UNIVERSITY OF SOUTHAMPTON

An Investigation into the Evolution of Hierarchy and its consequences for Evolvability

by

Frederick J. Nash

A thesis submitted for the partial fulfilment of
degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

June 2022

Research Thesis: Declaration of Authorship

Print name: Frederick James Nash

Title of thesis: An Investigation into the Evolution of Hierarchy and its consequences for Evolvability

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research. I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.
7. Parts of this work have been submitted for publication, but have not been publish.

Signature:

Date: 27th September 2021

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Frederick J. Nash

This thesis is concerned with the evolution of hierarchical modules in a model of gene regulation, and the consequences thereof for evolvability. Developmental processes map genotypes to phenotypes, and translate random variation at the genetic level into biased, selectable variation at the phenotypic level. These developmental processes are themselves subject to evolution by natural selection and it might be the case that natural selection favours developmental architectures that facilitate phenotypic variation that is adaptive and enhances evolvability. One manner of developmental organisation that has inspired much interest is modular hierarchy. Such hierarchy - where one gene directs many others - has the potential to be very important to evolvability because it effectively rescales the variability of phenotypes, enabling natural selection to search combinations of modules rather than combinations of individual genes. However, the conditions where natural selection favours hierarchical organisation and the conditions where its consequences enable such rescaling are not well understood.

Considering a developmental model based on a recurrent regulatory process, we describe conditions where natural selection favours the evolution of single-layer hierarchical modular structures, where independent ‘switch’ genes direct independent subsets of genes. We show that these structures increase evolvability by rescaling the genetic neighbourhood of phenotypes, from combinations of genes to combinations of modules, and that this makes high-fitness phenotypes more accessible to natural selection. This improved evolvability enables a micro-evolutionary process to better exploit a changing or static modular environment so long as sufficient long-term variation is maintained.

We then investigate the underlying cause of the evolution of hierarchy. Interestingly, we find that the observable increase in evolvability (in particular, the ability to rescale the variability of phenotypes) is not required for natural selection to favour hierarchy in this model. Rather, hierarchy evolves due to a selective pressure for efficient phenotypic expression and because it is an efficient organisation for increasing the expression of many genes given limited regulatory connections. Thereby, we show that - in some cases - the causes and consequences of developmental hierarchy are not the same. That is, hierarchy evolves - and it increases evolvabilty - but increased evolvability need not be the reason it was favoured by selection.

Contents

Nomenclature **xv**

Acknowledgements **xvii**

1 Introduction **1**

2 Background **7**
2.1 Evolvability 7
2.2 Development 9
2.3 The Evolution of Evolvability in Network Models 12
2.3.1 The Evolution of Modularity and Hierarchy in Network Models . . 13
2.4 Learning in Network Models 17

3 The Evolution of Hierarchy and its Consequences for Evolvability **19**
3.1 Introduction 19
3.2 Methodology 23
3.2.1 Developmental Model 24
3.2.2 Evolutionary Model 25
3.2.3 Model of a Modularly Changing Environment 27
3.3 Experimental Results 31
3.4 Conclusions 36

4 Rescaling Variability with Hierarchy **39**
4.1 Introduction 39
4.2 Partial Resets 41
4.3 Model of an Unchanging Modular Environment 44
4.4 Smooth MC 46
4.5 Spiky MC 50
4.6 A Continuous Hierarchical If-and-only-if 52
4.7 Discussion 58
4.7.1 Amount of Drift and Two Time-scales 62
4.7.2 Limits to Rescaling 63
4.8 Conclusions 66

5 The Efficiency of a Hierarchical Regulatory Topology **67**
5.1 Introduction 67
5.2 The Evolution of a Hierarchy with a dominant gene 68
5.3 Proof of the Optimality of Hierarchy under Assumptions 72

5.3.1 Assumptions . . . 72
5.3.2 Proof . . . 73
5.3.3 Discussion . . . 77
5.4 Fitness Gradients toward Hierarchy . . . 79
5.5 Saturation . . . 81
5.5.1 Saturation Points . . . 81
5.5.2 Changing Saturation Points . . . 83
5.5.3 The evolution of modularity . . . 90
5.6 Consequences of Model Parameters . . . 91
5.6.1 System Size . . . 92
5.6.2 Temporal Parameters . . . 92
5.6.3 The Importance of Discontinuity . . . 93
5.6.4 Other Saturation Point Considerations . . . 94
5.7 Comparisons between Mutations . . . 95
5.8 Conclusions . . . 96

6 The Evolutionary causes of Hierarchy **97**
6.1 Transient Dominance . . . 98
6.2 Feedback . . . 101
6.2.1 Single-Peaked Environment . . . 103
6.2.2 Rugged environment . . . 105
6.3 Discussion . . . 106
6.4 Conclusions . . . 109

7 Conclusions **111**
7.1 Future Work . . . 113

A Definitions and Proofs **117**
A.1 Proof of Bounded Phenotypic Expression . . . 117
A.2 Degree of Hierarchy . . . 118

B Musing on the MC problem **119**
B.1 Proof of equivalence of generalised quadratic MC with standard MC . . . 119
B.2 Partial resets without development . . . 121

C Select additional results **123**
C.1 Time to hierarchy with changing environmental conditions . . . 123
C.2 Changing environment model with different cost coefficients . . . 123
C.3 Changing environment with a strictly sub-additive cost of connections . . . 123
C.4 Fixed environment and initial gene expressions . . . 126
C.5 Spiky MC without a cost of connections . . . 127

D Gradient descent example **129**

E Partial hierarchies **131**

F Sources **133**

Bibliography **135**

List of Figures

3.1 Different regulatory networks have different consequences for the variability of the phenotype by changing the effects of variation in the genotype. 21
3.2 The squash function $tanh(x/2)$. 25
3.3 Module instances in a changing environment. 30
3.4 The evolution of independent internally hierarchical modules. 33
3.5 Hierarchy reflects evolved modules 34
3.6 Changes due to the evolution of hierarchy. 35
3.7 Hierarchy enables adaptation in a multi-peaked environmental instance . 36
3.8 Escaping local fitness peaks. 37

4.1 Cross-section of the 'Smooth' MC Problem 46
4.2 Terminal fitness distributions with the Smooth MC without cost of connections. 48
4.3 Terminal fitness distributions for the Smooth MC with cost of connections. 49
4.4 Cross-section of the Spiky MC Problem 51
4.5 Regulatory connection trajectories for the Spiky MC problem with L1 cost of connections. 52
4.6 Early switches for the Spiky MC problem with L1 cost of connections. . . 53
4.7 Phenotype patterns for the Spiky MC problem with L1 cost of connections. 53
4.8 Terminal fitness distributions for the GRN hillclimber with cost of connections in the Spiky MC environment. 54
4.9 Cross-section of the Spiky Continuous-HIFF 55
4.10 The Spiky Continuous-HIFF, Decomposed 56
4.11 Evolutionary trajectories of regulatory connections. 57
4.12 Evolution finds fitter phenotypes as the system becomes able to coordinate larger modules 58
4.13 Self perpetuating hierarchy solves CHIFF 59
4.14 Larger partial resets prevent over-fitting to sub-optimal solutions in the Spiky Continuous-HIFF. 60
4.15 Longer developmental periods allow better exploitation of evolved hierarchy. 61
4.16 Each layer of hierarchy reveals the next layer of modules. 61
4.17 Module coordination when $T = 8$ 65
4.18 Module coordination when $T = 14$ 65

5.1 Optimal assignments of s and r 70
5.2 Fitness curve for varying s proportion 70
5.3 'Identity' and Hierarchical Topologies. 71
5.4 Examples of sparse topologies. 78

5.5 Incidence of developmental-time dominance. 80
5.6 Time-to-hierarchy for varying duty cycle. 85
5.7 Time-to-hierarchy for varying duty cycle. 86
5.8 The evolution of hierarchy within episodes. 87
5.9 Example degree-of-hierarchy trajectories. 88
5.10 Time-to-hierarchy for varying c_L. 89
5.11 The evolution of independent modules. 91

6.1 Flavours of transient dominance 99
6.2 Plot of marginal fitness for different proportions of switched influence. . . 100
6.3 The B dominance constraint 102
6.4 Example trajectories in single-peaked environment without dominance constraints 104
6.5 Example trajectories in single-peaked environment with G dominance constraint 104
6.6 Example trajectories in single-peaked environment with B dominance constraint 105
6.7 Example trajectories in single-peaked environment with both B and G dominance constraint 105
6.8 Example trajectories in rugged environment with B dominance constraint 106
6.9 Example trajectories in rugged environment with G dominance constraint 107
6.10 Example trajectories in rugged environment without dominance constraints107
6.11 Example trajectories in rugged environment with both B and G dominance constraint 108

B.1 Module mean distributions after a reset ($Q = 20k$). 122

C.1 Time-to-hierarchy for different values of Z. 124
C.2 Time-to-hierarchy regressed against $1/|Z-1|$. 124
C.3 Terminal fitness distribution with varying Z. 125
C.4 Evolves GRNs with different choices of Z and λ. 125
C.5 GRNs evolved with a strictly sub-additive cost of connections. 126
C.6 Trajectories for a fixed environment and fixed G vector. 126
C.7 Terminal fitness distributions for the evolutionary process without a cost of connections in the Spiky MC environment. 127

E.1 Two-column partial hierarchy. 131

List of Tables

3.1 Parameters of the developmental model 25
3.2 Parameters of the evolutionary model 27
3.3 Parameters for the changing modular environment 32

4.1 Parameters for the MC Problem 47
4.2 Parameters for the evolutionary model without cost of connections 47
4.3 Parameters for the evolutionary process with cost of connections 48
4.4 Parameters for the Spiky Continuous-HIFF 56

5.1 Parameters for the duty-cycle experiments 84

6.1 Parameters for the feedback experiments 103

Listings

Nomenclature

N	The number of genes *OR* the number of modules in the MC problem.
G	The vector of initial gene expressions.
B	The developmental regulatory interaction matrix.
P	The vector of terminal (phenotypic) gene expressions.
K	The number of evolutionary steps per episode *OR* the number of genes per module in the MC problem.
σ	The developmental squash function.
h	The developmental squash constant.
ϕ	The regulatory interaction cost function.
λ	The cost coefficient.
η_m	The module-benefit-function for module m.
c_m	The module-benefit coefficient for module m.
Z	The frequency of multi-peaked environmental module instances.
Q	The number of neutral (reset) mutations at the start of each epoch or episode.
d_f	The decay factor for hierarchical environments.
D	The duty cycle of high-fitness environmental conditions.
c_H	The high benefit coefficient.
c_L	The low benefit coefficient.
M_G	The magnitude of mutations applied to G.
M_B	The maximum magnitude of mutations applied to B.
R_B	The probability of mutating B rather than G.
f'	The first derivative of a unary function f.
f''	The second derivative of a unary function f.

Acknowledgements

This work was supported by an ECS DTP Studentship.

The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

This work was greatly influence by invaluable discussion with David Prosser, Jamie Caldwell, Christoph Thies, Laura Mears, and Danesh Tarapore. None of this would have happened without the endless guidance of Richard Watson. Thanks must also be extended to the wildlife around Southampton, in particular the corvids and house sparrows.

Finally, my thanks to Tafta Nugraha, the University of Southampton Wireless Society, and the ECS Programming Help-desk.

Chapter 1

Introduction

Evolution is the process by which species change over many generations. A simple mechanism based on heritable variation in reproductive success goes a long way to explain the emergence of organisms that are well-suited to the environments in which they exist. However, evolution is a 'local search' process: it is inherently limited in its capacity to escape from 'local fitness peaks' (local optima), which may trap it in sub-optimal regions of the phenotype space. If an evolutionary process is to find high fitness phenotypes, it is necessary that it has suitable variability, else those phenotypes will be inaccessible, regardless of how long the process spends searching. This dissertation is concerned with how such variability may emerge as the result of the evolution of hierarchy in a gene regulation network, and its consequences for 'evolvability'.

The term evolvability bares various definitions (Houle, 1992; Lynch, 2007; Wagner, 2005, 2008; Wagner and Altenberg, 1996; Hansen, 2006), and some authors have sought to make a distinction between the evolvability of populations and the evolvability of individuals (Wilder and Stanley, 2015); of genotypes and of phenotypes (Wagner, 2008; Hansen, 2006). Here we consider the general definition provided by Payne and Wagner, "the ability of a biological system to produce phenotypic variation that is both heritable and adaptive" (Payne and Wagner, 2019). To a great extent, this ability is direct function of the systems variability. One notion of evolvability is that random mutation can be guided to produce biased variation. Further, evolution itself can change the nature of this variation, implying that evolvability could itself be evolved: how this might occur is an open question (Pigliucci, 2008).

A significant factor in the discussion of evolvability is the role of developmental processes ('evo-devo'), which can be abstracted as mappings between genotypes and phenotypes (Laland et al., 2015; Watson and Szathmáry, 2016; Waddington and Robertson, 1966). These genotype-phenotype maps ('GP maps') impose a developmental bias upon the distribution of phenotypes that can be produced by variation in genotypes due to random mutation, and thereby present an opportunity for a lineage to adapt, by incrementally updating what amounts to a genetic model of the phenotype space. This 'modelling' ability is revealed by the capacity for a GP map to translate homogeneous genetic variation into adaptive variation in the phenotype: while mutations inflicted upon an individual may be random, their phenotypic consequences may be a function of the individual's genetic background, and so its genetic history. As such, the GP map represents an important means by which evolvability might evolve. This dissertation focusses on modularity and hierarchy in the genotype-phenotype map, which both provides means by which evolvability might be achieved (Hansen, 2006; Kashtan et al., 2009; Clune et al., 2013; Mengistu et al., 2016; Watson et al., 2011c; Kouvaris et al., 2017).

Modularity is a prominent feature of many natural and artificial systems (Kashtan et al., 2005). A modular network is one containing "highly connected clusters of nodes that are sparsely connected to nodes in other clusters" (Clune et al., 2013). Modularity is generally recognised as beneficial in facilitating maintainability in 'nearly-decomposable systems' (Simon, 1969), as it allows parts with different jobs to be modified independently: a change in component should not interfere with another. The observation of functional equivalence between genes in different animals suggests that evolution is capable of maintaining modular developmental structure over long timespans (Carroll, 2008), and it has been suggested even that such structure can evolve without providing a fitness advantage (Force et al., 2005; Lipson et al., 2002).

Hierarchy, which is also found in natural systems (Riedl, 1977; Force et al., 2005), suggests a manner of organisation wherein a set of nodes have a role in coordinating some subset of the broader system. Hierarchy is often multi-layered, such that a function is divided into smaller and smaller sub-functions, with nodes coordinating some set of subordinate nodes which themselves coordinate their own (disjoint) sets of subordinate nodes (Force et al., 2005). This tree structure minimises the number of interfaces between components, avoiding polynomial growth of system complexity with the number of components. This conferred scalability may be necessary for the development of larger systems.

Both modularity and hierarchy in a system imply a limited degree of coupling, and facilitate local (independent) variation. This enables phenotype components to change and improve independently (Force et al., 2005), which can enable simple adaptive processes to readily respond to change despite the undirected nature of genetic variation: variation in the genotype is inherited, but fitness is a function of the phenotype and so a product of the genotype-phenotype map Watson and Szathmáry (2016).

Gene regulation is a classic example of a complex developmental process (Carroll, 2008), has inspired many models by various authors (Payne et al., 2014; Ho and Charleston, 2011; Siegal and Bergman, 2002; Wilder and Stanley, 2015), and provides one way to look at modularity (Kouvaris et al., 2017) and hierarchy. Some such models map onto a type of 'Recurrent Neural Net' called a 'Hopfield Network' (Hopfield, 1982; Watson et al., 2011b), and parallels have been drawn between the evolution of higher-order developmental parameters and the explicit models of machine learning (Watson and Szathmáry, 2016; Watson et al., 2016). In this context, it has been shown that evolution can mimic Hebbian Learning (Watson et al., 2014), which amounts to learning recurring correlations in a system. The exploitation of higher-order non-linear dynamics by the evolutionary process in these biologically inspired models provides a powerful tool to investigate spontaneous learning, and suggests that techniques employed by machine learning theorists and practitioners could be applicable to these natural processes (and vice versa).

One pertinent method in machine learning is regularization, which can involve augmenting an objective function with additional 'entropy' terms that impose some constraint on the system parameters. One style of regularisation that originated in regression models is to impose a cost on model coefficients (e.g. the linear (L1) LASSO Tibshirani (1996), which is particularly relevant to this work). Clune et al. (2013) and Mengistu et al. (2016) showed that an analogous cost of connections could promote the evolution of modularity and hierarchy in neural networks, while Kouvaris et al. (2017) demonstrated that applying a cost of connections to a standard model of gene regulation (Siegal and Bergman, 2002; Spirov and Holloway, 2013) can promote the evolution of modular gene regulation networks (GRNs).

The work described in this dissertation primarily comprises computer simulations and interrogation of their dynamics, following from Kouvaris et al. (2017) and Kounios et al. (2016). We consider the same developmental model (based on a recurrent regulatory process as described by Siegal and Bergman (2002)), and describe conditions where natural selection favours the evolution of hierarchy and modularity. We show that these structures can be promoted by the inclusion of a linear cost of connections, and facilitate evolvability by making useful phenotypes more accessible. This evolvability is achieved by essentially reducing the important parameter space of the initial conditions of the developmental process: modules of genes change independently, and a directed by a single 'lead' gene (a 'switch like function' (Erwin and Davidson, 2009)). This allows mutations which affect this single gene to determine the phenotypic expression of modules of genes, effectively changing the level at which variation operates from individual genes to modules of genes.

Chapter 2 will briefly outline some of the key ideas that will be explored in this dissertation, and the past works that have inspired their use and exploration. We will review the meaning and importance of 'evolvability', and explore the abstractions that we use to relate our computer simulations to the real world. This chapter will not discuss any model in detail, but will introduce some of the previous literature that has used the same (or similar) models of gene regulation, and discuss how the concepts of modularity and hierarchy can be related to such a model.

Chapter 3 will start by discussing the distinction between 'rate' evolvability (the ability of a system to evolve accessibly phenotypes more quickly) and 'access' evolvability (the ability of a system to evolve previously inaccessible phenotype). While both have been variously demonstrated by multiple authors (Wagner and Altenberg, 1996; Clune et al., 2013; Mengistu et al., 2016; Kashtan et al., 2005; Kouvaris et al., 2017; Parter et al., 2008)) it can be difficult to separate the two. We attempt to show the evolution of access evolvability specifically. To do so, we evolve a gene regulation network (GRN) in a changing environment comprising independent modules which differentially reward complementary phenotypes: sometimes it is 'easy' for a micro-evolutio process to find the fitter phenotype; sometimes it is not, and the process becomes trapped at a locally (and not globally) optimal phenotype. We find that - so long as there is sufficient long term variation - the system evolves independent modules consistent with the variation in the environment as in Kouvaris et al. (2017). Furthermore, these modules evolve a regulatory topology that facilitates 'module flips', and consequently the system is able to fully exploit the changing environment: it no longer becomes trapped.

Chapter 4 will transfer the observations from Chapter 3 to a modular but unchanging environment, and a hierarchical unchanging environment. The environments used are rugged, and so contain many local optima (fitness peaks) that will readily trap a micro-evolutionary process: the fittest phenotypes are inaccessible from much of the genotype space. The evolution of hierarchy facilities a rescaling of the level at which the evolutionary process operates from genes to modules. This provides a biologically plausible mechanism by which evolution can exploit modularity and hierarchy to access previously inaccessible phenotypes, in much the same way as has been implemented in previous studies (e.g. Iclanzan and Dumitrescu (2007) and Watson et al. (2011c)). We go on to discuss the importance of long-term episodic variation in our models, and extend the results of Kounios et al. (2016).

Chapter 5 will explore the main causes of the hierarchy that evolved in the Chapters 3&4, and show that they can be separated from the evolvability benefits conferred. Crucially, we reject selection for evolvability as a necessary condition for the evolution of hierarchy. We shall discuss the observation that a hierarchical topology which produces a 'dominant' gene consistently evolves in the presence of a linear (L1) cost of connections, and prove that this topology is the one that maximises fitness. We shall also argue that when there is a dominant gene that there is a gradient toward the hierarchical topology, indicating that direct selection for high fitness could provide an explanation for the evolution of hierarchy. We settle on the idea that hierarchy evolved because it is an 'efficient' regulatory topology for producing strongly expressed genes. The understanding acquired in this work enables us to cast the evolution of hierarchy as consequence of differential selection on regulatory connections when a module becomes 'saturated'. It is possible to characterise the effect of changing many model parameters in terms of how they influence the amount of time the system spends saturated, which allows us to model how quickly hierarchy can evolve. We further discuss the importance of the ‘strong-selection weak-mutation’ assumptions in our model, and how the circumstance in which hierarchy is evolved may be different if these are relaxed.

Chapter 6 shows that the conditions to promote hierarchy can occur even when the efficiency benefits of hierarchy are limited by removing the ability for the regulatory network to up-regulate genes to different extents. In this case, selection for hierarchy only occurs during transitionary periods where there is an imbalance between gene expression brought about by changes in the initial developmental conditions. In light of these results, we formulate the idea that the mechanism by which hierarchy evolves is through positive-feedback between dominance in gene expression and hierarchy in gene regulation, whereby genes which exert great influence become systematically more influential, until all genes (within their modules) are directed by them alone.

Chapter 7 will review the observations made in each of the chapters, and highlight the main conclusions drawn from them: hierarchy in gene regulatory networks can increase access evolvability; significant long-term ((episodic) variation is required to infer modular patterns in the selective environments; and that the causes and consequences of evolvability enhancing hierarchy need not be the same. The dissertation will conclude by outlining some potential directions for future work.

Chapter 2

Background

2.1 Evolvability

The term 'evolvability' was originally used in a precise manner in quantitative genetics, but has acquired various definitions over the years. Multiple authors have provided different definitions for the term 'evolvability' in accordance with the targets of their research (Lynch, 2007; Hansen, 2006), and some have become vogue (Wagner, 2005). For instance, Houle (1992) talks about "the ability of a population to respond to natural or artificial selection", discussing the rate at which a population can respond to some selective pressure, and recognises that it is not feasible to define a general measure of the evolvability of a population. Wilder and Stanley (2015) make a clear distinction between evolvable individuals and evolvable populations, concluding that the evolvability of populations - the number of accessible phenotypes - is of paramount concern, which is reflected in various works (Mayer and Hansen, 2017; Jiménez et al., 2015). It is clear that each author has a particular focus, and adopts a meaning which allows them to proceed to discuss their particular concern without confounding matters with unnecessary detail. In this dissertation we focus on the evolvabilty of genotypes. For our purposes, the definition of evolvability provided by Wagner and Altenberg (1996) is salient, and suggests an interpretation whereby

> Evolvability is the genome's ability to produce adaptive variants when acted upon by the genetic system. This is not to say that the variants need to be "directed" (Foster and Cairns 1992) for there to be evolvability, but rather, that they cannot be entirely "misdirected," that there must be some small chance of a variant being adaptive.

Hansen, in suggesting that evolvability is a property of genotype-phenotype maps and not of populations, highlights the distinction between variance and variability, suggesting that statistical measures of evolvability based on variation of traits within populations are misleading (Hansen, 2006). He incorporates a notion of robustness in a definition of evolvability, which he explicitly relates to genotype-phenotype maps:

> Evolvability is the ability of the genetic system to produce and maintain potentially adaptive genetic variants.

Hansen, in particular, writes about epistasis, which he sees an essential tool for understanding evolvability (Hansen, 2013). Epistasis refers to the idea that the fitness effect of a genetic mutation will depend on the genetic background: a mutation may be deleterious in one genotype but provide a fitness benefit in another. As such, the presence of epistasis means that it is possible for past evolution to influence the nature of future evolution. In general, epistasis can be introduced into models of evolution by means of a non-additive genotype-fitness (G-F) map. For the purposes of our discussion, epistasis may be introduced either in the genotype-phenotype (G-P) map, or the phenotype-fitness (P-F) map, the former of which can be thought of as development, the later as fitness evaluation (how successful the developed individual is in the environment in which it lives).

Along with the various definitions of evolvability, there have inevitably been suggested various schemes for measuring the evolvability of a system or population. Simple definitions have been provided which are applicable to systems of discrete phenotype spaces, which involve counting the number of phenotypes that are accessible by mutation from points of genotype or phenotype space. These metrics where introduced and employed to suggest that while robustness and evolvability may seem at odds, when a distinction is drawn between genotype and phenotype, it is not unreasonable to expect ‘phenotypic robustness’ (small changes in genotype space rarely resulting in a change in phenotype) should co-occur with high ‘genotypic evolvability’, whereby small changes in genotype can result in many new phenotypic structures (Wagner, 2008). Such observations are applicable to various biologically inspired dynamic processes, as presented in Steinacher et al. (2016) with a model of gene regulation, though recent discussion has shown that the situation cannot be so simple: Mayer and Hansen (2017) indicates that in general there cannot be a strong propensity for robustness and evolvability to co-occur, and that previous observations may have been limited to biologically unreasonable scenarios. While these later observations do not undermine any previous work, they highlight an ever-present concern with research in this field, that seemingly small details can have profound implications, and that the diversity of ways in which additional complexity may be introduced together present a concern as to the biological applicability of such work.

2.2 Development

'Development' is a slightly friendlier term than 'ontogenesis' often used to refer to the growth of an embryo – a relatively simple entity with a complete genetic makeup – to an adult organism capable of reproduction. Loosely, it is a mechanism by which the genotype of an individual is transformed into a phenotype, as an egg grows into a pigeon.

Developmental processes have long been recognised as a potential mechanism to facilitate evolvability by means of directed variation, and particular attention was brought to its significance by Waddington and Robertson (1966), who compared development to a ball rolling down an 'epigenetic landscape': the initial position of the ball is genetically determined, and the shape of the epigenetic landscape will determine where the ball rolls (which terminal phenotype is produced). He used this idea to explain how a phenotype may become robust to genetic variation, as a small change in the ball's initial position won't influence its final resting position if the path is sufficiently 'canalized' (a landscape with deep canals will direct the ball the same way when its initial position is perturbed slightly). Hansen (2006) described canalization in terms of epistasis, such that the fitness gradient in a region of genotype space is shallow.

Research into the evolution of developmental processes continues under the title 'evo-devo' (Müller, 2007) and as part of the broader 'Extended Evolutionary Synthesis' (Laland et al., 2015). Some elements of the developmental process are sometimes cast as a genotype-phenotype mapping (G-P map), where the genotype is the genetic information inherited from its parent(s) and the phenotype determines its fitness. The separation of genotype and phenotype is important because the G-P map may translate random, undirected variation in the genotype to biased variation in the phenotype. The biases in the developmental process may significantly limit the space of accessible phenotypes, leading to "developmental constraints" (Smith et al., 1985):

> A developmental constraint is a bias in the production of a variant phenotype or a limitation on phenotypic variation caused by the structure, characteristics, composition, and dynamics of the developmental system

Such constraints have a direct evolvability consequences: they may be beneficial in preventing the developmental of unviable types (improved robustness), may facilitate useful phenotypic variation, or may unhelpfully limit the accessibility of fit phenotypes. Crucially, the developmental process itself is a product of past evolution, and so the nature of the constraints on a lineage may change over time (Watson and Szathmáry, 2016).

Observations like Hansen's highlight an important consideration when a separation between genotype and phenotype is employed: evolution cannot strictly change the genotype-fitness map. Consequently, we must be careful when we describe an evolutionary change in the genotype which appears to produce a change in the genotype-phenotype map; rather, the evolutionary process is exploring a different region of the genotype-fitness map, one with different properties (of particular importants to Hansen, the nature of the local epistasis may be different). None-the-less, it is instructive to make this separation, as it allows us to discuss the evolution of specific phenotypes, rather than deferring to changes in fitness.

Hansen (2013) discusses how non-linear developmental processes introduce epistasis, and that ignoring development is akin to ignoring the higher-order components of a series expansion (i.e. only considering the tangent of a fitness curve). While useful predictions can be made about short-term evolution without considering the developmental process, this omission will fail to generalise as you move further from the original genetic background (the tangent tells you little about the global properties of a curve). Furthermore, in the absence of some other source of epistasis (e.g. in the determination of fitness), such a model is unable to model evolvability. This important observation provides some indication that the various complexities introduced to better model biological systems (e.g. development, niche-construction) are necessary to make inferences about the real world.

One aspect of development that has been treated as form of G-P map is the regulation of gene expression. Many models of gene regulation have been used (Ho and Charleston, 2011; Spirov and Holloway, 2013), including boolean nets (Payne et al., 2014), petri-nets, Bayesian nets, Hopfield Networks (Hopfield, 1982), and other systems of differential equations (Hopfield, 1982; Watson et al., 2014; Siegal and Bergman, 2002; Steinacher et al., 2016). Such models usually have a small state space, and encode, along with the initial conditions, some information about transitions between states (e.g. how genes influence one-another). These transitions may be discrete or continuous, but often the dynamics allow only a limited number of stable terminal states, each reachable from multiple embryonic states, which provides a directly analogy to developmental constraints (Watson et al., 2014; Smith et al., 1985) and canalization Waddington and Robertson (1966); Siegal and Bergman (2002).

This dissertation is concerned with a variation on the continuous model of gene regulation introduced by Siegal and Bergman (2002), which has previously been compared to a Hopfield network (Kounios et al., 2016). In this model, the expression of each gene is represented as a vector of gene expression levels, and the change in expression levels is governed by a matrix of interactions. Siegal and Bergman (2002) used this model to demonstrate how the principles of canalization (from Waddington and Robertson (1966)) could depend on the nature of gene regulation, and to show that no special evolutionary explanation is needed for properties like robustness and evolvability: they may be an inevitable consequence of development.

While this continuous model is capable of producing oscillations (i.e. the developing gene expressions may not settle in some cases), in practise the regulatory networks that are explored tend to be consistent - if not symmetric - and guide the developmental process to a handful of possible (stable) states. These developmental constraints were explored by Watson et al. (2014), showing that a gene regulation network could evolve to reliably reproduce one of two possible patterns, but that the regulatory network imposed significant constraints on what could be developed from different initial conditions despite the theoretically large space of possible phenotypes. They considered two versions of the gene regulation model: one 'linear'[1] with no epistatic effects, and one 'non-linear' with ample opportunity for epistasis. They found that the non-linear regime was able to evolve a 'developmental memory' of multiple previously evolved phenotypes (corresponding to previous selective environments); the linear model could not (as indicated by Hansen (2013)). This 'memory' was a consequence of what amounted to correlation learning in the regulatory connections: the strength of the regulatory connection between each gene was proportional to the average 'signal' (the proportion of the time the genes were positively correlated minus the proportion of the time the genes were negatively correlated). This was compared to Hebbian learning, and the GRN was described as a 'self-modelling system' (Watson et al., 2011a,c; Power et al., 2015): it evolves in such a manner as to reproduce previously experienced states.

Kounios et al. (2016) proceeded to explore the same (non-linear) model of gene regulation as Watson et al. (2014). Kounios et al. (2016) evolved the initial expression of the genes and the connections between them in a fixed, epistatic environment, but with regular 'resets', representing periods of genetic drift. Because the system was regularly partially reset, it would explore many different (locally optimal) phenotypes, and so the evolving connections in the GRN would again reflect the average signal. Though the selective environment was chosen to have only *local* epistatic constraints (disparate genes experienced almost no signal on average), so long as the rate at which the GRN evolved was slow enough, the system was able to integrate these signals and ultimately canalized the globally optimal phenotypes[2].

Models of gene regulation with a non-continuous representation of gene expression (such as the model from Wagner (1994) whence the Siegal and Bergman (2002) model is derived, and that in Crombach and Hogeweg (2008)) can show many of the same behaviours as the continuous models; however, because the non-continuous models have an inherently limited space of phenotypes (they only record whether a gene is strongly or weakly expressed, rather than to what extent they are expressed), they do not lend themselves to self-modelling as described by Watson et al. (2011c): in discontinuous models, fitness evaluation may, for example, be performed by computing the hamming distance between

[1] The terminal gene expression is a linear function of the initial gene expression; it is a polynomial function of the connections in the GRN.

[2] The model had 2 complementary globally optimal phenotypes, both stable states for the same GRN.

the phenotypic gene expression and a target phenotype (Wagner, 1994) or by the number of inter-gene constraints that are satisfied. In a continuous model, fitness evaluation may depend on how strongly the genes are expressed, and so there is opportunity for directional selection: even if all genes are strongly expressed (as opposed to weakly), there my be a fitness benefit to making them *more* strongly expressed. This property is crucial to the results in Watson et al. (2011c) and others that follow (Watson et al., 2014; Kounios et al., 2016; Kouvaris et al., 2017).

2.3 The Evolution of Evolvability in Network Models

The evolution of evolvability is of interest not only because it may help to explain the adaptability of biological systems, but also because of the seemingly implausible notion that natural selection should favour beneficial future variability when it can only select from present variation. While many features have been suggested that may increase the evolvability of organisms, it remains an open question how many of these features might evolve: one challenging question is to what extent evolvability evolves in response to selection for evolvability itself (Pigliucci, 2008; Payne and Wagner, 2019; West-Eberhard, 2019; Wagner and Draghi, 2010).

Draghi and Wagner (2008) discuss some general suggestions as to why selection for evolvability may be infeasible, and cite evidence to the contrary, suggesting that selection for evolvability may indeed be a significant factor. Addressing the idea that evolution is myopic and incapable of anticipating the (unknown) future, they say

> ... evolutionary biology contains several frameworks for understanding adaptation, such as geometric mean fitness (Stearns, 2000) and lifetime reproductive success, in which selection, by integrating information about the past, appears to anticipate the future. Seen in this context, this objection to the evolution of evolvability is simply an empirical question about how well past environments predict future ones, and not a logical paradox.

Watson and Szathmáry (2016) make a similar observation, comparing evolution to a learning process, and suggest that the evolution of evolvability is no more mysterious than the ability of (human designed) machine learning models to generalise over a training set, and thereby achieve good performance against a (previously unseen) test set drawn from the same distribution of problems. In the evolutionary sense, this distribution of problems could be collections of temporally or spatially separated environmental conditions which provide fitness landscapes with ‘structural regularities’: experience in one set of such environmental conditions may enable a population to evolve a ‘model’ of these regularities, and consequently be better able to adapt in newly - or repeatedly

- experienced environmental conditions. It may even be that the evolution of the model could occur under any sort of adaptive selection.

Riedl (1977) describes a possible mechanism by which evolvability may be increased as a result of selection preserving mutations which have a positive epistatic effect: if an evolvability enhancing mutation is accepted into a population for any reason (perhaps even drift), it may remain in the population if other mutations occur which depend on its epistatic contributions. Another mechanism by which evolvability could be directly selected is presented by Pavlicev et al. (2011), where the propensity of mutations to occur is determined by a 'G' matrix of coefficients, which effectively determine the principle directions in genotype space in which variation operates.

A more general mechanism by which evolvability might evolve is lineage selection, a phenomenon whereby diversity in a population permits selection for variation with long-term – rather than immediate – fitness consequences, provides a possible explanation for selection for evolvability Kirschner and Gerhart (1998); Virgo et al. (2017); Nunney (1999); Watson (2020). Lineage selection can be understood if we consider a heterogeneous population containing two types, one of which is better able to respond to a changing environment than the other. If the environment changes, the more responsive lineage will out-compete the other during the transitional period where the whole population is responding. Consequently, in the absence of recombination, the genes which conferred the greater evolvability will proliferate so long as they have a non-deleterious fitness effect or the environment continues to vary with a sufficiently high frequency that drift or selection against the evolvability increasing genes can be overcome.

These different mechanisms are largely compatible, and as such even if any one of them may be present in a given scenario, it can be difficult to rule out the others. This can make it difficult to discerne why evolvability emerges both in models and in the real world.

2.3.1 The Evolution of Modularity and Hierarchy in Network Models

Though this dissertation is primarily concerned with the evolutionary causes and consequences of hierarchy, past and present work on hierarchy is often also concerned with modularity, and this dissertation is no exception. Both modularity and hierarchy are related to the concept of sparsity (Espinosa-Soto, 2018), where there are few connections between components, and are natural concepts in the discussion of the topologies of networks and graphs. In the context of a gene regulatory network, the components may be genes (or groups of genes) and the connections the various regulatory interactions between them. Per Clune et al. (2013), "Networks are modular if they contain highly connected clusters of nodes that are sparsely connected to nodes in other clusters". The prevalence and significance of modularity has long been recognised in biological systems, and specifically in gene regulation networks and other developmental systems (Wagner and Altenberg, 1996). More abstractly, the value of modularity in problem solving has previously been demonstrated by work relating to the Building Blocks Hypothesis (Goldberg, 1988), which shows that modularly decomposable problems (or 'nearly-decomposable systems' (Simon, 1969)) can be solved by recombinative evolutionary processes so long as there is strong linkage within modules and weak linkage between them. The challenge for an evolutionary system, then, is to infer the appropriate linkage within and between modules. Intuitively, a modular gene regulatory network would be one there genes which contribute to particular function may be closely connected by regulation, while genes with disparate fitness effects will be largely decoupled. Consequently, mutations may significantly affect just one aspect of morphology or behaviour, making it possible to change different parts of the body or behaviour independently in a modular GRN.

Hierarchy imposes some assumption of asymmetry such that some components in a network may direct the action of many other components. This is also recognised as a potentially significant property of biological networks (Koza, 1994; Hallinan, 2004; Erwin and Davidson, 2009), and it has been shown that biological regulatory networks (which are inherently recurrent systems) can include large hierarchies (Force et al., 2005). One simple - though important - consequence of hierarchy is 'switch-like behaviour' in gene regulation (Yu and Gerstein, 2006): the expression of one gene may determine the expression of many others. On a larger scale, hierarchies can enable reuse and independent evolution of sub-tasks (Yu and Goldberg, 2006; Mengistu et al., 2016; Kashtan et al., 2005) over evolutionary time and as such is an important consideration for research on evolvability. Because hierarchies limit the number of connections between components, they provide the basis for a scalable organisation: potentially many components can coordinate without the need for reliable multi-way communication, which would be problematic when channels of communication are slow or costly to maintain.

Wagner and Altenberg (1996) and Wagner (1996), in discussing possible explanations for the evolvability of modularity, suggest that modularity should not be taken for granted and discussed the mechanisms whereby it might evolve. Using the language of pleiotropy (the propensity for individual genes to influence multiple phenotypic characters (Pavlićev, 2016)), Wagner and Altenberg (1996) suggested that there are two general processes that can lead to modularity: 'integration' and 'parcellation'. Integration involves the emergence of connections between related components (e.g. greater pleiotropy between genes in a GRN), while parcellation involves the loss of associations between components. Wagner and Altenberg (1996) suggested that these two processes can occur due to differential selection on pleiotropy between components, making a distinction between 'acquisition' and 'suppression': there may be selective pressures to increase pleiotropy or to reduce pleiotropy; either could lead to the evolution of modularity if the initial conditions are appropriate (e.g. integration assumes an initially sparsely coupled system; parcellation assumes an initially tightly coupled system).

One possible explanation for the differential selection described by Wagner and Altenberg (1996) would be evolution under 'Modularly Varying Goals' (MVG) (Kashtan et al., 2005), which refers to tasks with multiple problem instances, each of which has the same underlying modular structure. In evolutionary terms, MVG might occur as a consequence of a regularly changing environment where different aspects of the environment challenge different aspects of the morphology or behaviour of the organisms that live there. For instance, the environment may be variously hot or cold, selecting for short or long fur, while food may be abundant above or below ground, selecting for burrowing and foraging behaviours. The idea is that there will generally be selective pressure to acquire or suppress pleiotropy in order to produce a fit individual, and that the pattern of selection will be consistent within modules, and inconsistent between modules. It has been shown that exposure to such modularly varying goals can promote the evolution of modularity in simulated networks (Wagner and Altenberg, 1996; Kashtan et al., 2005, 2009), and represent a class of problems that promote an acceleration in the rate of evolution (Kashtan et al., 2007), but also that modularity in the problem by no means guarantees the evolution of a modular or efficient solution (Clune et al., 2013; Kouvaris et al., 2017).

Following from Kashtan et al. (2005) (which introduced the term MVG), Kashtan et al. (2007) describes how the long term evolutionary consequences of exposure to modularly varying goals affects multiple different evolutionary scenarios, and compared how quickly evolution operates under MVG than with randomly varying goals. One of the models used was that of a multi-layer neural network, where fitness was determined by the networks ability to classify different stimuli. The stimuli comprised multiple inputs, and in the modular case, the relations between the inputs varied in a modular manner, such that the problem could be modularly decomposed. Using the same model, Clune et al. (2013) and Mengistu et al. (2016) showed that a system of modularly varying goals could be 'solved' by a non-modular recognition network, but that the inclusion of a cost of connections between components promoted the evolution of a modularly and hierarchical structured networks. These sparse networks where able to find fitter solutions more quickly than their non-modular cousins when the goal changed in manner consistent with previous variation, providing a stark demonstration of the evolution of evolvability.

Kouvaris et al. (2017) used the same (non-linear) model of gene regulation as Watson et al. (2014) and Kounios et al. (2016) discussed above, where connections between genes are modelled as either have a positive or negative regulatory effect. They evolved the initial conditions of the genes and the connections between them in a modularly changing environment. Importantly, they used a somewhat 'unbalanced' environment, where the different environmental modules did not vary completely independently: different modules had a significant co-variance. As in Clune et al. (2013) and Mengistu et al. (2016), they showed that the inclusions of a cost of connections provided the necessary selective pressure to suppress pleiotropy to evolve a suitably modular (in their case regulatory) network.

Notably, Kounios et al. (2016) (and other similar investigations (Watson et al., 2011c)) employ directional selection, such that there is always a way to increase fitness, if only marginally, by evolving in a particular 'direction'. The directional selection on *genotypes* is a consequence of the use of a continuous GRN (where, in practise, the gene expressions can always be increased) and directional selection on phenotypes. These investigations certainly result in canalization of fit phenotypes. The use of selection toward 'extreme' (often unobtainable) phenotypes was identified as a common weakness in such investigations by Rünneburger and Le Rouzic (2016). Importantly, the use of a cost of connections by Kouvaris et al. (2017) changes selection on genotypes (but not phenotypes) to stabilising selection (selection toward a 'peak', where any deviation from the peak would produce a reduction in fitness), which has the effect of reducing the extent of canalization.

Accepting that modularity may confer evolvability, Clune et al. (2013) address the question of how such modularity might evolve as a consequence of selective forces, without wishing to assume that the long-term benefits of evolvability will be recognised by short-sighted selection:

> We investigate an alternate hypothesis that has been suggested, but heretofore untested, which is that modularity evolves not because it conveys evolvability, but as a by-product from selection to reduce connection costs in a network.

They proceed to demonstrate the evolution of modular network topologies for classification problems using a stochastic variation on the Non-dominated Sorting Genetic Algorithm version II (NSGA-II) (Deb et al., 2002), which employs the concept of Pareto dominance to optimise multiple objective functions within a single population. Such methods are employed in genetic algorithms for optimisation problems, and increase the diversity of the population without repeatedly exploring unfit regions of the phenotype space. As already noted, modularity bounds, to some extent, the complexity of a system, and this paper explicitly encodes the intuition that by punishing complexity we can promote the evolution of meaningful structure: a 'parsimony pressure' (a cost of connections) is included in the multi-dimensional objective function on the individuals in the population.

Following therefrom, Mengistu et al. (2016) shows the evolution of hierarchical implementations of binary functions in essentially the same model as Kashtan et al. (2005).

Again, a cost of connections is employed as a parsimony pressure, and they showed that it induced the system to evolve a hierarchical topology. Furthermore, they tested the evolvability of the hierarchical topologies by making localised changes to the problem, and showed that, when an appropriate representation was evolved, the system was able to quickly evolve a high fitness phenotype due to weak coupling of independent components.

