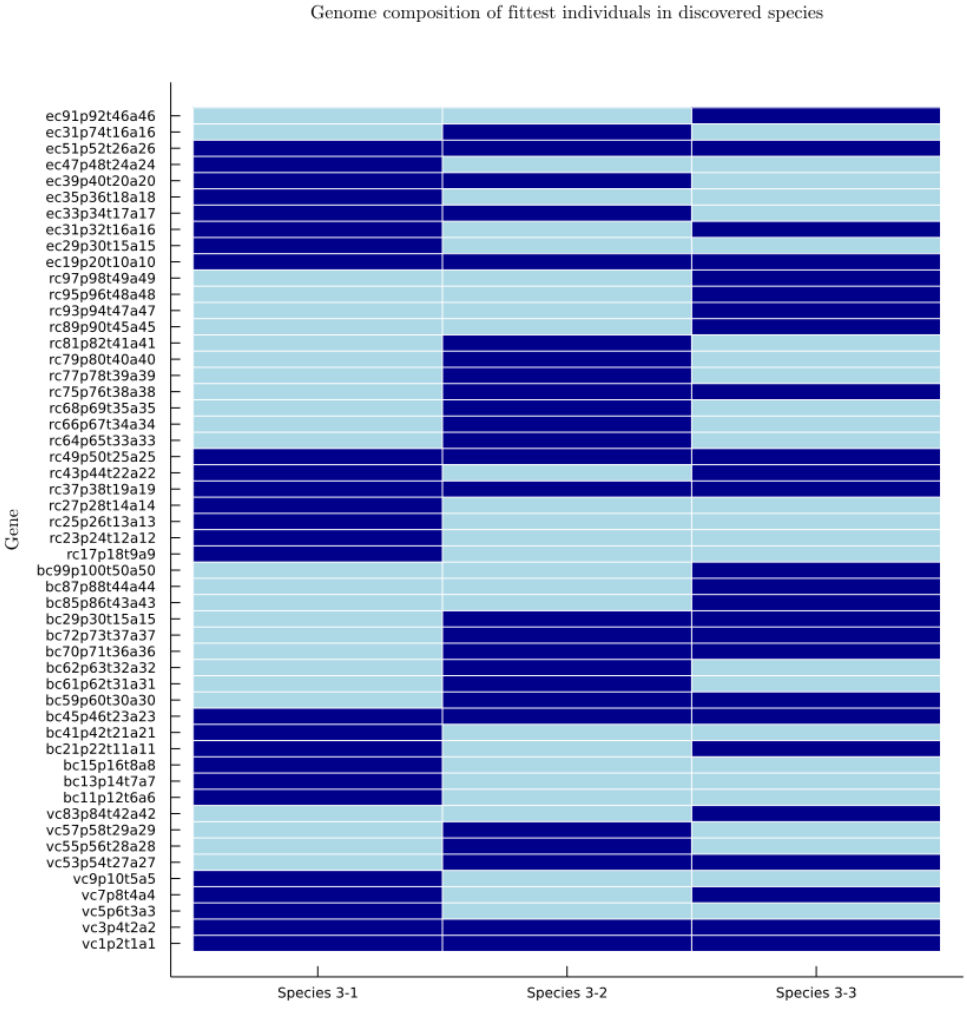

Figure 4.11: Experiment 2: All discovered species belonging to the 3-Colour family

4.3.2 Organism Properties

4.3.2.1 Temporal Stability of Forms

The experimental results associated with the temporal stability test are summarised in Figure 4.12, Figure 4.13 and Figure 4.14 for species 3-1, 3-2 and 3-3 respectively. As expected, the most stable organisms appear to be those which achieved the highest comparative fitness. Whilst 3-2 and 3-3 both demonstrate stability, interestingly it appears as though 3-2 had a higher propensity to increase its fitness as development was allowed to run for longer. In comparison, 3-3 tended to maintain the fitness it had achieved at $t = 800$. Species 3-1 demonstrates a higher degree

of instability - its interquartile (25 - 75%) range includes the full range of fitness depreciation, including death of the organism.