Cost of connections were further explored by Kouvaris et al. (2017), wherein they show that a simple developmental model with a recurrent network topology evolves independent modules of correlated traits under MVG when combined with a linear (L1) costs of connections. They conclude that the relatively sparse, modular networks produced confer an evolvability benefit as they provide a generalisation capability. They also make the suggestion that a quadratic (L2) cost of connections provides a lesser advantage, and Parter et al. (2008) find in their model that a cost of connections hampers the ability of their developmental system to facilitate useful variability. It is apparent, therefore, that a cost of connections by no means guarantees the evolution of a useful developmental mapping.

2.4 Learning in Network Models

While this dissertation is primarily concerned with concepts in evolution, it is instructive to briefly discuss its relation to 'learning', concepts from which our work (and prior works Watson and Szathmáry (2016); Kouvaris et al. (2017)) have borrowed. Learning is the ability for a system to improve its performance in some task as a result of past experience: genomes do not read books, but they are the result of continuous exposure to a variety of environments, and therefore evolutionary processes can – in theory – learn (Watson and Szathmáry, 2016). The evolution of evolvability can be considered a learning process, as a population or lineage becomes better at adapting to its environment after prolonged exposure. In the simplest case, this might only entail reproducing previously generate phenotypes more precisely and rapidly (robustness), or may be a dramatic change in variability that changes the accessibility of fit phenotypes. A change in variability that conferred robustness is presented in Kouvaris et al. (2017), and the authors directly invoke the concept of regularisation in machine learning to explain this: just as neural networks and other machine learning devices can learn to generalise over an incomplete training set when appropriately configured, so can non-trivial evolutionary models: the use of a suitable model (a GRN) and regularisation (a linear cost of connections) enabled the evolution process to 'learn' the environmental associations between genes.

Because learning requires continuous or repeated exposure to something (the 'training set' (Watson and Szathmáry, 2016)), it is necessary for a process to have some long-term storage capability if it is to improve its performance. In some models which have shown evolvability enhancing changes in variability, the storage medium is not so clear. Parter et al. (2008), for instance, presents the evolution of evolvability in a model of RNA folding, where the short-term and long-term consequences of evolution are tightly coupled: if there are two time-scales at play here, it is much less apparent, and no mechanism of learning can be inferred.

As discussed above, this dissertation will focus on the models employed by Kouvaris et al. (2017) and Kounios et al. (2016) derived from Siegal and Bergman (2002), where a matrix of regulatory connections provides an ideal medium for long-term accumulation of information. The learning mechanism involves integrating small changes (beneficial mutations) on the regulatory connections over many exposures, which has led to a notion of the importance of having two time-scales in such models: one time-scale exploits the information already integrated in the regulatory connections to evolve and express fit phenotypes; the other integrates the signal from directional selection (Kounios et al., 2016). In practise, it seems that necessary for the first time scale to be much shorter than the latter (i.e. the stored information have short-term consequences but long-term causes).

Chapter 3

The Evolution of Hierarchy and its Consequences for Evolvability

3.1 Introduction

As discussed in Chapter 2, the term 'evolvability' has many definitions (Pigliucci, 2008; Lynch, 2007; Brown, 2013), but here we consider the general definition provided by Payne and Wagner, "the ability of a biological system to produce phenotypic variation that is both heritable and adaptive" (Payne and Wagner, 2019). There are at least two senses in which the ability of natural selection to find fit phenotypes might be improved. The first is an increase in rate: the ability of one process to to reach a particular high fitness phenotype more quickly than another. This sense presupposes that the two process can reach the same high fitness phenotype. The second is an increase in access: high fitness phenotypes that are unreachable with one evolutionary system (e.g. due to wide fitness valleys) are reachable with another (Watson, 2020). Loosely, a phenotype needs not only to be evolvable (access) to be useful, but also evolvable in a timely manner (rate). While much prior work discusses rate (directly or otherwise), here we focus on access in the context of a changing environment, where the ideal phenotype changes regularly.

Many biological mechanisms have been identified as possible facilitators of such evolvability, with some of these relying on a developmental process to provide a non-trivial mapping from genotype to phenotype (Payne and Wagner, 2019). Such mechanisms can introduce functional epistasis, with the consequence that the phenotypic effect (and hence fitness effect) of a single genetic substitution may depend heavily on the genetic background. Consequently, the same mutation may produce little change in the adult phenotype in one genetic background, but in another may effect a dramatic change. The developmental processes that might effect such phenotypic variability are themselves genetically encoded, and so subject to random variation and selection. As such, the genotype-phenotype mapping may itself be shaped by natural selection (Watson and Szathmáry, 2016; Dall et al., 2015). We must also recognise that while a given genotype-phenotype map may be beneficial in some circumstances - perhaps enabling rapid evolution of viable phenotypes (rate or access evolvability) or minimising the occurrence of deleterious types (robustness) - it may have negative consequences in others.

Two properties of systems that are regularly considered are modularity (the ability of a system to vary individual sub-systems independently of one-another) and hierarchy (the arrangement of a system such that a subset of components direct the action of others). Both have long been recognised as potentially significant to evolvability (Wagner and Altenberg, 1996; Wagner et al., 2008; Mitteroecker, 2009; Berg, 1960; Hansen et al., 2007; Lipson et al., 2002; Lipson, 2007; Melo et al., 2016; Conrad, 2017; Hansen, 2003; Yu and Gerstein, 2006; Hallinan, 2004). Both can facilitate reusability and separation of concerns in systems designed by humans: properties which can permit robustness in natural systems (Lipson, 2007). Figure 3.1C shows how a modular regulatory network might map multiple disparate genotypes to similar phenotypes. The clusters of phenotypes approximate neutral nets in the sense that individual mutations have little phenotypic - and so fitness - consequence. This is in stark contrast to a regulatory process without developmental epistasis (e.g. Figure 3.1A) where changes in genotypes have correspondingly significant changes in phenotype. A regulatory network with epistasis but no semblance of decoupled modules (Figure 3.1B) may be robust to variation in some dimensions, but is limited in the variety of phenotypes it can actually produce.

Figure 3.1D shows how a system of internally hierarchical modules, where one gene determines the developmental trajectory of the whole module. This topology retains much of the robustness of the more densely connected modules in Figure 3.1C (where most mutations have little fitness consequence), but in addition maps some single-point mutations (those which affect the controlling gene) to large changes in phenotype. These large moves in genotype space have the potential to 'jump' fitness valleys that may exist between viable phenotypes (e.g. on the boundaries of neutral nets), allowing an evolutionary process to explore variation at the module level directly. The fundamental difference between the 'dense' modules (Figure 3.1C) and the hierarchical modules (Figure 3.1D) is that in the case where both clusters are fit, the hierarchical topology is able to move from one to the other as it is not obstructed by any unfit phenotypes that may separate the two. Note that this is explicitly a case of a difference in access: it is immaterial how many mutations might be required for a system with dense regulatory modules to move through the phenotype space if any sequence of mutations necessarily involves an initial reduction in fitness before reaching a potentially fitter alternative phenotype.

Some prior work on the study of the evolution of evolvability characterise evolutionary process using only a mapping between genotype and fitness. Others decompose this into a mapping between genotype and phenotype, and a mapping between phenotype and genotype. Doing so enables one to talk about evolvability in terms of the ability

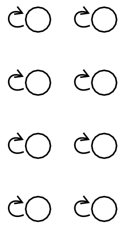
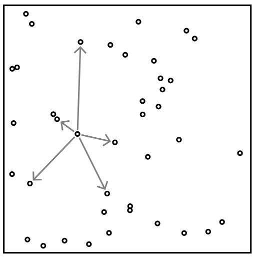
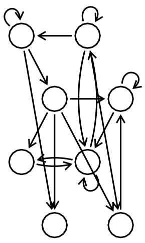
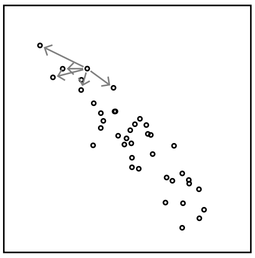
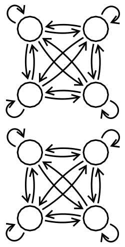
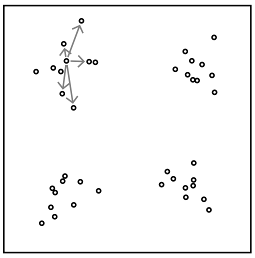
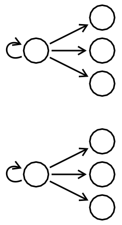
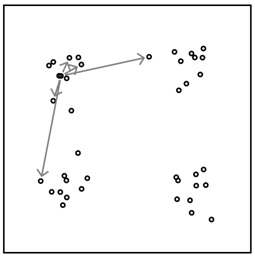
G-P Map	Gene regulatory network	Phenotype distribution	Properties
A One-to-one mapping			◆ Unconstrained
B Unstructured constraints			◆ Mono-modal ◆ Robust
C Modular constraints			◆ Multi-modal ◆ Robust
D Hierarchical modules			◆ Multi-modal ◆ Semi-robust ◆ Good access evolvability

Figure 3.1: **Different regulatory networks have different consequences for the variability of the phenotype by changing the effects of variation in the genotype.** Some examples of genotype-phenotype mappings that might occur in a gene regulatory network and their consequences for variability. **(A)** A one-to-one mapping effects no constraint on phenotypic gene expression. In the absence of external factors (e.g. noise, environmental influence) it preserves embryonic gene expression levels. Small changes in the genotype map directly to small changes in the phenotype. **(B)** A tightly coupled unstructured gene regulatory network constrains the phenotype significantly. Random variation in the genotype maps to biased variation in the phenotype, and may influence the rate at which the genotype space can be traversed. **(C)** Independent strongly connected modules restrict the phenotype within each module. Mutations in fit genotypes produce little variation in the phenotype. **(D)** Independent hierarchical modules impose the same constraints and robustness within modules but provide access to other phenotypes by mapping some gene-level mutations in the genotype to module-level variation in the phenotype.

and rate at which a system can evolves phenotypes. This is the approach we take here, as it allows us to discuss access evolvability. Various works demonstrate the ability of a population with a genotype-phenotype map or complex genotype-fitness map to evolve some manner of evolvability (Kashtan et al., 2005; Parter et al., 2008; Clune et al., 2013; Mengistu et al., 2016; Kashtan et al., 2009; Rünneburger and Le Rouzic, 2016; Crombach and Hogeweg, 2008); however, it is often not wholly apparent whether a measured increase in evolvability is due to a change in access (because fit phenotypes were previously inaccessible), or rate (the time to reach fit phenotypes was prohibitive).

A successful line of enquiry for evolvability research is the use of Modularly Varying Goals (MVG) (Kashtan et al., 2005). MVG entails the exposure of a system to multiple 'goals' (e.g. environmental conditions) which have the same underlying modular structure. Parter et al. (2008) use this paradigm to demonstrate ideas concerning 'Facilitated Variation' (Gerhart and Kirschner, 2007; Kirschner and Gerhart, 2005), providing vivid demonstrations of the potential for a non-trivial genotype-phenotype map to enable useful variability. They found that a population of Boolean-networks selected in rapidly changing environments becomes better able to adapt to new environments. The increased evolvability was due to the phenotypic neighbourhood becoming 'enriched' with novel types which represent appropriate evolutionary targets for environments which share the same modular structure as those in which the population evolved. Furthermore, their results with a model of RNA folding illustrate the dramatic consequences that a non-trivial developmental process can have on phenotypic variability, by facilitating adaptive structural changes in the phenotype with minimal variation in the genotype (Parter et al., 2008). In these works, it is shown that novel phenotypes are evolved, which could be interpreted as access evolvability; however, it is not clear that these phenotypes could not be reached given sufficient time.

Clune et al. (2013) suggested that evolvability could evolve as a side-effect of selective forces favouring parsimonious (simple) systems. To explore this, they investigated the evolution of feed-forward recognition networks (as originally employed by Kashtan et al. (2005) to investigate MVG). They demonstrated the consistent evolution of modular structures under a cost of connections when a population of networks is evaluated on a modular recognition problem (Clune et al., 2013). This line of inquiry was continued by Mengistu et al. (2016), where they found that the same cost of connections can promote the evolution of sparse hierarchical networks. The hierarchical topology provides an evolvability benefit, enabling a population to adapt more readily than more dense networks to a new problem with the same underlying structure. The precise cause of the evolution of hierarchy they observe is not apparent - as the use of a population and a diversity maintenance mechanism would provide ample opportunity for lineage selection - though it is clear that the cost of connections is a significant factor (Mengistu et al., 2016).

Kouvaris et al. (2017) employed MVG and a cost of connections to demonstrate the evolution of modular structures in a developmental regulatory network. They found that applying a cost of connections resulted in the evolution of relatively sparse modular networks. By removing dependencies between modules, the system permits modular variability which allows the development of a whole class of phenotypes, and provides robustness against deleterious mutations.

A parsimony pressure will not always promote evolvability: Parter et al. (2008) found that a cost of connections hindered facilitated variation in two models, and Kouvaris et al. (2017) showed that different types of parsimony pressure could result in different extents of pleiotropy, affecting the ability of evolved modules to vary independently (i.e. weak linkage to a greater or lesser extent). Just as with regularisations in learning, different cost functions produce different results.

3.2 Methodology

We use variations on a model previously employed by Kounios et al. (2016) and Kouvaris et al. (2017). It describes asexual individuals whose phenotypes are a deterministic function of their individual genotypes. An individual's fitness is a function of the suitability of their phenotype to the environmental conditions and a cost term computed from their genotype that models the cost of gene-regulation (the cost of connections). A hill-climber model provides an approximation of a population undergoing a micro-evolutionary process with strong-selection weak-mutation assumptions (Gillespie, 1984) which imply genetic homogeneity. Short-term variation is supplied through random mutation of the genotype each evolutionary step, and we assume perfect elitism such that no genotype will ever be displaced by a less fit mutant genotype.

In some experiments, additional variation will be introduced at regular intervals, called episodes, as in Kounios et al. (2016), simulating periods of neutral selection where the genome is subject to drift. In others, the environmental conditions experienced by the hill-climber change each episode, with the variation following a modular pattern per MVG, as in Kouvaris et al. (2017). Different environments are used to explore different dynamics. The fitness landscapes will routinely contain local fitness optima, which would trap an incremental hill-climber operating in the phenotype space directly. Crucially, the inclusion of a genetically controlled developmental step allows for the evolutionary process to change the genotype-phenotype mapping, and thereby change the accessibility of different phenotypes over time (future variability will depend on past selection).

3.2.1 Developmental Model

The developmental mapping between genotype and phenotype (the G-P map) is a variation on a standard model of gene regulation (Siegal and Bergman, 2002; Vohradsky, 2001; Spirov and Holloway, 2013), which has been used by various works from which this dissertation follows (Watson et al., 2014; Kounios et al., 2016; Kouvaris et al., 2017). The developmental process guides an initial 'embryonic' state of gene expressions to a phenotype of terminal 'adult' gene expressions. The genotype of an individual comprises a vector of initial gene expressions G and a matrix B that describes the interactions between each gene during development. A vector P describes the levels of gene expression in the adult phenotype. Development is deterministic, so the phenotype is solely a function of the individual's genotype (it is not a function of the environment or any random noise). The vectors P and G have length N (the number of genes), and the matrix B is square with dimensions N. Elements in G are binary (except at the very start of the simulation, when they are initialised to zero), either -1 or $+1$, while the values in B vary continuously and are unbounded.

The developmental process is the same used by Kouvaris et al. (2017), a slight variation on that introduced by Siegal and Bergman (2002). It is an iterative regulatory process, which is run for a fixed number of time-steps. The levels of expression of the i^{th} gene expression at time-step t is denoted by $y_i(t)$. The process is described by an update equation, which gives the expression of the i^{th} gene at time $t+1$ as a function of the regulatory matrix B, decay term $\tau = 0.2$, squash function $\sigma(x) = tanh(x/h)$, squash constant $h = 2$, and the gene expressions at time t.

$$y_i(t+1) = (1-\tau)y_i(t) + \sigma\left(\sum_j^N B_{ij}y_j(t)\right) \tag{3.1}$$

The phenotypic gene expression is then given by $P = \tau Y(T)$ where $Y(t)$ describes the vector of all gene expressions at time t and T is the number of developmental time-steps. The initial state is $Y(0) = G$, and we set $T = 10$ in most experiments for consistency with previous literature (Kouvaris et al., 2017).

By selecting σ such that $\sigma(x) \in (-1, +1)$, and recalling that the initial gene expressions $y_i(0) \in [-1, +1]$, we can show that the gene expressions are bounded such that $y_i(t) \in (-1/\tau, +1/\tau)$. Consequently, the phenotypic gene expressions P_i (which are the terminal conditions scaled by τ) are in the range $(-1, +1)$. A proof of this is provided in Proof of Bounded Phenotypic Expression for completeness.

Each entry B_{ij} of the regulatory matrix B describes the regulatory effect of gene j on gene i. A positive regulation coefficient contributes a change in the expression of gene i which is consistent with the sign of the expression of gene j; a negative value produces the opposite effect (because negative expressions are permitted, these do not map directly to activation and inhibition in a biological gene network). The greater the magnitude of B_{ij}, the larger the additive contribution of gene j's expression to the term inside the 'squash' function σ, and so the more influence it has on gene i. We use $\sigma(x) = tanh(x/2)$, as it is a suitably bounded continuous and monotonically increasing function. Being odd (symmetric), it produces no bias between positive and negative values which might influence the system. Because the output has the same sign as the input, it doesn't disrupt directional selection.

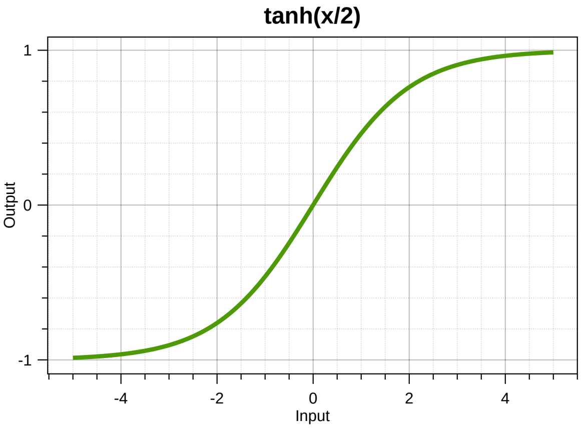


FIGURE 3.2: The squash function $tanh(x/2)$ is (odd) symmetric around $x = 0$, monotonically increasing ($x' > 0$), and asymptotically approaches -1 and $+1$.

Table 3.1 provides a summary of the developmental parameters. Apart from the number of genes N and number of developmental timesteps T, these will be the same in all experiments, and the same as those found in Kouvaris et al. (2017) (though τ_2 is renamed τ and the update-rate $\tau_1 = 1$ is omitted for simplicity).

Parameter	**Symbol**	**Values**
Number of genes	N	Variable
Number of developmental timesteps	T	Variable (usually 10)
Squash Function	σ	$\sigma(x) = tanh(x/h)$
Squash Constant	h	2
Gene Expression Decay Rate	τ	0.2

TABLE 3.1: Parameters of the developmental model.

3.2.2 Evolutionary Model

The evolutionary process is modelled as a hill-climber, which represents a population undergoing a micro-evolutionary process of strong-selection and weak-mutation assumptions (Gillespie, 1984) with elitism. These assumptions approximate the behaviour of a population where any mutation that occurs will either rapidly fix in the population or be lost before another mutation occurs, restoring genetic homogeneity.

Algorithm 1 Pseudocode for the developmental process.

```
procedure DEVELOP($G, B, T, \sigma, \tau$)
    $t \leftarrow 0$
    $Y(0) \leftarrow G$
    while $t < T$ do
        for $i \in N$ do
            $y_i(t+1) \leftarrow (1-\tau)y_i(t) + \sigma(\sum_j^N B_{ij}y_j(t))$
        end for
        $t \leftarrow t + 1$
    end while
    return $\tau Y(T)$
end procedure
```

The hill-climber operates on an individuals which comprise the genetic information $\langle G, B \rangle$ and the developed phenotype P. The state of the hill-climber is a 'current' individual representing the homogenous population, and each evolutionary step this current individual's genotype is cloned and mutated to produce a new 'candidate' genotype. This candidate genotype is developed to produce a phenotype, and together they comprise the candidate individual. The current and candidate individuals are both awarded a fitness based on their suitability to the current environmental conditions and a cost of regulation (a function of B). If the candidate individual's fitness is greater than the fitness of the current individual, then it replaces the current individual; otherwise, the current individual is retained for that evolutionary step: beneficial mutations fix immediately, and neutral or disadvantageous mutations are lost.

During mutation, either the matrix of regulatory coefficients B or the vector of initial gene expressions G is mutated. B is mutated with probability R_B, which equals one-half in experiments where both components are free to change. In some experiments we will fix G, and so the probability of mutating B will be $R_B = 1$.

Excepting those in Chapter 6, the manner of variation (simulating mutates) is as follows. When B is mutated, a single entry is chosen at random to be modified by addition of a random number. The random number is sampled uniformly from the range $[-M_B, +M_B]$. When the vector of initial gene expressions G is mutated, a single element is chosen at random to be mutated. In some models, a small number sampled uniformly from the range $[-M_G, +M_G]$ is added to this element, and clamped to the range $[-1, +1]$. In others, the element is assigned the value -1 or $+1$ with equal probability. While the possible values in G are consequently limited, the entries in B are unbounded. Unless otherwise specified, the values in B and G are initialised as all zero.

The fitness f of an individual is the difference between a benefit term b (a function of the phenotype and environment) and a cost term λc (a function of the genotype). The benefit term describes how well suited a phenotype P is to the current environmental conditions and is detailed in the next section. The cost term simulates a cost of gene regulation, which is known to be costly in biological organisms (Lang et al., 2009; Lynch and Marinov, 2015). We model this cost as the mean of some function ϕ over the magnitudes of the regulation coefficients found in the regulation matrix B (Eq 3.3) scaled by the parameter λ (Eq 3.2): this is the 'cost of connections'.

$$f = b - \lambda c \tag{3.2}$$

$$c(B) = \frac{1}{N^2} \overset{N,N}{\underset{i,j}{}} \phi(|B_{ij}|) \tag{3.3}$$

Table 3.2 provides a summary of the evolutionary parameters used throughout this thesis.

Parameter	**Symbol**	**Values**
Number of genes	N	Variable
Number of simultaneous G mutations	C_G	Variable
G vector mutation magnitude	M_G	Variable
G vector mutation type		Binary or Uniform
B matrix mutation probability	R_B	0.5 or 1
B matrix mutation magnitude	M_B	Variable
B matrix mutation type		Uniform
Regulation Cost Coefficient	λ	Variable
Regulation Cost Function	ϕ	Usually $\phi(\lvert x\rvert) = \lvert x\rvert$

TABLE 3.2: Parameters of the evolutionary model.

It should be noted that there is some redundancy between these parameters and those described in Section 3.2.1. In the absence of a cost of connections, the squash coefficient h from the developmental model, the B mutation magnitude M_B can be changed in proportion without changing the model (a property that is largely respected by the software implementation); for a linear (L1) cost of connections, λ must also be changed in proportion. The squash coefficient h and cost coefficient λ can be subsumed into the squash function σ and cost function ϕ respectively but are preserved for interpretability and consistency with previous literature.

3.2.3 Model of a Modularly Changing Environment

The benefit term b in Eq 3.2 is a function of the expression of phenotypic gene expressions P and the environmental conditions η it experiences. In this Chapter, we consider individuals of size $N = 16$, and – for the purposes of selection only – logically divide the phenotype into 4 'modules' of 4 genes. The environmental conditions regularly change in a modular manner, such that coordination between genes within a module

Algorithm 2 Pseudocode for mutation.

```
procedure MUTATE($G, B, M_G, M_B, R_B, BinaryG$)
    $G' \leftarrow G$
    $B' \leftarrow B$
    if $rand([0,1)) < R_B$ then
        $i \leftarrow rand(\{1..N\})$
        $j \leftarrow rand(\{1..N\})$
        $\mu \leftarrow rand(U[-M_B, +M_B])$
        $B'_{ij} \leftarrow B'_{ij} + \mu$
    else
        $i \leftarrow rand(1..N)$
        if BinaryG then
            $\mu \leftarrow rand(\{-2, 2\})$
        else
            $\mu \leftarrow rand([-M_B, +M_B])$
        end if
        $G'_i \leftarrow clamp([-1,1], G'_i + \mu)$
    end if
    return $G', B'$
end procedure
```

Algorithm 3 Pseudocode for evolution.

```
procedure EVOLVE($G_0, B_0, E, \lambda, epochs, K, M_G, M_B, R_B, BinaryG, Z, c_L, c_H, c_0$)
    $G \leftarrow G_0$
    $B \leftarrow B_0$
    $epoch \leftarrow 0$
    while $epoch < epochs$ do
        $Randomise(E, Z, c_L, c_H, c_0)$
        $evolutionary\_step \leftarrow 0$
        while $evolutionary\_step < K$ do
            $G', B' \leftarrow Mutate(G, B, M_G, M_B, R_B, BinaryG)$
            $f \leftarrow Evaluate(G, B, E, \lambda)$
            $f' \leftarrow Evaluate(G, B, E, \lambda)$
            if $f' > f$ then
                $G \leftarrow G'$
                $B \leftarrow B'$
            end if
            $evolutionary\_step \leftarrow evolutionary\_step + 1$
        end while
        $epoch \leftarrow epoch + 1$
    end while
    return $G', B$
end procedure
```

Algorithm 4 Pseudocode for randomising the modularly changing environment.

```
procedure RANDOMISE($E, Z, c_L, c_H, c_0$)
    for $\langle c_m^+, c_m^- \rangle \in E$ do
        $multipeaked \leftarrow rand([0,1)) < Z$
        $positive \leftarrow rand(0,1) = 0$
        if positive then
            $c_m^+ \leftarrow c_H$
            if multipeaked then
                $c_m^- \leftarrow c_l$
            else
                $c_m^- \leftarrow c_0$
            end if
        else
            $c_m^- \leftarrow c_H$
            if multipeaked then
                $c_m^+ \leftarrow c_l$
            else
                $c_m^+ \leftarrow c_0$
            end if
        end if
    end for
end procedure
```

Algorithm 5 Pseudocode for evaluating an individual.

```
procedure EVALUATE($G, B, E,$)
    $P \leftarrow Develop(G, B)$
    $benefit \leftarrow Judge(E, P)$
    $cost \leftarrow {}_{i,j}^{N,N} B_{i,j}$
    return $benefit - \lambda \times cost$
end procedure
```

is consistently rewarded, but there is no such pattern between modules. We describe a class of environments with the following benefit function (Eq 3.4), where M is the set of modules, $m \in M$ is the set of genes in a module, and η is a collection of 'module benefit functions' $\eta_m : R \rightarrow R$.

$$b(P, \eta) = \frac{1}{|M|} {}_{m \in M} \eta_m \left(\frac{1}{|m|} {}_{i \in m} P_i \right) \tag{3.4}$$

This computes the benefit of the phenotype P as the mean of the independent contributions of each module, which is in turn some function of the mean phenotypic gene expressions within each module. We use a piecewise-linear module benefit function η_m to describe a (potentially epistatic) fitness landscape based on the current environment conditions related to the module m. We define this function as the maximum of two contributions: one positive product with coefficient c_m^+, and one negative product with coefficient c_m^-.

$$\eta_m(x) = \max(x \times c_m^+, -x \times c_m^-) \tag{3.5}$$

This rewards a positive mean gene expression when c_m^+ is high, and a negative mean gene expression when c_m^- is high. Three values for c_m^+ and c_m^- are used: 1.0, 0.7, and -1.0. We restrict the combinations of these to four environmental 'instances', the module benefit functions for which are depicted in Fig 3.3.

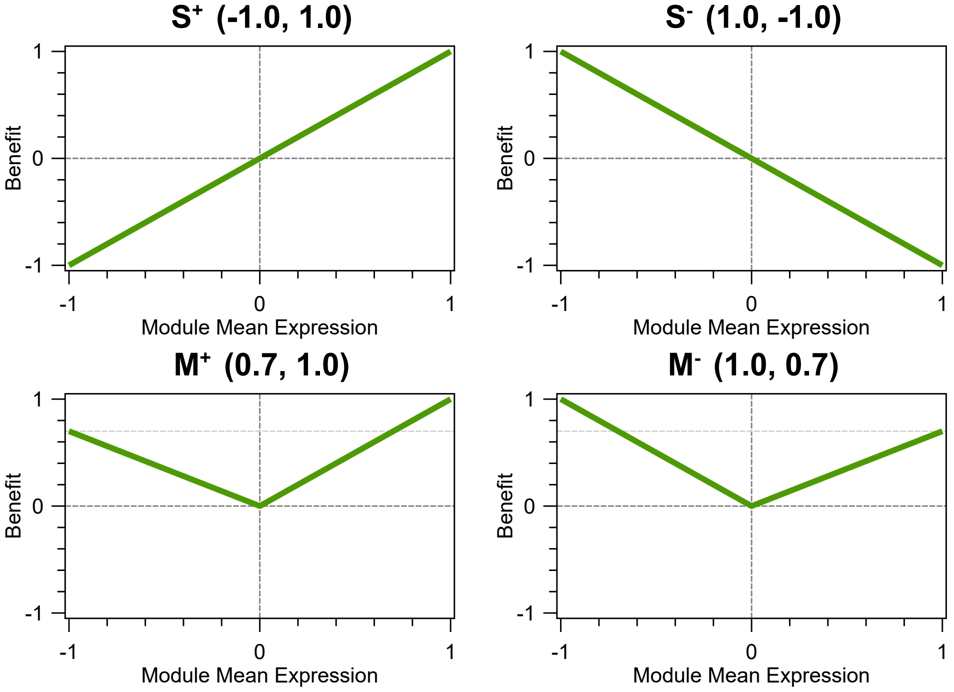


FIGURE 3.3: The four module instances that may occur. Headers show the values of c_m^- and c_m^+ respectively in parentheses. At any time, the environment comprises four of these (one for each module) selected independently at random according to the parameter $Z \in [0, 1]$. **Top**: the single-peaked instances, where there is only one optimum for the module. The optimum can easily be found by a local search process operating in the phenotype space. Because the module-benefit-functions are completely linear, there is no epistasis in these instances. **Bottom**: the multi-peaked 'trap' instances, where a local-search process can easily become trapped at the local optimum because the global optimum is made inaccessible by the fitness valley with minimum at 0. These instances introduce epistasis, because the marginal benefit of a mutation on a gene depends on the phenotypic expression of the other genes in its module: you should 'go with the majority'. The 'trap' instances occur with frequency $\frac{Z}{2}$, and the 'easy' instances with probability $\frac{1-Z}{2}$

Instances S^+ and S^- describe an easily solved 'single-peak' problem, where there is a positive benefit gradient toward the global optimum everywhere. A positive mean gene expression is rewarded in S^+; a negative mean gene expression is rewarded in S^-. Instances M^+ and M^- pose a problem for a local-search process – such as our hill-climber – because they describe conditions where there are two local optima, one being a global optimum that provides a significantly greater fitness than the other. Whereas the single-peaked environments are easily solved no-matter the initial population, the multi-peaked environments represent a 'trap': a naïve hill-climber with a positive mean gene expression will be 'trapped' at the worse local optimum if placed in the M^- conditions; a hill-climber with a negative mean gene expression will be 'trapped' at the worse local optimum if placed in the M^+ conditions.

The environment is randomised every $K = 1000$ evolutionary steps, whereupon each module is assigned one of the four instances independently: some modules may be presented with a single-peaked instance, while others find themselves confronted with a multi-peaked instance. We refer to each period of K evolutionary steps as an 'episode'. We control the incidence of the different type of module configurations with a parameter Z, which is the probability of any module being assigned a multi-peaked instance in any episode. There is an equal probability of a 'positive' or 'negative' instance (i.e. S^+ is observed just as often as S^-). Note that the maximum attainable benefit is constant in any environment, as the benefit at the global optimum of any instance is always 1.

One possible interpretation for this type of environment is one in which resource availability changes regularly, and where specialisation is rewarded. For instance, an insect ridden environment may benefit birds with a narrow probing beak, while a nut-rich environment will favour birds with broader crushing beaks. The abundance of either resource may occasionally decline significantly, necessitating re-adaptation if a single population is to survive; when both resources are available, there remains a benefit to specialising on the more abundant. Consequently, a population that is able to readily re-specialise will be able to fully exploit the environment. By taking each module as independent, we assume that the fitness consequences of other characters are independent (epistasis is restricted to within the modules).

3.3 Experimental Results

In order to demonstrate the evolution of independent modules of single-layer hierarchies, we simulate many thousands of episodes (each of $K = 1000$ evolutionary steps) with a linear (L1[1]) cost of connections (equivalent to a LASSO regularisation (Tibshirani, 1996)), cost coefficient $\lambda = 0.1$, and variable choice of Z (the probability of a multi-peaked module instance). For sufficiently low values of Z, we consistently observe the evolution of hierarchical modules, which increase the accessibility of fit phenotypes, and enable the hill-climber to always find the globally optimal phenotype in any of the possible environmental conditions. Below we show results for a run with $Z = 0.95$ randomly chosen from 40 replicates (independent runs with different random seeds) for illustrative purposes.

[1] $\phi(|x|) = |x|$, the linear sum of absolute values

Parameter	**Symbol**	**Values**
Number of episodes		400000
Evolutionary steps per episode	K	1000
Number of genes	N	16
Number of modules		4
Number of genes per module		4
B matrix mutations probability	R_B	0.5 or 0
B matrix mutation magnitude	M_B	0.001
B matrix mutation type		Uniform
Regulation Coefficient	λ	0.1

Table 3.3: Parameters for the changing modular environment.

The evolution of independent hierarchical modules

Hierarchy evolves dependably when λ is small (but not too small) and Z sufficiently smaller than 1. Simulating 100,000 episodes (each of $K = 1000$ evolutionary steps), with a multi-peaked module configuration probability of $Z = 0.95$ and cost of connections $\lambda = 0.1$, we observe the evolution of independent modules with internal hierarchical structure in all of 40 independents replicates. 720 additional runs with values of Z in the range $[0, 0.9]$ also evolved independent hierarchies corresponding to the environmental structure. The modules facilitate adaptation to changing environmental conditions by increasing the accessibility of potentially fit phenotypes. Figure 3.4 shows how the regulatory coefficients in the matrix B evolve.

Initialised as zero, the regulatory connections quickly form dense regulatory modules of many genes each regulating each other. This initial growth doesn't always produce independent regulatory modules consistent with the modules of the environment (Figure 3.4B), and the following episodes are marked by parcellation events, where previously connected modules disassociate in a dramatic fashion (Figure 3.4C&D). While the modules are disassociating – and for thousands of episodes thereafter – they are also slowly 'disintegrating' (pleiotropy is lost within the module) from the fully-connected 'dense' configuration (Figure 3.4D) toward a sparser 'hierarchical' configuration (Figure 3.4E). This hierarchical configuration is characterised by a single gene directing all genes within its regulatory module. Like the dense module disassociations, the tendency toward hierarchy is initially gradual, but later accelerates dramatically. This acceleration is a consequence of the benefits of hierarchy, which we present in the next section.

The disassociation of modules is not perfect, and for difficult selective environments (e.g. higher frequency of multi-peaked instances Z and greater cost of connections λ) can result in the evolution of one or more hierarchical 'super-modules' commanding more than one of the modules defined by the environment (Figure 3.5, additional results in Appendix C.2). In an extreme case, every gene may become part of a single regulatory modules, and the system fails to model all the variation expressed in the environment.

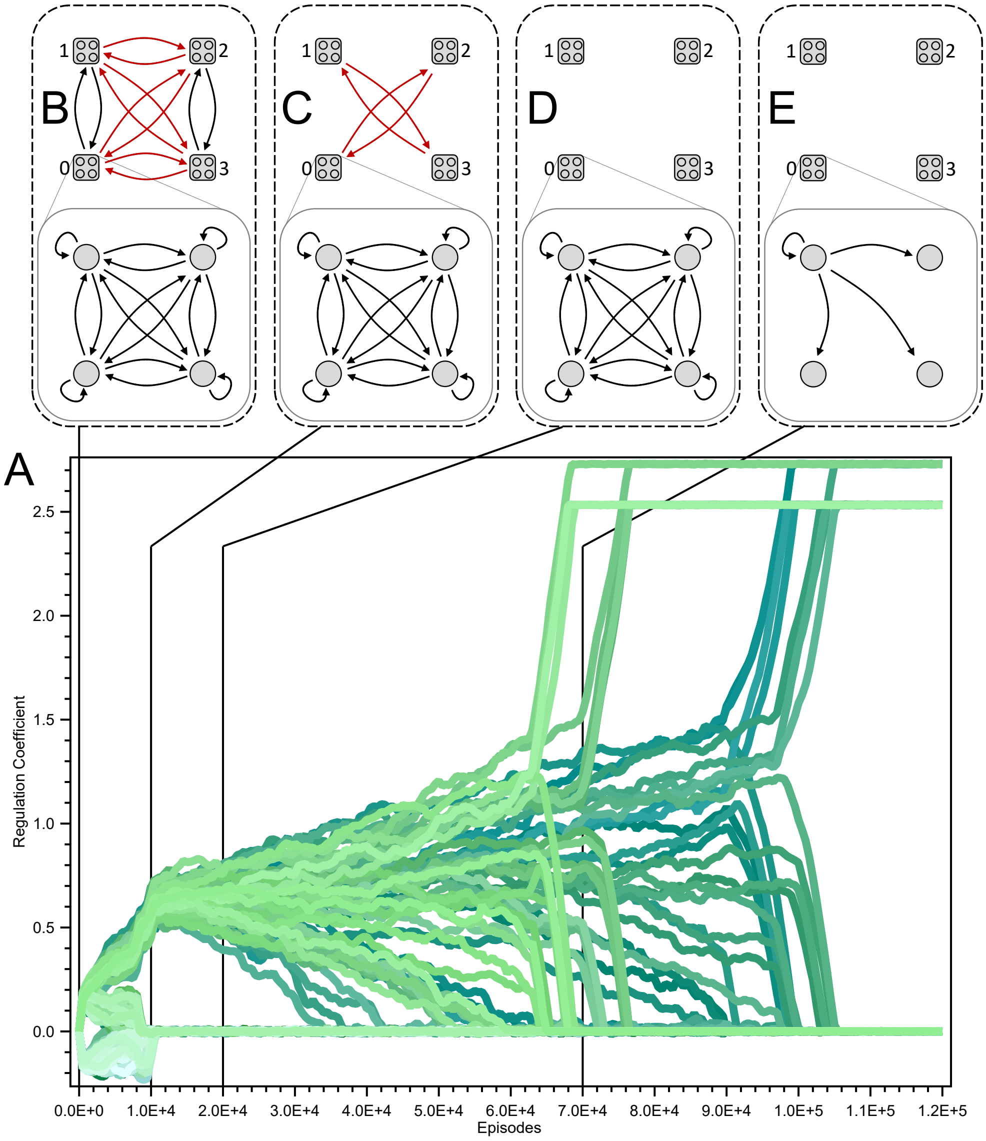


FIGURE 3.4: The evolution of independent internally hierarchical modules in a regulatory network. The trajectories of each regulatory connection **(A)** and partial illustration of the regulatory network at 500 **(B)**, 10,000 **(C)**, 20,000 **(D)**, and 70,000 **(E)** episodes for a single run ($Z = 0.95$, $\lambda = 0.1$), showing the sequence of events that leads to the evolution of four independent 'dense' modules **(D)**, which ultimately become hierarchical **(E)**. Each line in **(A)** corresponds to a single entry in the B matrix. Shades of green are used only to aid in visual discrimination, and should otherwise be ignored. Each network diagram (**(B-E)**) shows the presence or absence of inter-module connections between all modules, and the intra-module connections for the module which happened in this simulation to evolve a hierarchical configuration most quickly. By around 110 thousand episodes, all modules are completely hierarchical, and the trajectory is reduced to 3 groups: many inter- and intra-module connections with weight 0, 4 self-connections on the lead (or dominant) gene (one for each module) with weight 2.7, and 12 other intra-module connections with weight 2.5. Red arrows between modules correspond to negative connections between modules in the two modules; all connections between genes within modules are positive per the environmental conditions.

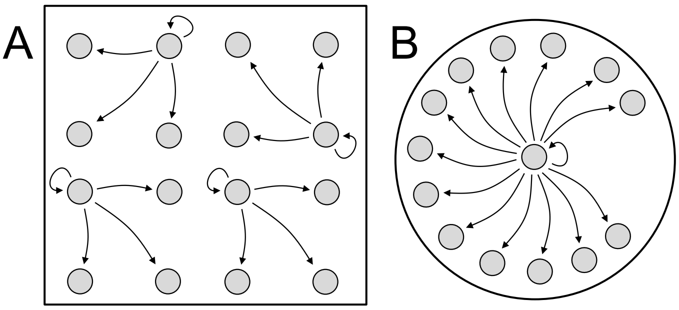


FIGURE 3.5: Hierarchy reflects evolved modules. **(A)**: In a modularly varying environment which induces variation between episodes (i.e. when Z isn't too large), each module evolves into a separate 1-layer hierarchy. **(B)**: In a non-changing environment, or one where the environmental variation does not permit significant phenotypic variation between episodes (e.g. when $Z = 1$), a single hierarchy directing all genes may evolve.

Evolved hierarchy enables macro variation

The emergence of independent hierarchical modules has dramatic consequences for the end-of-episode fitness achieved by the hill-climber. Figure 3.6C shows how the initially variable end-of-episode fitness becomes consistently higher as the degree of hierarchy within each module increases (Figure 3.6B). This occurs because the hill-climbing process becomes able to 'switch' whole modules with a single mutation. Consequently, genotypes that would previously be trapped producing phenotypes at a lower-fitness local optimum when the environmental conditions change are able to 'solve' the problem, by jumping the fitness valley that separate the local optima of a multi-peaked module instance. Each 'step' in Figure 3.6D – which depicts the frequency of switch events – represents a module which is no-longer trapped when an environmental change leaves it mismatched (i.e. the sign of the genes in a module do not correspond to the global optimum of the environmental module instance). Each jump corresponds to jump in Figure 3.6C, which shows the frequency with which the globally optimal phenotype is evolved. A 'partial' hierarchy (where the lead gene controls somewhat over half of the total connection weight) is sufficient to permit these switches, and the ability to switch coincides with the sudden increase in the rate at which the degree of hierarchy increases for each module (see Figure 3.6B&D).

The ability of the system to switch the modules is a direct consequence of the evolved hierarchy: independence of modules (as observed in Kouvaris et al. (2017)) is not enough. In these single-layer hierarchies, the state of one 'switch' gene ultimately determines the state of all others. As such, only the 'switch' gene of a module need be switched in order to effect a change in the sign of the phenotypic expression of the whole module. Consequently, a single-point mutation in G is sufficient to switch a whole module in the phenotype and so jump the fitness valley in the multi-peaked module instances (as depicted in Figure 3.7). Switch events can be observed by plotting the per-evolutionary step fitness of the hill-climber as it responds to a change in environment, as depicted

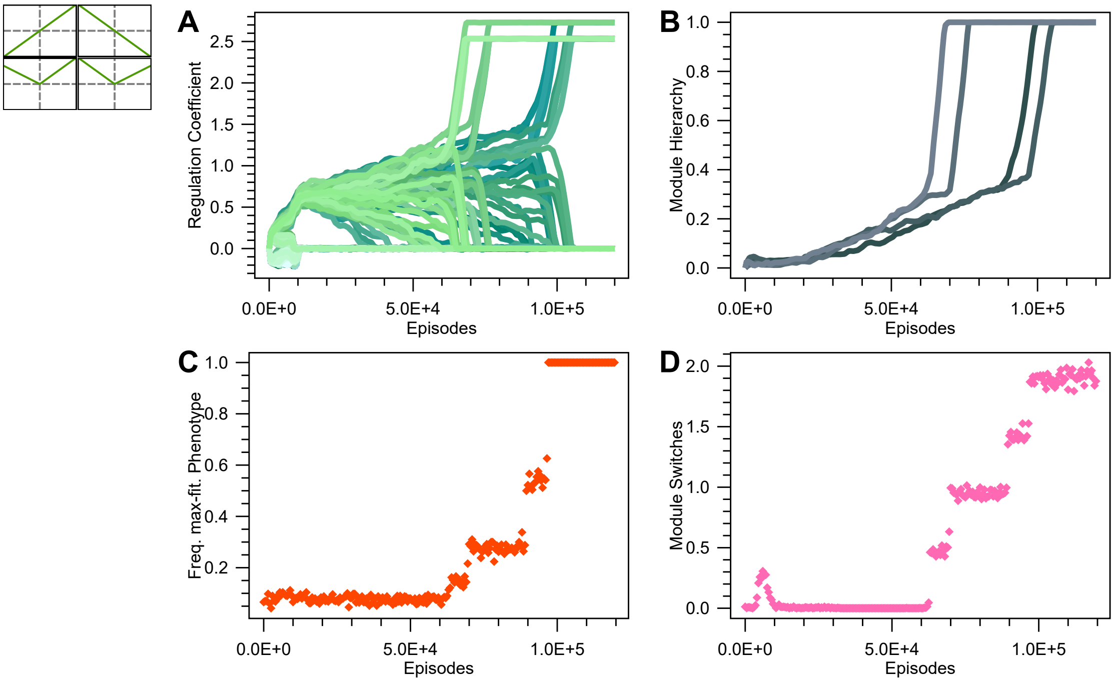


Figure 3.6: The evolution of hierarchy enables the system to achieve a high fitness under any of the possible conditions. **(A)**: trajectories of each regulation coefficient for a single run with $Z = 0.95$ and $\lambda = 0.1$ (the same run as Figure 3.4) **(B)**: the degree of hierarchy for each of the 4 modules over time (1 sample per 500 episodes); see Degree of Hierarchy for a definition of the degree of hierarchy d_h, which reflects how much influence the dominant gene has within its module. **(C)**: the proportion of episodes where a phenotype is evolved such that the sign of terminal gene expressions matches the environmental conditions (1 averaged data-point per 500 samples, 1 sample per episode). When a module becomes sufficiently hierarchical, it no longer becomes trapped by multi-peaked environmental instances. Consequently, the number of modules that may become trapped decreases, and the frequency with which the globally optimal phenotype is found suddenly increases (roughly $(1 - Z/2)^x$ where x is the number of non-hierarchical modules). **(D)**: the mean number of multi-peak module switches within each module per episode over time, with 'steps' (corresponding to those in **C**) revealing when each module becomes sufficiently hierarchical to start switching. Each module switches within a multi-peak module instance during $Z/2 = 47.5\%$ of episodes once it becomes sufficiently hierarchical, giving an expected rate of $4 \times Z/2 = 1.9$ switches per episode when all four modules are hierarchical (1 averaged data-point per 500 samples, 1 sample per episode). For $Z = 0.95$, the system produces such a result in each of 40 repeats.

in Figure 3.8B. A ‘dense’ module configuration (Figure 3.4B-D) - where each gene in a module regulates all others in its module with roughly the same interaction coefficient - would require at least half of the genes to switch simultaneously to achieve such a switch. Such an eventually is incompatible with the assumptions of strong selection and weak mutation.

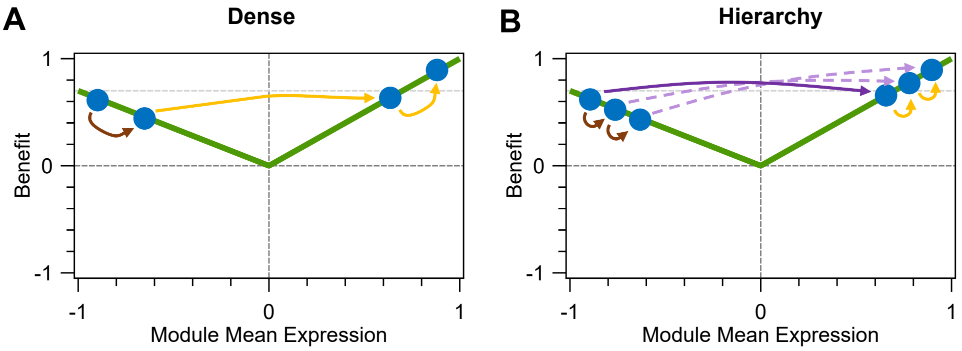


Figure 3.7: Hierarchy enables adaptation in a multi-peaked environmental instance: the phenotype space and fitness-contribution for a dense **(A)** and hierarchical **(B)** module in the D^+ environmental module instance. Blue dots represent accessible phenotypes in a 3-gene module. Arrows between phenotypes show mutations which switch a negative (-1) initial gene expression to positive $(+1)$. The horizontal displacement between phenotypes has been exaggerated for the purpose of clarity. **(A)**: With a dense regulatory module and initially negative gene expressions, a single-point mutation switching any gene’s initial expression (brown arrow) always results in a loss of fitness for regulatory module with more than 1 gene, as it is working contrary to the other genes in the module. **(B)**: With a hierarchical regulatory module, a single-point mutation on a subordinate gene (brown arrows) results in a small negative change in fitness, but switching the dominant gene (purple arrows) inverts the phenotypic expression of every gene in the module, and is able to achieve a higher fitness despite the lack of agreement in the initial state. This does not completely switch the phenotype, as the initial expression of the subordinate genes does have a small additive contribution to their final expression, but once the dominant gene switches it becomes beneficial to switch them also. Consequently, there is a sequence of mutations (including the orange arrows) which lead from one local peak to the other with a monotonic increase in fitness.

3.4 Conclusions

This chapter has introduced the evolutionary and developmental model that will be used throughout the dissertation, and one style of modularly changing environment. Our results demonstrates that the evolution of hierarchy can change the level at which an evolutionary process is operating from gene space to module space. This produces an increase in the accessibility of useful phenotypes, as it enables switch like behaviour, where single-point mutations in the G vector of initial gene expression to effect whole-module switches.

The benefit of this re-scaling is most apparent when considering an environment where Z is close to 1. In any given episode, multiple environmental modules are likely to be

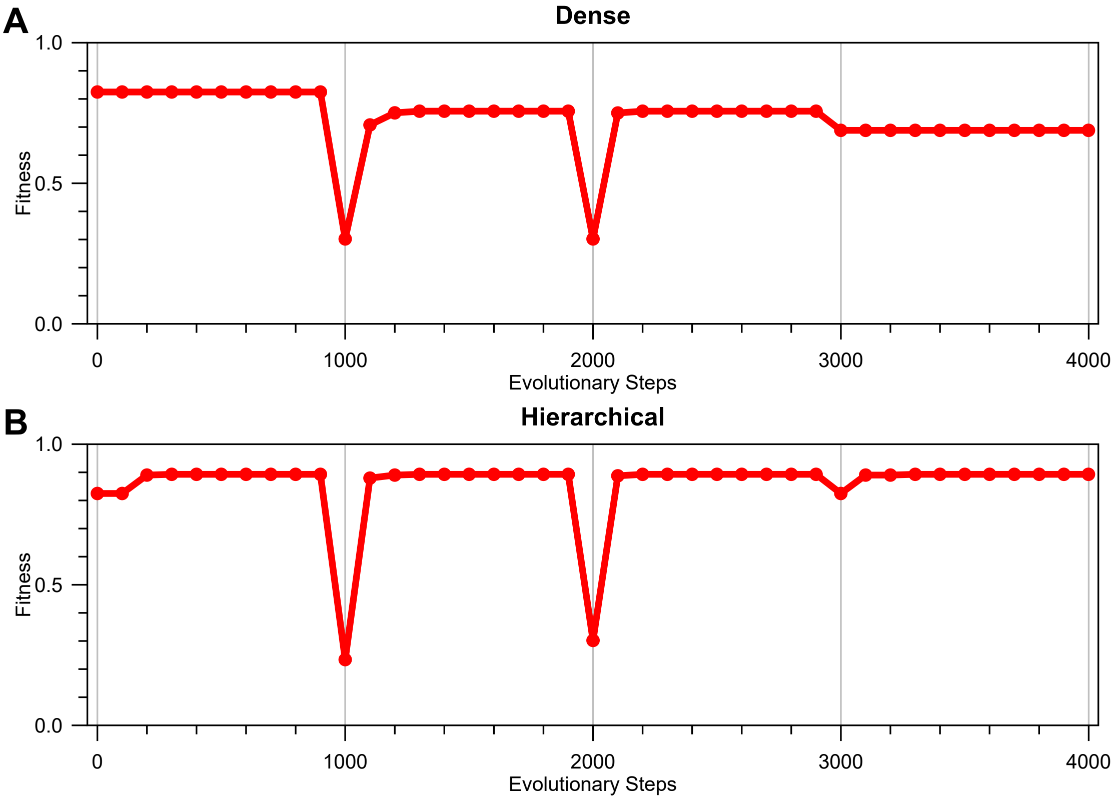


FIGURE 3.8: Hierarchical regulatory networks are able to escape local fitness optima after an environmental change. The response of a hill-climber with a dense regulatory network **(A)** and hierarchical regulatory network **(B)** to the same changing environment with $Z = 0.95$. Every 1000 evolutionary steps, a new episode begins, and the environmental conditions are randomised. With the dense regulatory network, the system is only able to respond to the dramatic reductions in fitness that coincide with a mismatch against a single-peaked module instance, as occurs at evolutionary steps 1000 and 2000. In these instances, there is no local optimum to trap the module. In contrast, a hierarchical regulatory network allows the system to quickly respond to any change in environment, quickly attaining a high fitness regardless of the environment conditions in which is finds itself, as at evolutionary steps 0 and 3000. 1 sample every 100 evolutionary steps.

multi-peaked and so present traps. For a restart hill-climber with a non-hierarchical regulatory network performing single-point flips in G, the expected number of attempts to find the global optimum is exponential in the number of modules, because each restart can only explore the local phenotypic optimum it happens to find for each module. Once evolved, a hierarchical regulatory network can enable the evolutionary process to readily find the global optimum in any environment, regardless of its initial conditions. This enables it to achieve a consistently high fitness over evolutionary time by finding the globally optimal phenotype each episode.

This represents the evolution of evolvability as it enables a micro-evolutionary process to better exploit a changing environment where previously it would become trapped in local optima. The hierarchical modules effectively reduce the parameter space of the genetic process, as only the initial states of the dominant genes in each module have a significant bearing on the terminal phenotype after development. This provides a concrete example of an *in silico* evolutionary system achieving greater access evolvability by changing its level of operation from gene space to an adaptive module space by means of evolved co-variation mediated by a developmental process.

It is notable that the evolution of hierarchy precedes the change in variability that the hierarchy confers. This is the first indication that the reasons for which hierarchy evolves, and consequences of its evolution for evolvability, may be distinct. We will expand on this thought in Chapter 5.

Chapter 4

Rescaling Variability with Hierarchy

4.1 Introduction

In this chapter, we use the same developmental model as in the previous chapter to show how the evolution of hierarchy can help to solve a modular constraint-satisfaction problem by rescaling the level of variation from genes to modules, enabling a micro-evolutionary process to search in the space of modules, and consequently satisfy inter-module dependencies. Using a generalised modular constraints problem (Watson et al., 2011c), we explore the properties of the fitness landscape that make such a problem solvable by an evolutionary process that does not exploit hierarchy (Kounios et al., 2016), and discuss how the inclusion of a cost of connections promotes the evolution of hierarchy that can solve the problem in a different way, by evolving independent hierarchical modules (as in Chapter 3) which enable the system to coordinate these modules to satisfy a global (i.e. between module) fitness constraint. We find that the evolution of hierarchy is somewhat scale-invariant, as hierarchies can evolve in modules of differing size; however, there are limits as to this process, and to where hierarchy can provide an evolvability benefit. We show that the evolution of hierarchical modules at one 'layer' of a problem enables the evolutionary process to exploit structure from the layer above, which can motivate the evolution of hierarchy at the layer above enabling the evolutionary process to 'solve' a hierarchically structured problem.

As in the previous chapter, variation in the G vector of initial gene expressions over the course of many episodes allows the evolutionary process to explore a rugged fitness landscape. In this chapter, the fitness benefit landscape is defined by epistatic interactions within and between a number of modules, unlike in the previous chapter where the fitness benefit of each environmental module was independent. When hierarchy evolves in these modules, the variability conferred enables the evolutionary process to satisfy the (weaker) epistatic constraints between modules as it becomes able to make whole-module switches which reveal the fitness gradients these constraints produce. In the previous chapter, the inclusion of single-peaked environmental instances (controlled with parameter Z) provided between-episode phenotypic variation by permitting the evolutionary process to make 'easy' adjustments (i.e. inducing it to invert a module without requiring the variability afforded by the not-yet-evolved hierarchy). In this chapter, the environment will be fixed, and the long-term variation in G will be induced through 'partial-resets': a large number of mutations will be applied between each episode without selection, this representing periods of genetic drift as per Kounios et al. (2016).

We will use explicitly modular problems derived from the Modular Constraints (MC) Problem of Watson et al. (2011c). The original MC problem uses quadratic epistatic interactions within modules, and the nature of the shallow gradients reveals information about the inter-module constraints when the evolutionary process to the genes contributing to undecided (i.e. internally inconsistent) modules. This property means that the MC problem can be solved without a self-modelling process (see Appendix B.2 for some discussion on that matter). We introduce a generalised version of the MC problem with which we define a problem with non-smooth (piecewise-linear) epistatic interactions, but otherwise the same modular properties: we will refer to the MC problem with quadratic epistatic interactions as the 'smooth' MC, and the MC problem with the piecewise-linear epistatic interactions as the 'spiky' MC (for reasons that will become clear). The spiky MC problem is more comparable with the model used in Chapter 3, and does not so readily reveal information about global constraints as the smooth MC because the gradients produced by within-module epistatic interactions are steep almost everywhere (and so dominate the shallow between-module constraints). We show that both the smooth and spiky MC problems can be solved by an evolutionary process that is able to evolve independent hierarchical modules, and find that the hierarchy scales up to produce a single 'grand' hierarchy (where one gene directs all other genes in the genotype: one big developmental module) once it is able to coordinate between modules, and so has reduced phenotypic variability between modules.

Based on the observation that the hierarchy will scale-up with the phenotypic variability, we shall then extend our results with the generalised MC problem to a hierarchical 'multi-layer' modular constraints problem based on the Hierarchical If-and-only-if (HIFF) (Watson and Pollack, 1999) problem, showing that the evolution of hierarchy is somewhat scale-invariant, and can 'bootstrap' itself: smaller hierarchies can facilitate larger hierarchies, which facilities even larger hierarchies. We discuss the limits of this process, which affirms the observation in the previous chapter that hierarchy evolves even when it cannot provide an evolvability benefit, and sets the scene for Chapter 5, where we will explore in detail why hierarchy evolves.

4.2 Partial Resets

In the previous chapter, phenotypic variation between episodes was introduced into the system by changing the environment at the start of each episode of K evolutionary step: the inclusion of single-peaked environmental instances S^+ and S^- with frequency Z ensured that the evolutionary process wasn't trapped at a single point in phenotype space for too long. In this chapter, however, we consider a static but rugged fitness landscape: one that doesn't change over evolutionary time, and that contains many local optima. Because the environment does not change, we use an explicit *partial reset* of the state of the evolutionary process (simulating an extended period of genetic drift per Kounios et al. (2016)) to introduce the long-term variation necessary for the system to infer problem structure. While the environment has many local-optima which can trap an incremental local search process, the frequent partial resets ensure that many different local-optima are visited over evolutionary time. Different local optima have different correlations between modules, and so with sufficient variation the average correlation will be zero. Kounios et al. (2016) demonstrated that an evolutionary model including a GRN (the same developmental model as described in Section 3.2) with partial-resets (where G is randomised completely but B is left unchanged) was able to solve both a 2-dimensional local constraints problem (the 'Concentric Squares' problem). The same evolutionary process could also find high-fitness phenotypes when selection was on the basis of random (potentially inconsistent) constraints. Both of these problems were modelled with a constraints matrix, such that the benefit was given by

$$b = P^T C P \tag{4.1}$$

Starting from an empty B matrix, the system would quickly reach a local optimum in the rugged fitness landscape after each reset. While the B matrix has only small values, it has little effect on the evolution of G, and the system explores local optima at random, though the distribution may be biased to 'find' higher fitness local optima with greater frequency than lower fitness local optima depending on the problem. Owing to the directional selection on the phenotype (i.e. the more strongly the genes are expressed, the higher the potential fitness) there is a benefit advantage to be obtained by increasing the magnitude of the connection strengths in the B to reflect the correlations in the phenotype. In Kounios et al. (2016), an explicit 'delta-rule' update process is used, which updates each entry in the B matrix of regulatory connections in the direction that most increases fitness at the end of each episode. If the 'learning rate' of the delta-rule updates process is low enough, then after many episodes the B matrix would steadily evolve to reflect the constraint matrix representing the problem. When the connections in the B matrix are strong enough, they begin to influence the space of representable phenotypes. The more constrained the system becomes, the more consistently it 'finds' relatively good solutions to the problem where many of the constraints expressed in the C matrix are satisfied.

In those problems where the global structure is not revealed by the constraint matrix (i.e. the C matrix only includes local constraints), the B matrix would eventually reflect not only the explicit local constraints, but also the implicit global structure in the problem (i.e. constraints between genes implied but not explicit in the C matrix). As the strength of the connections in B grew, it constrained the developmental process more and more, until it was only able to represent high-fitness solutions (those that satisfied many constraints). In one of the constraint problems they considered (the 'concentric squares' problem), this corresponded to modelling the correlations in the optimal phenotype. Initially the evolutionary process would become trapped at low-fitness local optima in each episode (that only satisfied some local constraints), but after many exposures with sufficient phenotypic variation and a low enough learning rate, the evolved B matrix would cause it to always evolve a globally optimal phenotype regardless of the state of the G vector after a reset. Note that it is impossible to canalize a specific phenotype, only a pattern of correlations, as the model is completely symmetric (inverting the values in G will always produce an inverted P with the same B). Accordingly, complementary phenotypes have the same fitness in these scenarios.

In the same paper, Kounios et al. (2016) compared the evolutionary process including the GRN model to one in which there is a one-one G-P map. Without the genetically controlled developmental step, the evolutionary process is unable to improve its performance over time because it undergoes a total reset: its whole state is G (there is no B), and G is randomised each episode. This is in contrast to the GRN model where B provides additional state which is not lost each episode. Because the GRN model has *partial* resets, the system is forced to explore disparate regions of the genotype and phenotype spaces, but is still able to integrate useful information over time. Similar behaviour was demonstrated in the random constraints problem.

To demonstrate the importance of partial resets, and to address some concerns raised by reviewers with the model presented in Kounios et al. (2016), the model we use (as described in Section 3.2) differs in the following ways:

- B **mutates along with** G**.** In Kounios et al. (2016), the 'delta-rule' update process is used to evolve the entries in the B matrix. This process explicitly separates the process of hill-climbing to a local optimum and updating the weights in B. In our evolutionary model there is no separation: B accumulates random mutations subject to selection through each episode. Each evolutionary step, a mutation is performed in either B or G with equal probability as described in Chapter 3.

  This change addresses the concern that the 'delta-rule' update process is not biologically realistic: it is an explicit learning rule, and though it is intended to simulate long periods of slow evolution, the conditions where this might be the case are not clear.

- **All resets are partial, and parameterized.** At the start of each episode, rather than resetting G and leaving B unmodified, a fixed number (Q) of mutations are applied without selection. This includes mutations in B. The parameter allows us to vary how large the partial resets are: a parameter that directly controls how much phenotypic variation the system experiences over evolutionary time. In this role, it is somewhat analogous to the frequency of multi-peaked instances Z in Chapter 3 (see discussion in Section 4.7.1).

  This addresses the concern that drift in B would undermine the products of evolution, though this is primarily a matter of timescales: if periods of drift are short compared to periods of strong selection (as simulated by the delta-rule), then the assumption that B doesn't change may be reasonable. In any case, this change shows that the model is not too sensitive to mutation in B.

  Additionally, the ability to parameterize the *amount* of drift will help to confirm that the description of the mechanism in Kounios et al. (2016) by which local (and ultimately global) problem structure is inferred is accurate; specifically, we will be able to show that the ability to 'solve' a problem depends directly on the size of the partial-resets at the end of each episode.

- **Use of an explicitly modular environment.** In Kounios et al. (2016), two environment models are used to demonstrate the evolutionary consequences of evolving a GRN, the first of which is a local-constraints problem (the concentric squares problem), and the second of which is a random-constraints problem. Here we use explicitly modular problems (described in Sections 4.3 and 4.6) where inference at one level of variation will reveal structure at a higher level.

  Neither problem employed by Kounios et al. (2016) lends itself to being solved by a hierarchical process (there is no hierarchy in the problem), but the second problem (the random constraints) is related to the modular problems we will use. Unlike the random constraints problems, the modular problems we use are well characterised, and contain easy-to-identify structure, which will help us to understand the evolutionary dynamics that occur.

A further difference is that we change the model initialisation for experiments where we include a cost of connections such that B begins with a strong set of self-connections (an identity matrix). This change enables us to use a high cost coefficient λ which causes the evolutionary system to come to completion more quickly and reliably. Without this change, the system exhibits too unfortunate tendencies: for reasonable choices of the mutation rate M_B, it it tends to rapidly canalize a single module assignment, and so loses necessary variability (as in the high Z case in Chapter 3); for high λ the system becomes trapped in the 'pit of despair' (see Figure 5.1 in a later chapter), where the cost of connections suppresses the evolution of any connections in the B matrix from the outset because the marginal benefit is very small when the B matrix is weak. The entries on the diagonal of the matrix cause the system to start near saturation of the developmental curve, which avoids the shallow region of the developmental function (which produces the pit of despair) and prevents a disproportionate increase in connection weights in the first few episodes that can bias future variability according to the limited variability observed by the system. It seemed to be the case that the system behaved well for lower values of λ and M_B; however, the long run-times made it infeasible to comprehensively explore this scenario.

4.3 Model of an Unchanging Modular Environment

The MC (Modular Constraints) problem Watson et al. (2011c) describes a class of 'correlation' problems, where the benefit awarded to a phenotype P is proportional to $\quad_{ij} C_{ij} P_i P_j$, where C is a square matrix of the same dimensions as P, and represents the benefit constraints between genes. The MC problem is of interest because it provides an idealised modular constraints problem: the benefit constraints between genes is large within modules, and small between modules. Traditionally, the problem is described by 3 parameters:

- N, the number of modules (we will generally not use this symbol in the text as it conflicts with the number of genes N used in other parts of this document)
- K, the number of genes in each module (conflicts with the number of evolutionary steps per episode; we will use alternative notion were possible)
- p, the relative magnitude of the inter-module benefit constraint

A phenotype vector of length $N \times K$ is divided into N equally sized 'modules' of K genes. A large benefit is awarded to phenotypes where the genes within modules agree (i.e. all -1 or all $+1$): the stronger the agreement, the greater the reward. A weaker benefit (proportional to $p < 1$) is also awarded for agreement between modules (or, more precisely, between the genes in *every* module).

Though it is nicely characterised with matrix products, and can be implemented with a constraints matrix as in Equation 4.1, here we define a generalised MC problem as a sum of explicit epistatic module contributions. Each module contribution is a non-linear function of the mean of the expression of its component genes, $\eta(x)$, the 'module-benefit-function', multiplied by a coefficient c_m. The mean expression of the genes within each module is a value in the range $[-1, 1]$, and the module-benefit-function produces a result in the range $[0 - 1]$. Unlike in Chapter 3, the module-benefit-function is the same for all modules, is symmetric (no preference for positive or negative gene expression, only agreement between genes), and the coefficient c_m for each module $m \in M$ does not change over time. For no particular reason, the sum of these contributions is normalised into the range $[0-1]$ to compute the benefit, b:

$$b(P) = \frac{\sum_{m \in M} \left(c_m \times \eta \left(\frac{1}{|m|} \sum_{i \in m} P_i\right)\right)}{\sum_{m \in M} c_m} \tag{4.2}$$

As in Chapter 5, a cost term will be included in some experiments, the mean cost of each individual entry in the B matrix:

$$c(B) = \frac{1}{N^2} \sum_{i,j}^{N,N} \phi(|B_{ij}|) \tag{4.3}$$

In this chapter, the cost-function ϕ will always be the identity $\phi(x) = x$. The cost is scaled by a parameter λ, with the final fitness of an individual given by:

$$f(B,P) = b(P) - \lambda c(B) \tag{4.4}$$

In addition to the 'sub-modules' of K genes, the set of modules M include an additional 'super-module' that contains every gene. This module provides the inter-module fitness contributions, and will have a different choice of c_m that the sub-modules modules, for which $c_m = 1$. Rather than using the traditional p parameter for this purpose, we will instead use a more general 'decay factor' d_f, which is proportional to $1/(1/p-1)$ (roughly p for p small), of which we will make greater use in Section 4.6.

Note that the denominator of Equation 4.2 is a constant factor (the sum of the module benefit coefficients) included to scale the benefit into the range $[0,1]$ given the module benefit function $\eta(x) \in [0,1]$ for $x \in [-1,1]$.

In our experiments, we use an MC problem where positive correlations are rewarded everywhere. In general, an MC problem may reward a (consistent) combination of positive and negative correlations multiplying the phenotype by a 'target' vector S of -1 and $+1$ values determining the pattern. Such a modification does not change the behaviour of our model; using an all-positive problem makes it easy to discuss and interrogate the system without losing any generality in the result.

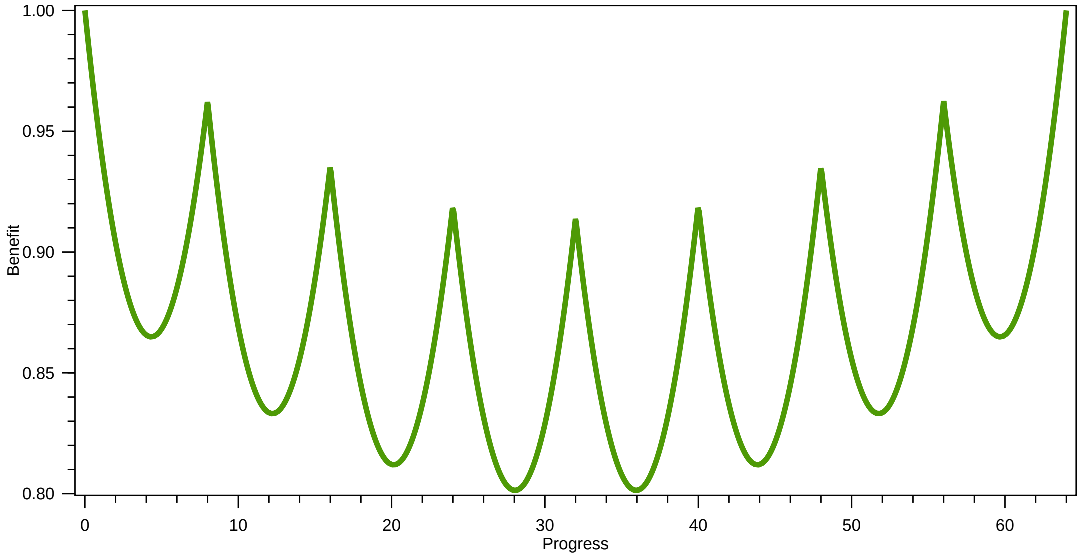


Figure 4.1: This plot shows how the fitness benefit of a phenotype changes as it transitions from a vector of all -1 to all $+1$, one trait (or equivalently, module) at a time. Each peak represents a locally optimal phenotype, where the traits within every module are strongly expressed (all have magnitude 1) and of the same sign. The height of each peak is a quadratic function of the number of modules which are all of the same sign (that is, the degree of between module concurrence). Though traditionally characterised by a correlation matrix, the MC problem is equivalent to a sum of quadratic epistatic module contributions, where all modules - apart from making their individual contributions - are considered part of an additional 'super-module' which contains every trait. Crucially, the shallow gradient in the middle of the quadratic function means that 'undecided' modules (i.e. modules whose mean expression is close to zero) are influenced by the gradient in the 'global' module: the basin of attraction for phenotypes is larger if there is more agreement between modules.

4.4 Smooth MC

We refer to the MC generalisation where the module-benefit-function is $\eta(x) = x^2$ as the 'Smooth MC' problem. Figure 4.1 provides some indication as to the nature of the benefit landscape produced by the smooth MC: the quadratic choice of module-benefit-function function produces the smooth, rounded basins, and makes the problem equivalent in shape to an MC problem. A proof of the equivalence between this description of the MC problem as a sum of independent module-benefit-functions and the traditional matrix formulation is given in Appendix B.1, but the idea behind the equivalence is that the benefit within a module (i.e. a constraint matrix where each entry is has the same positive value c) is equal to

$$\sum_{ij} cP_iP_j = c\left(\sum_i P_i\right)^2 \tag{4.5}$$

i.e. the sum of the module expressions squared times the constant. The constraint matrix for the whole MC problem can be decomposed into the sum of one fully populated matrix where $c = p$ is small, and one with block-diagonal entries where $c = 1 - p$. The first represents the large module of all genes, and the second all of the smaller sub-modules.

The evolutionary process was run with various neutral (i.e. without selection) mutation counts Q to investigate the ability of the system to 'learn' the problem structure for different amount of simulated genetic drift. Two configurations were used: one with a continuous G space and no cost of connections (see Table 4.2); another with a binary G space and L1 cost of connections (see Table 4.3). Each configuration underwent 40 replicates (complete runs with different random seeds). The parameters for the smooth MC probably are summarised in Table 4.1).

The distribution of fitnesses at the end of the simulations for each evolutionary process are shown in Figures 4.2&4.3. It is clear that both evolutionary processes are able to find higher fitness phenotypes more often when the neutral mutation count Q is greater (though if the values were too large the noise introduced into B may reverse the trend).

Parameter	**Symbol**	**Values**
Number of genes	N	64
Number of modules		8
Number of genes per module		8
Generalised decay factor	d_f	0.75
Super-module correlation	p	$1/(1 + 64^2/(0.75 \times 8^2)) \approx 0.012$
Smooth Module benefit function	η	$\eta(x) = x^2$
Spiky Module benefit function	η	$\eta(x) = \lvert x \rvert$

Table 4.1: Parameters for the MC Problem. The equivalent p value is computed from the actual choice of $d_f = 0.75$.

Parameter	**Symbol**	**Values**
Evolutionary steps per episode	K	50000
Episodes		10000
Number of developmental time-steps	T	10
Gene Expression Decay Rate	τ	0.2
G vector mutation magnitude	M_G	0.1
G vector mutation type		Uniform
B matrix mutations probability	R_B	0.5
B matrix mutation magnitude	M_B	2×10^{-6}
B matrix mutation type		Uniform
Cost Coefficient	λ	0
Neutral Mutation Count	Q	Variable

Table 4.2: Parameters for the evolutionary model without cost of connections.

Parameter	Symbol	Values
Evolutionary steps per episode	K	2000
Episodes		100000
Number of developmental time-steps	T	10
Gene Expression Decay Rate	τ	0.2
G vector mutation type		Binary
B matrix mutations probability	R_B	0.5
B matrix mutation magnitude	M_B	0.1
B matrix mutation type		Uniform
Regulation Function	ϕ	$\phi(x) = \frac{\lvert x\rvert}{N^2}$
Regulation Coefficient	λ	2
Neutral Mutation Count	Q	Variable

Table 4.3: Parameters for the evolutionary process with cost of connections.

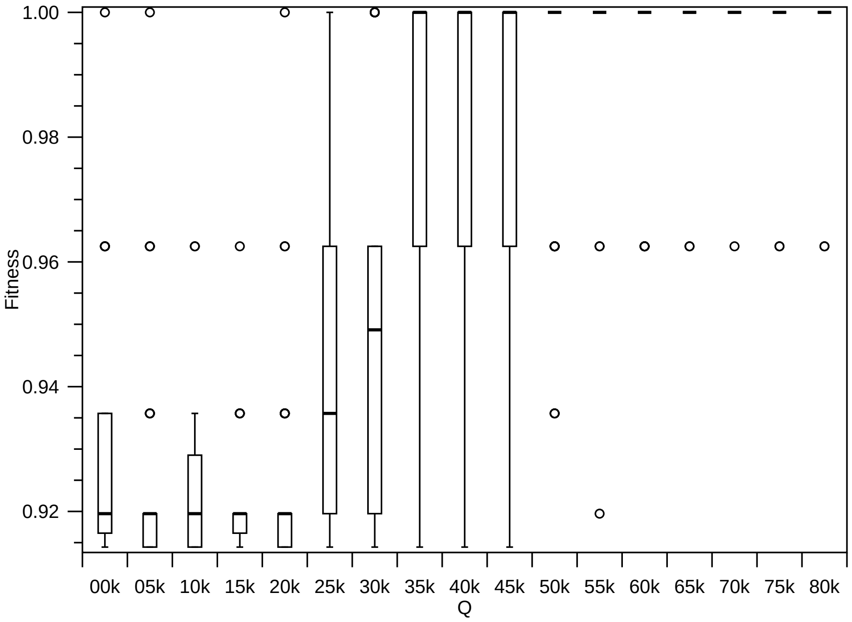


Figure 4.2: Terminal fitness distributions for the Smooth MC without cost of connections. For low neutral mutation counts (Q small), there is insufficient variation between episodes, and the system fixes to a locally optimal phenotype that is not the global optimum. For large neutral mutation counts, the variability in observed phenotypes enables the system to learn the problem structure, and ultimately canalizes the pattern of gene expression of the globally optimal phenotypes in most cases. In this figure, the fitness is computed on the sign of the phenotype: this reveals the change in distribution without the distraction of the absolute magnitudes of the phenotype.

When either evolutionary model is applied to the smooth MC problem, it quickly ‘learns’ the genes that make up each module, creating a developmental module of positive intra-module interactions, just as in Chapter 3. These modules in B provide a developmental constraint that practically limits the space of representable phenotypes to those where the genes are consistent within the modules. While this may provide valuable robustness, it is not necessary in order for the system to find fit phenotypes: the hill-climber without a cost of connections (which does not evolve hierarchy) is able to search locally to

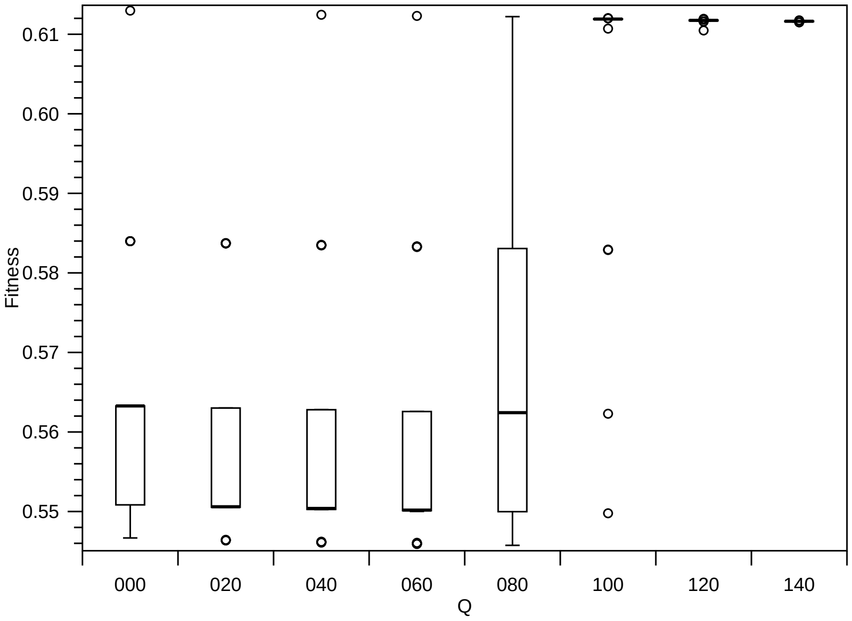


Figure 4.3: Terminal fitness distributions for the Smooth MC with cost of connections. Just as without the cost of connections (Figure 4.2), for low neutral mutation counts (Q small), there is insufficient variation between episodes, and the system fixes to a locally optimal phenotype that is not the global optimum. For large neutral mutation counts, the variability in observed phenotypes enables the system to learn the problem structure, and ultimately canalizes the globally optimal phenotypes in most cases. Compared with Figure 4.2, the comparative low fitnesses and small choices of Q are due to the inclusion of the cost of connections and binary nature of G respectively.

resolve conflicts within modules when the B matrix is empty or the within-module connections are removed, and the evolutionary process can still solve the problem (find the global optima) if we prevent mutations on the within-module regulatory connections (such that they remain zero). The ability of the evolutionary process without a cost of connections to ultimately canalize the globally optimal phenotypes is a consequence of the system visiting higher fitness local optima more often than low fitness phenotypes. These high fitness phenotypes are characterised by having more modules in agreement (their component genes have the same sign for their phenotypic expressions) than lower fitness phenotypes: the larger the majority, the higher the fitness. Evolution changes the entries in B to reinforce the correlations in these local optima which, on-average, are positive, and so correspond to the environmental inter-module constraints. As the magnitude of the connections in the B matrix grows, this limits the variability of the evolutionary system further, and transforms the originally unreliable mechanism for locating high fitness phenotypes (by following the weak inter-module gradients when possible) into a reliable one where *only* high fitness phenotypes can be represented, and genes evolve to coordinate with the majority.

Were all local optima visited with equal probability (i.e. independent of their benefit), the frequency with which any two modules in a phenotype would be coordinated would be exactly half. This is not the case, because the basin of attraction - the region of genotype space from which a local search process like out hill-climber may reach it - is larger for phenotypes which greater between-module agreement. This is a consequence of the quadratic nature of the MC problem (i.e. $\eta(x) = x^2$): the differential of the benefit contribution of a module with respect to any entry in G it contains is proportional to the magnitude of the mean expression in the module $\eta'(x) = 2x$. As such, when a submodule is undecided (the mean gene expression is close to zero), the sign of differential of the whole benefit function may be determined by the - potentially larger - differential with respect to the super-module. This effect moves the local minima (phenotypes where any small change will increase the fitness regardless of direction) closer to low-fitness local optima. This means that the basin of attraction is larger for fitter phenotypes, and consequently a fine-grained evolutionary process is more likely to visit them after a partial reset. The 'fine-grained' point is important, and the offset must be comparable to the smallest step that can be made by the evolutionary process.

Again, as in Kouvaris et al. (2017), the evolution of the modules in B in this module reveals low-level problem structure and confers robustness, but in this case they do not help the system to avoid or escape local optima. Crucially, the modules do not enable it to exploit the apparent problem structure (unless they become hierarchical). This ability of the evolutionary process without a cost of connections to solve the problem is rather due to the – much weaker – between-module connections that reflect the disparity in basin size. Indeed, the evolution of strong intra-module connections may preclude the evolution of consistent super-module connections, as it has the effect of normalising the basin sizes: the ability to observe the wider basins of fit combinations of modules in *phenotype* space is diminished as the intra-module connections grow stronger, as even a small deviation from the mid-point in *genotype* space can produce a large deviation in *phenotype* space.

4.5 Spiky MC

As in Chapter 3, the use of a linear (L1) cost of connections can induce the evolution of hierarchical modules, which we could expect to increase evolvability by enabling the system to switch whole modules and, consequently, hillclimb in the space of modules rather than genes. This ability would make it easy for the system to find a globally optimal arrangement in short order. However, the same signal that enables the hillclimber without L1 cost of connections to solve the smooth MC problem is present with the L1 cost of connections. Consequently, it is possible for the evolutionary process with cost of connections to solve the smooth MC problem relying solely on the signal from differing basin sizes described in the previous section, rather than exploiting the variability afforded by hierarchy. This makes it hard to determine whether it is the signal from the distribution of evolved phenotypes or the evolvability enhancing effects of hierarchy that enables the evolutionary process with a cost of connections to find high fitness phenotypes.

In an effort to eliminate the signal, and to provide better consistency with the model from Chapter 3, we consider a modification of the generalised MC problem with the explicit intention of making the basin of attraction for all modules the same size. The variation in basin size in the smooth MC problem is a consequence of the shallow gradient near the minimum of the quadratic module-benefit-function, so we substitute the quadratic module-benefit-function for a piecewise-linear module-benefit-function, $\eta(x) = |x|$. This produces a problem with the same ideal properties as the traditional MC problem (strong intra-module constraints, weak inter-module constraints), but where the gradient of the within-module epistatic effects is steep everywhere (specifically, steeper than the between-module epistatic effects). This change prevents a hillclimber operating in the phenotype space from observing the super-module gradient for small decay factor d_f. Figure 4.4 provides some indication as to the nature of this modified fitness landscape.

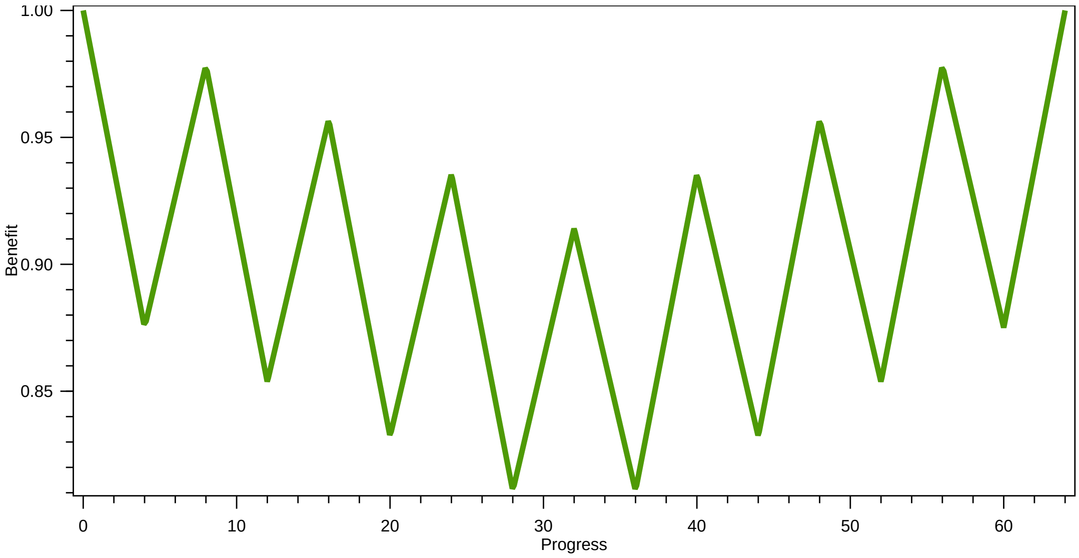


FIGURE 4.4: Here, the MC problem is modified, such that the epistasis function within each module (including the all-encompassing super-module) is an even-symmetric piecewise linear function, rather than an even-symmetric quadratic function. Because this function has the same gradient everywhere, it is harder to 'see' the weaker super-module correlations. The effect is to make the basin of attraction for each locally optimal phenotypes the same, regardless of its fitness benefit.

In this spiky environment, there is no signal in phenotype space to bias the distribution of explored local optima, which restricts the ability of the system to integrate useful information about the problem early in evolutionary time.

With the linear (L1) cost of connections, hierarchy readily evolves (Figure 4.5). As each module becomes sufficiently hierarchical that module flips become viable, it begins to align with the other modules (Figure 4.6). The effect is that the distribution of end-of-episode phenotypes (and genotypes) changes from essentially random to biased toward patterns with a clear majority (Figure 4.7). Notably, the hierarchical modules are able to align with non-hierarchical modules: the system does not need to wait for all modules to become hierarchical before it can begin integrating the between-module signal.