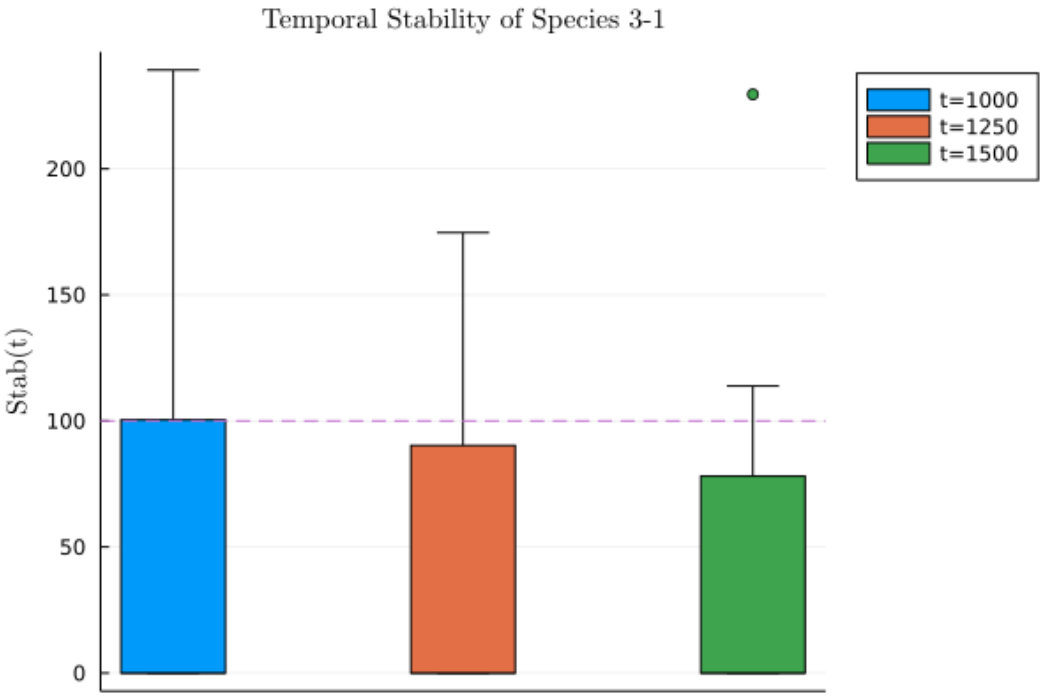

Figure 4.12: Experiment 2: Temporal stability of Species 3-1

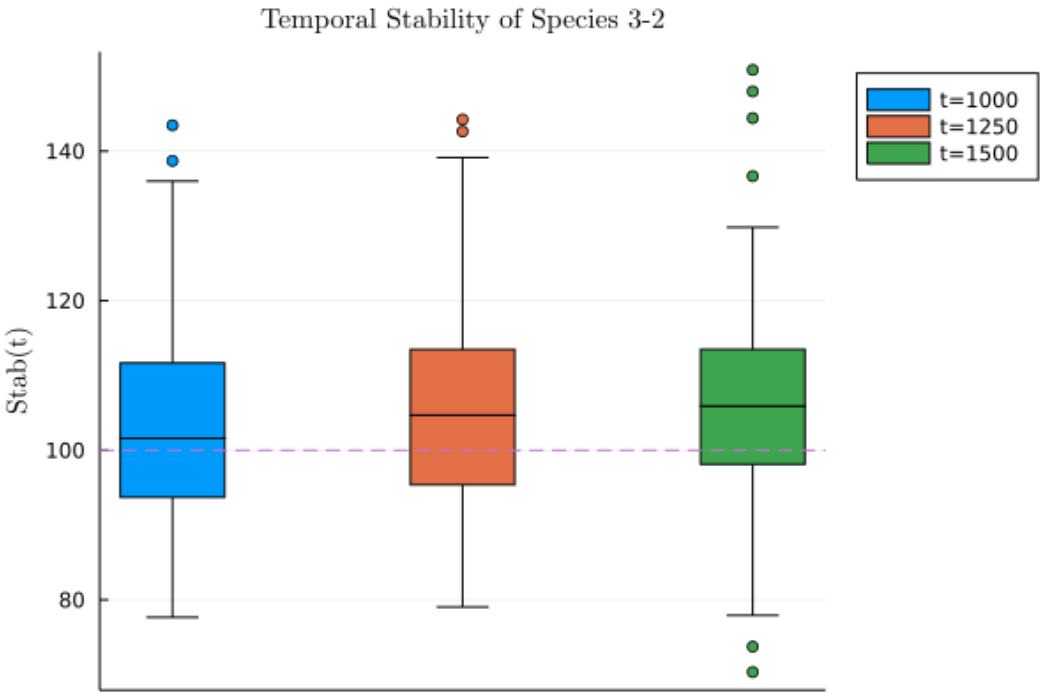

Figure 4.13: Experiment 2: Temporal stability of Species 3-2

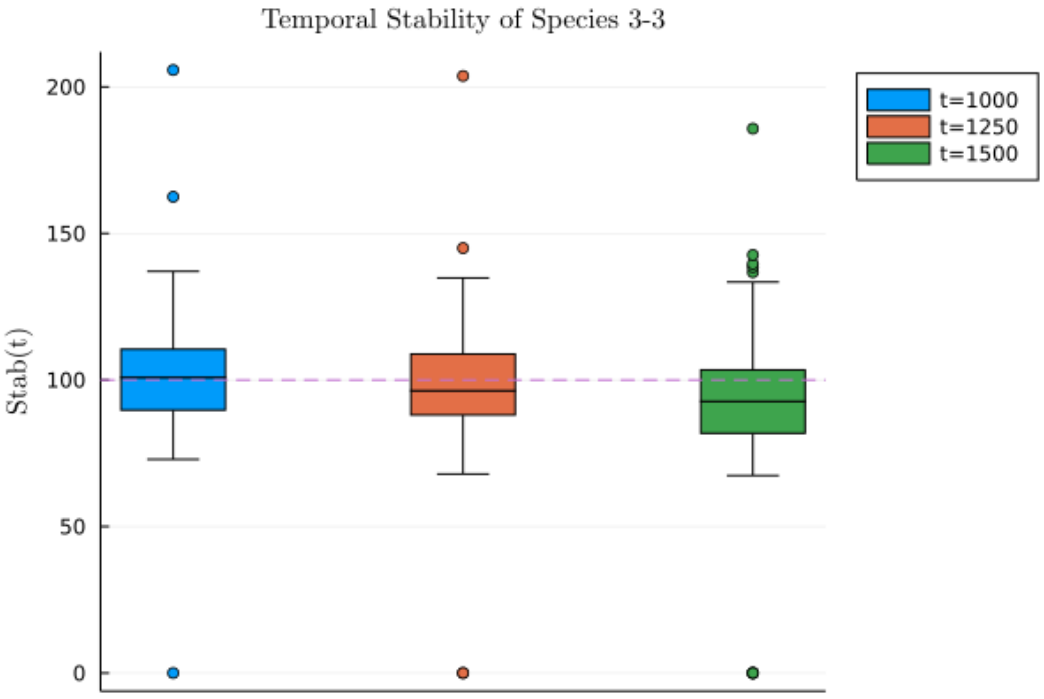

Figure 4.14: Experiment 2: Temporal stability of Species 3-3

4.3.2.2 Response to Phenotype Damage

The regeneration capabilities of species 3-1, 3-2 and 3-3 are summarised in Figure 4.15, Figure 4.16 and Figure 4.17 respectively. Similar to the stability results, species 3-1 demonstrated the worst regeneration capability across the damage severities. Species 3-2 and 3-3 appear to have excellent regeneration capabilities - even after 75% of cells had been destroyed, the median regeneration response resulted in over 75% of the original fitness been re-obtained. Damage of up to 50% of the cells resulted in almost perfect regeneration of the original fitness for the median of responses. Examination of the outliers also indicates that phenotype damage can result in an increase in ultimate fitness. Examples of regeneration for each species are shown in Figure 4.18.