Once there is a consistent majority, the signal for inter-module connections goes from providing no signal on average (and so is suppressed by the cost of connections) to each module tending to be directed by strong modules that align with it (the reasons for this will be explored in Chapter 5). Ultimately, a grand-hierarchy emerges, where a single gene controls all other genes (i.e. all genes in the super-module, as in Figure 3.5B). This results in a wholly consistent pattern of modules in the phenotype (Figure 4.7).

As with the Smooth MC, the ability to 'learn' the right modules depends on the amount of drift between episodes Q (Figure 4.8): low values of Q cause the evolutionary process to fix on an incorrect pattern of modules.

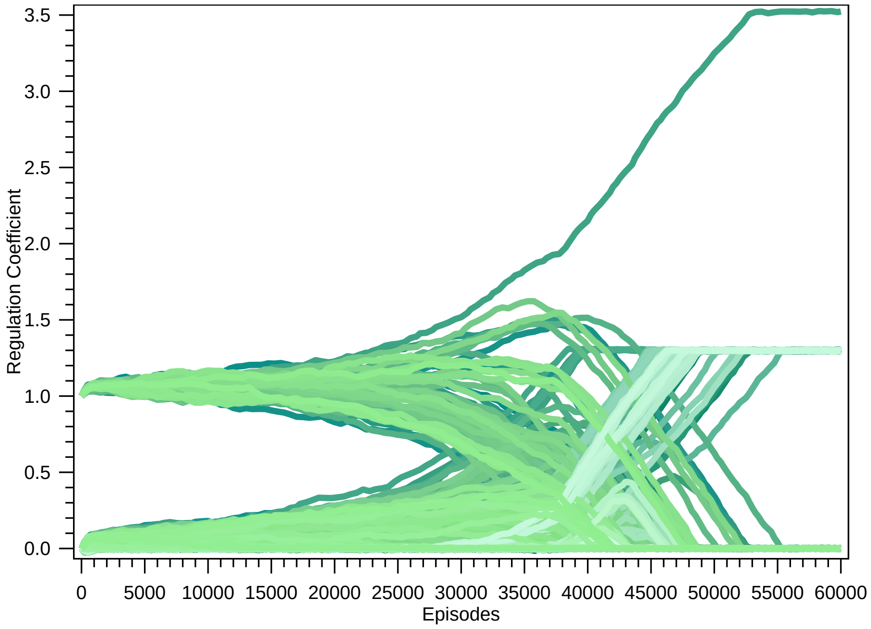


FIGURE 4.5: Regulatory connection trajectories for the evolutionary process with L1 cost of connections in the Spiky MC environment and $Q = 140$ (1 sample per 250 episodes). Initially the system evolves 8 mostly independent modules, but by around 40 thousand episodes a grand hierarchy begins to be emerge, with a single gene controlling all others.

4.6 A Continuous Hierarchical If-and-only-if

The transition from independent hierarchical sub-modules to a single grand-hierarchy in our experiments with the MC problem indicates the evolution of hierarchy can motivate a change in variability that can motivate the evolution of hierarchies at a 'higher-level' (i.e. modules that contain sub-modules). This corresponds to a rescaling from hill-climbing in the sub-modules of the MC problem by flipping individual genes, to hill-climbing in

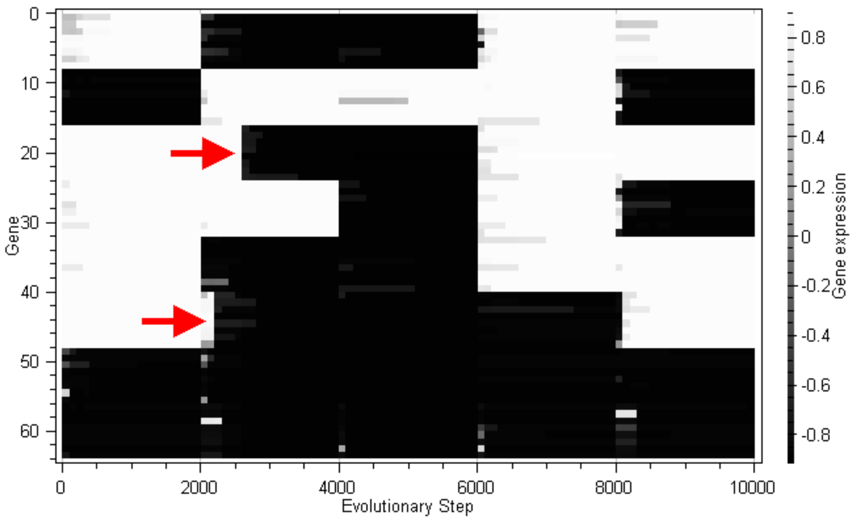


FIGURE 4.6: Phenotype patterns for the evolutionary process with L1 cost of connections in the Spiky MC environment and large $Q = 140$ at 30 thousand episodes (1 sample per 100 evolutionary steps, a new episode begins every 2000 evolutionary steps). By 30 thousand episodes, some modules are sufficiency hierarchical that they can start switching in order to better coordinate with the other modules. Two whole-module switches (indicated in red) can be seen in the second episode of this trace: the first (lower red allow) creates a majority, and the second (upper red arrow) joins it. Note that the same modules do not switch in later episodes: while they can switch under some circumstances, it is not yet reliable.

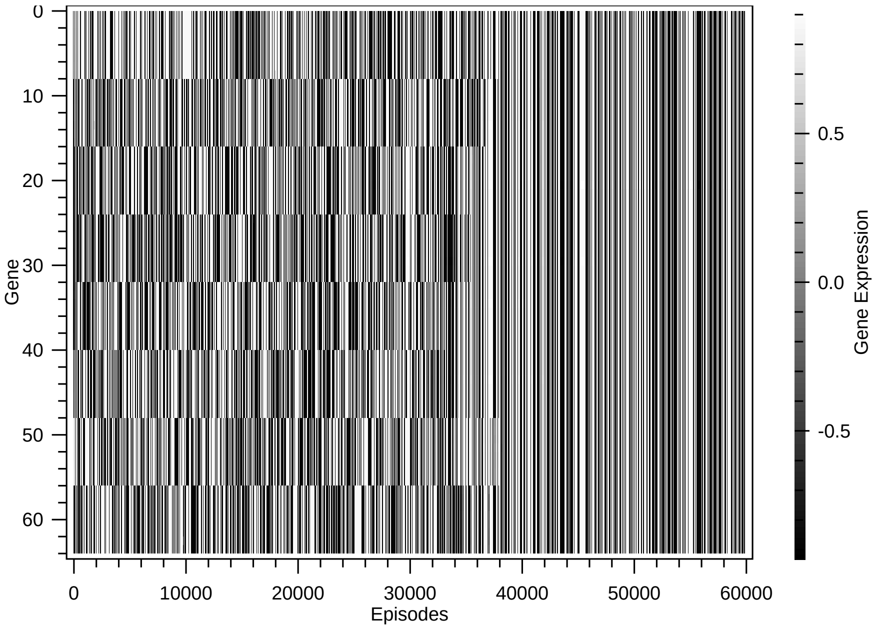


FIGURE 4.7: Phenotype patterns for the GRN hillclimber with L1 cost of connections in the Spiky MC environment and large $Q = 140$ (1 sample per 250 episodes). Initially there is no coordination between the phenotypes found by the evolutionary system: each module is assigned effectively randomly by the reset operation. After about 35 thousand episodes, there is some apparent consistency between a number of modules. After about 38 thousand episodes, the inter-module variation is lost and all the modules are always coordinated, producing a maximally fit phenotype.

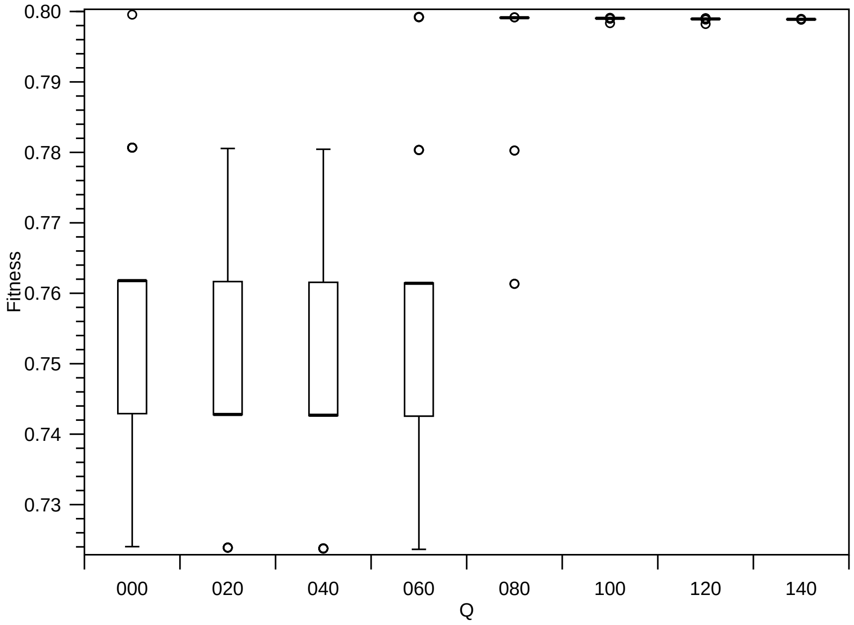


FIGURE 4.8: Terminal fitness distributions for the GRN hillclimber with L1 cost of connections in the Spiky MC environment. For low neutral mutation counts (Q small), there is insufficient long-term variability, and the system over-fits to a locally optimal phenotype that is not the global optimum. For large neutral mutation counts, the variability in observed phenotypes enables the system to learn the problem structure, and ultimately canalises the globally optimal phenotypes.

the super-module by switching sub-modules. There is no obvious reason why we should not be able to perform these transitions at multiple levels, and this possibility is what we shall explore next. The 'Hierarchical if-and-only-if (HIFF)' (Watson and Pollack, 1999) problem is a non-continuous binary problem designed to test the ability of systems to solve hierarchically problems, where it is necessary to 'solve' the small component sub-problems before any signal from the large super-problems can be seen, and provides a good basis for a problem to test whether hierarchy can perform this 'scale-invariant re-scaling': combining modules into a larger modules at any level of variation.

Because our model of development is a continuous (i.e. not discrete) process and we depend on directional selection for increasing the magnitude of gene expression (our phenotypes are not binary), we need to use a continuous version of HIFF. This landscape will be based on the spiky variation on the generalised MC problem we described in section 4.3, but with smaller sub-modules and additional modules introduced 'between' the small sub-modules and the all-encompassing super-module. These additional 'layers' of modules won't introduce any local-optima into the fitness landscape, but will change the relative benefit of different fitness optima, and further disrupt the signal from the all-encompassing super-module. Plotting a cross-section of the benefit curve for phenotypes starting from all negative, and ramping individual gene expressions linearly to positive, the continuous HIFF problem looks like Figure 4.9. We can visualise the decomposition by showing the benefit functions for each module at each 'layer' of modules, as in Figure 4.10. The maximum benefit attainable from each module decreases (according to the decay factor d_f) as the modules become larger: the contribution of each layer is the product of the contribution of the layer 'below' (i.e. the payoff of a single sub-module) and the decay factor. By choosing $d_f \leq 1.0$ we guarantee that, for any number of layers, the gradient (where it exists) of the whole benefit function will have the sign of the module-benefit function of the smallest module contributing: making infinitesimal changes in phenotype space, it is better to coordinate the smallest modules than to try to coordinate the larger modules.

As in Section 4.5, we use a piece-wise linear module benefit function $\eta(x) = |x|$, so the basin of attraction for each local optimum is the same size in phenotype space. In the experiments presented, the smallest sub-modules are of size 2, and each super-module comprises two sub-modules. There are 32 sub-modules of size 2 (64 genes in total), and so 6 'layers' of modules. Model parameters for the Continuous HIFF experiments are specified in Table 4.4.

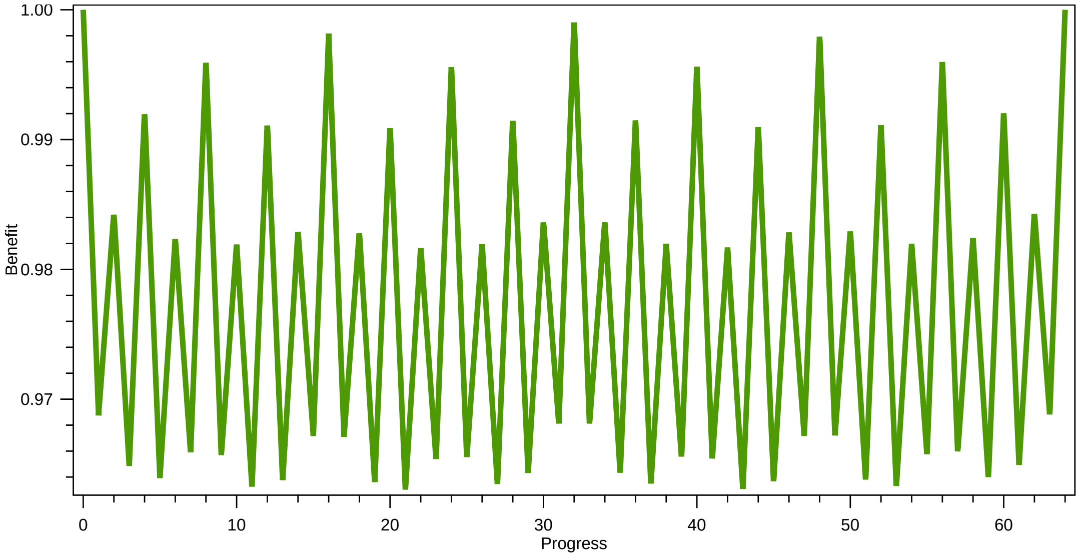


Figure 4.9: Cross-section of the Spiky Continuous-HIFF, revealing the many locally optimal phenotypes, where modular constraints in different layers of the hierarchy are not satisfied. The basins of attraction for the highest fitness phenotypes correspond to a very small region of the phenotype space, and are unlikely to be found by change.

As for the MC problems, a spread of experiments was run with varying partial reset size (more or fewer neutral mutations, Q) with the Spiky Continuous HIFF to simulate different amounts of inter-episode drift. Figure 4.14 shows how the distribution of fitnesses for 40 replicates after 50 thousand episodes of evolution changes with the number of neutral mutations Q. When Q is zero, the system simply climbs to a locally optimal phenotype and remains there for the rest of the simulation. Because there is no variation introduced between episodes, it is impossible for it to infer any problem structure, and what evolves is a single grand-hierarchy that efficiently produces one of the 2^{32} possible

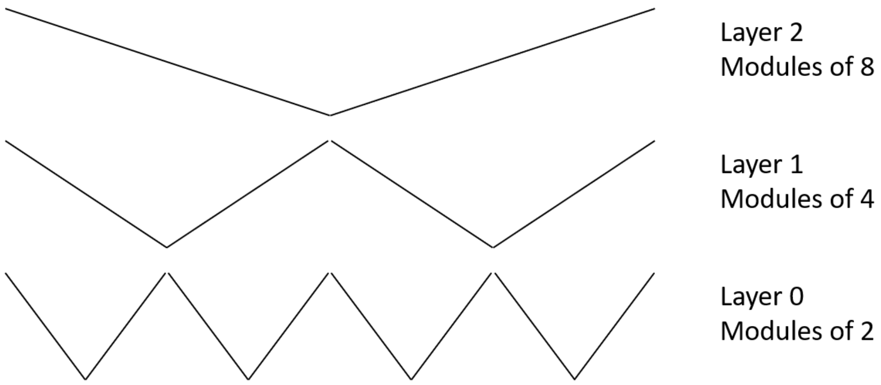


FIGURE 4.10: The fitness landscape is the sum of the fitness contributions of multiple modules of different sizes. Modules in the 'upper' layers are the union of modules in the 'lower' layers.

Parameter	**Symbol**	**Values**
Number of modules	N	32
Number of developmental time steps	T	*variable*
Decay factor	d_f	0.5
Evolutionary steps per episode	K	4000
Episodes		100000
B mutation magnitude	M_B	0.005
B mutation probability	R_B	0.5
Cost function	ϕ	$\phi(x) = \lvert x \rvert$
Cost coefficient	λ	8
Neutral 'reset' operations	Q	*variable*

TABLE 4.4: Model parameters for the Spiky Continuous-HIFF.

module assignments. Of these possible assignments, only 2 will provide the maximum benefit, and so the odds of it occurring by chance are slim (approximately 2^{-16}).

When Q is sufficiently large, the system explores many local optima, and evolves independent hierarchical modules corresponding to the smallest modules (of size 2 in these experiments). Once any one module is sufficiently hierarchical, the afforded variability of switching the whole module with a single-point mutation means that the module will begin 'aligning' with its partner module in the layer above. The lack of variation between the modules permits a hierarchy to evolve across both of them, which produces a further change in variability, enabling this super-module to align with its partner module in the *next* layer up. This process continuous until either all modules are integrated into a single hierarchy, it makes a mistake (e.g. due to noise), or it is no longer beneficial to perform module switches (the marginal benefit of a switch falls below zero).

This last effect (the marginal benefit of switching being non-positive) is most easily demonstrated with an additional set of experiments where we vary the number of developmental time-steps T (Figure 4.15). In these experiments, the partial resets were an explicit G reset (each entry is assigned either -1 or $+1$ with equal probability; B is left untouched): this is a less biologically plausible assumption, but avoids conflating parameter effects, and produces cleaner (for easy interpretation) evolutionary trajectories. The trajectories shown below are from one of these experiments with the maximum number of developmental time-steps $T = 20$, as in the experiments with variable Q. Figure 4.11 shows one such experiment, and Figure 4.12 the end-of-episode fitness and frequency with which a phenotype with a maximum benefit is evolved.

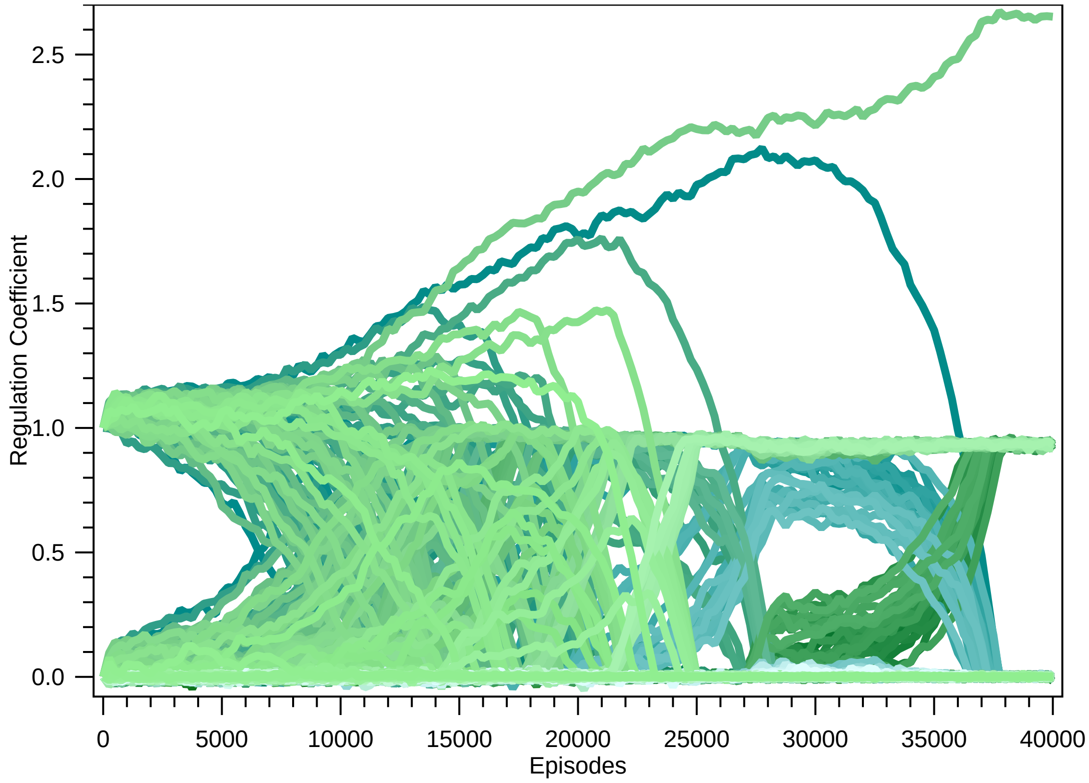


FIGURE 4.11: Evolutionary trajectories of regulatory connections for one experiment with $T = 14$ and $Q = 500$. It takes a long time for the smallest modules to decay from a near-identity topology to a degenerate hierarchy. Thereafter, however, the system rapidly combines pairs of modules into super-modules, until two modules remain, which takes somewhat longer to combine into a single grand-module. The strong self-connection is an indication of the dominance of the lead gene. The many self-connections that grow (lines above 1) and then fall-off are leaders of sub-modules that are subsumed into another module with a stronger (more dominant) self-connection.

The spread with variable T shows how hierarchy does not evolve for $T < 2$ (it requires recurrence, as will be discussed in Chapter 5), and in fact the regulatory connections are removed altogether, reducing the cost significantly. The larger T, the larger the modules that the system can combine, and so the more layers the evolutionary process can model from the environment, and so the greater fitness it attains on average. The experiments with varying Q use a value of $T = 10$, which is why they are not able to evolve a grand hierarchy, and do not always find the highest fitness phenotype.

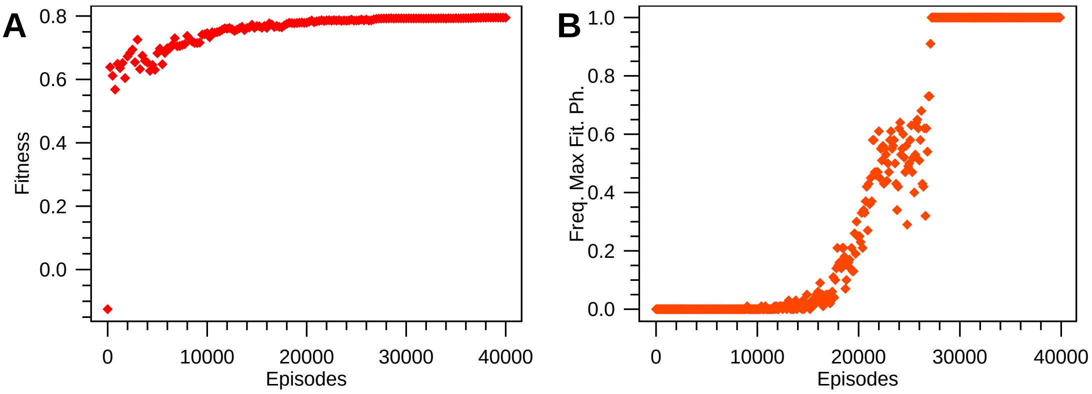


FIGURE 4.12: Evolution finds fitter phenotypes as the system becomes able to coordinate larger and larger modules. **A**: the absolute fitness at the end of each episode increases over evolutionary time as higher fitness phenotypes are found and developed more efficiently. The very low-fitness dot at the start is due to the large cost of the initial entries on the diagonal of B. (1 sample per 5 episodes) **B**: The frequency with which evolution finds the fittest phenotypes (i.e. ones where all genes are in agreement) is initially very low (theoretically on the order of $2/2^{32}$. It flat-lines at 1 once the evolutionary process is able to coordinate within the larger module: note that this is before it evolves the grand hierarchy (the grand-hierarchy does not confer any valuable variability (see Figure 4.13) but evolves anyway).

4.7 Discussion

In Chapter 3 we showed that the evolution of hierarchy can enable an evolutionary process to better exploit a modularly changing rugged environment. In this Chapter we have extended this to show the same evolved hierarchy can enable the evolutionary process to perform module switches to satisfy inter-module constraints, which we simulated with larger modules with relatively weak interactions. At the start of the experiments, the system was unable to satisfy the high-level constraints of the super-modules as it could only observe the strong gradients from the small sub-modules given the available phenotypic variability supplied by the B matrix: each gene would coordinate with its neighbours to maximise the benefit conferred by the lowest layer of modules. By using the Spiky MC and Spiky Continuous-HIFF problems we minimised the opportunity for genes to coordinate between modules by making the basins of attraction for each locally optimal phenotype the same size, which is not the case with the traditional quadratic module-benefit-functions.

As in Chapter 3, the evolution of hierarchy is induced by the use of a linear cost of connections, and to be useful requires that it evolves in modules that correspond to the epistasis in the environment: in this case, the modules correspond to the smaller, lower-level modules which must be combined to satisfy the higher level modules. By performing a ‘partial reset’ (a fixed number of mutations without selection to simulate drift) we are able to introduce enough long-term variation that these modules can be inferred by the evolutionary process. With the inclusion of a linear cost of connections, these

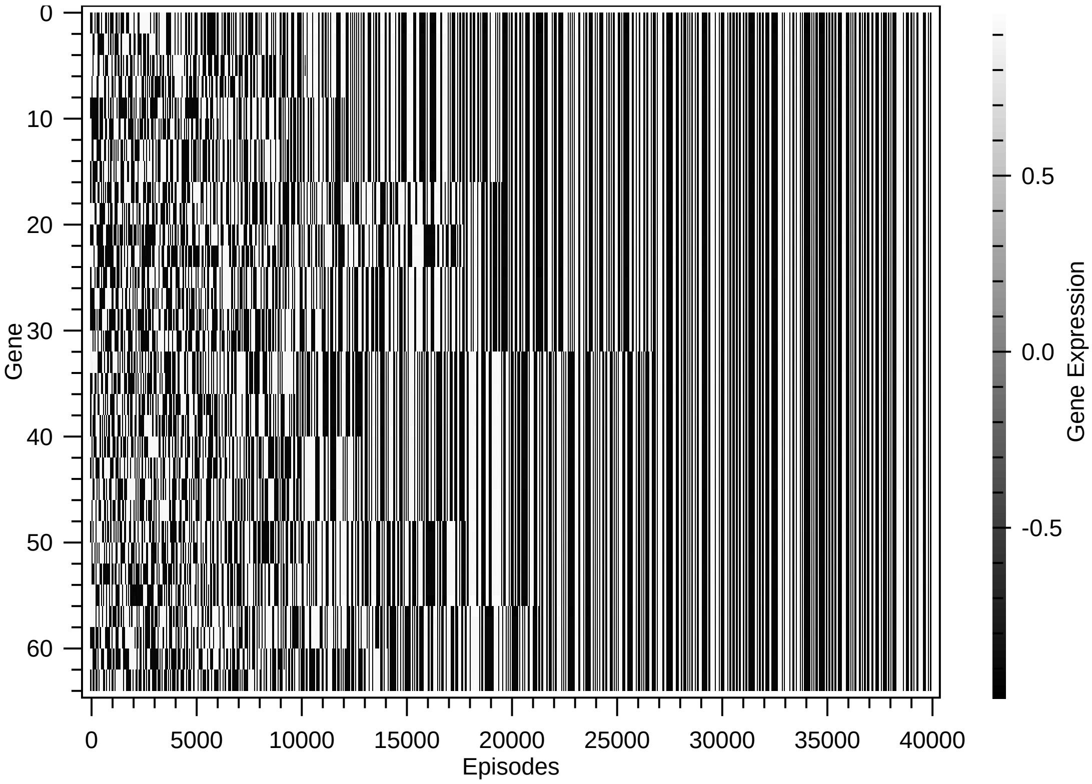


Figure 4.13: The phenotypes that evolve reflect the modularity and hierarchy in the gene regulatory network: neighbouring genes are consistent when they are members of modules, or when their module is a member of a module that contains a hierarchical module that can coordinate with its partner. As modules are combined into hierarchical super-modules, more of the genes 'agree'. This is visually apparent, as the modules are (superficially) spatially grouped.

modules evolve as hierarchies with the same key property as those described in Chapter 3: a single-point mutation on the dominant gene effects a near-total whole-module switch in the expressed phenotype. This switching ability enables the *independent* hierarchical modules to coordinate with the other modules (i.e. align with the majority) by following the (now observable) shallower inter-module benefit gradients. This changes the inter-module signal from being totally random (i.e. the probability of any two modules 'agreeing' is initially one-half) to being biased, such that the modules within a super-module agreed more often than half the time. This agreement implies a reduction in variation, and so a new module forms that is the union of its (now coordinating) sub-modules. The progression can be seen in Figure 4.16, where the B matrix of regulatory interactions is shown along with the auto-correlation between evolved phenotypes when the G vector of initial gene expressions is randomised. This 'Hebbian response' indicates which genes are usually correlated and foreshadows the evolution of hierarchy: correlations appear at the layer above the already evolved hierarchy.

Despite our efforts to design a selective regime that does not reveal associations at higher levels from the outset, it is possible to evolve high-fitness phenotypes with a

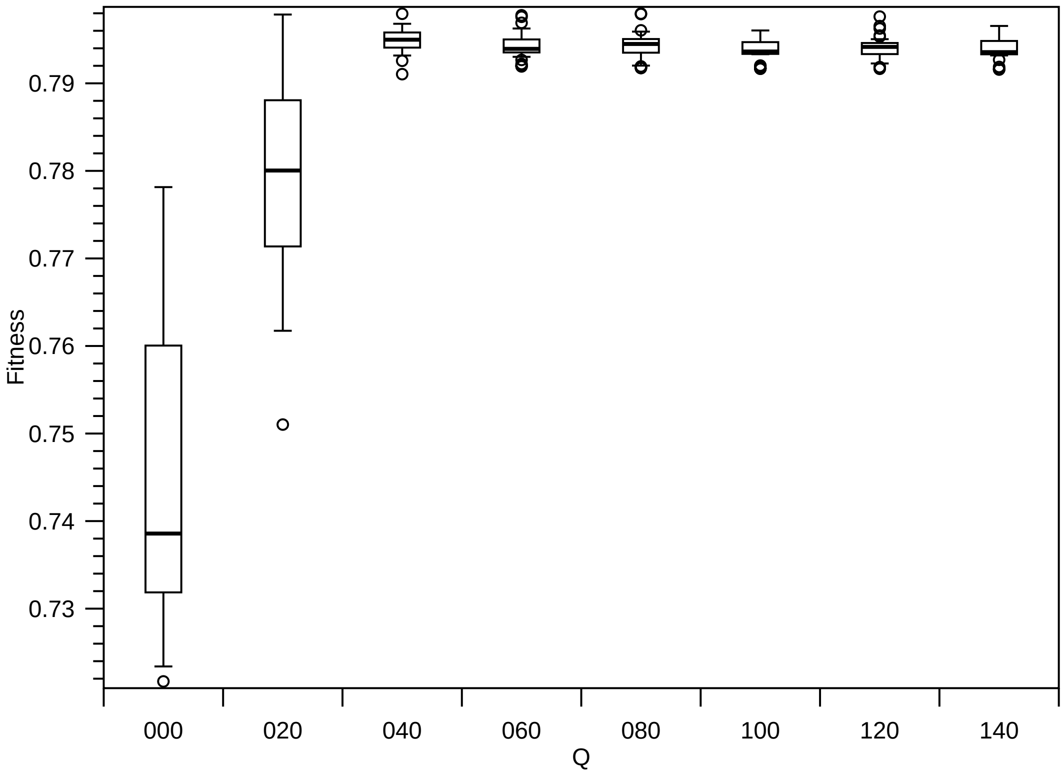


Figure 4.14: Larger partial resets - as controlled by parameter Q prevent over-fitting to sub-optimal solutions in the Spiky Continuous HIFF. As in the spiky MC problem, it is necessary to introduce sufficient variation in G that the system explores many different phenotypes for it to evolve the 'correct' modules. When Q is low, the system readily fixes on a sub-optimal phenotype.

mechanism that does not depend on hierarchy. The mechanism is not entirely clear, and we will present no results concerning it. Though the phenotype-fitness landscape explicitly has uniform-width basins in phenotype space, the gradients within the basins of attraction for low-fitness local optima are shallower than those of high-fitness local optima. The introduction of noise into the B matrix of regulatory connections distorts the genotype-fitness landscape, and this can cause the basins of attractions to become different widths in genotype space. This effect is largely removed by the use of a binary G vector of initial expressions, but not completely: if enough noise is present, it can move the minima of the valleys between local optima enough that it can change the widths of the basins, and so bias the evolutionary process toward evolving higher fitness phenotypes. The evolutionary process may fix on whatever phenotypic correlations it observes most often, and this may well produce phenotypes of an above-average fitness. Importantly, the amount of noise that is required to effect such a change is different for different pairs of local-optima: if two modules should be positively correlated to produce a fit phenotype, then to modify the landscape so as to bias the system to produce more phenotypes where they are negative correlated requires a stronger set of negative connections in the B matrix than to bias the system to produce phenotypes

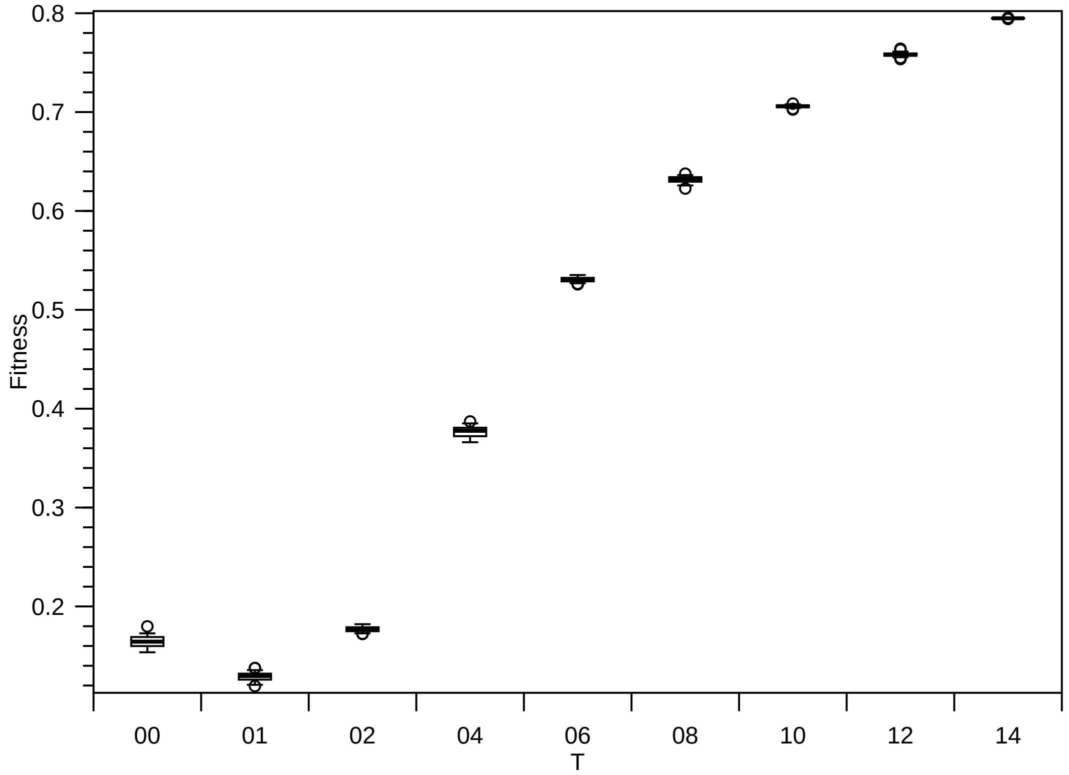


Figure 4.15: Longer developmental periods allow better exploitation of evolved hierarchy. For number of developmental time-steps $T \leq 1$, hierarchy does not evolve. Where hierarchy does evolve, more time-steps reduce the additive loss associated with module flips. This loss is larger for larger modules in our continuous-HIFF environment, and consequently module flips on large modules are not viable when T is small: the larger T, the larger modules can be exploited, and so the better end-of-episode fitness after the system has fixed.

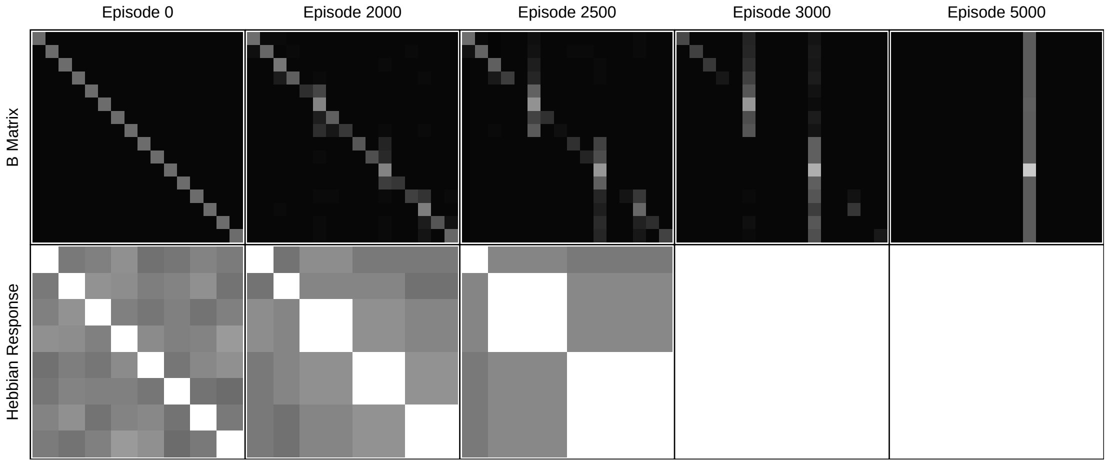


Figure 4.16: Using a smaller problem (for easy of interpretability), we can see how the changing regulatory matrix changes the ‘Hebbian response’ between genes and modules. There are clear positive correlations between genes in the evolved phenotypes where they are able to coordinate within. These correlations correspond to where new (hierarchical) modules will evolve, and emerge where a hierarchical module is able to coordinate with its partner module in the next layer.

where they are positively correlated. This suggests that it may be possible to 'solve' these multi-layer problems without any need for hierarchy, as the inter-module signal can be detected by exploration of the phenotype space due to drift in the gene regulatory connections. This proposed mechanism is close to the same mechanism that makes the Smooth MC solvable: rather than experiencing 'noise' in the B matrix, it integrates the weak inter-module signal already experience by the initial gene expressions in G, but it is still necessary that the 'correct' regulatory connections more readily influence the basin sizes (as is the case due to the relative steepness of the gradients within the basins).

It may also be the case that Spiky MC problem is trivially solvable with binary G when the number of genes in the sub-modules is odd, because the benefit curve is sampled either side of the minimum rather than the minimum itself. This has not been explored, but would in effect allow the evolutionary process to compare the gradients either side of the minimum, and so increase the basin size of the fitter phenotypes.

4.7.1 Amount of Drift and Two Time-scales

Varying the amount of drift each episode with the parameter Q shows how sufficient variation between episodes is necessary for the evolutionary process to evolve suitable modules, which the cost of connections motivates to become independent and hierarchical. We can relate Q from this Chapter with the frequency of dual-peaked environmental instances Z from Chapter 3: larger values of Q increase variation in the long term by shuffling G to a greater or lesser extent, while smaller values of Z increase variation in the long term by trapping the evolutionary system less often (being trapped means that the phenotype must be consistent with the previous episode).

Previous work has characterised the important relations between G and B as being along the lines of "G evolves quickly, and B evolves slowly" (Kounios et al., 2016). Just as important is the matter that G is restricted to $[-1, +1]$, while B is unconstrained. It is necessary that the partial resets in G produces a distribution around zero, but that resetting B does not. In all experiments the values are explicitly constrained to $[-1, +1]$, such that random mutations scramble (to a greater or less extended depending on Q) the vector of initial gene expressions. Random mutations on the B matrix of gene regulator interactions, however, introduce isotropic noise, and so, on average, don't do anything: they introduce no bias, and do not systematically undermine the products of past evolution so long as the number of mutations is small. This same distinction holds for the experiments in Chapter 3, as it is necessary that G approaches fixation during any episode: G must be bounded so that the selection from previous episodes can be readily 'undone' in a short time-frame. The opposite must be true of B: the system must integrate changes over a longer time-frame, so it must be practically unconstrained. Similarly, it is necessary that G is regularly scrambled (by the changing environment), but the amount of noise introduced into B during the transitionary periods (where G doesn't necessarily reflect the environmental conditions) must be small compared to the 'signal' from directed evolution when G does reflect the environmental conditions.

The 'speeds' with which G and B evolve are important: G must evolve quickly enough to produce phenotypes that reflect the environmental constraints and B must evolve slowly enough to not 'over-fit' immediately; however, these 'speeds' can be arbitrarily parametrised, and it is the *consequences* of the relative 'speeds' in the short term that determines the evolutionary outcomes: G must approach fixation in the short term (though it need not fix), and B must not over-fit in the short term. The short time-scale is determined by the episode duration K, and the other parameters can only be interpreted knowing K. The parameters Q and Z to some extent modulate this time-scale by changing how quickly G varies: it is necessary that it change frequently so that on the long time-scale over which B becomes more influential that it sees a somewhat balanced distribution of module assignments (however, we know from Kouvaris et al. (2017) that truly balanced sample is not necessary when a suitable cost of connections is employed). A further complication is that the Smooth MC problem can be solved by the evolutionary process without cost of connections if G does not evolve, so strictly the G timescale doesn't matter in such a scenario where the variability conferred by B is not necessary.

These additional details are worth exploring as they allow us to make better predictions about the evolutionary dynamics. For instance, in Chapter 3, we do not find a failure to evolve independent modules except for the extreme scenario of $Z = 1$: it is clear that we would see a failure to evolve independent modules for values of Z close to 1 but not equal to 1 (or similarly, increase K); however, scaling back M_B (the magnitude of mutations on regulatory interactions in B) or by changing squash constant h and cost coefficient λ in proportion would 'undo' the effect of reducing the amount of variation, though the evolutionary dynamics will be slower by virtue of the effectively reduced mutation rates. We shall reiterate and expand on this discussion in Section 5.6.2.

4.7.2 Limits to Rescaling

The evolution of hierarchy at any layer enables the evolutionary system to explore combinations of modules at that scale. However, as in the experiments with the changing environment, it is necessary that the hierarchy is sufficiently dominant that there is minimal loss of fitness due to inconsistency within the switching module: when the marginal benefit of switching is non-positive, the module will not align with the other modules in the layer above. Even in a total hierarchy (where the only gene with any influence is the lead gene), there is still an additive loss due to the initial expressions of the subordinate genes. The larger the module being flipped, the larger the loss due to the (counterproductive) additive contribution of subordinate genes as there are relatively more sub-ordinate genes. The effect is to make it less favourable to switch larger modules, as a larger proportion of the additive contribution isn't controlled by the lead gene. This effect is exacerbated by the nature of our hierarchical problem, in that the benefits awarded for coordinating larger modules are smaller than those of coordinating smaller modules. Consequently, the system is able to consistently integrate modules up to a certain size but becomes stuck as it is unable to coordinate larger, harder-to-switch modules that provide less benefit. These effects can be controlled independently: one is a property of the developmental process and is determined by the decay rate τ and number of developmental time-steps T; the other is a property of the environment and is determined by the decay factor d_f.

The regulatory decay rate τ and the number of developmental time-steps T together determine the additive contribution of each subordinate gene in a 'total' hierarchy (where all connections are from the lead gene). For each subordinate gene, the initial gene expressions of magnitude 1 decay to magnitude $(1-\tau)^T$ during development, but otherwise have no influence on the terminal gene expression. Because the developmental process is linear, this contribution is additive, and for $\tau = 0.2$ and $T = 10$, this contribution is $(1-\tau)^T/\tau \ = 2\%$. For the larger value of $T = 20$ it is about 0.2%, for $T = 30$ about 0.02%, etc.. This is one of two reasons why the system is able to achieve higher average fitnesses in the Spiky Continuous HIFF when T is larger: the additive loss of switching is smaller for larger T, so larger modules can be switched. This in turn means that the system can coordinate larger modules, and so satisfy more layers of the fitness landscape (as reflected by the small variance in Figure 4.15 for larger T). The other (more boring) reason is simply that larger T provides a slightly greater expression overall for a given cost, so high *benefit* phenotypes provide a greater fitness all else being equal (as reflected by the higher peaks in Figure 4.15 for larger T).

It should be noted that while the sign of the marginal benefit of switching a module is determined solely by the coordination with the modules 'partner' module, the magnitude of the benefit depends on the coordination between all modules as – unlike a non-continuous HIFF (Watson and Pollack, 1999) – the signal for upper-layers is always present. This is another complication in describing the conditions in which the evolutionary process will be able to coordinate modules, as modules switches effectively observe the magnitude of the benefit change of the single mutation (not just the sign of the benefit change of switching the whole modules): whether a particular module switch is beneficial or not depends on the genetic background. This includes the expression of genes in other modules (it is generally easier to switch early in an episode when some of the other genes in a module may not have flipped), and the interactions between them.

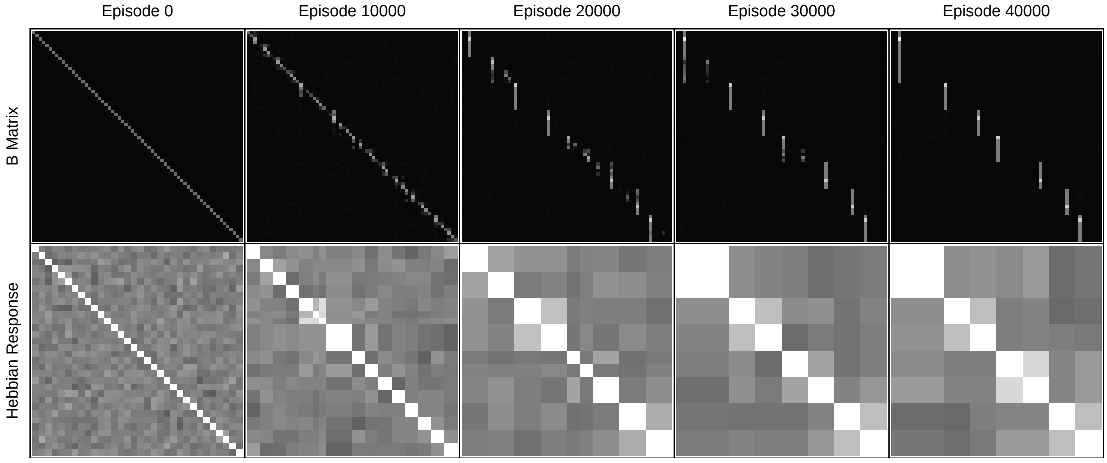


Figure 4.17: For $T = 8$ on a large problem with parameters per 4.4, the evolutionary process cannot reliably switch modules of size 8. Because it cannot always coordinate modules of size 8, it does not observe the modules of size 16 consistently, and so gets stuck without solving the problem. Note that while it is not reliable, the system is often able to coordinate the modules of size 8, as evidenced by the wrong positive inter-module Hebbian response and single module of size 16: it just isn't consistent enough to cause a new hierarchy to evolve in every case.

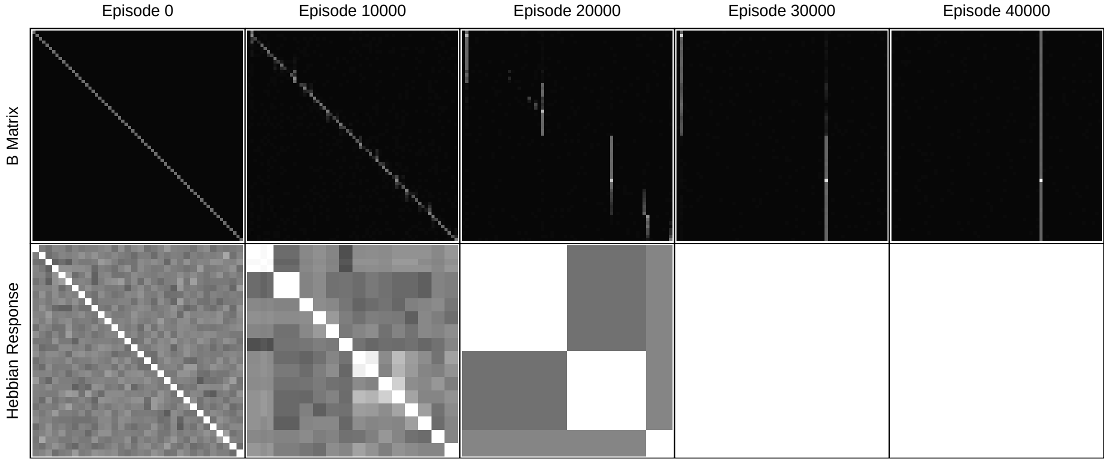


Figure 4.18: For $T = 14$ on a large problem with parameters per 4.4, the evolutionary process is able to coordinate modules of any size necessary to solve the problem, and so ultimately evolves a grand-hierarchy where the phenotype is effectively determined by the initial expression of a single gene.

4.8 Conclusions

In this chapter, we have extended the results from Kounios et al. (2016) to an explicitly modular problem (a generalised MC problem) using a slightly modified version of the model. Mostly significantly, there is no time separation between mutations in the G vector of initial constraints and the B matrix of regulatory coefficients. We characterised the conditions in the fitness landscape of the traditional MC problem that permit an evolutionary process to observe the inter-module correlations that reveal the global structure of the fitness landscape, and based on these conditions, we created a 'spiky' version of the MC problem. This spiky version if more consistent with the experiments from Chapter 3, and ensures that the basin width of each locally optimal phenotype is the same, regardless of fitness. Using our Spiky MC, we show that the evolution of hierarchical modules enables an evolutionary process to solve the MC problem by another means: the evolution of hierarchical modules can enable the evolutionary process to follow comparatively shallow inter-module benefit gradients to find the most fix phenotypes in the MC problem. By relating the amount of drift (Q) in this chapter to the frequency of dual-peaked module instances (Z) from the Chapter 3 we provide some additional insight as to the general principles of the developmental model under episodic selection that enables an evolutionary process to infer the modular structure of a problem.

The experiments in this chapter with the Spiky Continuous Hierarchical If-and-only-if reveal the capacity for the evolution of hierarchy to operate at multiple scales, and hence for a micro-evolutionary process to bootstrap its own evolvability: the evolution of hierarchy in sub-modules permits the coordination of these modules within larger modules, which themselves become hierarchical, enabling coordination in larger modules. This shows that the evolution of hierarchy is somewhat scale invariant, but we have discussed the limitations relating to the additive loss from subordinate genes when switching, and provided an example of how this limit can be parameterized in the developmental model.

The fact that a 'grand-hierarchy' can evolve in these experiments, and the observed limits to the bootstrapping process both provide additional evidence that the evolution of hierarchy may not necessarily be motivated by selection for evolvability, as the 'last layer' of modules becomes hierarchical despite it not providing an evolvability benefit. In the next chapter, we will discuss in some detail why the hierarchy evolves.

Chapter 5

The Efficiency of a Hierarchical Regulatory Topology

5.1 Introduction

In Chapters 3 and 4, we showed that the evolution of internally hierarchical modules has dramatic consequences for variability, and so evolvability. In this chapter we explore why the hierarchy evolves. In particular, we will show that hierarchy can emerge as a result of selection for efficient gene regulation, and that selection for evolvability is not necessary.

The question as to how the evolution of evolvability comes about is an interesting one because of the seemingly implausible notion that natural selection should favour beneficial future variability when it can only select from present variation: how is it to anticipate what variability will be beneficial in the future if the fitness consequences are not visible presently? In this chapter, we will specifically address whether evolvability evolves in response to selection for evolvability itself (Riedl, 1977; Payne and Wagner, 2019; West-Eberhard, 2019; Wagner and Draghi, 2010; Pigliucci, 2008) in our model.

It is also unclear to what extent a cost of connections is necessary or sufficient to promote modularity or hierarchy: Mengistu et al. (2016) acknowledge that selection for evolvability could be necessary to fix hierarchy in their model (indeed, they document that a population with an explicit diversity maintenance mechanism was needed to evolve suitable hierarchies) but suggest that the cost of connections might provide a necessary initial impetus. Many other models showing the evolution of evolvability also employ a population of genetically distinct individuals, often with some sort of diversity maintenance mechanism (Kashtan et al., 2005; Parter et al., 2008; Clune et al., 2013; Mengistu et al., 2016; Rünneburger and Le Rouzic, 2016; Crombach and Hogeweg, 2008). Consequently, it can be difficult to make strong statements concerning the causes of observed evolvability, because lineage selection remains a possible explanation. Virgo et al. (2017) noted that Kounios et al. (2016) use a population of one individual to evolve robustness, which shows that lineage selection certainly isn't always necessary for the evolution of evolvability.

What is apparent is that selection for evolvability is not necessary for the evolution of hierarchy in our models, because hierarchy will emerge in circumstances where the evolvability benefits described in Chapter 3 are not exploitable. Indeed, a grand hierarchy (where one gene directs all others) will evolve when the initial gene expressions fixed (removing any opportunity to switch modules) in an unchanging environment (removing any potential benefit of switching modules, see Appendix C.4). Furthermore, where hierarchy is exploitable, there is a systematic evolution toward a hierarchical module topology from the outset, when the hierarchy is too weak to enable single-point mutation module flips: evolution toward a hierarchy precedes the changes in variability it confers.

We can further largely reject lineage selection as a contributing factor, as our experiments deal with single-point mutations and populations of just one individual (as per Virgo et al. (2017) commenting on Kounios et al. (2016)). These properties limit the variation in the population significantly, making it impossible to select for a lineage at all, as only one lineage exists. It must be cautioned, however, that lineage selection is not the only means by which evolvability enhancing mutations could be directly selected, and it would to a certain extent be futile to try to address all possible mechanisms. As such, we devote the rest of this chapter to explaining how hierarchy can evolve as a consequence of its 'efficiency' (the capacity to produce fit phenotypes given a cost of connections) rather than its evolvability benefits. We will prove that hierarchy with 'dominance' (discussed below) is the most efficient regulatory topology under reasonable assumptions and argue that we can expect a fitness gradient toward such hierarchy in some relevant cases. We will then proceed to explain how the different experimental parameters can influence the rate at which hierarchy evolves by characterising their influence on the 'saturation time' of the evolutionary system.

5.2 The Evolution of a Hierarchy with a dominant gene

In this chapter we will assume entirely a narrow definition of hierarchy which reflects the structures that we observed evolving in the previous chapters. While there are many ways to construct a topology such that it is hierarchical (i.e. a sub-set of genes direct the developmental expression of other genes), we consistently see the evolution of 'columns' in our B matrices of regulatory connection weights - one gene directing all others - when evolved with a linear (L1) cost of connections. While much of the analysis in this chapter ignores the possibility of other hierarchical topologies (e.g. a chain is explicitly excluded by the 'two-parameter' characterisation to follow) we will find that the 'column' (or 'star') topology (which we will just refer to as a 'hierarchy') is the most efficient, and we will make the case that this is precisely why it evolves rather than any other topology.

It is also necessary to define the terms around dominance. 'Developmental dominance' is the feature that one gene in a module will tend to be more strongly expressed than the others during development. This developmental dominance may be a consequence of dominance in the B matrix of regulatory interactions or dominance in the G vector of initial gene expressions. Dominance in B is most apparent in the hierarchies that evolve in the experiments in the previous chapters, as the 'self-connection' on the switch gene is stronger than all the other connections: the switch gene up-regulates itself more than it up-regulates the other genes, and consequently has the greatest expression levels throughout development. Dominance in B comes about well before the evolution of any apparent hierarchy due to the stochastic nature of our simulations: any disparity in the connections within a module can produce developmental dominance.

Dominance in G is less apparent in our experiments with hierarchy because we use a clamped G vector. Rather than G providing slightly different initially expression levels, G determines the sign of the initial gene expression within a module, and it is expected that the evolutionary process will cause these to align after the perturbation at the start of each episode (either due to a change in environmental conditions, as in Chapter 3, or a period of drift, as in Chapter 4). The transitional period during which the G vector is aligned to the new environmental conditions, however, is of great interest, and provides 'transient dominance'.

While the transient dominance supplied by G is ephemeral, the evolution of hierarchy (in the B matrix) and dominance (in the B matrix) both persist over long time-scales, and both have long-term evolutionary consequences as a result: this 'regulatory dominance' will be the focus of this chapter; transient dominance in G will be discussed in depth in Chapter 6.

In our experiments with the linear cost of connections, the hierarchies that evolve can be characterised by the strength of the self-connection on the lead gene s, and the connection from the lead gene to its subordinates r. These hierarchies have dominance because it is always the case that $s \geq r$ where they evolve. This two-parameter characterisation provides an immediate clue as to the reasons for the evolution of dominance and hierarchy: given a hierarchical topology (which would provide an evolvability benefit in the absence of the dominance), there must be some direct fitness consequence as to the choice of s and r. Indeed, if we assume that s and r evolve to maximise fitness, then we are able to predict the terminal connection strengths by finding the assignment of s and r that maximises fitness numerically.

Figure 5.1 shows how the optimal assignment (that which produces the greatest attainable fitness) of s and r in a 4-gene module changes for different total weights. Figure 5.2

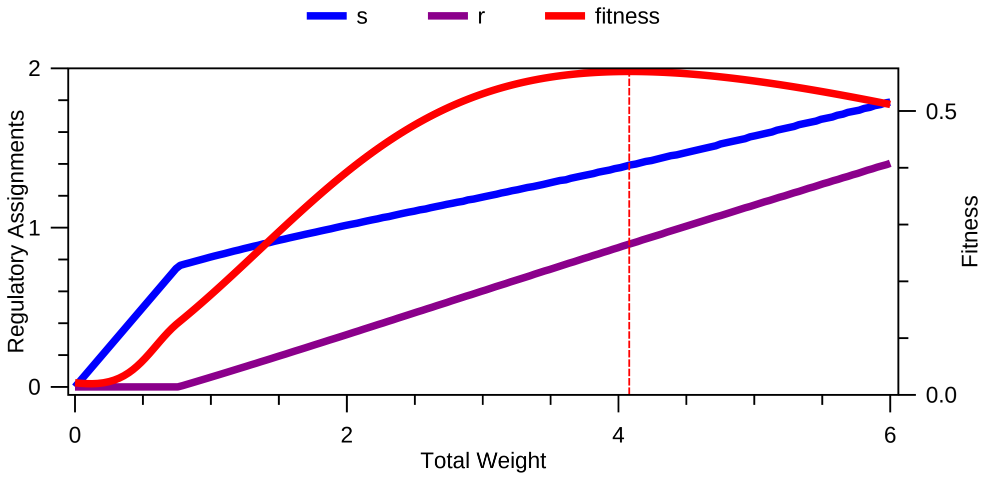


FIGURE 5.1: Optimal assignments of s and r for varying total weight along with the fitness they produce in a module of 4 genes with a benefit coefficient of 1, cost coefficient of $\lambda = 1$, and G vector of all $+1$. The fitness is highest with a total weight a little more than 4, with around 30% of the weight on s (25% would correspond to the no-dominance scenario of $s = r$). For any positive choice of total weight, the optimal assignment always has $s > r$. Though imperceptible in this figure, there is a dip in the fitness curve when the total weight is small. This is affectionately called the 'pit of despair', which precludes the evolution of anything in B for large values of cost coefficient λ (the larger λ, the wider the pit of despair) as the benefit curve is very shallow near zero. For small choices of λ, the pit will disappear completely; for some assignments, it is present but narrow, representing a hurdle that must be jumped before the evolutionary process can proceed.

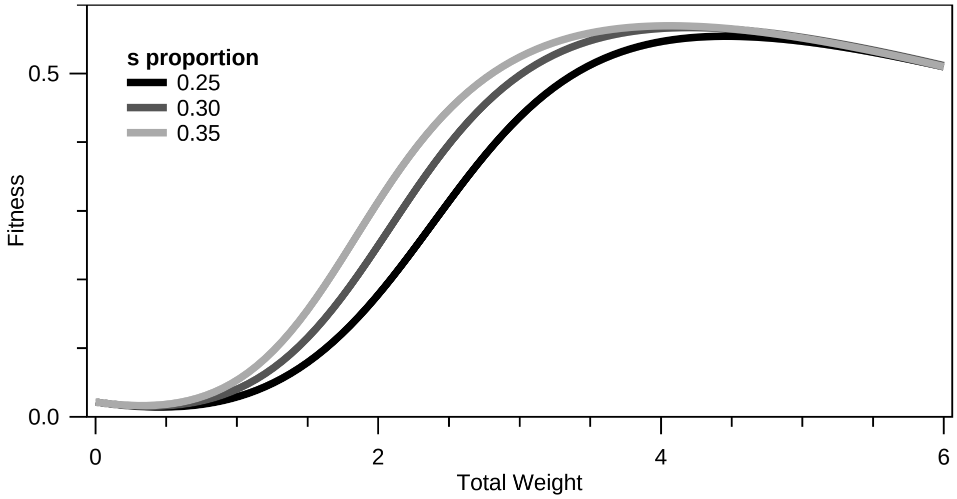


FIGURE 5.2: The fitness curve has a characteristic 'S' shape which for large values of λ initially has a negative gradient (the 'pit of despair') due to the initially shallow benefit curve, and has a maximum where the marginal benefit falls below the marginal cost as the benefit curve saturates due to the squash function σ. Shown are curves for hierarchical modules, varying the proportion of the total weight that is assigned to the self-connection s (same configuration as Figure 5.1). Because there are four genes in the module, the s proportion of 0.25 corresponds to a hierarchy without dominance ($s = r$).

shows the fitness curves for 3 ratios between s and r with the same total regulatory connection weight. Each curve is the sum of a monotonically increasing S-shaped ‘benefit’ curve, and the linearly decreasing fitness loss due to the linear cost of connections when the initial expression of all genes is $+1$, with a linear module-benefit function $\eta(x) = x$. It is clear that - in this configuration - the maximum fitness is attained for a configuration where $s > r$. Furthermore, we can infer that the maximum *benefit* is attained for the same assignment, because the cost is proportional to the total weight and so independent of the ratio between s and r.

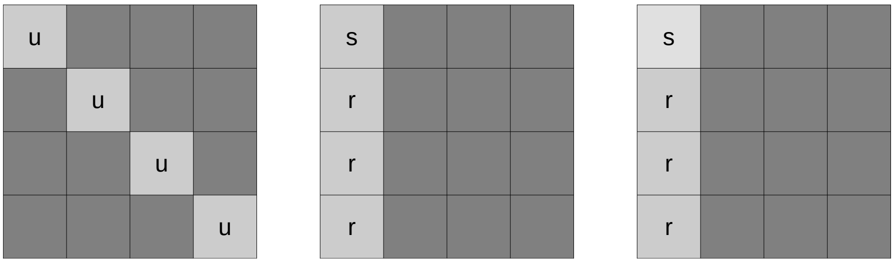


FIGURE 5.3: When the initial expression of all genes is the same, then the ‘identity’ with connections of strength u produces the same phenotype (and so fitness) as a hierarchy with $s = r = u$. A hierarchical topology can achieve a higher fitness with an assignment where $s > r$.

It is noteworthy that with the high cost coefficient of $\lambda = 1$, it is initially the case that adding weak connections *decreases* the fitness due to the shallow benefit curve (the ‘pit of despair’, apparent in Figure 5.2). In addition, because the benefit curve grows steeper initially, the optimal assignment early on is to place all the available weight on a single self-connection. This might lead one to assume that a hierarchy would emerge from the outset, but in practise this is not what happens, because the marginal benefit may still be positive for all connections in the module (see Figure 5.2). As we shall see in a later section, evolution only finds the optimal assignments for s and r once it has ‘saturated’ the whole module, such that indiscriminately increasing the weight on connections within a module no longer provides any benefit.

We can appreciate why s might evolve to exceed r if we consider the ‘identity’ topology where every gene has a self-connection of strength u, and no other connections are present (Figure 5.3A): in such a topology, there will be an optimal assignment for u where the marginal benefit of increasing gene expression is equal to the marginal cost of increasing the connection strength (the marginal fitness is zero). If we assume a linear module-benefit-function, then each gene is effectively independent (there is no pleiotropy or epistasis). Given this assumption, the optimal assignment of the self-connection will be the same for all genes: during development, the expression of each gene follows the same trajectory, each gene contributes the same benefit, and each connection has the same cost.

If we select one gene to be the leader, and instead arrange the same weight as a non-dominant hierarchical topology (where $s = r = u$, Figure 5.3B), we will find the exact same fitness: genes that previously up-regulated themselves are now up-regulated by another gene that follows the same developmental trajectory as itself. Keeping r the same, the marginal benefit for increasing the self-connection strength s will be greater than it was with the identity topology, as the increase in expression during development will not only increase the terminal expression of the lead gene, but will also increase the expression of the subordinate genes. Because the marginal cost of increasing s is the same as for r, it is clear that the hierarchical topology with regulatory dominance (Figure 5.3C) is more efficient than the identity topology under a linear cost of connections as it is able to achieve a greater benefit for the same cost.

5.3 Proof of the Optimality of Hierarchy under Assumptions

In this section, we prove - under reasonable assumptions - the intuition from the previous section: that a hierarchical topology with a dominant lead gene is the topology that maximises fitness of independent modules (assuming the initial gene expressions are well-matched to the environment).

We consider here a generalisation of the model, replacing the cost function with Equation 5.1. The function $\phi(|x|)$ controls how the cost of each connection varies with its strength (the magnitude of the connection coefficient). In the previous chapters, a linear cost of connections was used, which is the same as setting $\phi(|x|) = |x|$.

$$c(B) = \frac{1}{N^2} \mathop{}_{i,j}^{N,N} \phi(|B_{ij}|) \tag{5.1}$$

5.3.1 Assumptions

The assumptions made in this proof are as follows:

- Modules are independent, and the module of interest has only positive intra-module regulatory connections. Assuming this – and that the developmental functions are all symmetric around zero – we can ignore gene expressions less than zero.
- The developmental process must be recurrent: $T > 1$, where T is the number of developmental steps performed during development of the initial gene expression levels to the phenotypic expression levels.

- $\vec{y}(0) = G$ must be uniform and match the environment (all initial gene expressions of the same magnitude, and the appropriate sign to match the environmental conditions).
- The update term must pass through the origin, be (odd) symmetric, and grow monotonically with the magnitude of the weighted sum: $\sigma(0) = 0$, $\sigma'(x) > 0$ where σ' is the marginal of the squash function σ at x. This simply means that stronger gene expressions cannot result in lesser effects.
- The benefit term must grow monotonically with the expression of each gene: $\eta'_m(x) > 0$, where η_m is the module-benefit function, and its argument x is the mean expression level in the phenotype. This implies directional selection such that any increase in the expression of a gene 'in the right direction' is always beneficial, no matter how slight. Note that this holds in our piecewise linear environments when the population is at a local optimum, which is how the system spends most of its time for the values of K which we use.
- The cost term (neglecting any constant contribution) must grow monotonically and sub-additively with the magnitude of individual connection weights: $\phi(|a| + |b|) \leq \phi(|a|) + \phi(|b|)$. This means that it no more costly to lump weights together onto a single connection than to spread them out among many. The linear cost of connections, $\phi(|x|) = |x|$, is a special case of this, which evokes a resource allocation game: the cost of each unit of weight is constant, regardless of where it is assigned.
- The decay term must be positive and less than one: $0 < \tau < 1$. This is a fundamental assumption in the model, that when unregulated, gene expression levels tend toward zero.
- The cost coefficient must be non-negative: $\lambda \geq 0$. There must be some cost for connections for our result to have any meaning.

The final part of Part 3 requires either of the following stronger assumptions:

- The module-benefit-function η must be linear.
- The squash function has a non-positive second differential ($\sigma'' \leq 0$) locally *and* the cost function ϕ must be linear.

5.3.2 Proof

The proof proceeds in three parts, describing a process for modifying the assignment of a fixed total weight to the connections within a module without ever reducing the contribution of the module to the phenotype's fitness, each part performing a transformation which follows from the previous.

Part 1 shows that a dominant gene (which always has the greatest level of expression) can be selected, and that there is no fitness advantage to having any gene regulate it but itself. Part 2 shows that there is no fitness advantage to any gene other than the dominant gene regulating any gene, indicating that an optimal fitness can be achieved with a hierarchy. Part 3 shows that an optimal hierarchy can be described by only two parameters (as in Figure 5.1) if slightly stronger assumptions are made.

Part 1 Given a matrix describing regulatory connections within a single isolated module, we start by making all regulatory coefficients positive without changing any of their magnitudes to produce a matrix $\bar{B}$ that produces only positive inter-gene effects. Switching the sign of a connection produces no change in cost, and a system where all connections are coordinated will always outperform one with any contrary connections given the assumption of an odd (symmetric) squash function: $\sigma(x) = -\sigma(-x)$.

Next, we identify the gene in this module which has the greatest total in-coming connection weight and call it s (the index of the gene, not the strength of the connection).

$$s = \arg\max_{i} \quad {}^{N}_{j} \; \bar{B}_{ij}$$

Because all genes start with the same initial expression, and σ is assumed to be monotonic, this gene will have the greatest level of expression at time $t = 1$ (after one developmental step) given regulatory matrix $\bar{B}$, represented by $\bar{y}_s(1)$.

$$\bar{y}_s(1) \geq \bar{y}_j(1)$$

We then move all the incoming connection weight onto the self-connection B_{ss} to produce a new regulatory matrix $\tilde{B}$:

$$\tilde{B}_{ij} = \begin{cases} i = s, j = s & {}^{N}_{j'} \bar{B}_{ij'} \\ i = s, j \neq s & 0 \\ else & \bar{B}_{ij} \end{cases}$$

This effects no change in gene expression at time $t = 1$, as the product within the squashed part of the update equation is unchanged. However, we assume a recurrent system such that $T \geq 2$, and a necessary consequence of reinforcing the self-connection on gene s is that it has the greatest level of expression at any point in developmental time, as it is motivated by the most strongly expressed gene in the network (itself, s) via the strongest connection in the network (its self-connection, B_{ss}). This means that the expression of gene s may be greater for time $t > 1$ than any other gene.

$$\tilde{y}_s(t) \geq \tilde{y}_j(t)$$

Crucially, the expression of any gene i can only be greater at time t with $\tilde{B}$ than with $\bar{B}$, because none of its influencers will experience a reduction in expression: they can only experience an increase in expression due to a greater expression of gene (s).

$$\tilde{y}_i(t) \geq \bar{y}_i(t)$$

This change cannot increase the cost, as the cost function is assumed to be sub-additive, and we have concentrated all of the cost of regulating gene s on a single connection. Consequently, the transformation from $\bar{B}$ to $\tilde{B}$ can only improve the fitness, as it may improve the benefit (a monotonic function of total gene expression) and cannot increase the cost. We can therefore say that $\tilde{B}$ is no less efficient than $\bar{B}$:

$$\tilde{B} \succeq \bar{B}$$

Part 2 Next, we generate another regulatory matrix, $\hat{B}$, by moving all the weight for in-coming connections for every gene i so that it is solely motivated by gene s, which is a degenerate hierarchical module configuration (one gene directing itself and all others within the module).

$$\hat{B}_{ij} = \begin{cases} j = s & {}^{N}_{j}\,\tilde{B}_{ij} \\ else & 0 \end{cases}$$

Again, the sub-additive cost of connections means the total cost cannot be increased by this operation. The expression of gene s is unchanged, as it remains unaffected by any other gene, and we have not changed the strength of the self-connection B_{ss}. What has changed, is that now every gene is always motivated by the most strongly expressed gene. This may increase the expression of each gene at time $t \geq 2$ and cannot decrease it.

$$\hat{y}_i(t) \geq \tilde{y}_i(t)$$

Duly, the transformation from $\tilde{B}$ to $\hat{B}$ can only increase fitness.

$$\hat{B} \succeq \tilde{B} \succeq \bar{B}$$

Observe that the total incoming weight for each gene is unchanged in these transformations. A consequence of this is that given any regulatory matrix $\bar{B}$ we can generate a hierarchical regulatory matrix $\hat{B}$ with the same per-gene weight which is no worse. Consequently, if there is an optimal configuration, then hierarchy must be an optimal module configuration, as an optimal module configuration will have some particular distribution of weight per-gene, and a hierarchical assignment of these weights (with the most strongly regulated gene leading) achieves the highest possible fitness.

Part 3 Parts 1 & 2 determine that a dominant hierarchy (the strongest gene driving all genes exclusively) is optimal, thought does not prove that it is the only optimal topology: it may be that the transformations described have no effect on fitness. This final part shows that the topology can be described with just two parameters by two routes.

If we assume that the module-benefit-function is linear, then the subordinate genes are completely independent (no pleiotropy or epistasis) because they have no influence on any other gene or its benefit contribution: the initial gene expressions simply decay exponentially, providing an additive contribution to their own terminal expression (as discussed in Section 4.7.2). Because they are independent, the choice of connection strength that optimises the fitness contribution of one subordinate can be used for them all. As such, we can describe an optimal topology with just the strength of the self-connection B_{ss}, and the strength of all other (non-zero strength) connections B_{is} $(i = s)$.

If we wish to avoid assuming a linear module-benefit-function, then we can instead assume that the squash function has a non-positive second differential $\sigma''(x) \leq 0$ and that the cost function is linear $\phi(|x|) = |x|$. In this case, given a set of choices for the connection strengths between the lead gene and the subordinates, we can 'average them out' such the same total weight is preserved, but all subordinate connections have the same strength. This doesn't change the cost (because the cost function is linear) but may increase the benefit. The phenotypic expression of subordinate gene $i = s$ is given by the expression

$$y_i(T) = (1-\tau)^T + \sum_{p=1}^{T} (1-\tau)^{(T-p)} \sigma(B_{is} y_s(p)) \tag{5.2}$$

This is a linear combination of update terms, and because we assume $\sigma''(x) \leq 0$ locally (it can't be the case globally - unless it is linear - due to the assumption of odd symmetry), the whole expression must also have a non-positive second differential with respect to the regulatory connection B_{is}. Given any function f for which $f''(x) \leq 0$ and set of numbers $X = x_1, x_2, ..., x_n$ with mean $\bar{X} = (\sum x_i)/n$, Jensen's inequality says that

$$nf(\bar{X}) \geq \mathop{}_{i=1}^{n} f(x_i) \tag{5.3}$$

We can substitute f for the expression for $y_i(T)$ with respect to B_{is}, as it has a non-positive second differential, being the sum of terms with non-positive second differential, in which case the summation on the right-hand side of Equation 5.3 is proportional to the mean phenotypic gene expression within the module, which is in turn the input to the benefit function.

Consequently, setting B_{is} equal to the mean of subordinate connection strengths can only increase the mean expression within the module (and so benefit, because the benefit function is monotone), and does so without changing the total weight (and so cost, since we here assume the cost function is linear). As such, this pair of assumptions also lead to the result that we can produce an optimal hierarchical topology described by only two numbers: as before, these are the strength of the self-connection B_{ss}, and the strength of all other (non-zero strength) connections B_{is} $(i = s)$.

Finally, note that there can be no fitness advantage to a hierarchical module if all the connections are the same weight (it is equivalent to any topology where the sum of connections onto each gene is the same). As such, *if* there is an optimal topology (i.e. one that maximises fitness under our assumptions) then it must be a hierarchy with regulatory dominance $(B_{ss} > B_{is} \forall i = s)$.

5.3.3 Discussion

Most of the assumptions in this proof come directly from the model detailed in Chapter 3, though with mild relaxations. Here we will briefly discuss these assumptions and their implications.

The assumption of a recurrent developmental process $(T > 1)$ is reasonable (development in living organisms takes a long time). It should be noted that when a gene has no influence on any other gene, that its initial expression still makes an additive contribution to its terminal expression proportional to $(1-\tau)^T$ as discussed in Section 4.7.2: increasing T reduces this contribution, and so 'tightens' the grouping of phenotypes generated by a hierarchical developmental process (effectively an increase in robustness) allowing larger 'jumps' from one phenotype to its complement.

The assumption of a sub-additive cost of connections (of which the linear (L1) is an extreme example) corresponds to circumstances where it is no more expensive to increase the strength of strong connections than it is to increase the strength of weaker connections. In practise, hierarchy may still be optimal for a slightly super-additive cost function, or if there is strong positive epistasis in the benefit function. Figure 5.4 shows some examples of regulatory topologies that can evolve when a strictly sub-additive cost of connections is used, indicating that while hierarchy may be the most efficient regulatory topology, this does not imply that it will always evolve, though it is the case that a sparse topology (where each gene is influenced by only one gene) will evolve consistently).

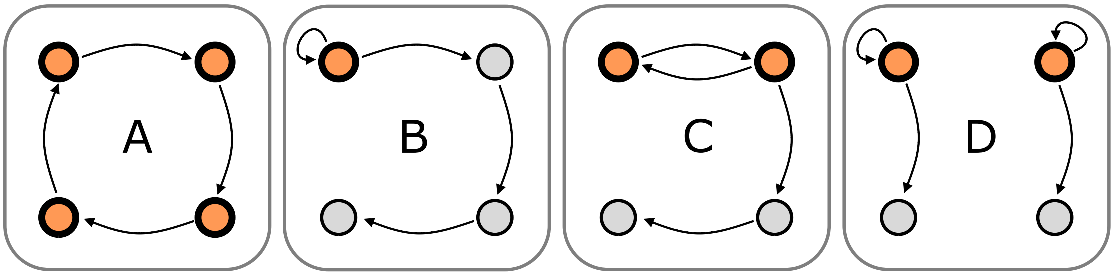


Figure 5.4: Examples of regulatory topologies evolved with a strictly sub-additive cost of connections (see Appendix C.3). While a 'column' hierarchy is the most efficient topology for any sub-additive cost of connections, a strictly sub-additive cost of connections produces many locally optimal topologies that are not hierarchical. The orange nodes with thick outlines are members of loops which (through positive feedback) can retain state (i.e. maintain the pattern of expression of its member genes) during development: if more than one node is able to retain state during development, then it is unlikely that the module can be switched by a single-point mutation.

Of note, is that parts 1 and 2 (and 3 with additional assumptions) holds for any benefit function which is locally monotonic. Apart from the piece-wise linear and quadratic module-benefit functions employed in our experiments, this would hold e.g. for a root-mean-squared benefit function for a broad class of B matrices. This implies that an optimal topology is hierarchy for any such benefit function, but it may be that it is not a two-parameter (e.g. when the per-connection cost function ϕ is non-linear). This possibility has not been tested, but suggests that the conditions under which hierarchy could emerge are quite general. An assumption of monotonicity is none-the-less an assumption of selection toward an extreme (i.e. directional selection). The tendency for experiments to assume directional selection (without considering selection for non-extreme phenotypes) is a concern raised by Rünneburger and Le Rouzic (2016), and indeed hierarchy may not evolve if the benefit function is not monotonic. Note that while the benefit function must be monotonic, this does not mean that the fitness function must be monotonic: indeed, as we shall see, it is necessary for the fitness function to be non-monotonic (as in Figure 5.2) to evolve hierarchy in our experiments.

Finally, note that a hierarchical topology is not sufficient to produce a high fitness phenotype: it must also have a doinant lead gene. Fortunately, the more hierarchical a topology, the more benefit may be gained by increasing the extent of the dominance (the more influence one gene has, the greater the return for increasing its own expression; and the more dominance in the connections, the greater the fitness can be obtained by increasing the influence of the dominant gene. The proof relies on both these properties in a ‘lump-transfer’ way, but if they hold incrementally as well, then this would provide a good indication that direct selection for high fitness can promote the evolution of hierarchy. If it is the case that dominance promotes selection for hierarchy and hierarchy promotes selection for dominance, there is the potential for positive feedback, and we will explore this further in Chapter 6.

5.4 Fitness Gradients toward Hierarchy

The previous section showed that hierarchy is the most efficient topology for a sub-additive cost of connections, but it does not indicate the hierarchy should necessarily evolve, or even that it can. In general, we cannot prove that there is always a gradient to a ‘column’ hierarchy, but if there exists a gene s, such that $y_s(t) \geq y_j(t) \forall j$, and all connection weights have the ideal sign (e.g. all positive for an all positive target and genotype), then there may be a gradient towards a hierarchy with dominant gene s if, in addition to the assumptions in Section 5.3, we assume an additive cost of connections $\phi(|a| + |b|) = \phi(|a|) + \phi(|b|)$, such that all regulatory networks B with the same total weight (sum of the magnitude of connection strengths) have the same cost.

We can express the developmental dynamics as an update term, showing the change in gene expression for gene i for each of the T developmental steps, as in Equation 5.4.

$$\Delta y_i = -\tau y_i + \sigma \left(\sum_j^N B_{ij} y_j \right) \tag{5.4}$$

We consider a single isolated module with always-positive gene expressions. The final fitness contribution of a module is a monotonically increasing function of the sum of $y_i(T)$ terms, so we ignore the module-benefit function and only consider the total gene expression. For any gene i, the differential of its update term with respect to the within-module connections that motivate it is given by Equation 5.5.

$$\frac{\partial \Delta y_i}{\partial B_{ij}} = \sigma' \left(\sum_j^N B_{ij} y_j \right) y_j = Q_i y_j \tag{5.5}$$

Q_i is then defined for gene i (common for all B_{ij}), and because σ is monotonically increasing ($\sigma' > 0$) we know Q_i is non-negative. Assuming that the target phenotype is one where all genes have a positive gene expression, and that $y_j > 0$, it is apparent then that the partial differential for Δy_i at each step is greatest with respect to the gene which, at that step, has the highest expression. Should one gene always have the highest level of expression, then there is always an advantage to moving weight from connections that are not outgoing from this gene as they provide a shallower expression-

(and so benefit-) gradient than those that are outgoing from this dominant gene. This essentially motivates incrementally moving weight from any row of the regulatory matrix onto one column, which will ultimately lead to a degenerate (single-layer) hierarchy, when directional selection provides a pressure to increase the magnitude of every gene when the phenotype matches the environment well. We do not present this as a proof, because changes to the expressions of genes early on in develop may violate our assumptions of developmental time dominance; however, this argument does suggest that if there is a gene which is significantly more strongly expressed than all others throughout developmental time, then this indicates that there is a greater marginal benefit to be gained from increasing its influence over any other, and that a micro-evolutionary process could in theory evolve a hierarchy.

The assumption of developmental time dominance is not one that always holds; though it is generally the case in our experiments, it is unclear whether this is just down to stochasticity in the simulation, or is a consequence of selection for dominance. Figure 5.5 shows how the number of modules with a dominant gene throughout development was usually four during an experiment from Chapter 3 with $Z = 0.5$ (where there were four modules in total), showing that this can be the case, and indicating that as the evolution progresses, the probability of any module having sufficient regulatory dominance to induce developmental-time dominance may grow.

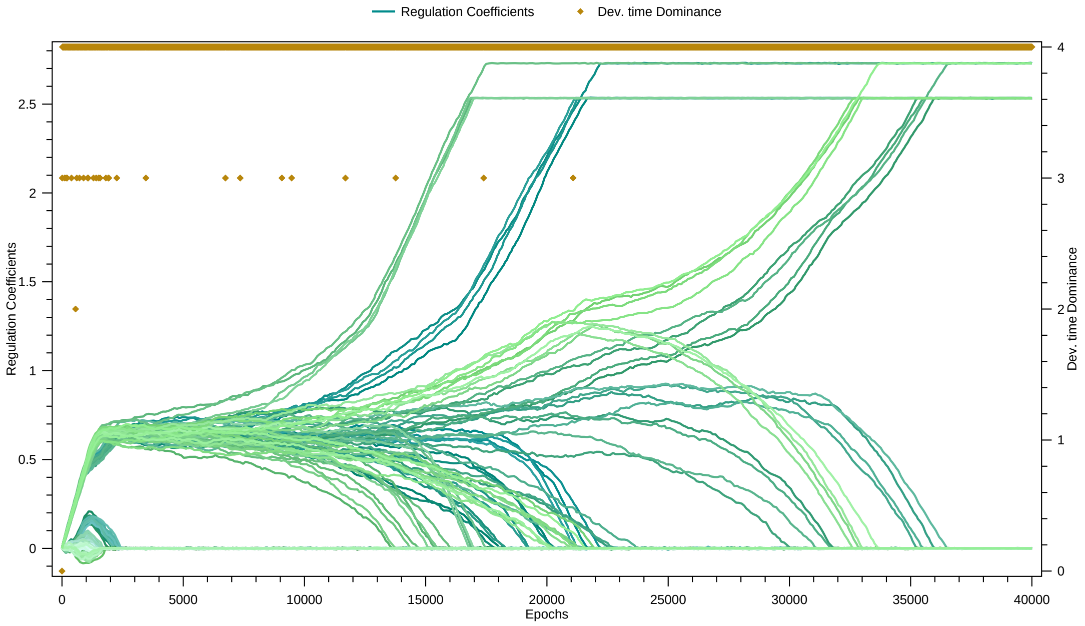


FIGURE 5.5: The number of modules with dominance, alongside regulatory coefficient trajectories, for an example simulation from Chapter 3 with $Z = 0.5$. One sample per 10 episodes.

5.5 Saturation

We have proved that an optimal module topology would be hierarchical (Section 5.3) and argued that in some cases a gradient exists that leads toward such a topology (Section 5.4); however, this does not explain why it evolves in our simulations. Furthermore, while the assumptions made allow us to make predictions about the circumstances in which hierarchy could evolve, they may not represent necessary conditions, and they say nothing about how long it should take to evolve.

It is possible for an explicit gradient descent mechanism to 'evolve' hierarchy (Appendix D), but this doesn't imply is must evolve in a dynamic environment with a stochastic evolutionary process. Note that - when talking about the evolution of hierarchy - we always presume the existence of a strongly connected module. This is the case in our experiments, where either the modules initially emerge as dense (fully connected) modules (Chapter 3), are initialised as the identity (all self-connections, Chapter 4), or evolve from two or more independent modules (Chapter 4). This is a subtle but important point: a module can only observe the fitness advantages of hierarchy in B when improving its efficiency is the only means by which to increase its fitness: a dense but weakly connected module can increase its gene expression by indiscriminately increasing the strength of its connections; a strongly connected module is confronted by the cost/benefit trade-off shown in Figure 5.2.

In this section we will explore the idea that there is only a tendency for hierarchy to evolve in a strongly connected module: one that approaches 'saturation' of its benefit function, such that the marginal benefit of indiscriminately increasing the strength of the connections within the module is equal to the marginal cost of doing so. Using a very simple environment (with some of the properties of that used in Chapter 3) we will observe steady evolution toward hierarchy within episodes and realise that the amount of time a module spends saturated largely determines the rate at which hierarchy evolves. We will use this insight to describe the 'saturation time' (how much time a module spends saturated) and explain how various parameters influence this value and consequently the rate at which hierarchy evolves.

5.5.1 Saturation Points

The 'saturation point' refers to the state where, for a given module and its internal interactions, indiscriminately increasing or decreasing the total connection strength does not provide an increase in fitness. The saturation point is important for explaining the evolution of modularity and hierarchy under single-point mutations, as there can be no comparisons made between mutations. If we allowed multiple mutations in B simultaneously (as is the case in Kouvaris et al. (2017)), then it would be possible for more beneficial mutations to accumulate more rapidly than less beneficial mutations even when not saturated. With single point mutations, this is not the case: each mutation is judged only on the sign of its induced fitness difference (does it help (positive) or hinder (negative)?). Given that we observe hierarchy evolving from dense modules, this means that there must occur situations where increasing the influence of a leading gene is beneficial, while decreasing the influence of a subordinate gene is also beneficial.