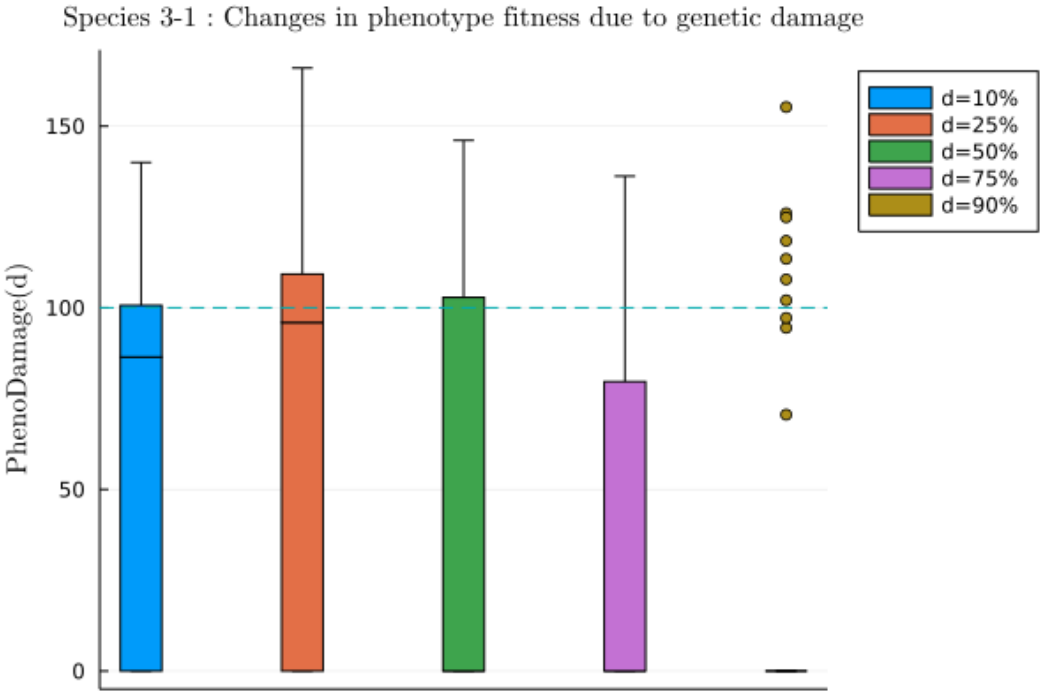

Figure 4.15: Experiment 2: Regeneration capability of Species 3-1

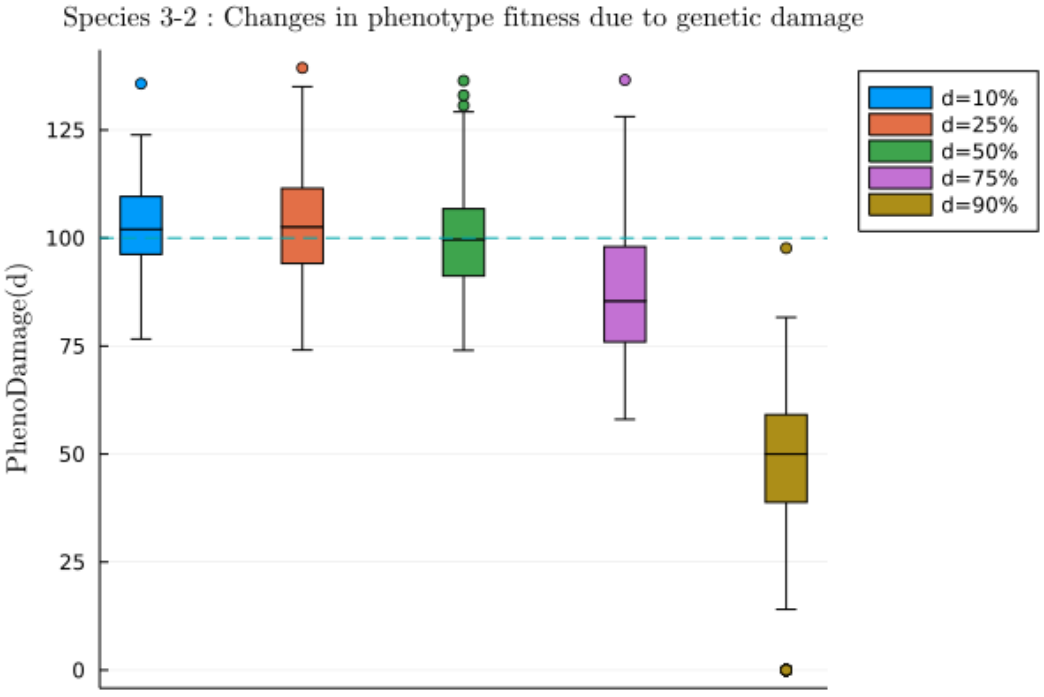

Figure 4.16: Experiment 2: Regeneration capability of Species 3-2

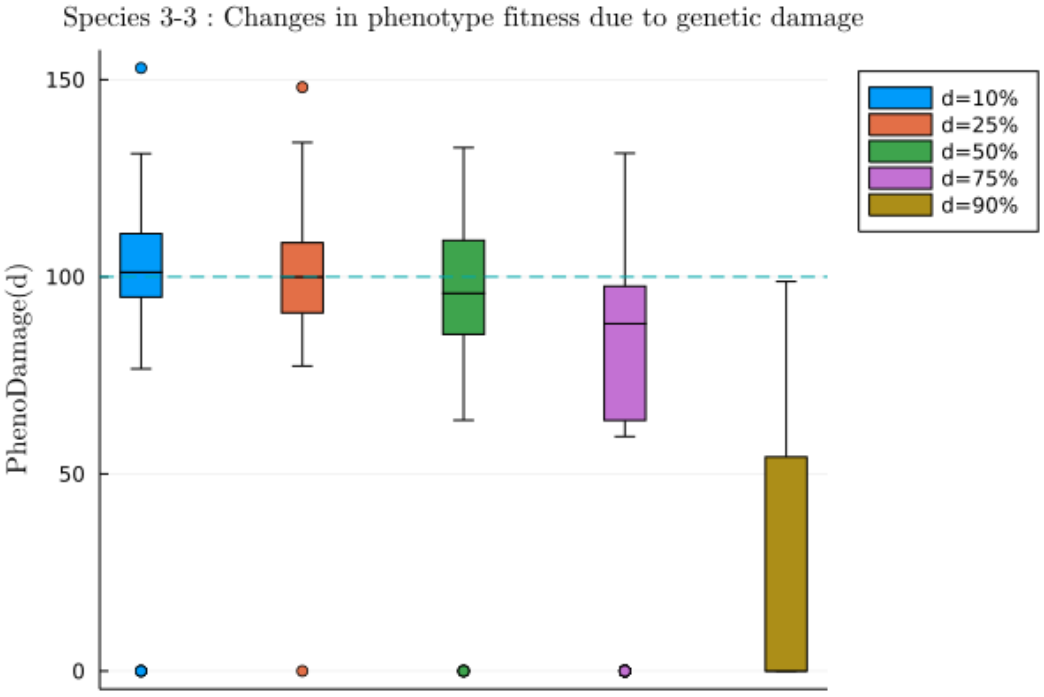

Figure 4.17: Experiment 2: Regeneration capability of Species 3-3

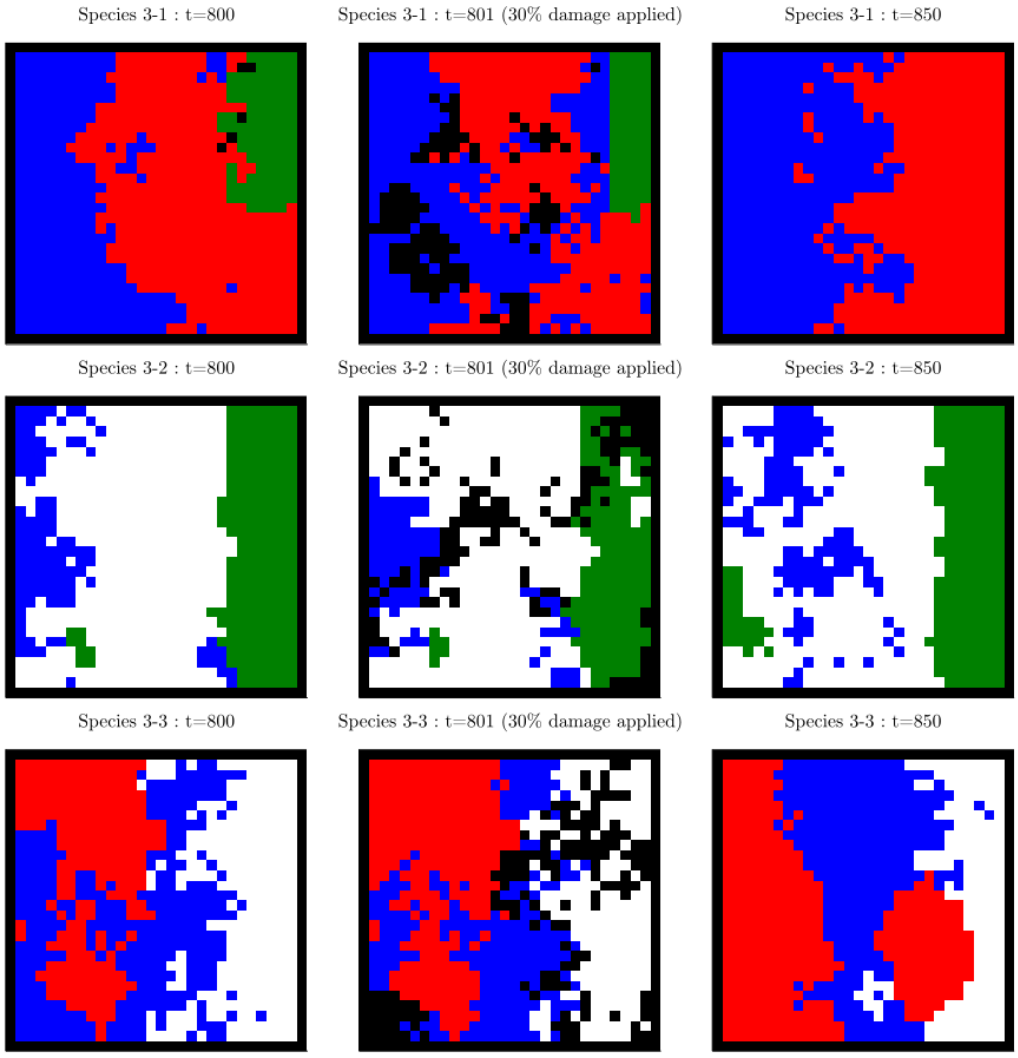

Figure 4.18: Experiment 2: Examples of regeneration across discovered species

The next part of the investigation revisits the energy-regeneration hypothesis explored in the first experiment. The relationship between system energy and system fitness is summarised in Figure 4.19, Figure 4.20 and Figure 4.21 for species 3-1, 3-2 and 3-3 respectively. As before, a strong inverse relationship between system energy and system fitness is apparent across species.

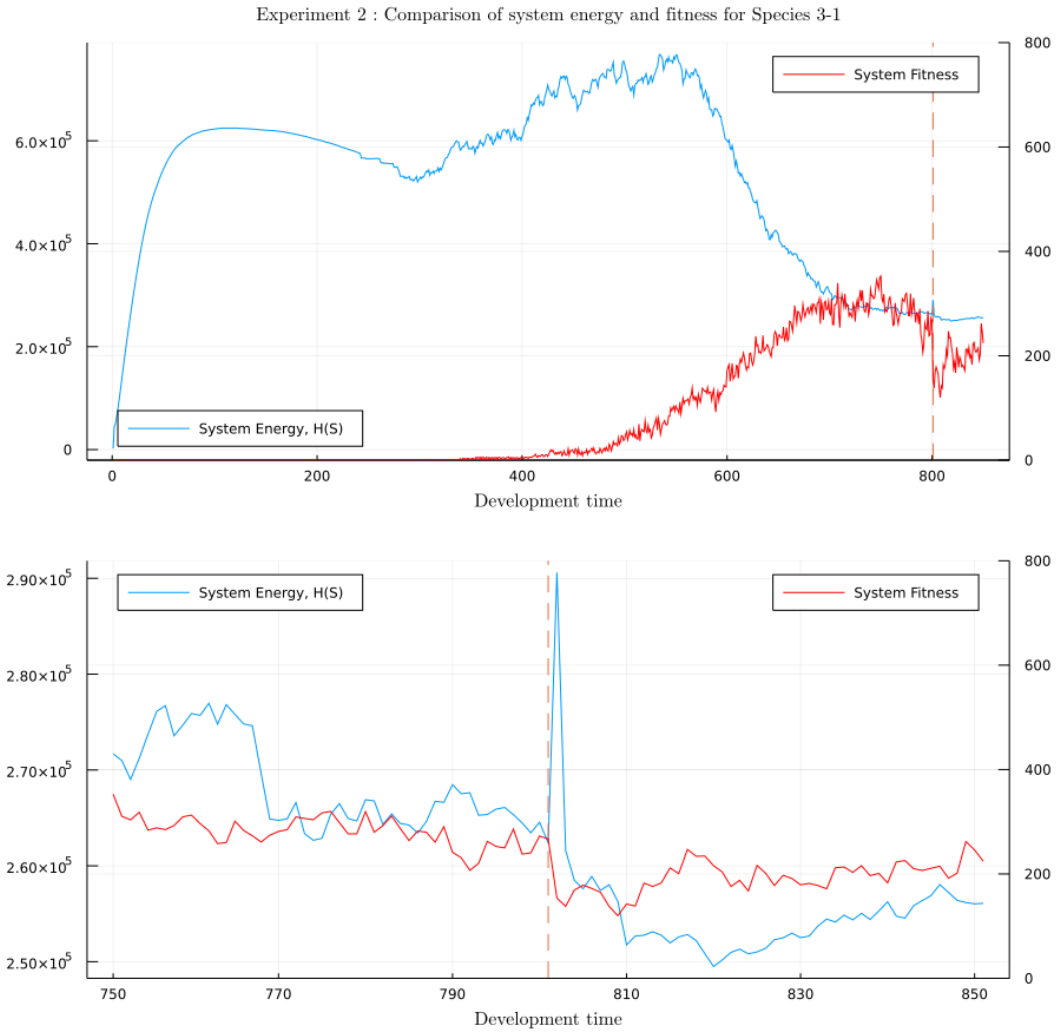

Figure 4.19: Experiment 2: Comparison of system energy and fitness with damage application for Species 3-1

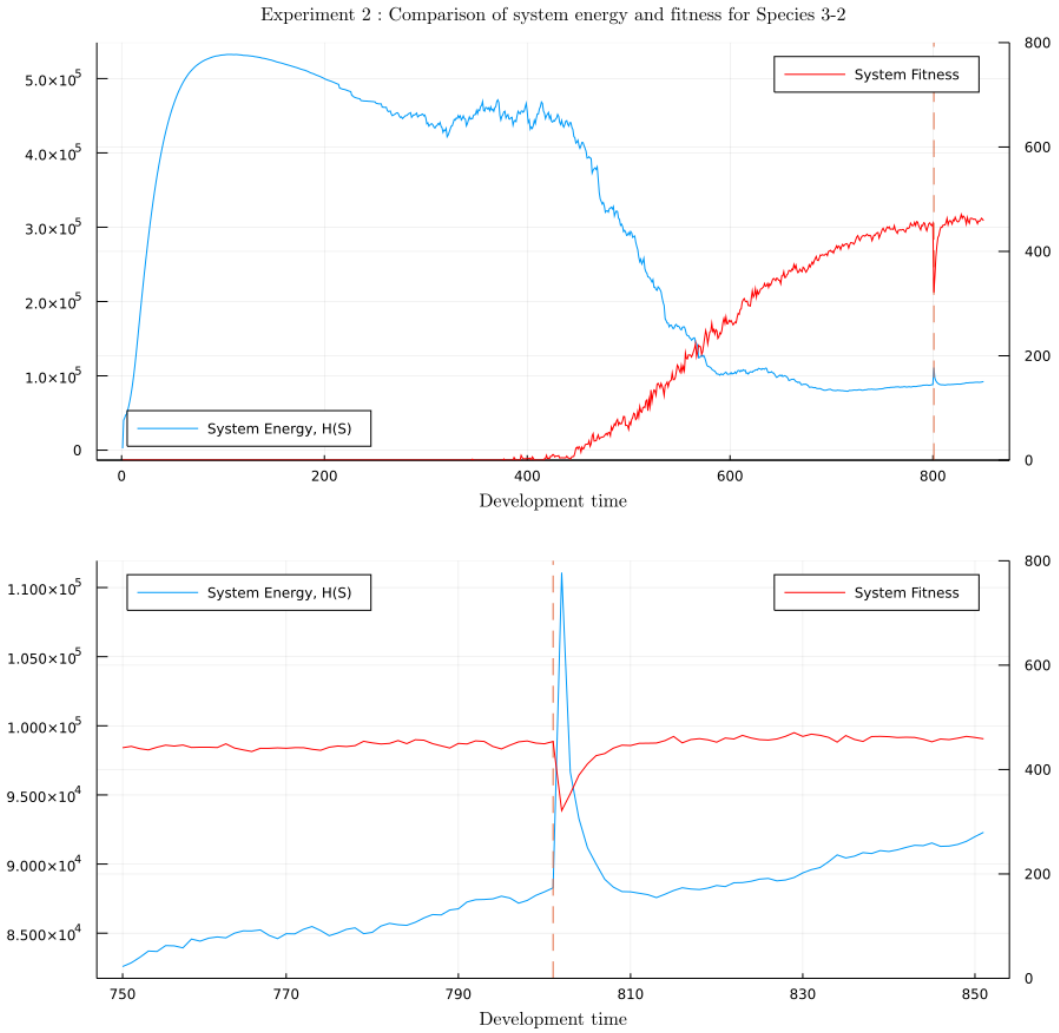

Figure 4.20: Experiment 2: Comparison of system energy and fitness with damage application for Species 3-2