We can describe a sub-set of connections C in B as at saturation when the marginal fitness ∂f of increasing the weight of all members is zero.

$$\underset{\langle i,j\rangle \in C}{} \frac{\partial f}{\partial B_{ij}} = 0 \tag{5.6}$$

Assuming a linear cost of connections, this is equivalent to saying that the marginal benefit of increasing the weight of all members by the same amount is equal to the cost coefficient λ.

$$\underset{\langle i,j\rangle \in C}{} \frac{\partial b}{\partial B_{ij}} = \lambda \tag{5.7}$$

Of course, this doesn't mean that the marginal benefit of the individual components need be the same: in the extreme case of a hierarchy, we have already determined that there is a fitness benefit to 'over-expressing' the dominant gene, such that the subordinate genes can be regulated by a stronger gene. Importantly, this definition says nothing about the individual connections: if we assumed that every connection had a marginal fitness of zero, then we have presupposed an efficient regulatory network. Rather, this definition indicates a network where indiscriminately increasing (or decreasing) connection weights won't provide a fitness benefit: to improve the network we must change the assignment of weights in different proportions or directions. The discussion above indicates that this can be achieved by making the network more hierarchical (assuming some measure of dominance): if we assume that evolution is able to perform such reassignments, then we should not be surprised if a module tends toward hierarchy when nearly saturated (the discrete nature of mutations prevents us from considering any ideal circumstances). Just as importantly, we should expect that when a module is *not* saturated that it should tend toward being saturated: it will increase or decrease the strength of individual connections in B to try to make their individual marginal fitness zero, which has the effect of saturating the whole module. Note that it need only be that the sign of some connections within a module are different for evolution with strong-selection weak-mutation assumptions to rebalance the distribution of connection weights, and the non-differentiable nature of the L1 cost of connections at zero means that the concept of saturation breaks down for weak connections; however, the notion of the saturation point is a useful tool because it is easy to see how it depends on experimental parameters.

5.5.2 Changing Saturation Points

To test the assumptions that hierarchy will increase when a module is saturated, we will use a simple experimental set-up designed to mimic some of the properties of the changing-environment model from Chapter 3 while removing much of confusing stochasticity. We will use the same developmental model, but a slightly changed evolutionary model, and a different environmental setup. We fix G so that it matches the environment (as in Appendix C.4): this eliminates any transient dominance in G, allowing us to be confident we only keep the effects of regulatory dominance from B. We replace the complex environment with a simple one which has only two possible conditions: either it awards a high benefit proportional to the mean gene expression, or it awards a low benefit proportional to the mean gene expression. We will switch between these two environmental conditions consistently and regularly, changing the relative duration of exposure with a 'duty cycle' parameter $D \in [0, 1]$: the higher benefit environment will remain for $2KD$ evolutionary steps before being replaced by the lower benefit environment for $2K(1 - D)$ evolutionary steps.

Because we are changing the benefit awarded (while keeping the cost function the same), the saturation point of the module will change as well. If the amount of weight that needs to be moved to switch between saturation points (the 'saturation gap') is large then it will take the evolutionary process a significant amount of time to respond to the change. If it is the case that the degree of hierarchy will increase when a module is saturated, then we would expect the rate at which hierarchy evolves to depend on how long the evolutionary process takes to respond to the changing environment: the wider the saturation gap, the more mutations will be required each episode to traverse the saturation gap, and so the shorter the 'saturation time'. Assuming that the saturation time is the only significant factor in determining the rate at which hierarchy evolves (something which we know can't be true, but may none-the-less be a useful characterisation) then we would expect hierarchy to evolve most quickly when the duty-cycle is 0 or 1 (i.e. when we only present the high or low environmental conditions). How transitional periods of non-saturation will influence matters is not immediately obvious.

Figure 5.6 shows that the number of episodes it takes to evolve hierarchy changes dramatically as we vary the duty cycle: it evolves quickly near 0 and 1 as expected, but for a duty cycles of 50%, hierarchy did not readily evolve within the 10000 episodes of the simulation. Figure 5.7 shows how the rate at which hierarchy evolves is proportional to $|D - 0.5|$.

Figure 5.8 shows the within-episode behaviour for a duty-cycle of $D = 0.8$ (meaning the higher-benefit environmental conditions are observed 80% of the time). With a duty cycle greater than 0.5, the high benefit environmental conditions predominant, such that the system spends most of its time at the high saturation point. Owing to the wide saturation gap, the system does not have time to saturate in the lower-benefit environmental conditions. When exposed to the lower-benefit environmental conditions, the fitness plummets, and weight is removed indiscriminately to reduce the cost paid for the low marginal benefit. Though the module is not saturated, the degree of hierarchy measure continues to increase because removing weight from all connections at the same rate has the effect of increasing the proportion of weight controlled by the dominant gene (i.e. its relative influence). This effect is stronger when the degree of hierarchy is greater. It doesn't effect a long-term increase in hierarchy, but does introduce some noise. This is because once the evolutionary system returns to the higher-benefit environmental conditions, there is a period of time where all regulatory connections again increase together, resulting in an apparent reduction in the degree of hierarchy. This undoes the prior reduction in weight and introduces more noise. Once the module has saturated again, then selection for efficient regulation takes over, and the degree of hierarchy increases as weight is steadily shifted from the connections of less influential genes to the connections of more influential gene. Importantly, the system never reaches the low saturation point. On average, the time spent *not* saturated can be expected only to introduce noise: the apparent increase in hierarchy when weights are suppressed should be undone by the apparent decrease in hierarchy when weights are free to grow. The noise is a consequence of the indiscriminate nature in which mutations occur and fix when not saturated (the number of mutations on each regulatory connection will not be totally consistent) and may serve an important purpose for 'symmetry breaking': some measure of asymmetry is need in the developmental network to permit selection for hierarchy.

Parameter	**Symbol**	**Values**
Number of evolutionary steps per episode	K	1000
Episodes		10000
B matrix mutations probability	R_B	1.0
B matrix mutation magnitude	M_B	1×10^{-3}
B matrix mutation type		Uniform
High module benefit coefficient	c_H	1.0
Low module benefit coefficient	c_L	0.7
High-benefit Duty cycle	D	Variable
Cost Coefficient	λ	0.1

TABLE 5.1: Parameters for the duty-cycle experiments.

Figure 5.9 shows that for duty cycles near 50% there is little systematic tendency toward hierarchy because the saturation-gap is wide enough that during the K evolutionary steps available to re-adjust to the changed environment, the evolutionary process cannot traverse the saturation gap: the system performs a random walk in the saturation gap and rarely saturates, so there is limited selection for hierarchy. Consequently, these specific duty-cycles tests do not indicate much about the saturation gap: the saturation time in these experiments is rather a function of how much time the system spends 'desaturating' away from the high saturation point (if the duty cycle is high) or low

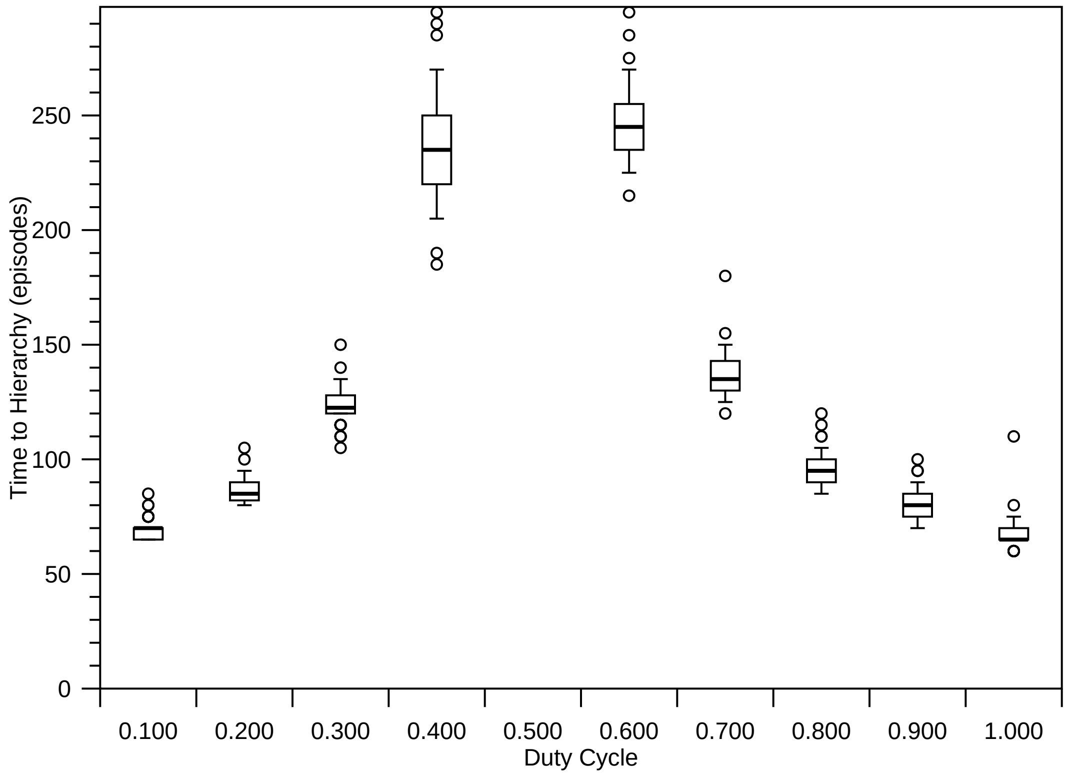


FIGURE 5.6: The number of episodes required to evolve a hierarchy ($d_h \geq 0.9$). For a duty cycle of 50%, hierarchy did not routinely evolve in the 10000 episodes allotted.

saturation point (if the duty cycle is low): the system will remain in the locality of whichever saturation point it most often experiences, and when the environment is (temporarily) in the more uncommon condition it will depart from this. Assuming that the saturation gap is much wider than can be traversed in K evolutionary steps, we can expect the saturation time to be proportional to $2|0.5 - D|$, and the rate at which hierarchy evolves should be approximately proportional to this.

Figure 5.10 shows how the time-to-hierarchy changes when we vary the 'low' benefit coefficient c_L. These are relatively large values of c_L; previous experiments have all used $c_L = 0.7$. In these experiments, the time-to-hierarchy depends directly on the saturation gap, as values of c_L closer to the 'high' benefit coefficient $c_H = 1$ moves the low saturation point closer to the high saturation point, tightening the saturation gap. Such values of c_L close to c_H enable the evolutionary process to traverse the saturation gap within the $K = 1000$ evolutionary steps of each episode: the more quickly the saturation gap is traversed, the more quickly the modules saturate, the longer the system spends at saturation, and so the more quickly the system tends toward hierarchy.

The duty-cycle and saturation gap experiments provide significant clues as to the dynamics in the experiments in Chapter 3, where the parameter Z effectively determines the frequency with which each module experiences a low saturation point. For $Z < 1$,

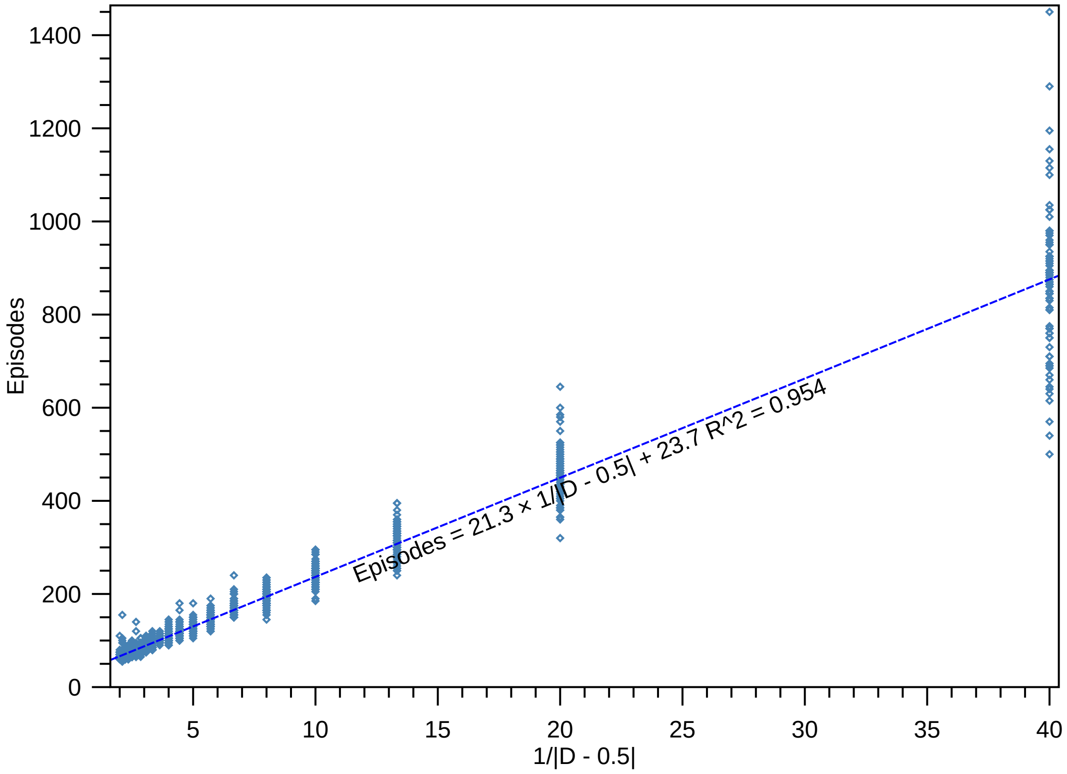


FIGURE 5.7: With a four gene module, the number of episodes elapsed prior to the emergence of hierarchy ($d_h \geq 0.9$ is a linear function of $1/|0.5-D|$. This suggest that the rate at which hierarchy evolves is approximately proportional to $|0.5 - D| \times 2$, which is proportional to how much more time the evolutionary processes is exposed to the more common environmental conditions (e.g. for $D = 0.8$, $|0.5-D|\times 2 = 0.6 = 0.8-(1-0.8)$). The intercept in the regression is consistent with the early period during evolution where the modules have not yet saturated.

in a randomly selected episode, any module will fix at a high fitness local-optimum (where the benefit coefficient is c_H) with probability $1 - Z + Z/2 = 1 - Z/2$: either it experiences an S-type environment with probability $1 - Z$, or it experiences a D-type environment consistent with its expression with probability $Z/2$. Crudely, we can treat this probability as the duty cycle of the system. This provides some explanation for why the experiments with lower Z evolve hierarchy more quickly (see Appendix C.1). It also provides an explanation for why the rate at which hierarchy evolves in each module increases dramatically when the module is sufficiently hierarchical to perform module switches: no longer does the module have to slide between the high and low saturation points; rather it can escape the low saturation point local optimum, and so always sees a high saturation point. Consequently, the duty cycle is effectively changed to 1 by the enhanced evolvability. Note that we can be confident that the experiments from Chapter 3 must be operating in this 'mode' (where the saturation gap cannot be traversed) because they used the same choice of $\lambda = 0.1$ and $M_B = 0.001$, use the low value of $c_L = 0.7$ as the experiments presented in Figure 5.9, have twice the probability of mutating B each evolutionary step, have 16 times as many entries in the B matrix

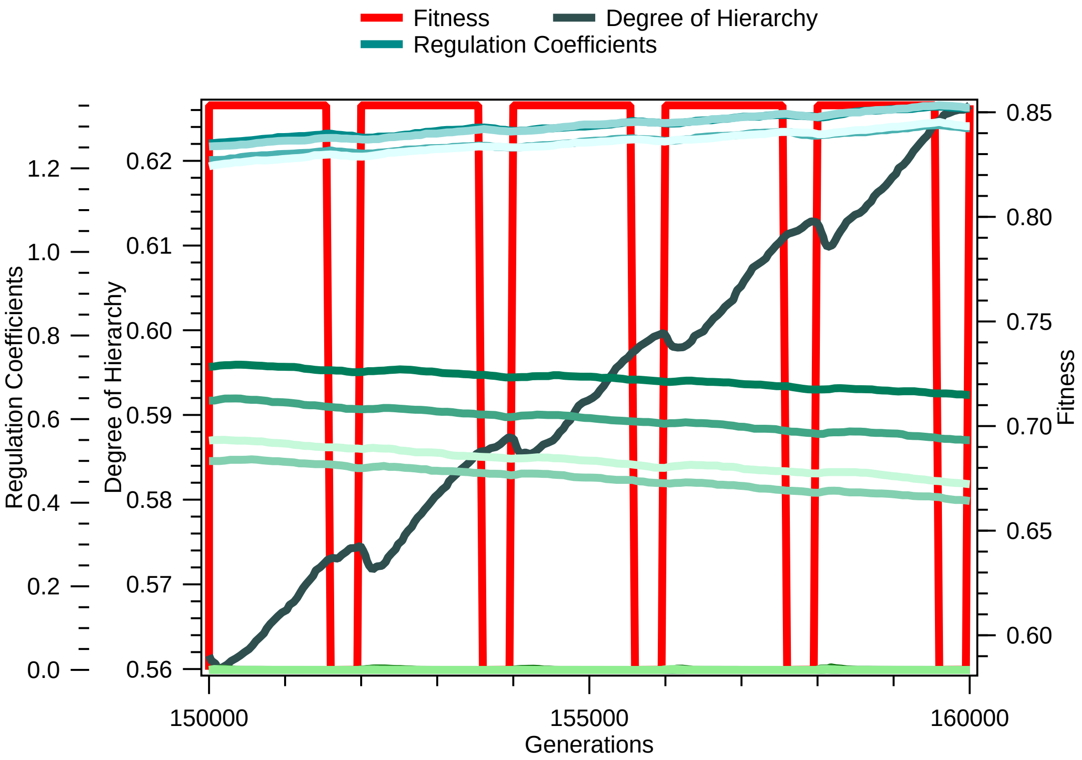


FIGURE 5.8: Short-term changes in the degree of hierarchy with a duty cycle of 80% across a number of episodes. Fitness is high in the high-benefit conditions, and because this is the most common condition, the system remains in the locality of the high saturation point. When exposed to the low-benefit conditions, all regulatory connections decrease in magnitude to compensate, taking the system away from the high saturation point. When the high-benefit environmental conditions return, the system takes approximately as long as to return to saturation as it spent in the low-benefit conditions, during which time the degree of hierarchy decreases by approximately the same amount as it increased while desaturating.

of regulatory connections, so we would expect it to take on average 64 times as long to traverse the saturation gap in each module, but they have (on average) half as long to do so.

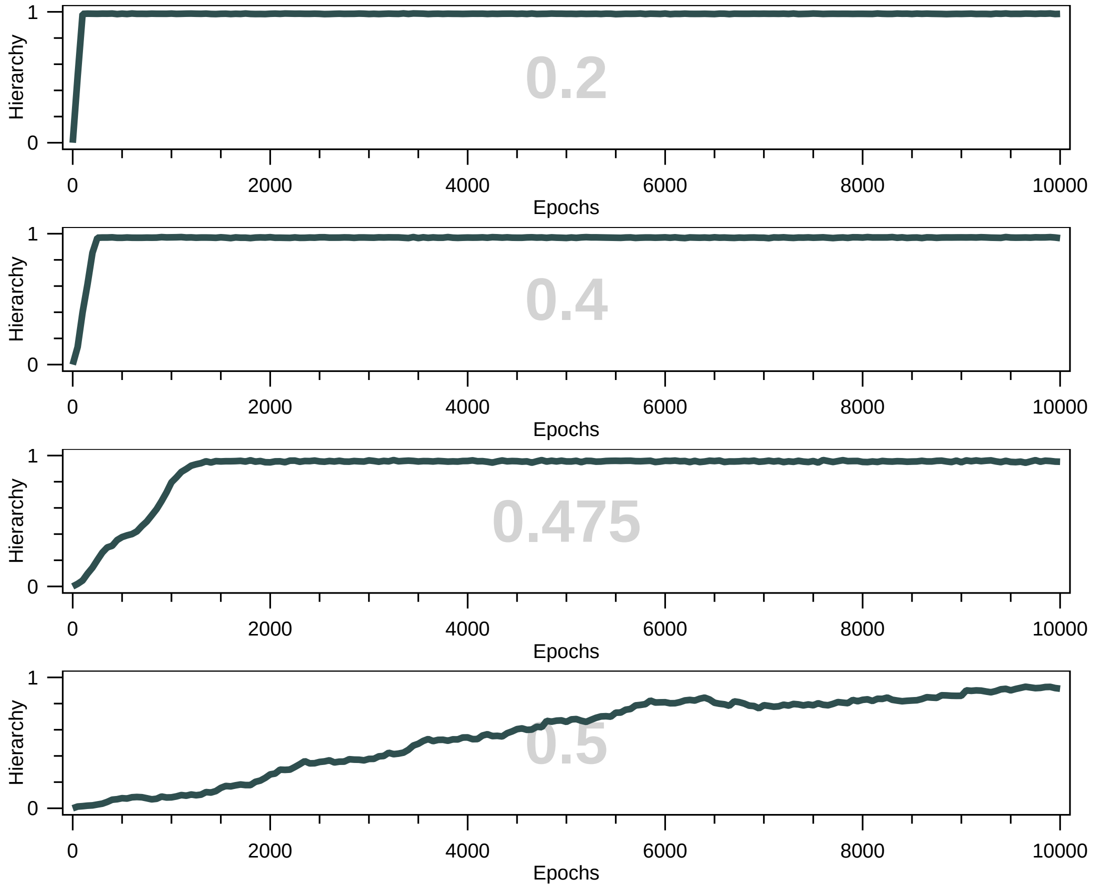


Figure 5.9: Plot of changing degree of hierarchy in a 4-gene module for a select number of duty cycles, showing the jittery behaviour for $D = 0.5$ where the system becomes 'lost' in the saturation gap.

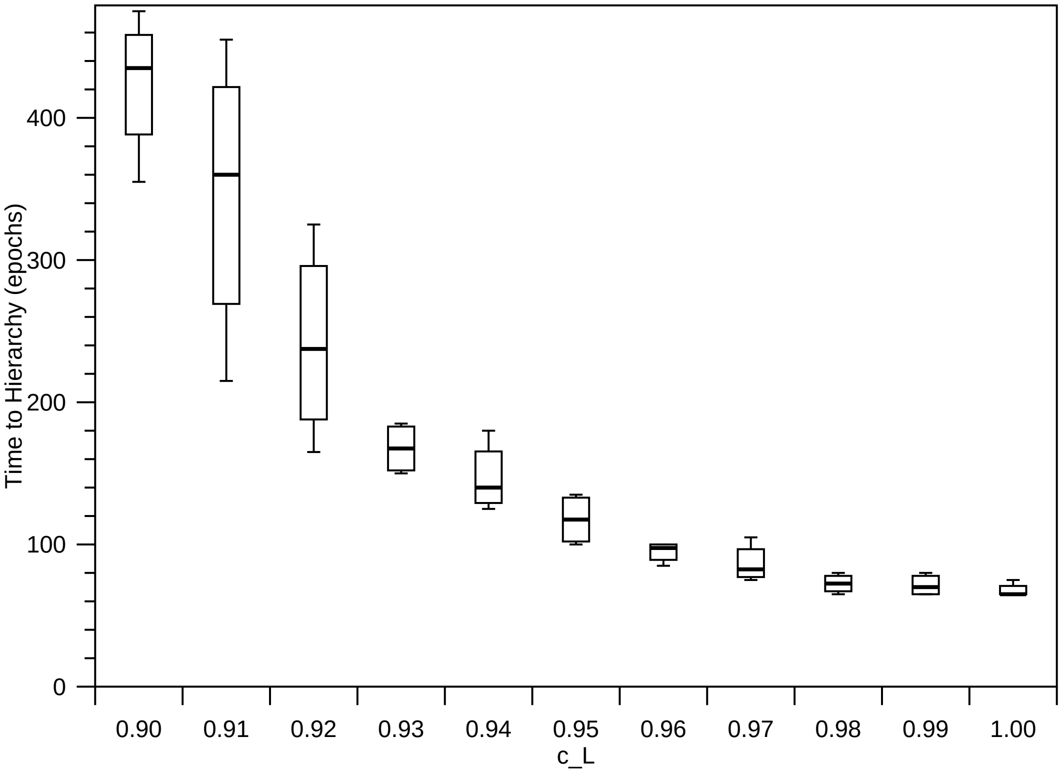


FIGURE 5.10: With a 4-gene module, the closer c_L is to $c_H = 1$, the more quickly hierarchy evolves. Hierarchy evolves readily when c_L is close to c_H despite the duty cycle of $D = 0.5$.

5.5.3 The evolution of modularity

This chapter is focussed on the causes of the evolution of hierarchy, but it is worth briefly discussing the cause of the modularity without which the hierarchy cannot provide any *useful* variability.

Just as individual modules can become saturated, so can combinations of modules. The experiments in Chapter 3 show that in some cases the evolving modules (which initially have many inter-module connections) go through some disassociation before independent modules emerge. These disassociations occur once the whole system saturates, because prior to this there is no way selection can prevent the evolution of inter-module interactions: when the B matrix is initialised as all zeros, then evolution behaves much like a Hebbian learner: all regulatory connections are constantly being changed according to whatever pattern of gene expression is currently expressed. During one episode, it may be that modules A and B 'agree', and so the interactions between the genes in the modules will grow more positive. In the next episode, they may disagree, and the interactions will grow more negative. If the distribution of phenotypes is balanced, then on average the inter-module connections will remain weak; however, if the distribution is not balanced, then - as discussed in Chapter 4 - the B matrix will reflect this. If the B matrix becomes too fixated on one pattern, it will start to bias the distribution of phenotypes observed and further selection will reinforce this pattern.

Indeed, the simulation shown in Figure 3.4 initially 'over-fits' onto a single pattern of expressions. Despite this, it is still able to disassociate the modules as is necessary to fully exploit the changing environment. The disassociations occur because the whole system saturates, stopping the yet unchecked growth of all interactions: from now on, an increase in magnitude of one connection will only come at the expense of another. For instance, the single-peaked (S type) environmental instances select against interactions that produce a phenotype ill-matched to the environment: selection actively reduces their magnitude, which moves the module away from the saturation point. Because the module is no longer saturated, evolution will re-enforce the other interactions (those which are consistent with the environmental conditions). Over many episodes, this moves weight from 'incorrect' connections to other connections. The dramatic completion of the disassociations observed in Chapter 3 occur once the B matrix no longer limits the variation in phenotypes, and so the interactions are also suppressed when they are inconsistent with the phenotype, regardless of the environmental conditions. Figure 5.11 shows a changing environment of only 2 modules and $Z = 0.7$, where the different stages in the evolution of the inter- and intra-module connections are more apparent.

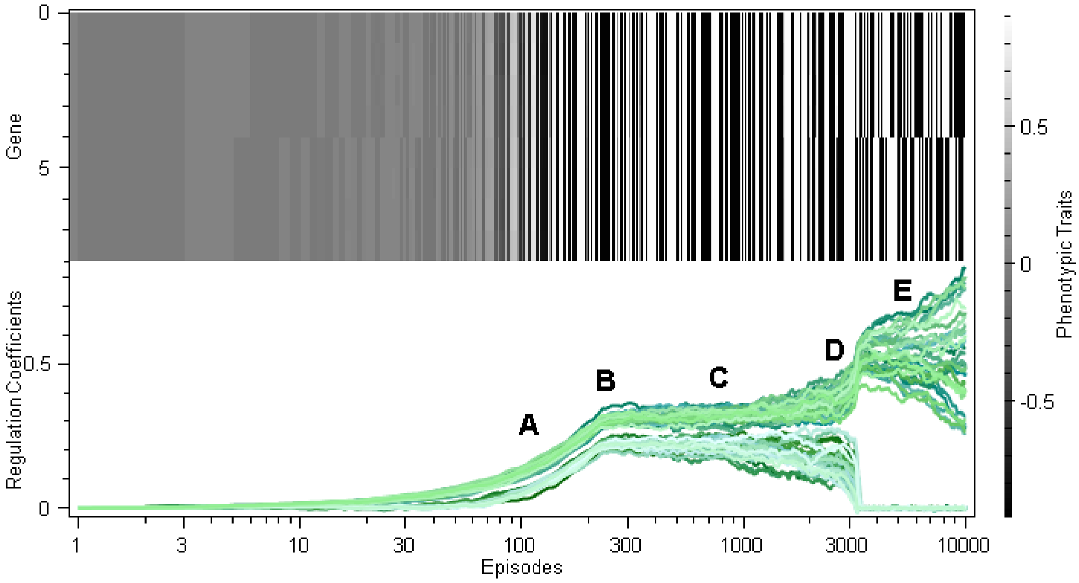


Figure 5.11: The evolution of independent modules in a changing environment of two modules with probability of multi-peaked module instance $Z = 0.7$ and B mutation magnitude $M_B = 0.0001$. Zebra stripes show the phenotype at the end of each episode, and the green lines the trajectories of the regulatory connections in the B matrix. A logarithmic time axis is reluctantly used to reveal the different stages of the dynamics; the apparent durations should be ignored. The upper bundle of connections are the intra-module connections; the lower bundle are the inter-module connections. Initially there is sufficient variation in the phenotype between episodes that the inter-module connections only drift slowly. However, by about 30 episodes they have become strong enough to influence the space of developable phenotypes, and the variation between the modules is lost. This leads to the rapid growth of the inter-module interactions along with the intra-module connections (**A**). Before 300 episodes (**B**), the system saturates and both bundles of connections stop growing. Now the process of slowly disassociating the modules takes over the dynamics (**C**). By 3000 episodes (**D**), the modules are sufficiently independent that the variation in phenotype returns, and the modules rapidly disassociate (the inter-module connections go to zero). Thereafter (**E**) the modules become steadily more hierarchical (it will take another 60000 epochs before this process completes).

5.6 Consequences of Model Parameters

We've seen above that the effective duty-cycle [1] of an evolutionary system can affect the saturation time directly. We've also seen how the saturation gap can affect the evolutionary process if the duty-cycle is near 50% [2]. In this section we will discuss how other parameters can affect the evolution of modularity and hierarchy.

[1] In Chapter 3, duty-cycle is determined by Z; it is unclear if this is especially meaningful in the context of Chapter 4.

[2] $Z = 1$ in Chapter 3, though the saturation gap is controlled by the c_L parameter; the variation in benefit coefficient is small in Chapter 4, and we have not tried to characterise it.

5.6.1 System Size

Changing system size (the number of genes N or the size of modules) has implications for the meaning of the benefit and cost functions, which are derived from those described in Kouvaris et al. (2017). The benefit function is defined as a function of mean phenotypic gene expressions, and the cost function as a mean of a per-connection cost. With a sub-additive cost of connections, the total connection weight influencing each gene is approximately the same (for sufficiently small modules or non-asymmetric regulatory topology), meaning the expected total connection weight will only grow linearly with the number of genes, while the per-weight cost is divided by the square of the number of genes.

The system size naturally changes the relative significance of the episode duration K, the mutation rates M_B and M_G, and the probability of mutation in B. The mutation rates are simple enough: each mutation only mutates a single entry, so when mutations are favourable on only a small number of genes (e.g. for large systems, most connections are not involved in the evolution of hierarchy) if there are more entries then we need to perform more mutations overall to 'find' the mutations that matter. This suggests that by increasing the number of modules, we would need to increase K to compensate, so that the same amount of information is integrated (as connections in the B matrix) each episode. Increasing K, however, and apart from changing the amount of between-episode variation, reduces the relative amount of time spent mis-matched to the environment (the transitionary period at the start of an episode), though we could compensate for this also by adjusting R_B, the probability of mutating B rather than G.

5.6.2 Temporal Parameters

In the model of a changing environment from Chapter 3, there are three key parameters that influence the long-term behaviour of the system: K, the episode duration; Z, the probability of a multi-peaked module instance occurring; and M_B, the mutation rate of the B matrix.

We know from Kouvaris et al. (2017) that K directly affects the ability of the system to generalise to a modular environment. If K is too small, then G does not have enough time to adapt to the environmental conditions, and so B does not evolve to reinforce well-formed samples. If K is too large, too much information may be integrated during early episodes, and the system can 'over-fit' (fail to disassociate the modules, resulting in a GRN with insufficient variability) due to the influence B exerts on G: this influence is positive feedback reflecting the phenotypes it has already experienced, and if it becomes fixated on a particular pattern of expression, it will fail to accommodate all of the environmental variability if hierarchy evolves before they can disassociate.

It is, however, possible to choose the other parameters such that the range of values of K for which the system is able to evolve independent modules is non-existent or arbitrarily large. This is because the reasons why a small or large choice of K are problematic are largely separable. The lower-bound for viable K is determined by the size of the system: we must provide enough time in each episode for all entries in G to respond, regardless how many of them actually respond, because the targets of mutations are random.

The upper-bound, on the other hand, depends on the rate at which the selective signal is integrated and how often the signal changes. The mutation rate M_B can be thought of as a learning rate (the rate at which the system integrates information). All else being equal, if this is too high, then the system over-fits to patterns it sees early on, or becomes too noisy to fix at all. Generally, the lower M_B, the more consistently the system integrates information from each exposure; however, the time to evolve any structure increases approximately proportionally to its inverse. The product $M_B \times K$ can be considered the amount of information integrated each episode. The concern with over-fitting is that the same signal is integrated too much at the start of the evolutionary process, so if the signal is different each episode (i.e. we observe a different phenotype) then the amount of information integrated each episode is the only factor that affects the propensity to over-fit. Consequently, if the signal is somewhat consistent between episodes, then the system will be more likely to over-fit because it integrates more than one episode-worth at a time.

In Chapter 3 the frequency of dual-peaked module instances Z directly affects the amount of variation in signal between episodes: if Z is large, then there are few single-peaked module instances to guide G to produce a different phenotype, and consequently the period over which the signal changes can be much longer than K. In Chapter 4 the propensity for the signal to change depends on Q: the larger the partial reset the larger the expected change in signal. By making a larger change to the signal, spurious consistency in the signal is lost more quickly.

Apart from determining whether it is possible to evolve independent modules, M_B and K directly affect how many episodes it will take for hierarchy to evolve, because it changes the saturation point as discussed in Section 5.5.2 and so influences whether or not a saturation gap can be traversed.

5.6.3 The Importance of Discontinuity

The evolution of hierarchy results in the squashing and stretching non-uniformly of the G-P map, such that the genetic distance between strongly-expressed phenotype and the basin of attraction for its complement is reduced: a smaller change is needed to escape from one phenotype to another, but there are inherent limits as to how small these distance can be. Indeed, it is essential to the exploitation of hierarchy that the mutations in G are large: if mutations in G were continuous and small, it would be impossible (with a continuous G-P map) to evolve any change in access evolvability in dual-peaked environmental conditions.

The nature of the bit-flipping mutations, which mostly change the initial expression of a gene, is essential to the access evolvability enabled by hierarchy: if G varied continuously or in small steps, then there could be no ability to 'flip' a whole module with a single-point mutation. Additionally, the targeted nature of the mutations is also important: if the same absolute change in initial gene expression were distributed over a whole module, it would again be impossible to flip such a module: the mutations must be axis aligned, or rather, the selective modules must align with the expressed genes.

It should also be noted that while the binary nature of G is necessary to exploit the hierarchy, it is unnecessary for it to evolve. Hierarchy will evolve happily in the model of Kouvaris et al. (2017) if given sufficient time to do so.

5.6.4 Other Saturation Point Considerations

One hereto largely ignored effect of different saturation points is that the less the total amount of connection weight involved, then the more quickly hierarchy can evolve because it simply requires fewer mutations. Any parameter that affects the saturation point can change this. All parameters which affect the saturation points will also affect the saturation gap (the space between saturation points): this includes the benefit function η, the cost function ϕ, the benefit coefficients, the cost coefficient λ, and the developmental process (which depends on T, σ, and h). We've seen that the saturation gap doesn't necessarily affect the saturation time, but we should enumerate some of the factors that change it none the less. Many parameters will influence the saturation point by changing the benefit and cost. The benefit coefficients of the environment have already been discussed: they directly modulate the benefit, and we have seen how we can narrow the saturation gap to increase the saturation time by changing them.

Because we use a non-linear squash function and perform many steps of the developmental update process, the mapping between connection weight in B and fitness is a fancy curve (Figure 5.2). Consequently, none of these parameters has an easily characterised effect on the saturation points, which are where the gradient of the various manifestations of this curve equals the gradient of the cost curve. The gradient is highly non-linear, so a linear change in cost or benefit (per the cost and benefit coefficients) does not have an easily predictable effect. Furthermore, the marginal benefit is small when the connections in B are weak (near 0), so it is possible for λ to exceed it (producing the 'pit of despair') that prevents the evolution of any regulatory connections in B at all: while there may be a nice saturation point further along the curve (the second differential is initially positive), the evolutionary process may be unable to reach it when it is initialised with an all-zeros B matrix as in Chapter 3.

5.7 Comparisons between Mutations

In this chapter, we explored how the evolution of hierarchy occurs when a module is saturated (i.e. the marginal fitness of increasing the connection strength of all intra-module connections is zero). This happens because the evolutionary process is forced to change the distribution of weight for regulatory connections in the B matrix to increase the efficiency of the module (to achieve high levels of gene expression with the low cost): evolution is forced to 'compare' the marginal benefit of each regulatory connection. This comparison is made indirectly by comparing the marginal benefit of each regulatory connection to its marginal cost: the marginal benefit depends on the other connections within the module, and the stronger the connections in the module the smaller the marginal benefit due to the saturating developmental curve. The reason hierarchy does not readily emerge in our model when modules are not saturated is because the evolutionary process cannot otherwise make comparisons between mutations: when the marginal fitness of all connections share the same sign then the mutations on them are similarly selectable.

In Kouvaris et al. (2017), variation in B was *not* provided by single-point mutations; rather, every entry in B was changed by a small amount in every generation (in addition to a single-point non-binary mutation in G). In these circumstances, any mutation to each regulatory connection in B is always evaluated in the context of many other changes, and so comparisons can be made: mutations on regulatory connections with a small marginal benefit will be compared to mutations on regulatory connections with larger marginal benefits: the larger changes will tend to determine whether *all* mutations are preserved. The choice to use single-point mutations in B in this dissertation was very deliberate as it better fits the assumptions of strong selection and weak mutation if it is presumed that the strength of regulatory connections can vary independently (i.e. mutations influence the regulatory effect of one gene on another without significantly affecting the interactions of other genes); none-the-less, it is important to recognise that this model 'detail' determines when selection for hierarchy is possible: in theory, the model of Kouvaris et al. (2017) could evolve toward hierarchy without first evolving dense (saturated) modules.

It is hence clear that the strong-selection weak-mutation assumptions adopted in this thesis encoded important assumptions about the meaning of the representation of the genome and have important consequences for the evolutionary causes and consequences of hierarchy. Because genes are well aligned with the environment (all genes make an equal contribution to their modules, and no contribution to any other module) and mutation is well aligned with the genes and their connections (mutations operate on a single entry in the G vector of initial gene expressions or B matrix of regulatory interactions) there is limited opportunity to compare the relative advantage of mutations: only the *sign* of the fitness consequence of a mutation is observed by selection (either a mutation increases fitness or does not); a mutation that produces a larger fitness benefit has no selective advantage over a mutation that produces are lesser (but still positive) fitness benefit.

5.8 Conclusions

In this chapter, we introduce the notion of developmental dominance, and discussed in some detail 'regulatory dominance' (motivated by asymmetries in the B matrix of gene interactions). We have shown that hierarchy is an efficient regulatory topology, one that maximises fitness for any sub-additive cost function will be a hierarchical one with a strong self-connection on the lead gene, and it is the regulatory dominance provided by this strong self-connection that is the key to this result. The model assumptions behind the proof of this property helped us to understand some of the conditions in which hierarchy may evolve, particularly if we assume selection will lead to an efficient regulatory topology. Crucially, we've shown that hierarchy can evolve without selection for evolvability (G was fixed in every experiment in this chapter) as indicated by observations in the previous chapters.

To better understand the mechanism by which hierarchy evolves in our models, we have introduced the concept of saturation, and acquired an understanding of how various model parameters can affect the 'saturation time' of a module. We have shown that hierarchy will emerge when comparisons can be made between the marginal benefit of individual regulatory connections, and this understanding provides clues as to how quickly hierarchy will evolve in a reduced model, and these results are consistent with observations in Chapter 3.

In the next chapter, we will explore dominance in G, and how it may provide another mechanism by which hierarchy can evolve: one that doesn't require regulatory dominance in B.

Chapter 6

The Evolutionary causes of Hierarchy

In the previous chapter we explored how regulatory dominance (dominance in the B matrix of regulatory connections) might explain the evolution of hierarchy (where one gene within a module directs the development of the whole module), and explored the possibility that selection for dominance and hierarchy in B to improve fitness at saturation would provide a sufficient explanation for the evolution of hierarchy as observed in Chapter 3 and Chapter 4. In this chapter, we develop a picture of a positive feedback between dominance and hierarchy, and test this by placing constraints on the model to limit two sources of developmental dominance ('transient dominance' in G and 'regulatory dominance' in B) and observing whether hierarchy evolves.

The previous chapter explained that developmental dominance (the property of one gene being more strongly expressed than all others during development) could be due to regulatory dominance owing to imbalances in B, and avoiding the issue of dominance in G (as occurs when the G vector of initial gene expressions changes in response to a perturbation of some sort) by treating it as fixed. In this chapter, we shall show that dominance in B is sufficient to motivate the evolution of hierarchy in simple scenarios (as indicated by results in previous chapters) but will focus on showing that transient dominance in G due to a changing environment is also sufficient to promote the evolution of hierarchy. To bring the two ideas together, we will describe the evolution of hierarchy as a consequence of positive feedback between the evolution of hierarchy (in B) and the evolution of dominance (due to B or G).

6.1 Transient Dominance

The regulatory dominance in B is a property of the gene interactions described in the B matrix, and can produce developmental dominance can persists over many episodes in our experiments due to the slow rate at which the interactions evolve. In contrast, the initial gene expression represented by the vector G change from episode to episode in our experiments, and in most cases rapidly fixes. Consequently, the only property that we can rely on is its consistency in removing any dominance produced by any perturbation. Indeed (in the absence of evolvability enhancing hierarchy) it is only during the period at the start of an episode in the experiments of Chapters 3 and 4 (e.g. after an environmental change or partial reset) that G shows any dynamics. In the case of a changing environment, it may be that the genes of a particular module must transition from all negative to all positive to satisfy an S^+ single peaked environmental instance from Chapter 3. Because the genes are mutated individually, during this transitional period there is opportunity for 'transient dominance': dominance induced by the transient imbalances in G. In the case of the partial resets of Chapter 4, the imbalance is imposed explicitly.

We will consider two flavours of transient dominance:

1. **Unbiased transient dominance:** where the order in which the genes in a module switch is random. This is the 'default' type of transient dominance, which occurs due to the normal dynamics of G responding to changed environmental conditions or a partial reset.
2. **Biased transient dominance:** where the order in which the genes in a module switch is biased, such that certain genes tend to lead others. Biased transient dominance is exemplified by hierarchy-enabled whole-module switches: the switch always starts with the switch gene, with all the subordinates following it wherever it goes. The contrast between this and unbiased transient dominance in the changing environment model of Chapter 3 is apparent in Figure 6.1.

Both types are able to motivate the evolution of hierarchy only when there are disparities in the relative influence of genes, where the 'influence' of a gene is the proportion of the regulatory weight that falls on the connections away from it (those to other modules and itself). Biased transient dominance is the easiest to observe: once a module is sufficiently hierarchical that whole-modules flips become viable (one gene having somewhat more than the total influence of all others), then when a module is placed on the worse peak of a dual-peaked environmental instance (e.g. a positively expressed module experiencing a D^- environmental instance from Chapter 3) or is mismatched with a neighbouring module (as in Chapter 4), the only gene in that module which can be switched to produce an increase in fitness is the lead gene of the hierarchy. During the resulting

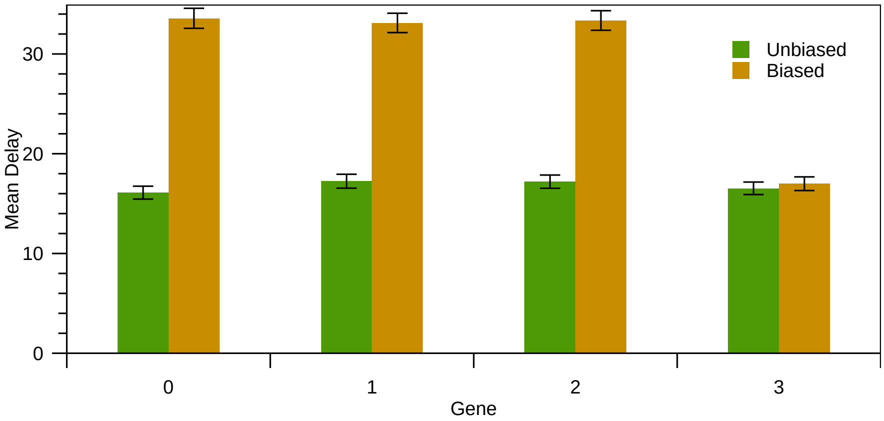


FIGURE 6.1: Plot of mean delay before an element in G switches with single-peaked and dual-peaked environmental conditions showing unbiased and biased transient dominance respectively. In the single-peaked environment, the order in which entries in G mutate is random and unbiased: the mean delay after the start of each episode for each gene to switch is roughly the same. In the dual-peaked environment, the order in which entries in G mutate is biased. This is most apparent when a hierarchical module is switched in a dual-peaked environment, where the switch gene (in this case, gene 3) is the only gene that can switch initially, and so the others must follow thereafter (indeed, the switch gene switches with the same mean delay as it would in a single-peaked environment, while the mean delay for the other genes is doubled, as they have to 'wait' for the switch gene to switch first).

period of transient dominance - during which other genes with influence are mis-matched - mutations in B that increase the influence of this gene or reduce the influence of other mis-matched modules will be more beneficial than those that have the opposite effect. The cost of connections will provide a threshold which determines which mutations will be accepted and which will be rejected (a state of transient saturation, if you will). The effect is that for a short period in each episode, there is a tendency to shift weight to the (transiently dominant) lead gene.

Transient dominance can also cause the evolution of hierarchy in the absence of any evolvability benefit in the form of unbiased transient dominance. When the environmental conditions change and a module becomes mismatched to its environment (e.g. a positive module experiencing an S^- single-peaked environment from Chapter 3), selective pressures act to weaken regulatory connections, as they drive development in the wrong direction. Selection also drives the G vector of initial gene expressions to invert itself, and there will necessarily be a transitionary period where some entries in the G vector have flipped, but not all. During this period, some genes will match the environment, and others will not. If the genes that match the environment have enough influence, then there will be a fitness benefit to increasing their influence because they pull the other genes in the right direction[1]. The threshold where increasing the influence of a group of genes is beneficial depends on many factors, but more influential genes are more likely to appear in a group of genes that collectively surpass the threshold by virtue of being more influential. This creates positive feedback, as groups of genes with more influence are more likely to increase in influence during a transitionary period. This effect is certainly applicable in a model of a changing environment where selection takes the evolutionary process from a local minimum to a local maximum (as in the single-peaked environmental instances in Chapter 3), as is shown in Figure 6.2.

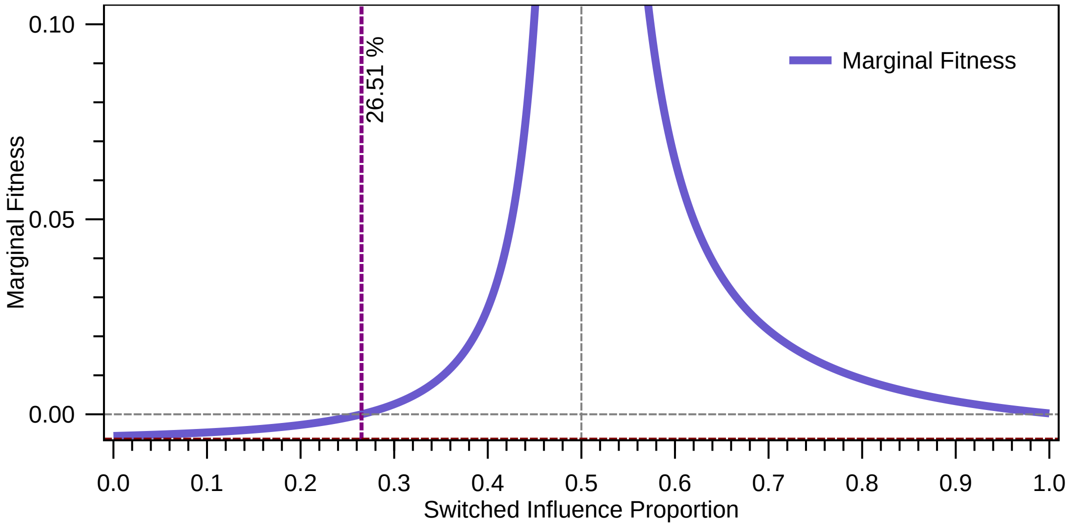


Figure 6.2: Plot of the marginal benefit of increasing the influence of switched genes in a single-peaked environment instance. When the genes that have switched have little influence (switched influence proportion is low), the benefit of increasing their influence is exceeded by the cost, so the marginal fitness is negative. There is a peak when one-half of the influence is switched (due to a bifurcation in the developmental process, see Appendix E). The marginal fitness approaches zero when all the switched genes have total control (indicating the module in this example is saturated). In this example, the influence of a gene (or group of genes) that controls more than about 26.5% of the module will be reinforced when they are switched toward the optimum because the marginal benefit of increasing their influence is positive. Generally the threshold depends on the total connection strength in the system, the benefit function, and the cost function. More influential genes are more likely to be members of a group of switched genes that (collectively) exceed the threshold. This promotes the evolution of hierarchy by increasing the influence of already influential genes during transitionary periods.

Importantly, this process doesn't depend on any variation in the order in which the elements of G switch (it can work with unbiased transient dominance) and operates independently of dominance that occurs in the B matrix. Without any local-optima in the landscape and assuming a regulatory topology that reflects the correlations in the environment, mutations in G that improve the match between G and the environment are

[1] It should be cautioned that this analysis assumes a module where all outgoing connections from a given gene have the same weight; without this assumption, the marginal benefits of different genes will vary within the groups of matched and unmatched genes, but it is not clear whether this is a significant effect in practice, and this chapter will explicitly avoid this complication.

always beneficial, so the order in which they switch is completely random (as indicated by the proxy measure of time-to-switch in Figure 6.1). Despite this, more dominant genes will become more dominant over time because they produce a greater (positive) impact when they happen to switch early in the transitionary period, and so are more likely to be members of a group of switched genes which together command enough influence (i.e. exceed the threshold as shown in Figure 6.2) that there is a fitness advantage to increasing their influence relative to the non-switched genes (who are pulling the module in the wrong direction). On average, the gene with the *most* influence is most likely to be a member of the group that passes this threshold, so this mechanism can readily maintain a hierarchy. However, to evolve new hierarchy, it requires sufficient disparities in the relative influences of each genes, as genes of similar influence can receive the same impetus from this mechanism due to the discrete nature of the groups that emerge. For instance, if two genes command 40% and 60% of the influence in a module, then they will always be in a sufficiently influential group if the threshold is below 40% as in Figure 6.2. As such, this mechanism depends on noise in the B matrix of interactions to produce disparities. For these reasons, this effect is somewhat unreliable, though it can be very visible.

The feedback with biased transient dominance in the case of hierarchy-enabled module switches is also very apparent, but the examples in our experiments require that an evolvability enhancing hierarchy is already present: this is particularly interesting, because it seems that biased dominance selects for evolvability. This will be discussed further in the discussion. Note that the 40%/60% scenario which would not trigger selection for hierarchy through unbiased transient dominance may well represent a sufficiently hierarchical topology that biased transient dominance occurs which can 'take over' the process. For now, a reminder that while feedback with biased transient dominance may seem a compelling explanation for the sudden increase in the rate at which hierarchy evolves in the experiments in Chapter 3, our previous explanation (that it changes the duty cycle) is probably the more significant contribution. If K were (relatively) smaller, this might change.

6.2 Feedback

Combining the assertions above regarding transitive dominance in G with dominance in B, we can paint a picture of the feedback between hierarchy and dominance in our experiments: hierarchy in B is motivated by dominance in B and G independently; and hierarchy motivates dominance in B (greater efficiency per Chapter 5), produces a signal for unbiased transient dominance, or facilitates biased transient dominance. Developmental dominance is required for a hierarchical topology to provide an efficiency benefit, and comes about due to dominance in B (i.e. uneven allocation of weight in the regulatory network) or (transient) dominance in G. The presence of a hierarchical topology in B also promotes dominance in B because a hierarchical topology is most cost effective when the lead gene is most dominant during development. Additionally, a hierarchical regulatory topology can produce biased transient dominance in some environmental conditions, and asymmetries in influence are necessary for feedback with unbiased transient dominance also, so there is opportunity for positive feedback through dominance in B or G to motivate the evolution of hierarchy. Sparing the details: developmental dominance promotes selection for hierarchy, which in turn promotes selection for developmental dominance.

In most of our experiments, it is perfectly possible that both dominance in B and transient dominance in G may be contributing to the evolution of hierarchy; however, we have already shown that dominance in G is unnecessary in experiments where G is fixed to all $+1$, as this removes any transitional period during which a disparity between genes can occur. Rather than removing mutations altogether, we could instead change the mutation regime for G such that every gene in a module switches together (rather than individually) and for comparison we shall do just that in this chapter to 'break' the transient dominance (and so preclude the systematic evolution of hierarchy through feedback with dominance in G).

We can similarly preclude within-module dominance in B with a modification to how the B matrix is mutated, such that 'columns' within modules (corresponding to all connections from one gene to those in its module) mutate together, rather than individual entries of the B matrix. This means that the sum of regulatory connections influencing each gene in a module will be the same (assuming we initialise the system to all zeros initially, which we shall) and so there can be no fitness advantage to a hierarchical topology when G matches the environment (as discussed in Chapter 5). Figure 6.3 depicts some examples of modules that might evolve with this constraint.

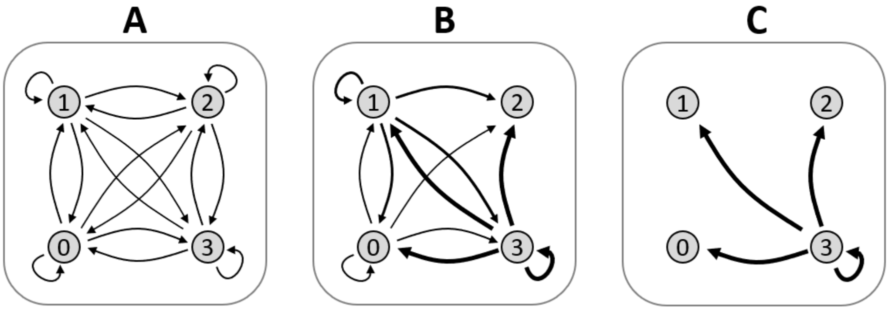


FIGURE 6.3: Examples of networks that might occur with the B dominance constraint. Rather than single-point mutations on individual entries of the B matrix, each mutation changes every outgoing connection from a gene by the same amount. This has the effect of ensuring the 'row-sum' (the total amount of incoming weight) is the same for all genes because the outgoing connections from each gene have the same strength (indicated by arrow thickness). When this is the case, and the initial expressions of all genes are initially the same, then the terminal expressions of all genes will also be the same. **A** represents an even distribution of weights (a 'dense' module), **B** represents an uneven distribution of weights, and **C** represents a hierarchy where gene 3 is the switch gene.

If our picture of feedback is complete, then we should expect that hierarchy can evolve when we apply either one of these modifications, but not both. We shall refer to them as the 'G dominance constraint' and 'B dominance constraint' and explore the four possible combinations of applying and not-applying them in a simple 4-gene system. We will use two environments to do so: one that comprises only single-peak environmental instances, and one that is a combination of single- and dual-peaked instances. The former will preclude biased transient dominance, as there will be no local optima to escape by way of a single-point mutations module flip; the latter will permit it.

6.2.1 Single-Peaked Environment

Our single-peak-only environment will switch between the S^+ and S^- environmental instances every $K = 1000$ evolutionary steps. This corresponds roughly to $Z = 0$ in the changing environment model of Chapter 3, but with a single module of four genes - rather than four modules of four genes - to ease interpretation. Table 6.1 summarises the other model parameters used for these experiments.

Parameter	**Symbol**	**Value**
Episodes		10000
B mutation magnitude	M_B	0.001
B mutation probability	R_B	0.5
Cost function	ϕ	$\phi(x) = \|x\|$
Cost coefficient	λ	0.1

Table 6.1: Parameters for the feedback experiments.

With neither dominance constraint, hierarchy evolves readily in this environment (Figure 6.4): there is no saturation gap, and there is ample time for G to fix during each episode. Including the G dominance constraint (Figure 6.5) doesn't change this, and in both cases, a hierarchy readily evolves in around 70 episodes in all of 10 replicates (runs with the same configuration but different random seed).

With 10000 episodes available, hierarchy evolved in 3 of 10 replicates with the B dominance constraint (Figure 6.6). With this constraint and the single-peak environment, unbiased transient dominance is the only feedback mechanism available. This indicates that it is possible to motivate (and maintain) the evolution of hierarchy given enough time, though it is highly stochastic and can take a long time owing to inconsistent signal between episodes and short transitioning periods. In many cases, two sets of regulatory connections will be effectively suppressed, but two will remain, seemingly trapped with roughly the same connection weight. How this state could be maintained is unclear (understanding thus far would suggest it should be an unbiased random walk); if either set of connections becomes too strong, then it will pass the influence threshold after which transient dominance will provide a positive feedback toward hierarchy.

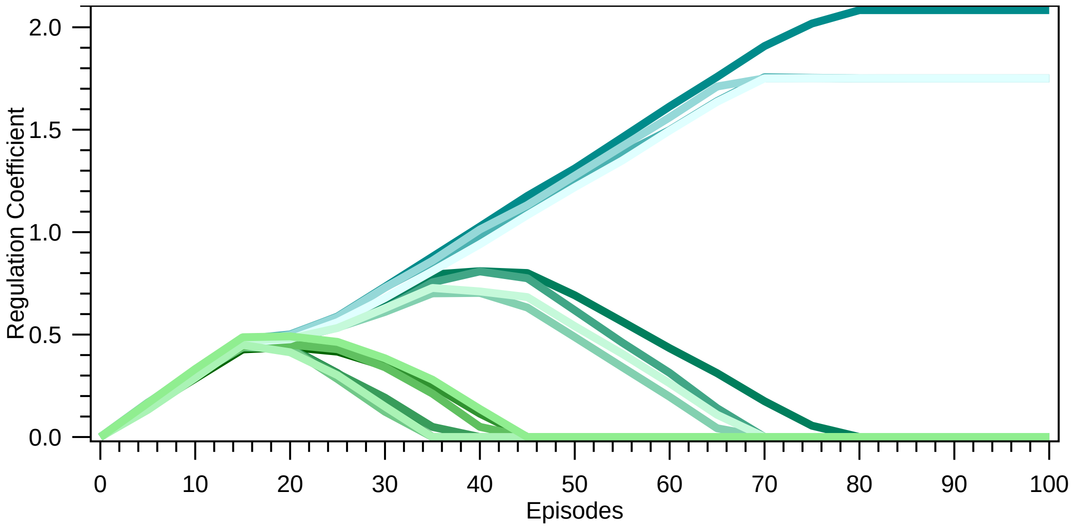


FIGURE 6.4: Representative plot of regulatory connection trajectories without any dominance constraint in the single-peaked environment. As in the experiments in Chapter 3, initially a dense module evolves (by around episode 16, all connections have similar non-zero weights of about 0.5), but the non-dominant genes eventually lose all their influence (3 bundles of lines that descend down to zero). What remains is a hierarchy with regulatory dominance in B (one strong self-connection, and 3 weaker connections to the other genes in the module).

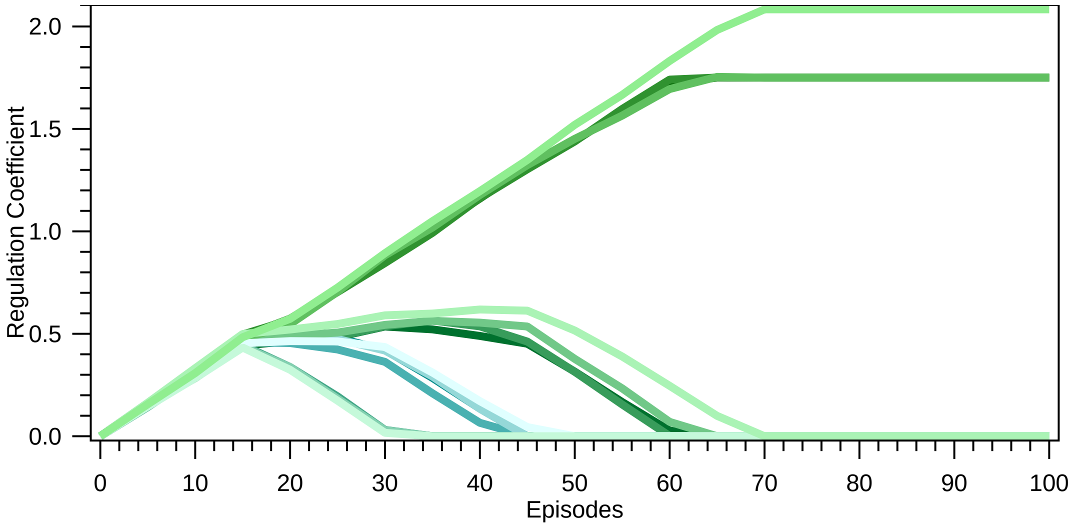


FIGURE 6.5: Representative plot of regulatory connection trajectories with the G dominance constraint, which is much the same as without the G dominance constraint (Figure 6.4). Transient dominance is evidently not essential for the evolution of hierarchy, and indeed does not contribute significantly with this combination of environmental instances and choice of K.

With both the B and G dominance constraints (Figure 6.7) hierarchy did not evolve within 10000 episodes in any of 10 replicates. Furthermore, with both constraints, evolution is not able to maintain an existing hierarchy: the trajectories of the regulatory connections are just random walks.

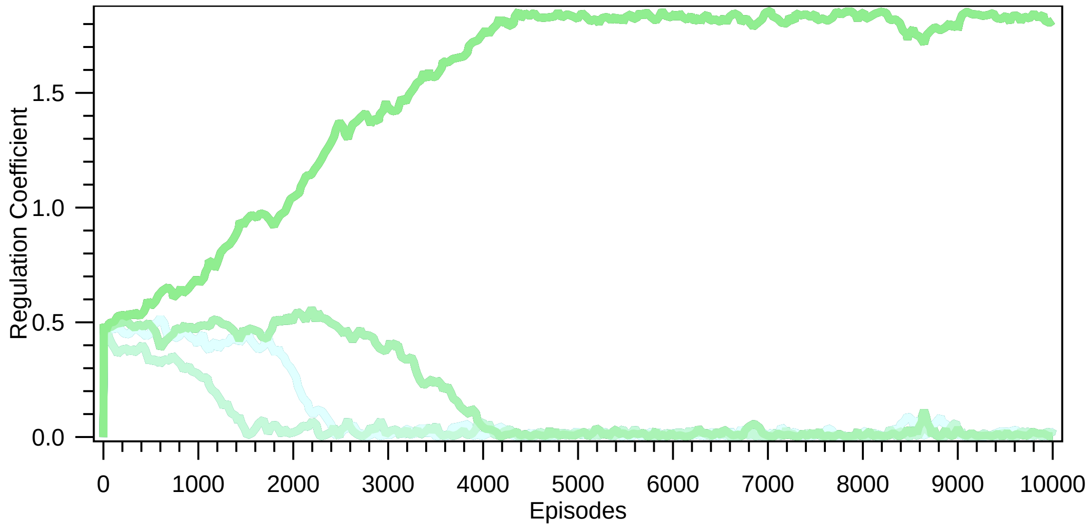


Figure 6.6: Example plot of regulatory connection trajectories with the B dominance constraint where hierarchy did evolve within 10000 episodes. Because of the B dominance constraint, there are only 4 lines: each corresponds to all 4 connections emanating from one gene (each 'column').

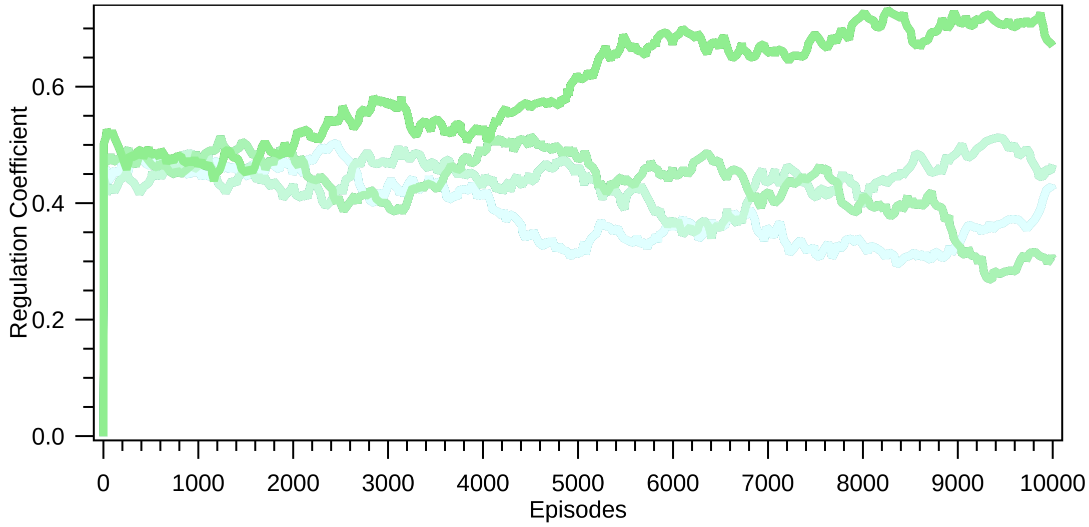


Figure 6.7: Example plot of regulatory connection trajectories with both the B and G dominance constraints. The regulatory connections effectively perform a random walk after evolving a dense module, remaining in saturation throughout evolution (the sum of connection weights is practically constant).

6.2.2 Rugged environment

Adding some dual-peaked environmental instances into the mix, we provide an opportunity for biased transient dominance. We set the switching period to $K = 1000$, and cycle through the instances in the order D^+, S^+, D^-, S^- (the total duration of each episode is the same as in the previous section). As in Chapter 3, the low benefit-coefficient is $c_L = 0.7$.

With the B dominance constraint (Figure 6.8), unlike with the single-peaked only instances, the biased transient dominance induced by the dual-peaked environments causes the system to evolve hierarchy rapidly once two sets of connections have been suppressed, as one of the remaining sets becomes able to direct the whole module. For this reason, all of 10 replicates evolved a hierarchy in under 10000 episodes.

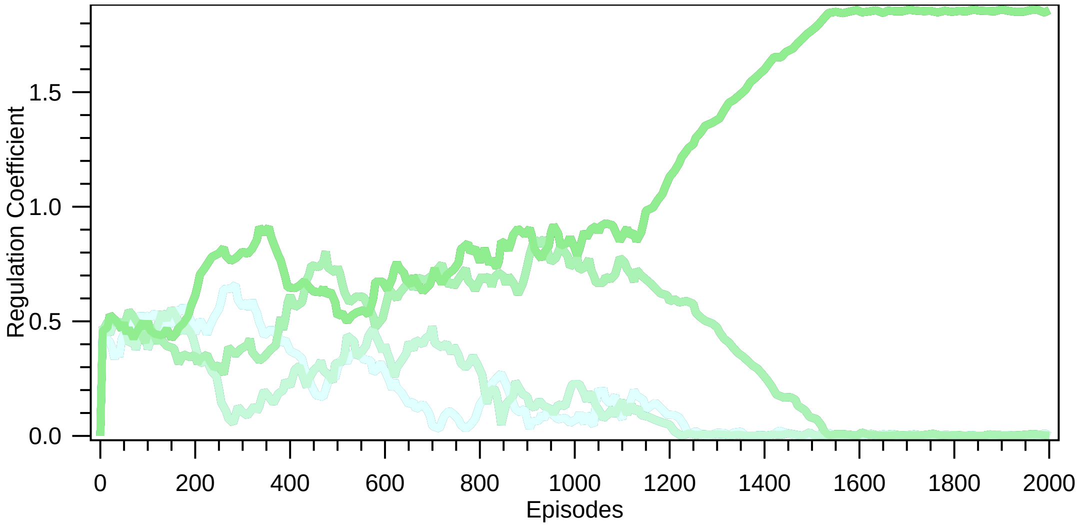


Figure 6.8: Example plot of regulatory connection trajectories with the B dominance constraint. Biased transient dominance causes the rapid evolution of hierarchy once one gene has enough influence to effect single-point module switches as eventually occurs by chance due to the stochastic nature of the simulations.

With the G dominance constraint (Figure 6.9), the dual-peaked environments are effectively ignored, as the evolutionary process can already perform whole-module switches with a single mutation: evolution proceeds essentially the same as it did in the single-peak-only environmental conditions.

This consistency is not the case when there is no dominance constraint (Figure 6.10). While hierarchy will evolve eventually, this environment effectively has a duty cycle of 50% (per Chapter 5), and so the wide saturation gap causes the evolution of hierarchy to proceed slowly until single-point module switches become viable and remove the saturation gap. This dramatic acceleration is the same observed in the larger experiments of Chapter 3.

With both constraints, the system again does not evolve or maintain hierarchy.

6.3 Discussion

We have seen that both (regulatory) dominance in B and (transient) dominance in G can motivate the evolution of hierarchy in our model and maintain it once evolved. In Chapter 5, we explained the evolution of hierarchy through dominance in B as being a

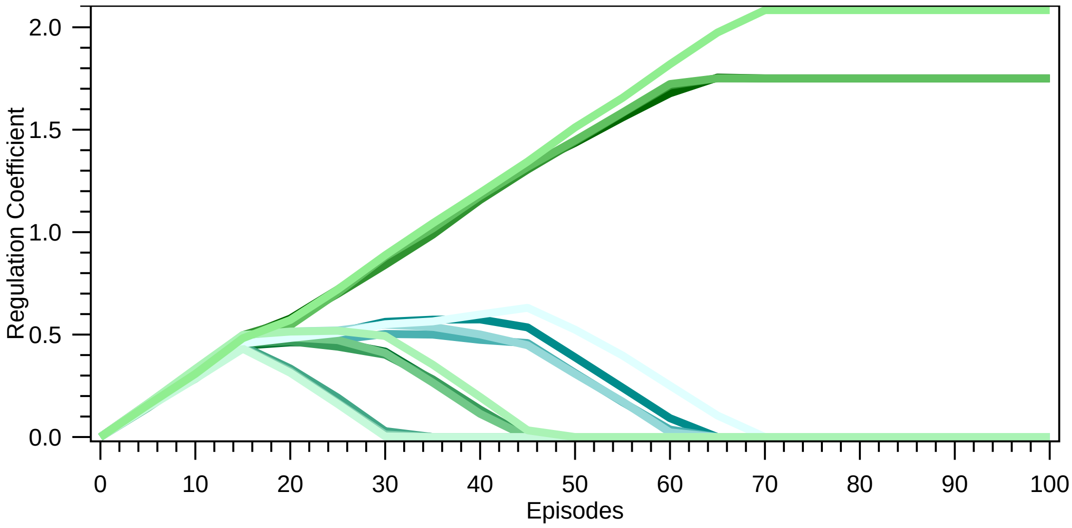


FIGURE 6.9: Representative plot of regulatory connection trajectories with the G dominance constraint, which is much the same as in the single-peaked environment (Figure 6.5).

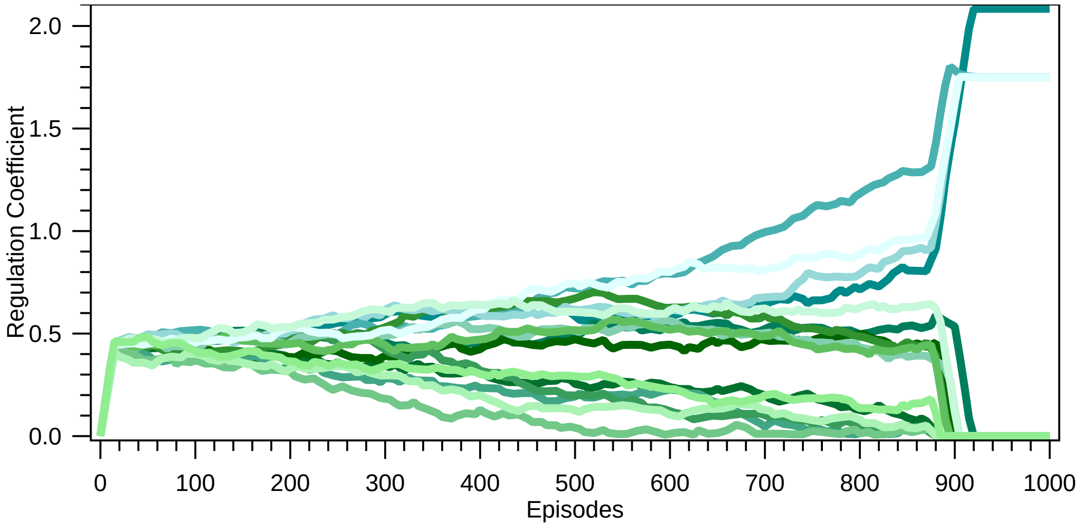


FIGURE 6.10: Representative plot of regulatory connection trajectories without any dominance constraint in the rugged environment. Due to the wide saturation gap, the evolution of hierarchy is slow until the single-point mutation module flips start occurring.

consequence of the efficiency of hierarchy. The experiments in this chapter with the B dominance constraint, show that regulatory dominance is not necessary (at least in these simple scenarios), though confirmed it is in some way sufficient. These experiments are qualitatively quite different from those in Chapter 5, and the dramatic effect of biased transient dominance appears like direct selection for evolvability: with biased transient dominance, we cannot separate the causes and consequences of the evolved hierarchy, because it requires that we can exploit the evolvability benefits of hierarchy before it

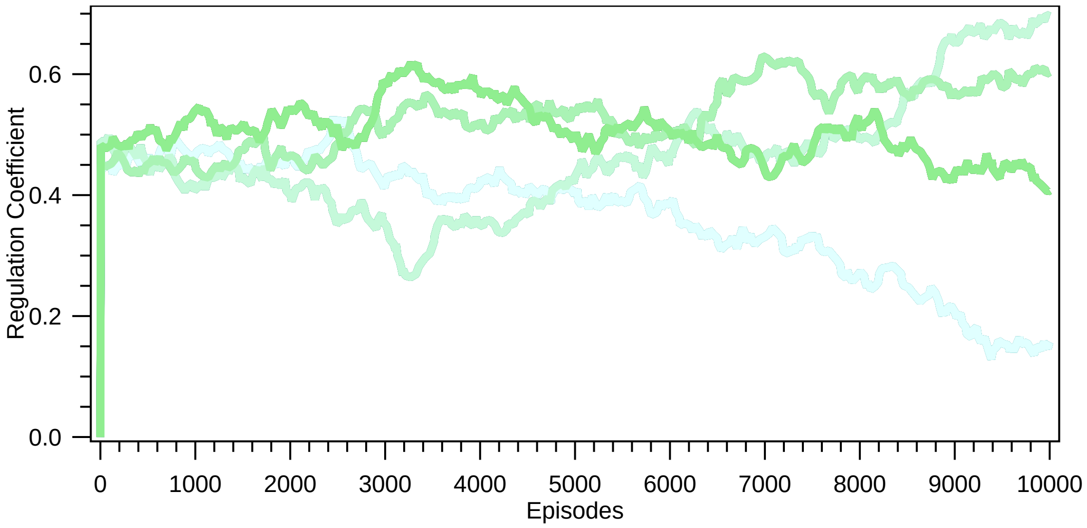


Figure 6.11: Example plot of regulatory connection trajectories with both the B and G dominance constraints. As in the single-peaked environment, the regulatory connections effectively perform a random walk.

comes into effect. This naturally raises the question as to what extent *selection for evolvability* may be considered to be contributing to the evolution of hierarchy.

Importantly, biased transient dominance is a special case of transient dominance, both of which require some luck (i.e. time and noise) to produce an asymmetric regulatory topology before they can motivate the evolution of hierarchy. A key difference is that biased transient dominance can be consistent across episodes: once module switches are possible, the lead gene will *always* switch first in a dual-peaked environment, while unbiased transient dominance relies only on providing a signal more often than not (e.g. in a hierarchy, unbiased transient dominance will provide no useful signal when the lead gene by chance switches last).

This feels close to the suggestion by Riedl (1977) regarding direct selection for evolvability; however, we suggest that the increase in hierarchy that comes about in these circumstances is, as in Chapter 5, ultimately due to feedback and selection for efficiency: the presence of dominance creates a scenario that favours hierarchy (and vice versa). The scenarios described in this chapter - where the dependent feature of dominance disappears routinely (it is transient) - does not align well with Rield's description of direct selection for evolvability. This is not to say that selection for evolvability cannot occur, but Riedl's description is not consistent with all our observations, and is not a complete explanation, and without a deep understanding of the developmental, evolutionary and selective model, has no real predictive power.

6.4 Conclusions

Hierarchy evolves because it exploits developmental dominance: the most efficient way to up-regulate genes within a module is for them all to be driven by the most strongly expressed genes. This developmental dominance may come about either through the uneven allocation of connection weight in the B matrix of regulatory interactions, or through variation in the G vector of initial gene expressions as a consequence of a changing environment or other perturbation. The evolution of hierarchy itself influences these causes, creating two somewhat independent feedback loops, which we are able to isolate by the use of 'dominance constraints'. By applying these constraints in small systems, we have shown that either feedback loop is sufficient to evolve hierarchy (though dominance in B is much more predicable), and that hierarchy does not readily evolve (nor is it maintained) in the absence of one of the feedback loops in a changing environment. We have confronted the question as to whether direct selection for evolvability may be present in our model, and the question remains as to whether a general notion of selection for evolvability is able to predict outcomes like ours without a detailed understanding of the dynamics, and it is not clear that it can in our model.

Chapter 7

Conclusions

This thesis has explored the causes and consequences of the evolution of hierarchical modules in a model of gene regulation: it is specifically concerned with how the introduction of a non-linear genotype-phenotype map can change the dynamics of evolution with respect to the evolution of evolvability. The gene regulatory model is a standard model, and the micro-evolutionary assumptions have remained consistent throughout the document.

Multiple selective regimes have been employed, all of which provide directional selection on phenotypes, and are specifically designed to minimise confounding factors that would make it difficult to assess the ability of hierarchy to affect evolvability. Combined with a cost of connections, this produces stabilising selection on the genotype. The evolutionary assumptions are specifically designed to be simple, so as to aid in interpretation of results. The experiments have primarily been concerned with the circumstances in which hierarchy can evolve, and the circumstances in which hierarchy provides an access evolvability benefit.

Based on our results and analysis, we draw these main conclusions:

- **Hierarchy in gene regulatory networks can increase evolvability by changing the level at which (random) variation operates.** In Chapter 3 we showed how independent hierarchical modules can evolve in a changing environment when a linear cost of connections is employed. We showed that this hierarchy can effect a dramatic increase in evolvability through a change in variability that enables the evolutionary process to 'jump' fitness valleys which would otherwise trap the evolutionary process in low-fitness regions of the fitness landscape. As a result, the evolutionary process is able to find high fitness phenotypes in any of the possible environmental conditions. The improved access evolvability is due to the change in the level at which mutations operate: mutations on the 'switch' genes that evolve effect change at the level of the module, as they determine the developmental trajectories of all genes within their module. Chapter 4 extended this result to a static (unchanging), hierarchically defined environment, where the change in phenotypic variability provided by hierarchy could lead to larger hierarchies: the evolvability benefits of hierarchy can bootstrap further evolution of hierarchy, and so further evolutionary benefits by repeatedly rescaling the level at which variation operates.

- **Significant variation in the genotype is necessary for evolution to infer problem structure, but not necessary for the evolution of hierarchy.** Chapters 3 and 4 build on observations from Kouvaris et al. (2017) and Kounios et al. (2016) respectively to show that variation can be induced by either a changing environment or periods of relaxed selection, and that this variation enables evolution to infer local and global problem structure. Chapter 3 relies on an environment where the conditions regularly permit a change (e.g. a single-peaked module instance) in the genotype to produce this variation; Chapter 4 relies on partial resets of the genotype (simulating periods of drift), which directly introduces variation at regular intervals. Though similar claims were presented explicitly in Kounios et al. (2016), we have taken measures to prevent the inference of problem structure as occurs in their work, and have addressed a concern regarding the necessity of 'resets' by parametrising the amount of drift introduced each episode to explore this directly. Using this, we have shown that the magnitude of the reset (i.e. the number of neutral mutations of drift) affects whether the evolutionary process can evolve a suitable regulatory network, such that an insufficient magnitude results in over-fitting of the regulatory network to a sub-optimal phenotype. Where evolution is able to infer problem structure (i.e. modules that correspond to the selective environment) it is also able to evolve hierarchy, but it is clear from our results that hierarchy will evolve whether the problem structure can be inferred or not.

- **The causes and consequences of evolvability need not be the same.** Hierarchy evolved in the experiments of Chapters 3 and 4 when a linear (L1) cost of connections is applied and does so regardless of whether the evolutionary process can find a high-fitness phenotype. Chapter 5 shows that hierarchy will readily evolve in circumstances where it provides no evolvability benefit and provides analytic evidence to support the idea that short-term selection for greater fitness when a sub-additive cost of connections is employed is sufficient to promote the evolution of hierarchy. Chapter 6 attempts to generalise the conclusions from Chapter 5 as a general feedback between a hierarchical topology in the regulatory network and dominance in the gene expression. By controlling the ability to evolve dominance directly, Chapter 6 shows that hierarchy can evolve as a result of dominance induced either by selection on embryonic conditions or by disparities in the strength of regulatory connections. The chapter concludes by returning to the original question as to what extent selection for evolvability may be responsible for the evolution of hierarchy: we conclude it is unnecessary in our model, and provides little insight in the situations where it may be argued to occur.

These conclusions are significant, as they provide evidence that the inclusion of a genotype-phenotype mapping (gene regulation) with a parsimony pressure (a cost of connection in our models) can enable a micro-evolutionary process to evolve valuable structure that reflects its environment, and that this structure can enable it to better exploit its environment in the long term as a direct consequence of increased evolvability.

In addition to these core claims and observations, we have explored the conditions under which hierarchy can evolve in depth. We have proved that hierarchy is an efficient regulatory topology and demonstrated a mechanism by which this efficiency can be selected considering the specifics of the model. This helps us to understand the conditions in which hierarchy evolves in our simulations, and to explain the rate at which does so. We have characterised the evolution of both modularity and hierarchy under strong-selection weak-mutation as consequences of 'saturation', a generalisation that enables us to make predictions about how different model parameters will affect the propensity for modularity and hierarchy to evolve. Finally, the characterisation of the evolution of hierarchy as feedback between selection for hierarchy and dominance in Chapter 6 may generalise to other models which include some form of dominance (i.e. non-uniform expression) and hierarchy (i.e. asymmetric control).

7.1 Future Work

There are a few natural ways that future work could extend the results of this dissertation:

1. **Relax the strong-selection weak-mutation assumptions.** As discussed in Chapter 5, this dissertation has employed a particular evolutionary model where small axis-aligned mutations are considered one at a time by a hill-climber. This has important consequences, and so relaxing the strong-selection weak-mutation assumptions (e.g. by performing multiple mutations or using a population) may change the dynamics significantly. For instance, if it is possible to compare the magnitude of the change in fitness produced by mutations (as opposed to just the sign), then evolution can select for a more efficient (e.g. hierarchical) regulatory topology prior to saturating the regulatory connections in a module. Furthermore, hierarchy could produce a significant change in rate evolvability by allowing selection to preferentially select mutations on more influential genes (i.e. the lead gene in a hierarchy). As such, it would specifically be interesting to explore allowing multiple mutations on G, either by performing mutations simultaneously

or by employing a population. Doing so is a natural route for future work, as it makes for a more complete model of evolution and it will be comparatively easy to untangle the effects of lineage selection from other selection in this model because we already understand many of its dynamics well. It should be noted that Kouvaris et al. (2017) performed mutations simultaneously on G and B, and the mutations on B were not axis-aligned: the later allowed selection for modularity and hierarchy without the need for saturation, but it is unclear that the former has any significant consequences in the scenarios they explored. It may also be instructive to change the relative magnitude and frequencies of mutations on different genes, which - while often modelled as indistinguishable in these regards - are not in living organisms (Erwin and Davidson, 2009).

2. **Exploring hierarchy in other network models.** This dissertation has exclusively explored the evolution of hierarchy with a genotype-phenotype map provided by a gene-regulatory network. This provides what Watson and Szathmáry (2016) describes as 'co-variation', where phenotypic characters change together under genetic variation. It would be interesting to see whether the same sort of hierarchical organisation could emerge in other network models, such as an ecological model of co-selection based on that of Power et al. (2015). Importantly, such a model would have the selective unit at the 'node' level of the network, rather than mutations and selection applying to the whole network. This would make it unlikely that selection for efficiency in static environmental conditions could motivate the evolution of hierarchy, but dynamics like those explored in Chapter 6 may be applicable.

3. **Multi-layer hierarchies in a gene regulatory model.** It is notable that we only evolve single-layer hierarchies (in contrast to e.g. Mengistu et al. (2016) who evolve a layered network model) which have limited expressivity: it can only represent correlations between the genes (i.e. gene A and gene B should have the same or opposite phenotypic expression). One possible way in which more exciting topologies might emerge would be due to an alternative cost function or mutational regime which encourages somewhat sparse regulatory topologies; however, preliminary investigations - for example, where the number of outgoing connections is limited - have not provided any promising results: as discussed in Chapter 5, indirection is selected against by the cost of connections.

4. **Determine whether the spiky continuous HIFF can be solved without hierarchy**. Chapter 4 introduced the Spiky MC and HIFF problems specifically to reduce the potential for a simple evolutionary process to dependably find high fitness phenotypes; however, it may be the case that high fitness phenotypes can be found in the continuous HIFF without the use of hierarchy (as is the case with the single-layer MC problem). Preliminary results have been inconclusive, and the mechanism by which this could occur is unclear, though some discussion is provided in Chapter 4. It seems that the mechanism is due to positive feedback

on drift, and consequently it will be computationally expensive to explore in the current model due to the slow mutations rates involved; it may be that imposing a variability constraint of the sort used in Chapter 6 to preclude dominance could allow us to probe the mechanism directly. Should it transpire that there is no mechanism which can reliably solve the Spiky MC or Spiky Continuous HIFF, then this would be a strong result; otherwise, there is potential to compare the relative time-complexity of such a mechanism with that described in Chapter 4 (the repeated rescaling of variability due to hierarchy).

5. **Explore the evolution of hierarchy without selection toward extreme values.** As identified by Rünneburger and Le Rouzic (2016), works like ours tend to select toward extreme characters: in our case, the linear and quadratic benefit functions provide consistent directional selection on phenotypes; however, the inclusion of a cost of connections means that the selection is not strictly directional on the B matrix part of the genotype. None-the-less, it may be instructive to investigate benefit functions which favour intermediate expression levels.

6. **Exploring the stability of hierarchy when subjected to changing environmental structure.** It is apparent that our model reveals structure that appears consistent in the experienced environmental conditions (changing or otherwise), and it is also apparent that the premature evolution of hierarchy can prevent the evolutionary process from surfacing this structure. It would be interesting to explore to what extent this is the case by introducing another time-scale to the model, such that after some number of episodes the underlying environmental structure is changed. We have shown that the evolution of hierarchy can provide evolvability benefits when the environmental conditions have a consistent structure, but it may be the case that this is at the expense of longer-term variability necessary to respond to a radically different environment.

Appendix A

Definitions and Proofs

This appendix provides some simple proofs and definitions that are not essential to the arguments in the thesis.