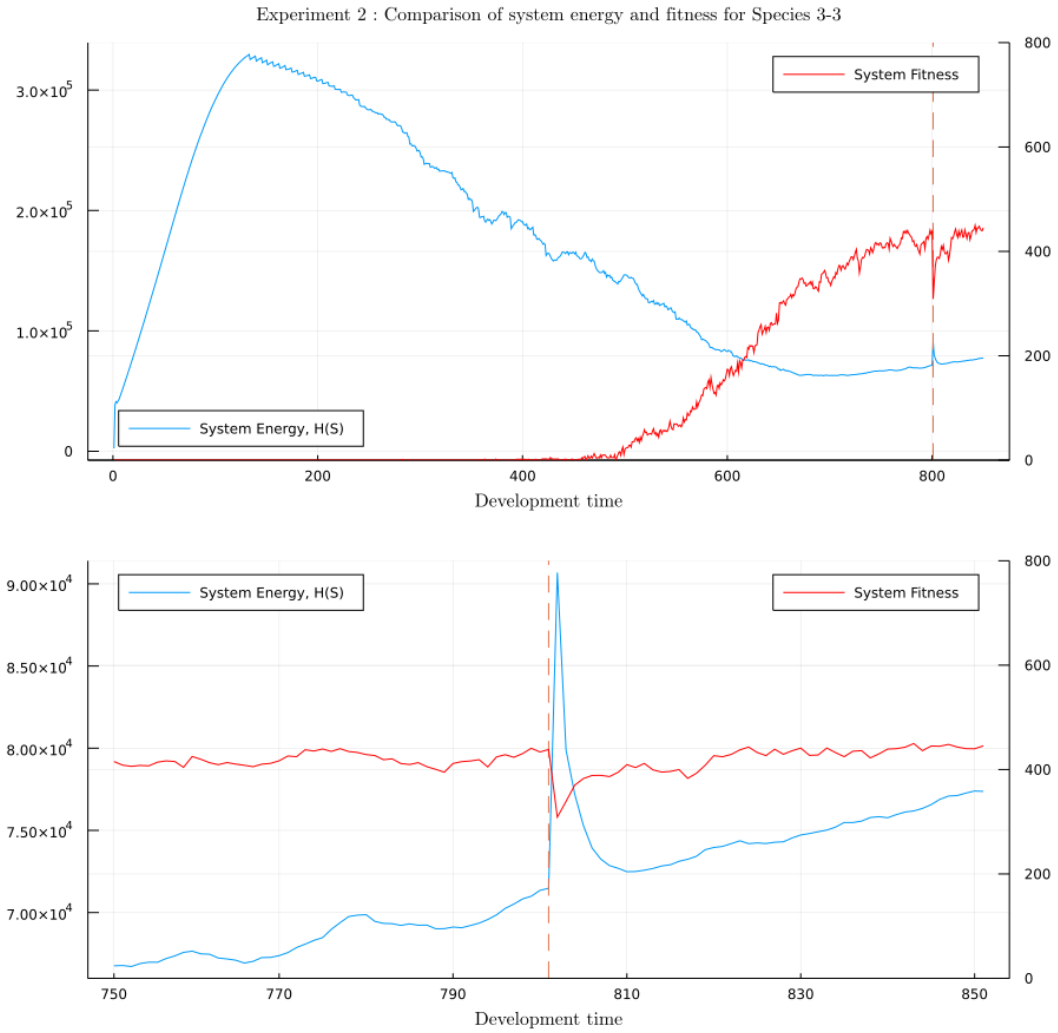

Figure 4.21: Experiment 2: Comparison of system energy and fitness with damage application for Species 3-3

In addition, it appears that the behaviour of the system energy trajectory between time steps 400 and 600 is useful in characterizing the comparative fitness of individuals, and their ability to regenerate after damage. The trajectories indicate that those organisms which display the strongest inverse correlation between system energy and system fitness during development achieve the highest regeneration ability. To test this hypothesis, the spearman correlation coefficient between the system energy trajectory and the system fitness was calculated across 100 development trials per species. The results are displayed in Table 4.4, and support the observation.

Species	Spearman rank coefficient	Average fitness at $t = 800$	Average PhenoDamage(30%)
3-1	-0.391	229.01	61%
3-2	-0.728	433.41	104%
3-3	-0.681	479.39	102%

Table 4.4: Experiment 2: Correlation between system energy and system fitness

4.3.2.3 Response to Genotype Damage

The experimental results associated with the genetic damage test are summarised in Figure 4.22, Figure 4.23 and Figure 4.24 for species 3-1, 3-2 and 3-3 respectively. Neither 3-1 nor 3-2 appear to be robust to the random deletion of genetic material. However, species 3-3 demonstrates a good level of robustness at the 10% deletion level, with both the median and inter-quartile range indicating minimum deviation from the fitness of the phenotype obtained with an undamaged genome. At the 20% deletion level its median response was reasonably robust, at 75% of the original fitness. However, the variability of the results does not suggest that the evolved genome is confidently robust to that level of damage.

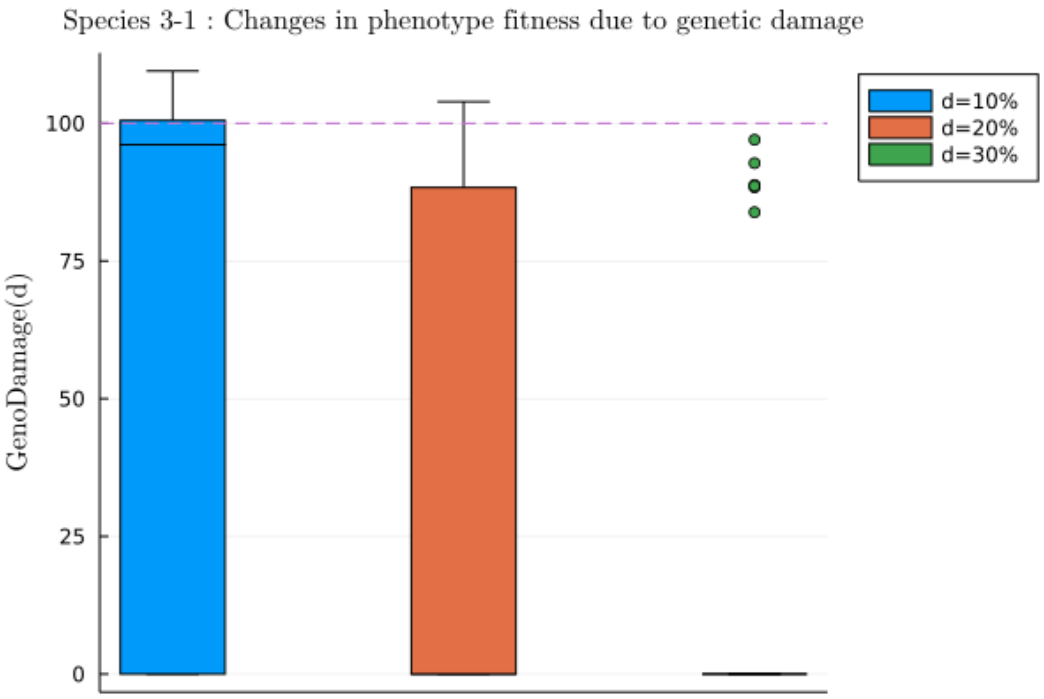

Figure 4.22: Experiment 2: Phenotype damage associated with various degrees of genetic deletion for Species 3-2

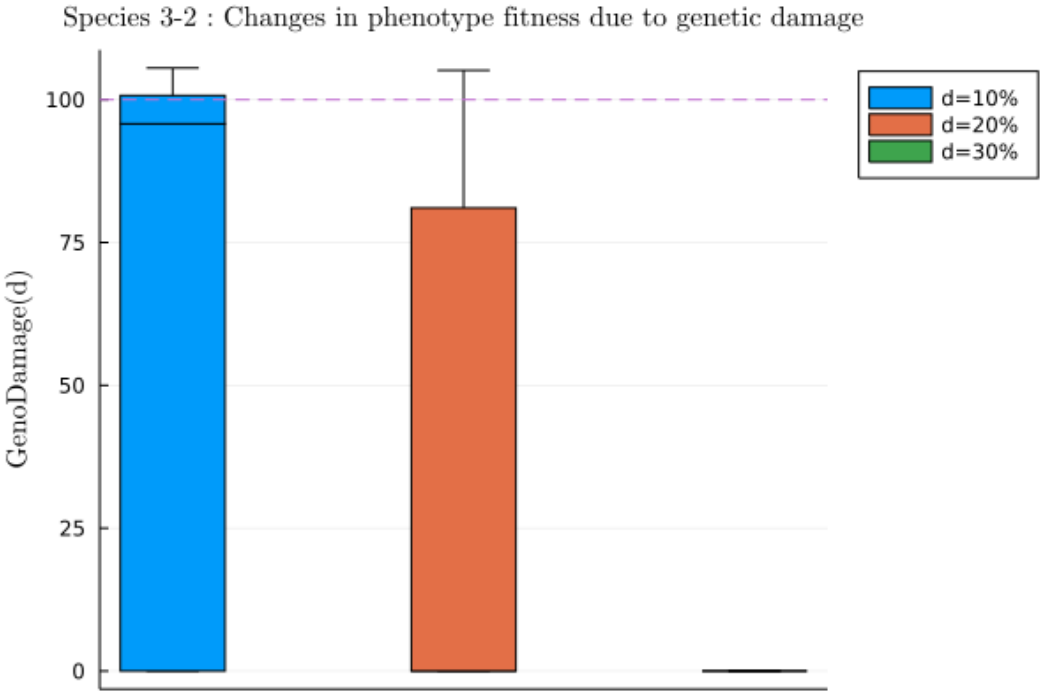

Figure 4.23: Experiment 2: Phenotype damage associated with various degrees of genetic deletion for Species 3-3