A.1 Proof of Bounded Phenotypic Expression

During the recurrent process, gene expression decays by a factor of $\tau \in (0, 1)$ and grows by some value produced by a squash function $\sigma(x)$ with maximum magnitude 1. Duly, we can prove that the terminal phenotype P, which is equal to τY_T has elements with maximum magnitude 1 given the initial conditions have magnitude less than $1/\tau$.

Consider the developmental update equation (Eq 3.1) and preconditions:

$$y_i(t+1) = (1-\tau)y_i(t) + \sigma\left(\mathop{\vphantom{\sum}}\limits_{j}^{N} B_{ij}y_j(t)\right)$$

$$|\sigma(x)| \leq 1 \qquad |y_i(0)| \leq 1$$

We want to prove the general case $|y_i(t)| \leq \frac{1}{\tau}$, so that $P_i = \tau y_i(T) \leq 1$. The base case of $y_i(t=0) \leq \frac{1}{\tau}$ is given by our initial conditions, and we proceed by induction to prove the general case.

$$|y_i(t+1)| \leq (1-\tau)|y_i(t)| + \sigma(x)| \tag{A.1}$$

$$|y_i(t+1)| \leq \frac{1-\tau}{\tau} + 1 \tag{A.2}$$

$$\therefore |y_i(t+1)| \leq \frac{1}{\tau} \tag{A.3}$$

A.2 Degree of Hierarchy

Degree of hierarchy is defined as the proportion of weight that it contributed by the module with the greatest total weight contribution (i.e. the weight of connections ‘from’ it, those on its column in the regulatory matrix) scaled between 0 and 1. Inter-module weights are ignored. For a module m containing n genes with dominant gene s, the degree of hierarchy d_h is given by Eq A.4.

$$d_h = \frac{1}{1-1/n} \cdot \quad \frac{{}_{i\in m} |B_{is}|}{{}_{i,j\in m} |B_{ij}|} - 1/n \tag{A.4}$$

Though this scalar metric has little descriptive power, it is valuable as an indicator of the extent to which a single a gene determines the trajectories of the whole. Indeed, if we consider a regulatory system where the weights of connections from each gene in a module to the other genes in its module are uniform (and consistent with all other connections), then if the proportion of weight ‘owned’ by the dominant gene is over one-half, and all genes start with the same magnitude of gene expression, the developmental process will drive all genes in the module to whatever sign the dominant gene bears.

Appendix B

Musing on the MC problem

This appendix provides some additional commentary on the Modular Constraints problem employed in Chapter 4.

B.1 Proof of equivalence of generalised quadratic MC with standard MC

The traditional definition of the MC problem is as a quadratic matrix product Watson et al. (2011c):

$$b_{MC} = P^T W P \tag{B.1}$$

Where W is a matrix of gene interaction strengths, such that $W_{ij} = 1$ if i and j are in the same module, and $W_{ij} = p$ otherwise. We can decompose the constraints matrix into two matrices, W_g and W_m, where W_g is a matrix where every entry contains the value p, and W_m is all zeros, except for the entries that correspond to genes within the same module, which have value $1 - p$. W is the sum of these two matrices.

$$W = W_g + W_m \tag{B.2}$$

W_g represents a 'grand' module which includes all entries; W_m represents the sub-modules. We can now re-write the matrix product from equation B.1 as the sum of two nested summations:

$$b_{MC} = P^T W_g P + P^T W_m P \tag{B.3}$$

$$b_{MC} = \sum_{i\in g}\sum_{j\in g} P_i p P_j + \sum_{m\in M}\sum_{i\in m}\sum_{j\in m} P_i(1-p)P_j \tag{B.4}$$

Where g is the set of all genes, and M is the set of all sub-modules $m_1, m_2, ...$, where m_i is the set of genes in that modules. This simplifies to

$$b_{MC} = p\left(\sum_{i\in g} P_i\right)^2 + (1-p)\sum_{m\in M}\left(\sum_{i\in m} P_i\right)^2 \tag{B.5}$$

Given $|g| = N$ and assuming all sub-modules are of size $|m| = K$ (per the traditional MC problem), and defining $T_g = \sum_{i\in g}/N$ and $T_m = \sum_{i\in m}/K$ for each module $m \in M$ (the mean phenotypic expression in each module, in range $[-1, +1]$), this can be re-written as

$$b_{MC} = pT_g^2N^2 + (1-p)\sum_{m\in M} T_m^2K^2 \tag{B.6}$$

$$b_{MC} = pN^2T_g^2 + (1-p)K^2\sum_{m\in M} T_m^2 \tag{B.7}$$

Factoring out $(1-p)K^2$, we can write this as:

$$b_{MC} = (1-p)K^2\left(\frac{pN^2}{(1-p)K^2}T_g^2 + \sum_{m\in M} T_m^2\right) \tag{B.8}$$

Now we compare this to the generalised MC, with module-benefit-function $\eta(x) = x^2$, where we use the same g and m notation to differentiate the grand module (with $c_g = d_f$) and sub-modules (with $c_m = 1$):

$$b_{GMC} = \frac{\sum_{m\in(M\cup g)} c_m \times \eta\left(\frac{1}{|m|}\sum_{i\in m} P_i\right)}{\sum_{m\in(M\cup g)} c_m} \tag{B.9}$$

$$b_{GMC} = \frac{d_f\eta(T_g) + \sum_{m\in M}\eta(T_m)}{d_f + |M|} \tag{B.10}$$

For $\eta(x) = x^2$, we have

$$b_{GMC} = \frac{1}{d_f + |M|}\left(d_fT_g^2 + \sum_{m\in M} T_m^2\right) \tag{B.11}$$

Ignoring the constant terms outside the brackets (which change the magnitude of the benefit, but not the shape), equations B.11 and B.8 are equivalent when we take the decay factor to be

$$d_f = \frac{pN^2}{(1-p)K^2} \tag{B.12}$$

Rearranging, we get an expression for p in terms of d_f.

$$p = \frac{1}{1 + \frac{N^2}{d_f K^2}} \tag{B.13}$$

For our experiments, we only used a value of $d_f = 0.75$, which gives a value of $p \approx 0.012$ for 8 modules of 8 genes, which is comparable to values used in previous literature. Ignoring the constant factors (which we did not equate) only changes the magnitude of the benefit function: it does not change its offset or shape, so this has no effect on the dynamics of the system with the inclusion of a per-connection cost-of-connection (as used in this thesis): this arbitrary benefit factor (which we choose such that the benefit is in the range $[0, 1]$ for convenience) can be equivalently simulated with a different choice of cost coefficient λ. As such, the generalised MC with $\eta(x) = x^2$ is equivalent to the traditional MC for our purposes.

Regarding the interpretability of p and d_f, neither is particularly intuitive, and both need to be chosen according to the number of modules in the super/grand-module. d_f is somewhat desirable for our purposes because it implies a consistent ratio in signal at each layer of the continuous HIFF in Section 4.6 (the super-modules in each layer comprise the same number of sub-module from the layer below at each layer).

B.2 Partial resets without development

In Kounios et al. (2016), a *total* reset hill-climber without a developmental mapping was compared to a partial reset hillclimber with a developmental mapping: the total reset hill-climber without developmental mapping failed miserably to learn, by virtue of having no persistent state. Though not relevant to our investigation into the evolution and evolvability of hierarchy, additional experiments were run to evaluate the power of a *partial* resets without a developmental process.

It transpires that it is possible for a carefully tuned partial-reset hill-climber *without* B to solve the 'concentric squares' problem described in Kounios et al. (2016), as well as some MC problems. In the case of the MC problem, this is due to the same signal causes the model with the developmental mapping to explore high-fitness phenotype more often than one would expect by change. The different sizes basins allow the model without developmental mapping to climb uphill in a 'ratchet' fashion: the magnitude of reset necessary to 'flip' a module that is ill-matched with the rest of the phenotype (and so achieve a higher fitness) is smaller than the magnitude of a reset necessary to 'flip' such a module back. The distribution of reset magnitudes is controlled by the G mutation magnitude M_G, the neutral mutation count Q, and the module size K, and careful tuning can produce a system which tends toward the global optimum.

It is noteworthy that the modules in the MC problem are only consequential because of variation in the variability: because the module-benefit-functions are taken over an average of the expression of each module, the fitness effect of a change in the phenotypic expression of a trait is the same for all traits within a module. Variation in module size, however, has a significant effect on the types of distributions of phenotypic expressions observed after a partial reset (as in Figure B.1): larger modules mean that the distribution more closely approximates a Gaussian whose mean approaches 0 as the size of the reset is increased (i.e. less information is retained about the previous state).

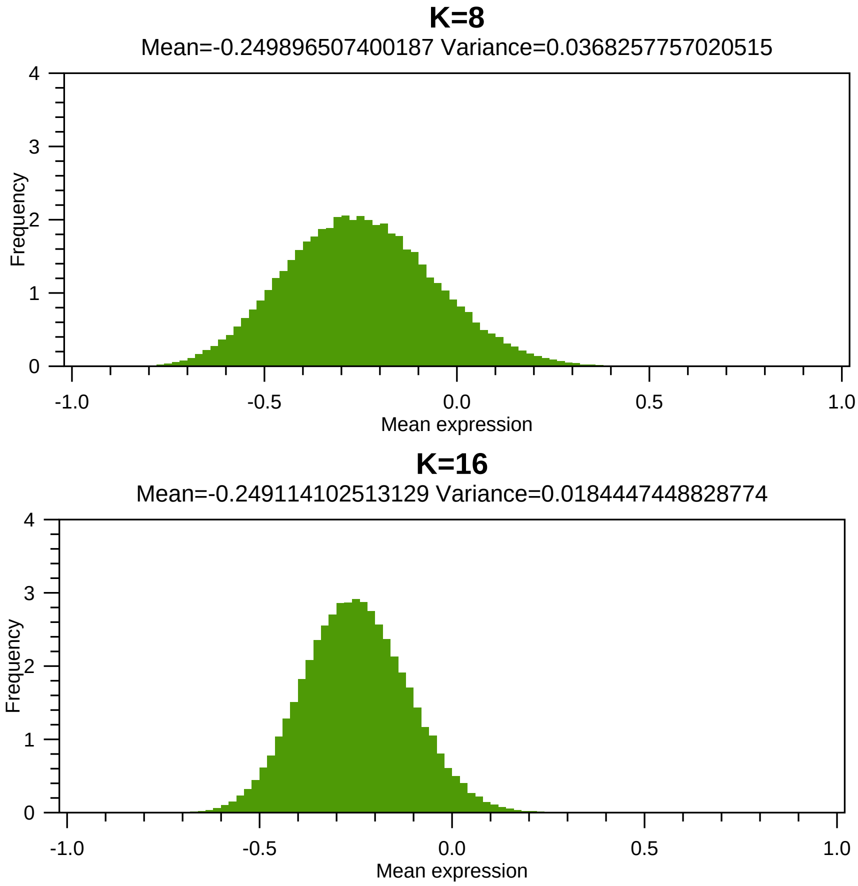


FIGURE B.1: Distributions of mean module expression after a $Q = 20k$ reset, starting from all entries being -1, for a genome of size 64 with modules of size $K = 8$ and $K = 16$. A larger module reduces the variance in the distribution of mean phenotype expression, which reduces the ratio of large module resets - which randomise the state of a module - to tight module resets, which move the system into a state of uncertainty where it is able to follow local gradients.

Appendix C

Select additional results

The experiments results presented in this dissertation represent a small proportion of the experiments run in search of understanding. This appendix includes some select additional results to support points of discussion.

C.1 Time to hierarchy with changing environmental conditions

Figure C.1 shows the time-to-hierarchy (how many episodes of evolutionary occur before 4 independent hierarchical modules emerge). Figure C.2 shows how this may be related to the choice of Z, the probability of multi-peaked module instance occurring.

C.2 Changing environment model with different cost coefficients

Chapter 3 focusses on the dynamics around the evolution of hierarchy, and considers a choice of cost coefficient λ. These are additional results with different choices of λ.

C.3 Changing environment with a strictly sub-additive cost of connections

Apart from the linear (L1) cost of connections used throughout the dissertation, a 'strictly sub-additive' cost of connections ($\phi(a + b) \leq \phi(a) + \phi(b)$) was also considered, and a spread of experiments were run with the changing environmental model. A grid of the evolved B matrices is presented in Figure C.5.

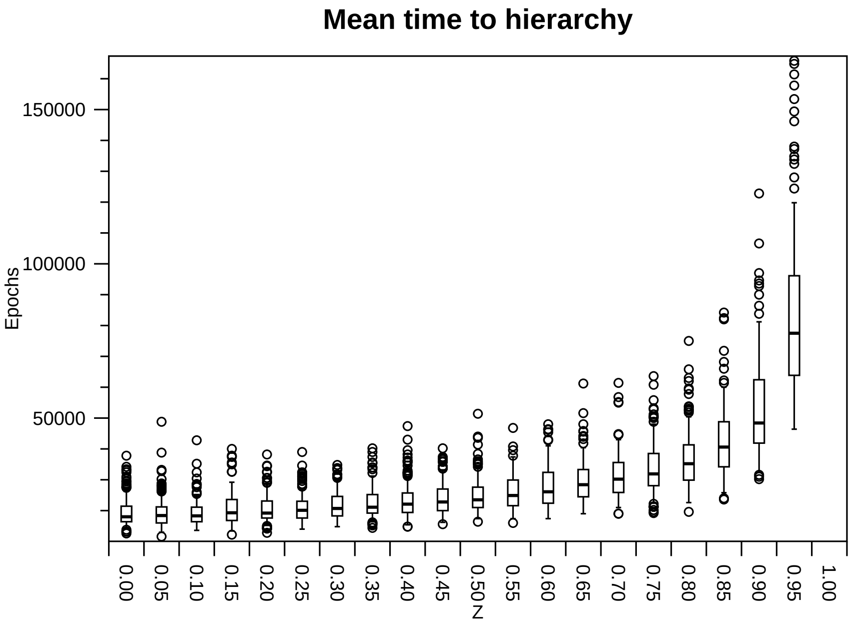


FIGURE C.1: Excepting the results for $Z = 1$ (where a grand-hierarchy evolves), all experiments with the changing environment model showed the evolution of four independent internally hierarchical modules.

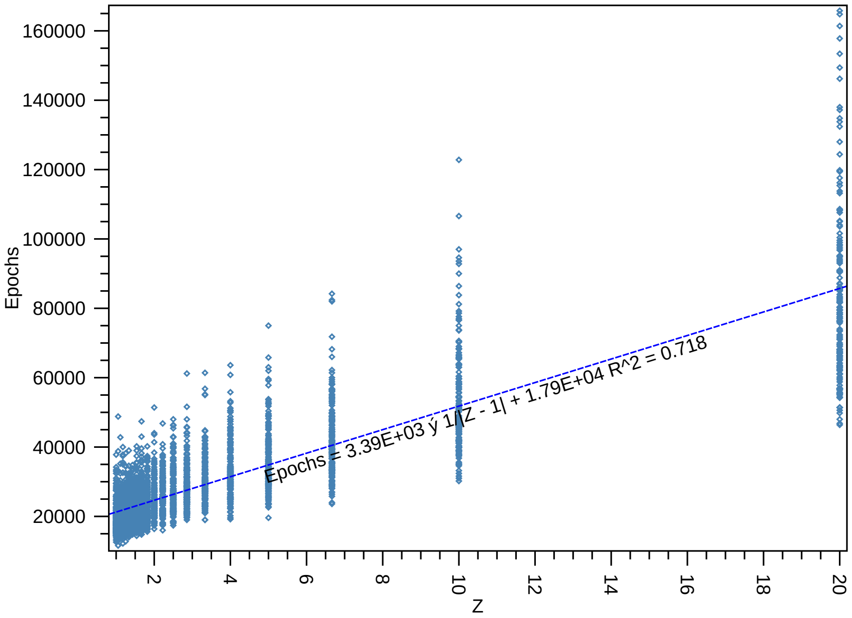


FIGURE C.2: Excepting the results for $Z = 1$, the time to hierarchy appears to be a linear function of $1/|Z - 1|$. Too much should not be read into this regression.

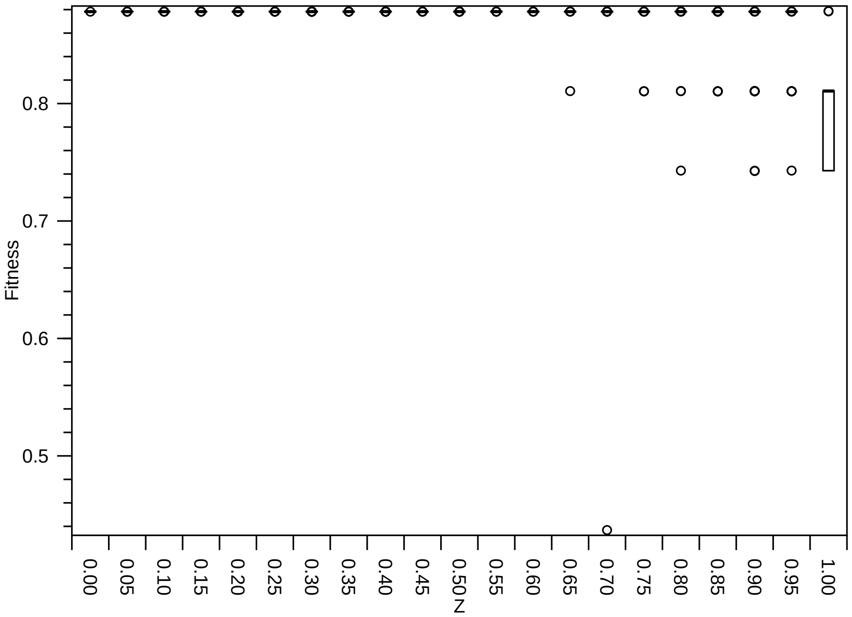


FIGURE C.3: Using a larger cost coefficient of $\lambda = 0.2$ (rather than $\lambda = 0.1$) the distributions of terminal fitnesses reveal that for large choices of multi-peak instance probability Z that the system does not always evolve independent modules, resulting in an inability to respond to full exploit the changing conditions.

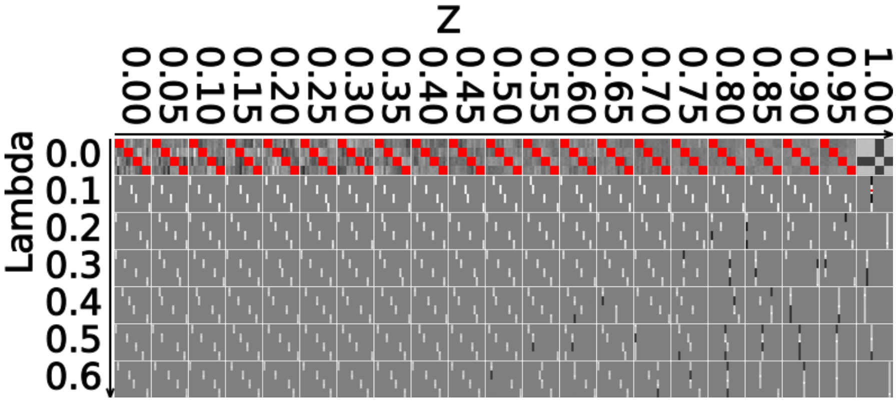


FIGURE C.4: Larger choices of λ and Z increase the probability of failing to evolve independent modules. Note that for $\lambda = 0$ there is no cost of connections, and while independent modules evolve, they do not become hierarchical. The fact that independent modules evolve is surprising as we explain in Chapter 5 that the ability to evolve modules for large values of Z is due to saturation, but saturation itself is due to the cost of connections. These experiments reveal that there is another way to saturate a system: by running out of numbers. This quite literally saturates the squash function in the benefit curve, and has much the same effect as the cost of connections.

The exact function used is given By

$$\phi(|x|) = 0.8|x| + 0.2|x|/(|x|+1) \tag{C.1}$$

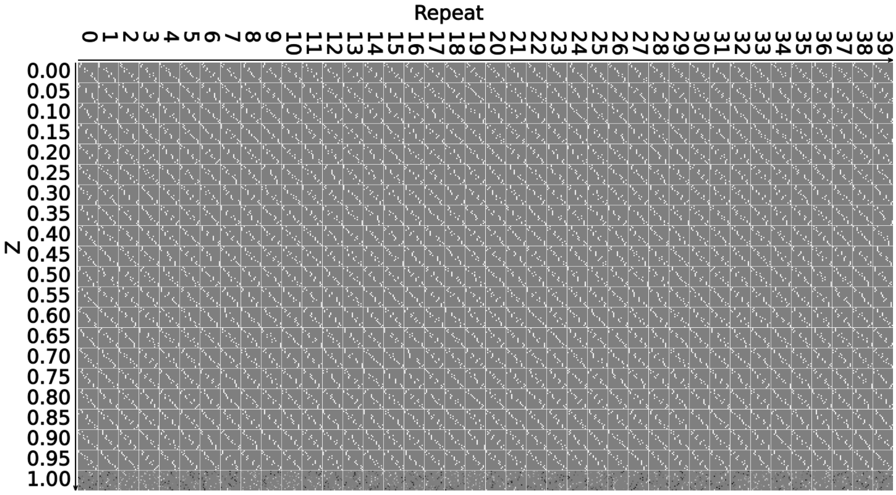


FIGURE C.5: GRNs evolved with a strictly sub-additive cost of connections. Excepting those for $Z = 1$ (where the evolutionary process is ongoing), each comprises *at least* 4 independent and sparse modules: The environmental modules are reflected in these modules, but it many cases the 4 genes that correspond to and environmental module will appear in more than one module (e.g. one gene may only have a self-connection, or a pair may mutually up-regulate each other). Light values present positive regulatory interactions; dark values represent negative regulatory interactions; most interactions have strength zero.

C.4 Fixed environment and initial gene expressions

Figure C.6 shows the trajectories of the regulatory interactions and the degree of hierarchy for a fixed environment and G vector of initial conditions. Experimental parameters are otherwise the same as those discussed in Chapter 3. Because the environment is unchanging and has no epistasis, there is no evolvability benefit to be attained by evolving the hierarchy.

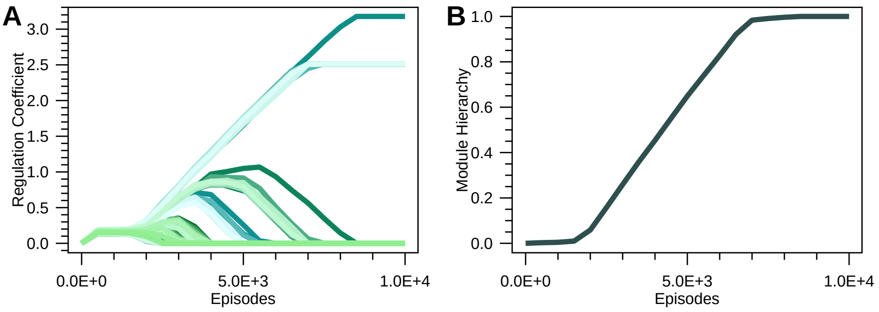


FIGURE C.6: As in the case of multi-peaked environmental instance frequency $Z = 1$, a single grand-hierarchy evolves; there are no independent modules.

C.5 Spiky MC without a cost of connections

Figure C.7 shows the terminal fitness distributions for the evolutionary process without a cost-of-connection on the Spiky MC. The Spiky MC is not readily solved this case, and there is no significant difference between performance for large and small numbers of neutral mutations Q, despite the ample between episode variation introduced for the large choices of Q.

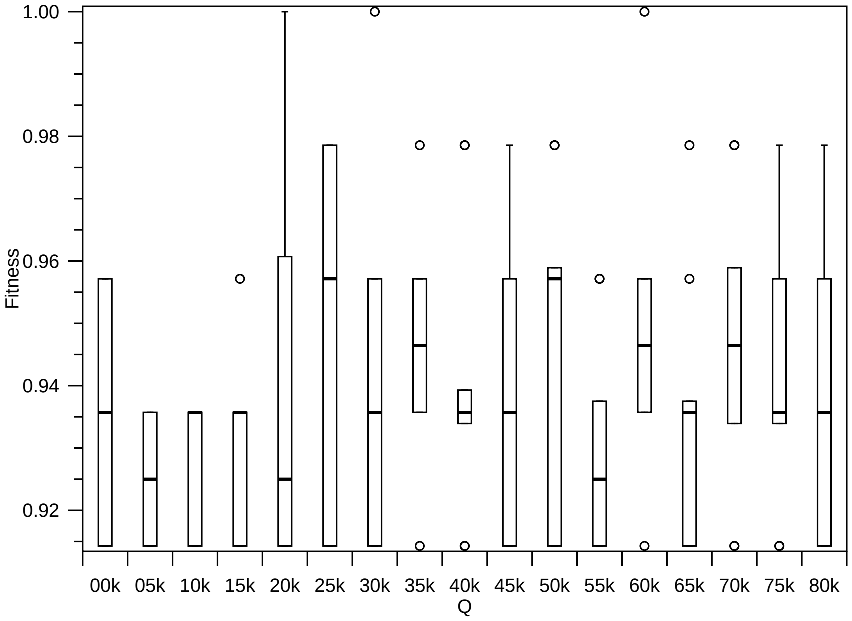


FIGURE C.7: Terminal fitness distributions for the evolutionary process without a cost of connections in the Spiky MC environment. There is no significant improvement in performance for different choices of neutral mutation count Q. In this figure, the fitness is computed on the sign of the phenotype: this reveals the change in distribution without the distraction of the absolute magnitudes of the phenotype.

Appendix D

Gradient descent example

```
// usage: dotnet fsi Script.fsx
#r "nuget: DiffSharp.Backends.Reference, 1.0.7"
open System
open DiffSharp

let step B P tau =
    let u = dsharp.matmul(B, P)
    let s = dsharp.tanh(u / 2.0)
    (1.0 - tau) * P + s

let develop B G T tau =
    let mutable P = G
    for t = 1 to 10 do
        P <- step B P tau
    P * tau

let benefit P =
    dsharp.sum(P) // S+

let cost B =
    dsharp.sum(dsharp.abs(B)) // linear (L1)

let N = 16
let G = dsharp.ones(N).reverseDiff() // G is all +1
let mutable B = dsharp.randn([ N; N ]).reverseDiff()
let lambda = 0.1
let tau = 0.2
let MB = 1e-2 // ’learning rate’
let T = 10

for iter = 1 to 100000 do
    let P = develop B G T tau
    let b = benefit P
    let c = cost B
    let f = b - lambda * c

    f.reverse()
    B <- (B.primal + B.derivative * MB).reverseDiff()

    if iter % 1000 = 0 then printfn $"{iter}\tf: {f} {B}"
```

Appendix E

Partial hierarchies

Figure E.1 shows developmental trajectories for a system of two genes, where gene has self- and outgoing-connections of strength s, and the other gene has connections of strength u. This represents a 'partial' hierarchy with two 'columns'. If $s > u$, then the terminal gene expression only depends on the initial gene expression of the lead gene.

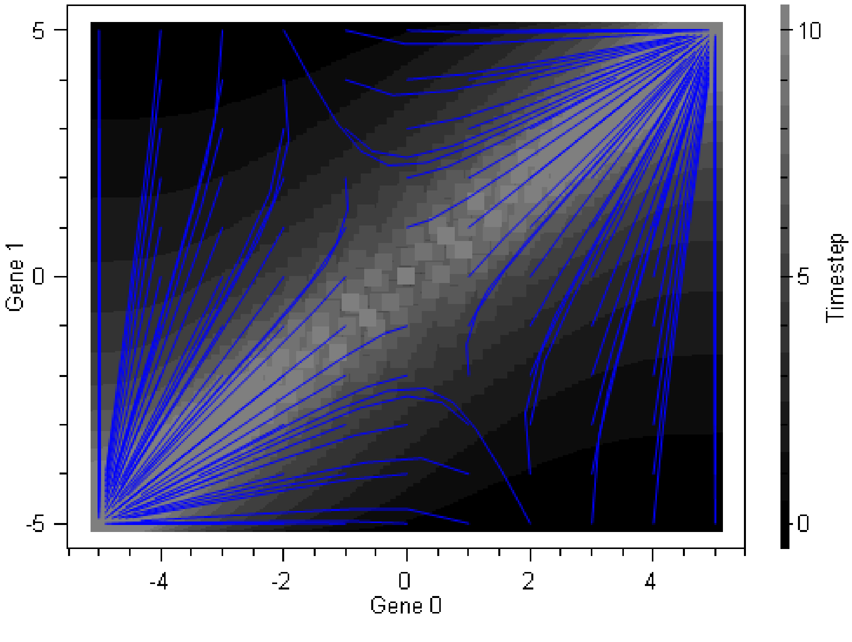


FIGURE E.1: Developmental trajectories for a two-gene module, where gene 0 has influence $s \approx 1.9$ and gene 1 has influence $u \approx 0.8$. Generally, the boundary between the basins of attraction to the two corners is given by $y_0/y_1 = -u/s$.

Appendix F

Sources

The source code and files to generate the key experiments presented in this dissertation can be found at `https://git.soton.ac.uk/fjn1g13/m4mtrim`. The code, being a product of 4 years of frantic incremental development by a single individual, is pretty dense and generally badly designed. If for whatever reason you would like to mess around with this implementation, feel free to contact me. The readmes should be useful.

A simple implementation of the changing environments model used in Chapter 3 can be found at `https://git.soton.ac.uk/fjn1g13/ivmctrim`. This includes the same developmental model as used throughout the dissertation. This is a better place to look if you want to understand the model. It also includes an (older) copy of the main sources, but they can be ignored.

Bibliography

R. L. Berg. The ecological significance of correlation pleiades. *Evolution*, 14(2):171–180, 1960. ISSN 00143820, 15585646.

Rachael L. Brown. What Evolvability Really Is. *The British Journal for the Philosophy of Science*, 65(3):549–572, 08 2013. ISSN 0007-0882.

Sean B. Carroll. Evo-devo and an expanding evolutionary synthesis: A genetic theory of morphological evolution. *Cell*, 134(1):25 – 36, 2008. ISSN 0092-8674.

Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. The evolutionary origins of modularity. *Proceedings: Biological Sciences*, 280(1755):1, 2013. ISSN 09628452.

Michael Conrad. The importance of molecular hierarchy in information processing. *Biological Processes in Living Systems*, 2017.

Anton Crombach and Paulien Hogeweg. Evolution of evolvability in gene regulatory networks. *Plos Computational Biology*, 4(7):e1000112, 2008. ISSN 1553-7358.

Sasha R.X. Dall, John M. McNamara, and Olof Leimar. Genes as cues: phenotypic integration of genetic and epigenetic information from a darwinian perspective. *Trends in Ecology & Evolution*, 30(6):327 – 333, 2015. ISSN 0169-5347.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, April 2002. ISSN 1089-778X.

Jeremy Draghi and Günter P Wagner. Evolution of evolvability in a developmental model. *Evolution*, 62(2):301–315, 2008.

Douglas H Erwin and Eric H Davidson. The evolution of hierarchical gene regulatory networks. *Nature Reviews Genetics*, 10(2):141–148, 2009.

Carlos Espinosa-Soto. On the role of sparseness in the evolution of modularity in gene regulatory networks. *PLOS Computational Biology*, 14(5):1–24, 05 2018.

Allan Force, William A Cresko, F Bryan Pickett, Steven R Proulx, Chris Amemiya, and Michael Lynch. The origin of subfunctions and modular gene regulation. *Genetics*, 170(1):433—446, May 2005. ISSN 0016-6731.

John Gerhart and Marc Kirschner. The theory of facilitated variation. *Proceedings of the National Academy of Sciences*, 104(suppl 1):8582–8589, 2007. ISSN 0027-8424.

John H. Gillespie. Molecular evolution over the mutational landscape. *Evolution*, 38(5): 1116–1129, 1984. ISSN 00143820, 15585646.

D. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning*. 1988.

Jennifer Hallinan. Gene duplication and hierarchical modularity in intracellular interaction networks. *BioSystems*, 74(1):51 – 62, 2004. ISSN 0303-2647.

Thomas F Hansen. Is modularity necessary for evolvability?: Remarks on the relationship between pleiotropy and evolvability. *BioSystems*, 69(2):83 – 94, 2003. ISSN 0303-2647.

Thomas F Hansen. The evolution of genetic architecture. *Annu. Rev. Ecol. Evol. Syst.*, 37:123–157, 2006.

Thomas F Hansen. Why epistasis is important for selection and adaptation. *Evolution*, 67(12):3501–3511, 2013.

Thomas F. Hansen, Christophe Pélabon, and W. Scott Armbruster. Comparing variational properties of homologous floral and vegetative characters in dalechampia scandens: Testing the berg hypothesis. *Evolutionary Biology*, 34(1):86–98, Jun 2007. ISSN 1934-2845.

J.W.K. Ho and M.A. Charleston. Network modelling of gene regulation. *Biophysical Reviews*, 3(1):1–13, 2011. ISSN 18672450.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

David Houle. Comparing evolvability and variability of quantitative traits. *Genetics*, 130(1):195–204, 1992.

David Iclanzan and Dan Dumitrescu. Overcoming hierarchical difficulty by hill-climbing the building block structure. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, pages 1256–1263, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-697-4.

Alba Jiménez, James Cotterell, Andreea Munteanu, and James Sharpe. Dynamics of gene circuits shapes evolvability. *Proceedings of the National Academy of Sciences of the United States of America*, 112(7):2103 – 2108, 2015. ISSN 00278424.

Nadav Kashtan, Uri Alon, and Jr. Curtis G. Callan. Spontaneous evolution of modularity and network motifs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(39):13773, 2005. ISSN 00278424.

Nadav Kashtan, Avi E. Mayo, Tomer Kalisky, and Uri Alon. An analytically solvable model for rapid evolution of modular structure. *PLOS Computational Biology*, 5(4): 1–14, 04 2009.

Nadav Kashtan, Elad Noor, and Uri Alon. Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences*, 104(34):13711–13716, 2007. ISSN 0027-8424.

Marc Kirschner and John Gerhart. Evolvability. *Proceedings of the National Academy of Sciences*, 95(15):8420–8427, 1998. ISSN 0027-8424.

M.W. Kirschner and J.C. Gerhart. *The plausibility of life: Resolving Darwin's Dilemma.* Yale University Press, (1)Department of Systems Biology, Harvard Medical School, 2005. ISBN 9780300108651.

L. Kounios, J. Clune, K. Kouvaris, G. P. Wagner, M. Pavlicev, D. M. Weinreich, and R. A. Watson. Resolving the paradox of evolvability with learning theory: How evolution learns to improve evolvability on rugged fitness landscapes. *ArXiv e-prints*, December 2016.

Kostas Kouvaris, Jeff Clune, Loizos Kounios, Markus Brede, and Richard A. Watson. How evolution learns to generalise: Using the principles of learning theory to understand the evolution of developmental organisation. *PLOS COMPUTATIONAL BIOLOGY*, 13(4), 2017. ISSN 1553734X.

John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4(2):87–112, 1994.

Kevin N. Laland, Tobias Uller, Marcus W. Feldman, Kim Sterelny, Gerd B. Müller, Armin Moczek, Eva Jablonka, and John Odling-Smee. The extended evolutionary synthesis: its structure, assumptions and predictions. *Proceedings of the Royal Society of London B: Biological Sciences*, 282(1813), 2015. ISSN 0962-8452.

Gregory I. Lang, Andrew W. Murray, and David Botstein. The cost of gene expression underlies a fitness trade-off in yeast. *Proceedings of the National Academy of Sciences of the United States of America*, 106(14):5755–5760, 2009. ISSN 00278424.

Hod Lipson. Principles of modularity, regularity, and hierarchy for scalable systems. *Journal of Biological Physics and Chemistry*, 7:125–128, 12 2007.

Hod Lipson, Jordan B Pollack, and Nam P Suh. On the origin of modular variation. *Evolution*, 56(8):1549–1556, 2002.

Michael Lynch. The frailty of adaptive hypotheses for the origins of organismal complexity. *Proceedings of the National Academy of Sciences*, 104(suppl 1):8597–8604, 2007.

Michael Lynch and Georgi K. Marinov. The bioenergetic costs of a gene. *Proceedings of the National Academy of Sciences*, 112(51):15690–15695, 2015. ISSN 0027-8424.

Christine Mayer and Thomas F Hansen. Evolvability and robustness: A paradox restored. *Journal of theoretical biology*, 430:78–85, 2017.

D. Melo, G. Marroig, A. Porto, and J.M. Cheverud. *Modularity: Genes, Development, and Evolution.*, volume 47 of *Annual Review of Ecology, Evolution, and Systematics*. Annual Reviews Inc., Laboratório de Evolução de Mamíferos, Departamento de Genética e Biologia Evolutiva, Instituto de Biociências, Universidade de São Paulo, 2016. ISBN 15452069.

H. Mengistu, J. Huizinga, J. Clune, and J.-B. Mouret. The evolutionary origins of hierarchy. *PLoS Computational Biology*, 12(6):23p., 2016. ISSN 15537358.

Philipp Mitteroecker. The developmental basis of variational modularity: Insights from quantitative genetics, morphometrics, and developmental biology. *Evolutionary Biology*, 36(4):377–385, Dec 2009. ISSN 1934-2845.

Gerd B Müller. Evo–devo: extending the evolutionary synthesis. *Nature reviews genetics*, 8(12):943–949, 2007.

Leonard Nunney. *Levels of Selection in Evolution*, volume 66. Princeton University Press Princeton, 1999.

Merav Parter, Nadav Kashtan, and Uri Alon. Facilitated variation: How evolution learns from past environments to generalize to new environments. *PLOS Computational Biology*, 4(11):1–15, 11 2008.

Mihaela Pavlićev. *Pleiotropy and Its Evolution: Connecting Evo-Devo and Population Genetics*, pages 1087–1096. Springer, Cham, 2016. ISBN 978-3-319-33038-9.

Mihaela Pavlicev, James M. Cheverud, and Günter P. Wagner. Evolution of adaptive phenotypic variation patterns by direct selection for evolvability. *Proceedings of the Royal Society B: Biological Sciences*, 278(1713):1903–1912, 2011.

Joshua L Payne, Jason H Moore, and Andreas Wagner. Robustness, evolvability, and the logic of genetic regulation. *Artificial life*, 20(1):111–126, 2014.

Joshua L Payne and Andreas Wagner. The causes of evolvability and their evolution. *Nature Reviews. Genetics*, 20:24–38, 2019. ISSN 1471-0064.

Massimo Pigliucci. Is evolvability evolvable? *Nature Reviews Genetics*, 9:75–82, 2008.

Daniel A Power, Richard A Watson, Eörs Szathmáry, Rob Mills, Simon T Powers, C Patrick Doncaster, and BłaŻej Czapp. What can ecosystems learn? expanding evolutionary ecology with learning theory. *Biology direct*, 10(1):69, 2015.

Rupert Riedl. A systems-analytical approach to macro-evolutionary phenomena. *The Quarterly Review of Biology*, 52(4):351–370, 1977. PMID: 343152.

Estelle Rünneburger and Arnaud Le Rouzic. Why and how genetic canalization evolves in gene regulatory networks. *BMC Evolutionary Biology*, 16(1):239, Nov 2016. ISSN 1471-2148.

Mark L. Siegal and Aviv Bergman. Waddington's canalization revisited: Developmental stability and evolution. *Proceedings of the National Academy of Sciences*, 99(16): 10528–10532, 2002. ISSN 0027-8424.

Herbert A. Simon. *The Sciences of the Artificial.* MIT Press, Cambridge, MA, 1969.

J Maynard Smith, Richard Burian, Stuart Kauffman, Pere Alberch, John Campbell, Brian Goodwin, Russell Lande, David Raup, and Lewis Wolpert. Developmental constraints and evolution: a perspective from the mountain lake conference on development and evolution. *The Quarterly Review of Biology*, 60(3):265–287, 1985.

Alexander Spirov and David Holloway. Using evolutionary computations to understand the design and evolution of gene and cell regulatory networks. *Methods*, 62(1):39–55, 2013. ISSN 1046-2023. Modeling Gene Expression.

Stephen C Stearns. Life history evolution: successes, limitations, and prospects. *Naturwissenschaften*, 87(11):476–486, 2000.

Arno Steinacher, Declan G. Bates, Ozgur E. Akman, and Orkun S. Soyer. Nonlinear dynamics in gene regulation promote robustness and evolvability of gene expression levels. *PLOS ONE*, 11(4):1–21, 04 2016.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246.

Nathaniel Virgo, Eran Agmon, and Chrisantha Fernando. Lineage selection leads to evolvability at large population sizes. *Proceedings of the European Conference on Artificial Life*, pages 420–427, 2017.

Jiri Vohradsky. Neural model of the genetic network. *Journal of Biological Chemistry*, 276(39):36168–36173, 2001.

C. H. Waddington and E. Robertson. Selection for developmental canalisation. *Genetical Research*, 7(3):303–312, 1966.

A Wagner. Evolution of gene networks by gene duplications: a mathematical model and its implications on genome organization. *Proceedings of the National Academy of Sciences*, 91(10):4387–4391, 1994. ISSN 0027-8424.

Andreas Wagner. Robustness, evolvability, and neutrality. *FEBS letters*, 579(8):1772–1778, 2005.

Andreas Wagner. Robustness and evolvability: a paradox resolved. *Proceedings of the Royal Society of London B: Biological Sciences*, 275(1630):91–100, 2008. ISSN 0962-8452.

G. P. Wagner and J. Draghi. Evolution of evolvability, 2010.

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

Gunter Wagner, Mihaela Pavlicev, and James Cheverud. The road to modularity. *Nature reviews Genetics*, 8:921–931, 01 2008.

Günter P Wagner. Homologues, natural kinds and the evolution of modularity. *American Zoologist*, 36(1):36–43, 1996.

R. A. Watson and J. B. Pollack. Hierarchically consistent test problems for genetic algorithms. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 2, pages 1406–1413 Vol. 2, 1999.

R.A. Watson, K. Kouvaris, A. Jackson, C. Cox, S. Tudge, A. Davies, L. Kounios, D. Power, R. Mills, C.L. Buckley, and S.T. Powers. Evolutionary connectionism: Algorithmic principles underlying the evolution of biological organisation in evo-devo, evo-eco and evolutionary transitions. *Evolutionary Biology*, 43(4):553–581, 2016. ISSN 00713260.

RA Watson, GP Wagner, M Pavlicev, DM Weinreich, and R Mills. The evolution of phenotypic correlations and "developmental memory". *EVOLUTION*, 68(4):1124 – 1138, 2014. ISSN 00143820.

Richard A. Watson. *Evolvability*, pages 1–16. Springer International Publishing, Cham, 2020. ISBN 978-3-319-33038-9.

Richard A Watson, Christopher L Buckley, and Rob Mills. Optimization in "self-modeling" complex adaptive systems. *Complexity*, 16(5):17–26, 2011a.

Richard A. Watson, Rob Mills, and C. L. Buckley. Transformations in the scale of behaviour and the global optimisation of constraints in adaptive networks. *Adaptive Behavior*, 19(4):227–249, May 2011b.

Richard A Watson, Rob Mills, and Christopher L Buckley. Transformations in the scale of behavior and the global optimization of constraints in adaptive networks. *Adaptive Behavior*, 19(4):227–249, 2011c.

Richard A. Watson and Eörs Szathmáry. Opinion: How can evolution learn?. *Trends in Ecology & Evolution*, 31:147 – 157, 2016. ISSN 0169-5347.

Mary Jane West-Eberhard. Modularity as a universal emergent property of biological traits. *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution*, 332(8):356–364, 2019.

Bryan Wilder and Kenneth Stanley. Reconciling explanations for the evolution of evolvability. *Adaptive Behavior*, 23(3):171 – 179, 2015. ISSN 10597123.

Haiyuan Yu and Mark Gerstein. Genomic analysis of the hierarchical structure of regulatory networks. *Proceedings of the National Academy of Sciences*, 103(40):14724–14731, 2006.

Tian-Li Yu and David E. Goldberg. Conquering hierarchical difficulty by explicit chunking substructural chromosome compression. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, page 1385"1392, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595931864.