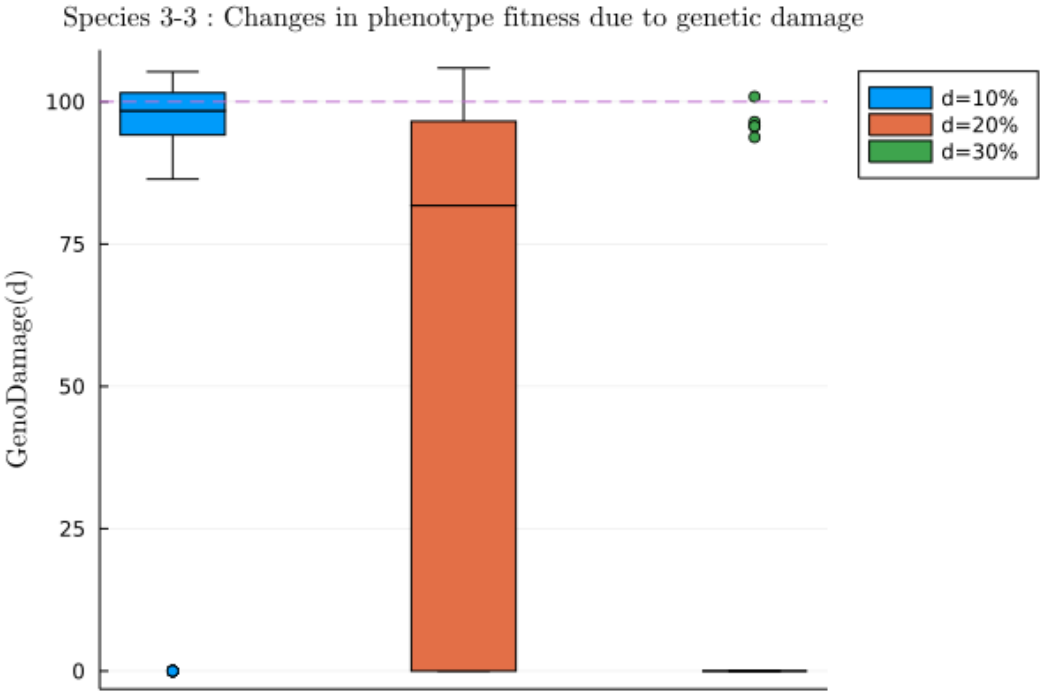

Figure 4.24: Experiment 2: Phenotype damage associated with various degrees of genetic deletion for Species 3-3

4.3.2.4 Fractal Induced Properties

The evolved differential adhesion values between cell types for species 3-1, 3-2 and 3-3 are presented in Figure 4.5, Figure 4.6 and Figure 4.7 respectively. Whilst a full investigation into the values and their significance for the CP dynamics was out of the scope of this report, a few observations are made. The following instabilities have been observed by the author in the organisms under study -

1. Species 3-1 struggles to maintain the volume of its green region, as is shown in Figure 4.18.

2. Species 3-2 struggles to maintain the volume of its blue region, as is shown in Figure 4.18

The red-green and blue-white differential adhesion values are extremely small in species 3-1 and species 3-2 respectively. Red and white are also their respective middle region cell types, and therefore this supports the hypothesis made in the first experiment that higher values help to maintain the size of edge regions.

Cell Type	Blue	White	Red	Green
Blue	0	0.035	0.097	0.097
White	0.035	0	0.062	0.062
Red	0.097	0.062	0	0
Green	0.097	0.062	0	0

Table 4.5: Experiment 2: Evolved differential adhesion values for Species 3-1

Cell Type	Blue	White	Red	Green
Blue	0	0.041	0.601	0.601
White	0.041	0	0.560	0.560
Red	0.601	0.560	0	0
Green	0.601	0.560	0	0

Table 4.6: Experiment 2: Evolved differential adhesion values for Species 3-2

Cell Type	Blue	White	Red	Green
Blue	0	0.119	0.614	0.618
White	0.119	0	0.505	0.500
Red	0.614	0.505	0	0.005
Green	0.618	0.500	0.005	0

Table 4.7: Experiment 2: Evolved differential adhesion values for Species 3-3

Table 4.8 presents the evolved receptor strengths for the location-specific external signal.

Receptor τ	$S(R_\tau, F_{\text{sub}})$: 3-1	$S(R_\tau, F_{\text{sub}})$: 3-2	$S(R_\tau, F_{\text{sub}})$: 3-3
Blue	0.099	0.099	0.099
White	0.047	0.047	0.047
Red	0.040	0.043	0.043
Green	0.034	1.000	0.034

Table 4.8: Experiment 2: Evolved receptor strengths for discovered species

The evolved chemotaxis strength values were $\mu_{3-2} = 0.844$, $\mu_{3-1} = 0.491$ and $\mu_{3-3} = 0.842$. Since this chemotaxis strength is calculate as a similarity between proteins, it is bounded between zero and one. Therefore, two of the three species evolved high levels of chemotaxis.

4.4 Finding General Flags Using a Type-Specific Signal

The previous experiment demonstrated that the proposed development system can be evolved to produce flag like organisms using a location-specific signal. The use of the MAP-Flags genetic algorithm with its flexible fitness function allowed multiple species to be discovered in a single evolutionary run. The third and final experiment extended the scope of the search to all of the *n*-colour families associated with the flag taxonomy. Furthermore, the location specific signal was changed to the type specific signal.

4.4.1 Evolutionary Results

The genetic algorithm was allowed to run for 50 generations. The resultant fitnesses associated with each of the *n*-colour families can be seen in Figure 4.25.

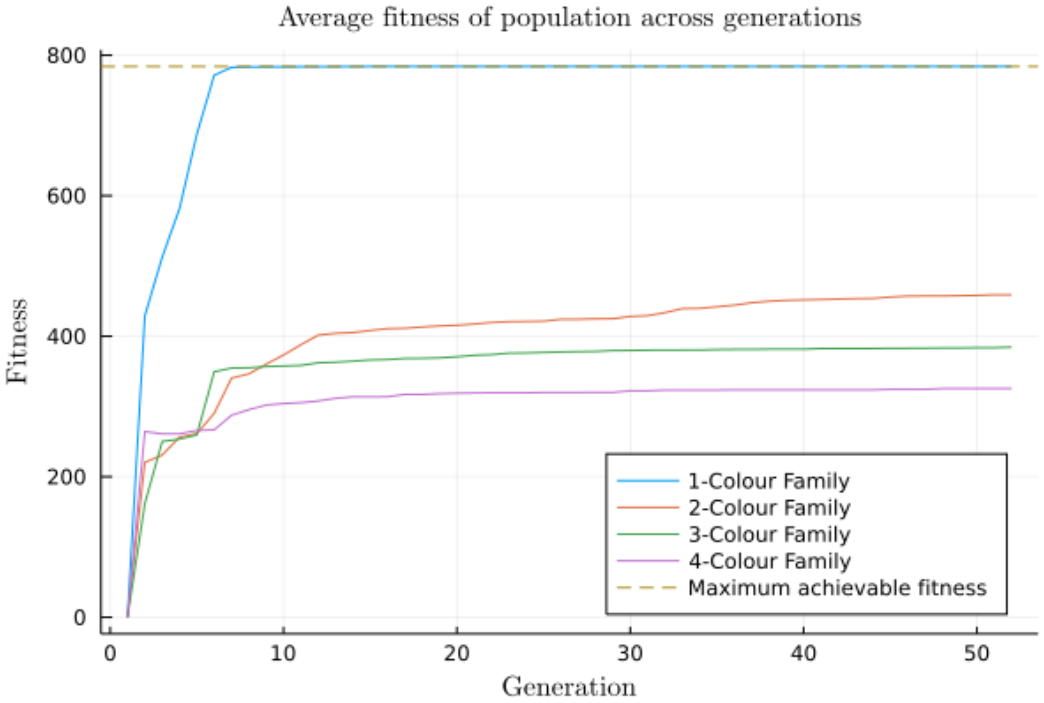

Figure 4.25: Experiment 3: Average fitness of population across generations

Unfortunately, time constraints meant that evolution could not run for longer. In spite of this, all possible species were discovered in the evolutionary run. Example forms of the fittest individuals from each of the species are displayed in Figure 4.26.

Aside from the 1-colour family who trivially obtained maximum fitness, none of the other families appear to be capable of developing flag like forms. However, there are some promising attributes, such as the appearance of cellular regions with homogenous cell type.

4.4.1.1 Lineage

Since the state of the final evolved population is significantly less mature than the population obtained in the second experiment, it is interesting to examine the lineage of organisms and note the differences between the mature and immature populations. Table 4.9 and Table 4.10 both support the hypothesis that inter-species

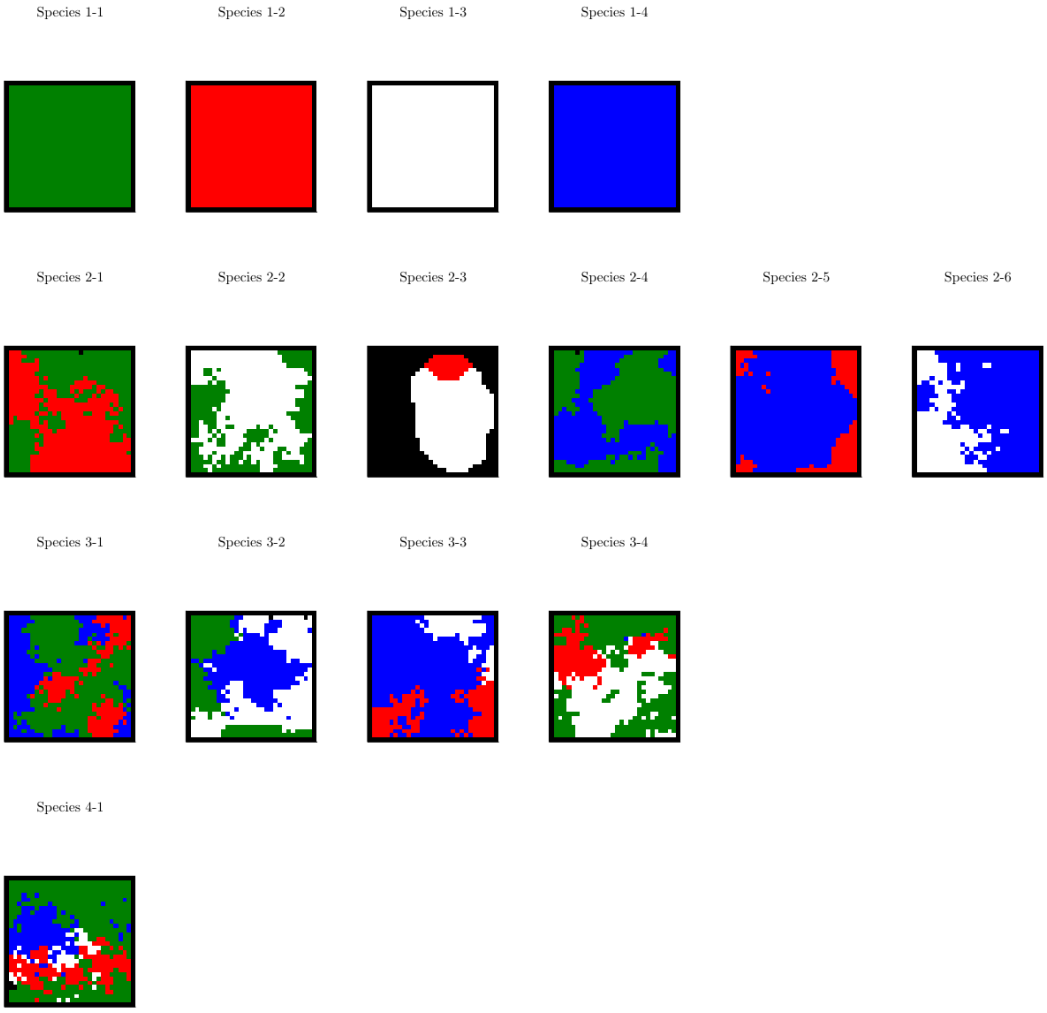

Figure 4.26: Experiment 3: All discovered species

crossover and the MAP-Flags algorithm aid in the search for fit organisms across species. This data suggests that in the earlier stage of evolution, there is healthy exchange of genetic material between species in addition to the gradual refinement that the competition within a species produces.

Species	2 parents within species	1 parent outside species	2 parents outside species
Species 1-1	60%	0%	40%
Species 1-2	60%	0%	40%
Species 1-3	50%	10%	40%
Species 1-4	30%	0%	70%
Species 2-1	100%	0%	0%
Species 2-2	90%	10%	0%
Species 2-3	100%	0%	0%
Species 2-4	90%	0%	10%
Species 2-5	80%	10%	10%
Species 2-6	100%	0%	0%
Species 3-1	90%	0%	10%
Species 3-2	50%	0%	50%
Species 3-3	70%	30%	0%
Species 3-4	60%	10%	30%
Species 4-4	30%	0%	70%

Table 4.9: Experiment 3: Origin of parents within the final population

Species	Highest Fitness	Generation Discovered	Parent Origin
Species 1-1	Max (784)	0	2 parents outside species
Species 1-2	Max (784)	0	2 parents outside species
Species 1-3	Max (784)	3	2 parents within species
Species 1-4	Max (784)	6	2 parents within species
Species 2-1	503	7	2 parents within species
Species 2-2	484	10	2 parents within species
Species 2-3	315	13	2 parents outside species
Species 2-4	498	14	2 parents within species
Species 2-5	530	14	1 parent outside species
Species 2-6	549	17	2 parents outside species
Species 3-1	386	19	2 parents within species
Species 3-2	397	19	1 parent outside species
Species 3-3	410	21	2 parents within species
Species 3-4	395	24	2 parents within species
Species 4-4	337	28	2 parents outside species

Table 4.10: Experiment 3: Origin of parents of fittest individuals

Chapter 5

Conclusions

5.1 Method

In Chapter 3, a link was formed between the fractal proteins within a cell and the following cellular behaviours: growth, chemotaxis, receptiveness to external signals, and the differential adhesion between cell types. In addition, a new spatial-temporal property was attached to environmental proteins - the speed at which they diffuse through the cellular substrate. This resulted in a high level of integration between the genetic and phenotype control mechanisms, forming an expressive development system with a compact genetic representation. In Section 3.2.3, new ways of expressing regulatory mechanisms were described, which aimed to reduce the number of user-set meta parameters whilst increasing the richness of the protein chemistry. In Section 3.6, a flag taxonomy was created. This enabled a novelty-search style genetic algorithm named MAP-Flags to be designed for the purpose of evolving organisms with flag-like characteristics. A single fitness function was described which allowed all organisms within the taxonomy to be flexibly evaluated, regardless of final colour composition or arrangement.

5.2 Experiments

Section 4.2 described the first experiment and its results. An organism which developed into a French flag was successfully evolved using the main development system proposed in Chapter 3 combined with a location specific signal. In this sense the experiment was a success, providing a proof-of-concept for subsequent work. The evolved organism demonstrated temporal stability, as well as a capacity to regenerate damaged designs. In Section 4.2.2.2, the origin of this regenerative capability was investigated, and a link between high fitness organisms and low energy cellular

configurations was evidenced. The evolved genome also demonstrated a reasonable level of robustness to gene deletions, however significant damage severely reduced the fitness of the developed phenotype.

The second experiment and its results were described in Section 4.3. The MAP-Flags genetic algorithm was used to discover three distinct species from the flag taxonomy in a single evolutionary run, with all species demonstrating visible flag like characteristics in their developed forms. Investigations into the genetic composition of the fittest individuals showed that specific genes existed across species, lending weight to the hypothesis that the sharing of genetic material through inter-species crossover whilst maintaining intra-species competition aided the evolutionary search.

Whilst the properties of evolved organisms varied across species, all demonstrated a reasonable capacity to maintain stable final forms and regenerate damaged areas. The link between system energy and organism fitness was further substantiated, with a comparison across species suggesting that quality of regeneration was positively associated with the degree of inverse correlation between system energy and system fitness across the development time frame. Cellular properties such as differential adhesion and degree of chemotaxis were all evolved through the fractal protein formulation, with commonalities identified across the fittest organisms from each species.

The final experiment and its results were presented in Section 4.4. This experiment removed the location-specific signal, and expanded the evolutionary search across the full flag taxonomy. Unfortunately the amount of evolutionary search time was limited due to project time constraints, and therefore evolved organisms did not reach a high level of fitness. However the initial results are promising, with all species within the flag taxonomy discovered in a single evolutionary run. It is expected that the removal of the location specific signal makes the task of evolving organisms with flag like characteristics considerably more difficult than in previous experiments. Future work will determine whether the capabilities of the development system and evolutionary strategies introduced in this work extend far enough for high fitness individuals to be discovered.

Chapter 6

Further Work

Whilst this work has introduced new ideas and offered promising lines of inquiry, there is plenty of further analysis which could be completed.

Firstly the adjustments to the fractal gene regulation formulation described in Section 3.2.3 should be investigated further, in particular to establish whether or not it can be experimentally verified that the new formulation presents advantages over the original. In this work, the adjustments were made in order to reduce the number of user set meta-parameters, whilst also increasing the complexity of the metabolism chemistry responsible for gene regulation. This increased complexity hypothesis is theoretical and moreover, it has not been experimentally confirmed whether this provides any benefits for certain regulation tasks.

Even though efforts were made to reduce the number of user set meta-parameters in the development system, there is still a large number. All of these parameters were set by the author based on trial and error or heuristic reasoning, and specific values for each experiment can be found in Appendix B. It is unknown to what degree the choices affect the capabilities of the system, and therefore work needs to be completed in order to increase transparency. A possible solution would be to make them accessible to evolution through further integration with fractal proteins. In particular the author believes it might be advantageous to reformulate the update rule for fractal protein concentrations, so that different types of reaction dynamics can be defined implicitly by properties of the reactant proteins, rather than by meta-parameters in a differential equation.

A more complete assessment of the MAP-Flags genetic algorithm should be completed. The ancestry of individuals within the population could be recorded throughout the evolutionary run, so that the transfer of genetic material across species can be specifically identified. In addition, user-set parameters such as the mutation frequency or inter-species crossover rate should be varied across experiments and the effects recorded. In addition, the evolutionary experiments should be ran for more generations. Firstly to improve the fitness of organisms, and secondly to see if more compact genomes can be evolved, as was done in [9].

The link between system energy and regeneration capability should be investigated in more detail. In addition, a characteristic that all of the trajectories demonstrated was an increase in system energy in the initial stages of development. An investigation into the effect of the temperature value T in the metropolis-style update should be made to clarify this phenomena.

Finally, whilst best efforts were made to produce a high performance software implementation of the development system, improvements could be made which would increase the throughput in the evolutionary experiments. The strategy in this work was to produce a high performance single-thread implementation for the evaluation of a single individual, and then distribute the evaluation of the multiple offspring across multiple parallel workers. Other ways of parallelizing the implementation were not explored. It should be noted however that one of the key factors contributing to the length of fitness evaluation is the stochastic nature of the development system, meaning that a reasonable number of repeat evaluations must be made to assess fitness in a robust manner.

Appendix A

Code

A.1 Overview

All work described in this paper was implemented in Julia [1]. There are two main repositories associated with the work in this report. The full implementation of the artificial development system and evolutionary strategies can be found at `https://github.com/harrybooth/FractalDevelopment`. Jupyter notebooks which reproduce much of the analysis included in this report can be found at `https://github.com/harrybooth/FractalDevelopmentExperiments`.

A.2 Selected Methods

A.2.1 Mitosis

```
function mitosis(which_cells::Array{CartesianIndex{2},1})

    # which_cells is an array containing the lattice positions of a cell

    cell_size = size(which_cells)[1] # number of lattice sites belonging to cell

    centre_of_mass = (Tuple(sum(which_cells)) ./ cell_size)

    xx = 0.
    xy = 0.
    yx = 0.
    yy = 0.

    for i in Tuple.(which_cells) # create covariance matrix of mean subtracted positions
```

[1] `https://julialang.org/`

```
15
16          xx += (i[1] - centre_of_mass[1])^2
17          xy += (i[1] - centre_of_mass[1])*(i[2] - centre_of_mass[2])
18          yx += (i[2] - centre_of_mass[2])*(i[1] - centre_of_mass[1])
19          yy += (i[2] - centre_of_mass[2])^2
20      end
21
22      # Calculate eigenvalues (quick as in 2D)
23
24      T = (xx + yy)/cell_size
25      D = (xx*yy - xy*yx)/(cell_size^2)
26
27      L1 = 0.5*(T + sqrt(T^2 - 4*D +  1e-10))
28      L2 = 0.5*(T - sqrt(T^2 - 4*D + 1e-10))
29
30      if yx != 0
31          normal_dir= (-yx/cell_size,max(L1,L2) - yy/cell_size)
32
33      else
34          normal_dir = (0,-1)
35      end
36
37      hyp_const = -sum(normal_dir .* centre_of_mass)
38
39      left::Array{CartesianIndex{2},1} = []
40      right::Array{CartesianIndex{2},1} = []
41
42      for i in which_cells
43          if sum(Tuple(i) .* normal_dir) + hyp_const >= 0 # evaluate which side of split axis
44              push!(left,i)
45          else
46              push!(right,i)
47          end
48      end
49
50      return left,right
51  end
```

A.2.2 Fitness Functions

```
1   function evaluate_fitness(state::Matrix{Float64},evo_config::POOLConfig)
2
3       colors = unique(state)
4       filter!(e->e!=0,colors)
5       n_color = size(colors,1)
6
7       matches = 0
8
9       for j in 1:size(state,2)
10          for i in 1:size(state,1)
11              if state[i,j] == evo_config.flag[i,j]
12                  matches += 1
13              end
14          end
15      end
```

```
16
17      return exp(-(n_color - 3)^2)*matches
18
19  end
20
21
22
23  function evaluate_fitness(state::Matrix{Float64},evo_config::EMAPConfig)
24
25      colors = unique(state) # colours in final phenotype
26      filter!(e->e!=0,colors)
27      n_color = size(colors,1)
28
29      if n_color in evo_config.families
30
31          section_step = Int(floor(size(state,2)/n_color)) # equal vertical stripes
32          choices = permutations(colors) |> collect  # all possible colour orderings
33          results = []
34
35          for layout in choices
36              matches = 0
37              start = 1
38              for section in 1:n_color # evaluate the matches in each of the sections
39                  begin_it =  (1 + (start - 1)*section_step) # start of section
40                  end_it =   min(start*section_step,size(state,2)) # end of section
41                  for j in begin_it:end_it
42                      for i in 1:size(state,1)
43                          if state[i,j] == layout[section] # colour of section set by current layout
44                              matches += 1
45                          end
46                      end
47                  end
48                  start += 1
49              end
50              push!(results,matches)
51          end
52
53          fingerprint = 1 # fingerprint value is an encoding of species
54
55          for i in colors
56              if i == 1
57                  fingerprint += 2^3
58              elseif i == 2
59                  fingerprint += 2^2
60              elseif i == 3
61                  fingerprint += 2^1
62              elseif i == 4
63                  fingerprint += 2^0
64              else
65                  fingerprint = "error"
66              end
67          end
68
69          return maximum(results),fingerprint # return max possible fitness given all orderings
70      else
71          return 0,1
72      end
73
74  end
```

Appendix B

Experiment Configuration Files

B.1 Experiment 1

```
grid_size = 30

flag = zeros(Float64,(grid_size,grid_size));
flag[:,1:Int(grid_size/3)] .= 1
flag[:,Int(grid_size/3)+1:Int(2*grid_size/3)] .= 2
flag[:,Int(2*grid_size/3)+1:end] .= 3

exp_user_spec = Dict{String, Any}("diffusion_alpha" => 1.,
                        "chemotaxis_mu" => 1000.,
                        "external_signal_type" => "location",
                        "external_signal_receptor_strength" =>  1.)

grn_user_spec = Dict{String, Any}("protein_size" => 15,
                        "max_iter" => 55,
                        "mandel_thresh" => 2,
                        "Ct" =>  0.2,
                        "Cs" =>  0.2,
                        "Cw" =>  -0.1,
                        "Cp" =>  20.,
                        "Ci" =>  1.5)

cell_user_spec = Dict{String, Any}("cell_size_upper_limit" => 100.,
                        "max_split" => 1.,
                        "area_lambda" => 10000.,
                        "temperature" =>  1.,
                        "chemotaxis_max_strength" => 5000.,
                        "grid_size" =>  grid_size)

evo_user_spec = Dict{String, Any}("n_total_genes" => 20,
                        "n_evaluation" => 20,
                        "time_steps" => 800,
                        "adult_pop_size" => 10,
                        "child_pop_size" => 8,
                        "n_pools" =>  20,
                        "elitism_percentage" =>  0.4,
                        "elitism_migration_percentage" =>  0.1,
                        "emmigration_rate" =>  0.15,
```

```
                        "immigration_rate" =>  0.15,
                        "crossover" => 1.,
                        "flag" => flag,
                        "genome_start" => [1,1,1,2,2,2,3,4,3])

exp_config = ExperimentalConfig(exp_user_spec)
grn_config = FGRNConfig(grn_user_spec)
cell_config = CellularConfig(cell_user_spec)
evo_config = POOLConfig(evo_user_spec)
```

B.2 Experiment 2

```
exp_user_spec = Dict{String, Any}("diffusion_alpha" => "evolve",
                        "chemotaxis_mu" => "evolve",
                        "external_signal_type" => "type",
                        "external_signal_receptor_strength" => "evolve")

grn_user_spec = Dict{String, Any}("protein_size" => 15,
                        "max_iter" => 55,
                        "mandel_thresh" => 2,
                        "Ct" =>  0.2,
                        "Cs" =>  0.2,
                        "Cw" =>  -0.1,
                        "Cp" =>  20.,
                        "Ci" =>  1.5)

cell_user_spec = Dict{String, Any}("cell_size_upper_limit" => 100.,
                        "max_split" => 1.,
                        "area_lambda" => 10000.,
                        "temperature" =>  1.,
                        "chemotaxis_max_strength" => 5000.,
                        "grid_size" =>  30)

evo_user_spec = Dict("n_total_genes" => 35,
                            "n_evaluation" => 20,
                            "time_steps" => 800,
                            "adult_niche_size" =>  10,
                            "child_map_size" =>  8,
                            "initial_pop_size" =>  500,
                            "families" => [1,2,3,4],
                            "elitism_percentage" =>  0.4,
                            "inter_species_crossover" => 0.2,
                            "normal_crossover" =>  0.8,
                            "non_elite_crossover" =>  0.2,
                            "genome_start" => [1,1,1,1,1,2,2,2,3,4])

exp_config = ExperimentalConfig(exp_user_spec)
grn_config = FGRNConfig(grn_user_spec)
cell_config = CellularConfig(cell_user_spec)
evo_config = EMAPConfig(evo_user_spec)
```

B.3 Experiment 3

```
exp_user_spec = Dict{String, Any}("diffusion_alpha" => "evolve",
                        "chemotaxis_mu" => "evolve",
                        "external_signal_type" => "type",
                        "external_signal_receptor_strength" => "evolve")

grn_user_spec = Dict{String, Any}("protein_size" => 15,
                        "max_iter" => 55,
                        "mandel_thresh" => 2,
                        "Ct" =>  0.2,
                        "Cs" =>  0.2,
                        "Cw" =>  -0.1,
                        "Cp" =>  20.,
                        "Ci" =>  1.5)

cell_user_spec = Dict{String, Any}("cell_size_upper_limit" => 100.,
                        "max_split" => 1.,
                        "area_lambda" => 10000.,
                        "temperature" =>  1.,
                        "chemotaxis_max_strength" => 5000.,
                        "grid_size" =>  30)

evo_user_spec = Dict("n_total_genes" => 35,
                            "n_evaluation" => 20,
                            "time_steps" => 800,
                            "adult_niche_size" =>  10,
                            "child_map_size" =>  8,
                            "initial_pop_size" =>  500,
                            "families" => [1,2,3,4],
                            "elitism_percentage" =>  0.4,
                            "inter_species_crossover" => 0.2,
                            "normal_crossover" =>  0.8,
                            "non_elite_crossover" =>  0.2,
                            "genome_start" => [1,1,1,1,1,2,2,2,3,4])

exp_config = ExperimentalConfig(exp_user_spec)
grn_config = FGRNConfig(grn_user_spec)
cell_config = CellularConfig(cell_user_spec)
evo_config = EMAPConfig(evo_user_spec)
```

Appendix C

Colophon

This document was set in the Times Roman typeface using LaTeX and BibTeX, composed with TeXstudio.

Appendix D

Acronyms

CP

Cellular Potts.

References

[1] Andrew Adamatzky. *Game of life cellular automata*. Vol. 1. Springer, 2010.

[2] Enrique Alba and Marco Tomassini. "Parallelism and evolutionary algorithms". In: *IEEE transactions on evolutionary computation* 6.5 (2002), pp. 443–462.

[3] Bruce Alberts et al. "Molecular biology of the cell". In: *SCANDINAVIAN JOURNAL OF RHEUMATOLOGY* 32.2 (2003), pp. 125–125.

[4] Vivi Andasari et al. "Integrating intracellular dynamics using CompuCell3D and Bionetsolver: applications to multiscale modelling of cancer cell growth and invasion". In: *PloS one* 7.3 (2012), e33726.

[5] Ariel Balter et al. "The Glazier-Graner-Hogeweg model: extensions, future directions, and opportunities for further study". In: *Single-Cell-Based Models in Biology and Medicine*. Springer, 2007, pp. 151–167.

[6] Wolfgang Banzhaf. "On the dynamics of an artificial regulatory network". In: *European Conference on Artificial Life*. Springer. 2003, pp. 217–227.

[7] Peter J Bentley. "Evolving beyond perfection: An investigation of the effects of long-term evolution on fractal gene regulatory networks". In: *Biosystems* 76.1-3 (2004), pp. 291–301.

[8] Peter J Bentley. "Evolving fractal proteins". In: *International Conference on Evolvable Systems*. Springer. 2003, pp. 81–92.

[9] Peter J Bentley. "Fractal proteins". In: *Genetic Programming and Evolvable Machines* 5.1 (2004), pp. 71–101.

[10] Peter J Bentley, Sanjeev Kumar, et al. "Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem." In: *GECCO*. Vol. 99. 1999, pp. 35–43.

[11] Natalie Berestovsky et al. "Modeling integrated cellular machinery using hybrid petri-boolean networks". In: *PLoS computational biology* 9.11 (2013), e1003306.

[12] Terry Bisson. “They’re Made out of Meat”. In: *VOICES-NEW YORK-* 39.1 (2003), pp. 66–68.

[13] Josh C Bongard and Chandana Paul. “Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution”. In: *From Animals to Animats: The Sixth International Conference on the Simulation of Adaptive Behaviour*. Citeseer. 2000.

[14] Chris P Bowers. “Formation of modules in a computational model of embryogeny”. In: *2005 IEEE Congress on Evolutionary Computation*. Vol. 1. IEEE. 2005, pp. 537–542.

[15] Terence A Brown. “The human genome”. In: *Genomes. 2nd edition*. Wiley-Liss, 2002.

[16] Tom B Brown et al. “Language models are few-shot learners”. In: *arXiv preprint arXiv:2005.14165* (2020).

[17] David M Bryant and Keith E Mostov. “From cells to organs: building polarized tissue”. In: *Nature reviews Molecular cell biology* 9.11 (2008), pp. 887–901.

[18] F Piccini Cercato, José CM Mombach, and Gerson GH Cavalheiro. “High performance simulations of the cellular Potts model”. In: *20th International Symposium on High-Performance Computing in an Advanced Collaborative Environment (HPCS’06)*. IEEE. 2006, pp. 28–28.

[19] Arturo Chavoya and Yves Duthen. “A cell pattern generation model based on an extended artificial regulatory network”. In: *Biosystems* 94.1-2 (2008), pp. 95–101.

[20] Arturo Chavoya and Yves Duthen. “Evolving cellular automata for 2D form generation”. In: *Proceedings of the Ninth International Conference on Computer Graphics and Artificial Intelligence 3IA*. Vol. 2006. Citeseer. 2006, pp. 129–137.

[21] K-H Cho et al. “Reverse engineering of gene regulatory networks”. In: *IET Systems Biology* 1.3 (2007), pp. 149–163.

[22] Jean Disset, Sylvain Cussat-Blanc, and Yves Duthen. “Self-organization of symbiotic multicellular structures”. In: (2014).

[23] Jean Disset et al. “A comparison of genetic regulatory network dynamics and encoding”. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2017, pp. 91–98.

[24] Peter Eggenberger et al. “Evolving morphologies of simulated 3D organisms based on differential gene expression”. In: *Proceedings of the fourth european conference on Artificial Life*. Citeseer. 1997, pp. 205–213.

[25] Wilfried Elmenreich and István Fehérvári. “Evolving self-organizing cellular automata based on neural network genotypes”. In: *International Workshop on Self-Organizing Systems*. Springer. 2011, pp. 16–25.

[26] Donald E Fosket et al. *Plant growth and development: a molecular approach.* Academic Press Inc., 1994.

[27] William Gilpin. “Cellular automata as convolutional neural networks”. In: *Physical Review E* 100.3 (2019), p. 032402.

[28] François Graner and James A Glazier. “Simulation of biological cell sorting using a two-dimensional extended Potts model”. In: *Physical review letters* 69.13 (1992), p. 2013.

[29] ES Harris et al. “Biased eukaryotic gene regulation rules suggest genome behaviour is near edge of chaos”. In: *Santa Fe Institute*. 1997, pp. 97–05.

[30] Marc Jakoby and Arp Schnittger. “Cell cycle and differentiation”. In: *Current opinion in plant biology* 7.6 (2004), pp. 661–669.

[31] Michał Joachimczak and Borys Wróbel. “Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension”. In: *European Conference on Artificial Life*. Springer. 2009, pp. 35–43.

[32] Michal Joachimczak and Borys Wróbel. “Open ended evolution of 3d multicellular development controlled by gene regulatory networks”. In: *Artificial Life Conference Proceedings 12*. MIT Press. 2012, pp. 67–74.

[33] Stuart Kauffman. “Gene regulation networks: A theory for their global structure and behaviors”. In: *Current topics in developmental biology*. Vol. 6. Elsevier, 1971, pp. 145–182.

[34] J Knabe, M Schilstra, and Chrystopher L Nehaniv. “Evolution and morphogenesis of differentiated multicellular organisms: autonomously generated diffusion gradients for positional information”. In: *Artificial Life XI* (2008).

[35] Sanjeev Kumar and Peter J Bentley. “Computational embryology: past, present and future”. In: *Advances in evolutionary computing*. Springer, 2003, pp. 461–477.

[36] Joel Lehman and Kenneth O Stanley. "Evolving a diversity of virtual creatures through novelty search and local competition". In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. 2011, pp. 211–218.

[37] Michael Levine and Eric H Davidson. "Gene regulatory networks for development". In: *Proceedings of the National Academy of Sciences* 102.14 (2005), pp. 4936–4942.

[38] Harley H McAdams and Adam Arkin. "Stochastic mechanisms in gene expression". In: *Proceedings of the National Academy of Sciences* 94.3 (1997), pp. 814–819.

[39] Barry M McCoy and Tai Tsun Wu. *The two-dimensional Ising model*. Harvard University Press, 2013.

[40] Georgios Methenitis et al. "Novelty search for soft robotic space exploration". In: *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*. 2015, pp. 193–200.

[41] Julian Francis Miller. "Evolving a self-repairing, self-regulating, french flag organism". In: *Genetic and Evolutionary Computation Conference*. Springer. 2004, pp. 129–139.

[42] Alexander Mordvintsev et al. "Growing neural cellular automata". In: *Distill* 5.2 (2020), e23.

[43] Jean-Baptiste Mouret and Jeff Clune. "Illuminating search spaces by mapping elites". In: *arXiv preprint arXiv:1504.04909* (2015).

[44] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. "Innovation engines: Automated creativity and improved stochastic optimization via deep learning". In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. 2015, pp. 959–966.

[45] Athanasios Polynikis, SJ Hogan, and Mario di Bernardo. "Comparing different ODE modelling approaches for gene regulatory networks". In: *Journal of theoretical biology* 261.4 (2009), pp. 511–530.

[46] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. "Quality diversity: A new frontier for evolutionary computation". In: *Frontiers in Robotics and AI* 3 (2016), p. 40.

[47] Dana Reichmann et al. "The molecular architecture of protein–protein binding sites". In: *Current opinion in structural biology* 17.1 (2007), pp. 67–76.

[48] Paula J Rudall. "Colourful cones: how did flower colour first evolve?" In: *Journal of experimental botany* 71.3 (2020), pp. 759–767.

[49] Nicholas J Savill and Paulien Hogeweg. “Modelling morphogenesis: from single cells to crawling slugs”. In: *Journal of theoretical biology* 184.3 (1997), pp. 229–235.

[50] Luís F Seoane. “Evolutionary aspects of reservoir computing”. In: *Philosophical Transactions of the Royal Society B* 374.1774 (2019), p. 20180377.

[51] Ilya Shmulevich, Edward R Dougherty, and Wei Zhang. “From Boolean to probabilistic Boolean networks as models of genetic regulatory networks”. In: *Proceedings of the IEEE* 90.11 (2002), pp. 1778–1792.

[52] Karl Sims. “Evolving 3D morphology and behavior by competition”. In: *Artificial life* 1.4 (1994), pp. 353–372.

[53] Lindsay I Smith. “A tutorial on principal components analysis”. In: (2002).

[54] Kenneth O Stanley and Risto Miikkulainen. “A taxonomy for artificial embryogeny”. In: *Artificial life* 9.2 (2003), pp. 93–130.

[55] Till Steiner, Yaochu Jin, and Bernhard Sendhoff. “Vector field embryogeny”. In: *PLoS One* 4.12 (2009), e8177.

[56] Alan Mathison Turing. “The chemical basis of morphogenesis”. In: *Bulletin of mathematical biology* 52.1 (1990), pp. 153–197.

[57] Anja Voss-Böhme. “Multi-scale modeling in morphogenesis: a critical analysis of the cellular Potts model”. In: (2012).

[58] Daniel C Weaver, Christopher T Workman, and Gary D Stormo. “Modeling regulatory networks with weight matrices”. In: *Biocomputing’99*. World Scientific, 1999, pp. 112–123.

[59] Dennis G Wilson et al. “Evolving Differentiable Gene Regulatory Networks”. In: *arXiv preprint arXiv:1807.05948* (2018).

[60] Margaret Wilson. “Six views of embodied cognition”. In: *Psychonomic bulletin & review* 9.4 (2002), pp. 625–636.

[61] Fa-Yueh Wu. “The potts model”. In: *Reviews of modern physics* 54.1 (1982), p. 235.

[62] N Wulff and J A Hertz. “Learning cellular automaton dynamics with neural networks”. In: *Advances in Neural Information Processing Systems* 5 (1992), pp. 631–638.