Master Thesis

How Genomes are Shaped by Direct and Indirect Selection Pressure: A Study in in silico Experimental Evolution

Brian Davis

Supervisors:

Bérénice Batut and Anika Erxleben

May 2020

Albert-Ludwigs Universität Freiburg

Bioinformatics Group

Professor Dr. Rolf Backofen

Abstract

Not all aspects of evolution are fully understood, and one area of pressing interest is reductive evolution, in which the genome of a species evolves to become significantly smaller over time. Among the more well-known examples of this phenomena is *Prochlorococcus*, a marine cyanobacteria whose genome has reduced up to 43% among some of its own ecotypes and around 36% smaller than some strains of its closest living relative, *Synechococcus*. To study the mechanisms behind reductive evolution in the lab would be too difficult, expensive, and above all, slow, so instead *in silico* evolution may be used to study this phenomenon. This method seeks to simulate organisms and their evolution in software, allowing for greater control of the environment, mutation rates, and other variables, as well as providing a full phylogenetic record of all organisms in a lineage. In this thesis the in silico tool Aevol is used to study reductive evolution, particularly by looking at how varying parameters (e.g. mutation and selection rates, population size) impacts the structure of the genome as well as measures such as robustness, evolvability, and fitness. Through these methods, a little light is shed on some of the underlying mechanisms which might lead to reductive evolution.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One way of speaking about evolution is with regard to the change in frequencies of alleles (variants of genes) in a population over time. In population genetics, the *Hardy-Weinberg principle* states that genetic variation (diversity of gene frequencies) in a population will remain constant from generation to generation in the absence of disturbances. There are five "Hardy-Weinberg assumptions" which must hold for this to be true: no mutation, random mating, no gene flow (i.e. the transfer of genetic material from one population to another), infinite population sizes, and no selection [8]. If any of these assumptions does not hold, then this may change the frequencies of alleles in a population, which allows evolution to occur. The mechanisms for evolution are, then, the opposite of these assumptions, namely: mutation, non-random mating, gene flow, finite populations, and natural selection. Pulling at any of these levers has an effect on the genetic variation of a population and thus impacts their genome.

Reductive evolution is the process of the average genome size of a species shrinking over time, with respect to both the number of base pairs and genes. For example, some species of bacteria have experienced reductive evolution over the course of millions of years, and this reduction in their genome has lead to a loss of genes, certain regulatory abilities, and more. One such bacterial species is the marine cyanobacteria *Prochlorococcus*, some of whose strains have experienced a reduction of nearly 40% of their base pairs when compared to larger strains of their closest living relative, *Synechococcus*. Even among strains of Prochlorococcus, the difference between some of the

larger and smaller strains is upwards of 38% [2]. Despite being extensively studied, the underlying mechanisms and full impact of reductive evolution are not fully understood and are an area of intense current research. Several competing explanations exist for these mechanisms, from the influence of population size, genetic drift (defined below), or selection [2].

Although it would provide more conclusive evidence, performing *in vivo* experiments with living organisms is often impractical because of the difficulty or impossibility of reproducing natural environmental conditions in a laboratory. Such experiments are often too costly in terms of both time and resources. As an alternative, *in silico* experimental evolution is one option that can be used to study the conditions under which an organism's genome may become reduced. In this method, organisms and their evolution over thousands or millions of generations are simulated in software. In this manner, one can control and evaluate every aspect of their evolution over time and a full record of their lineage may be maintained and studied, allowing one to go back and closely examine every step of the evolutionary period for a greater understanding of the factors that lead to specific effects on the genome. The in silico tool *Aevol* is one such platform which realistically models bacterial genomes and evolution, allowing one to draw conclusions about their real-world counterparts. In the following thesis, the results of experiments in artificial evolution are presented which aim to identify and evaluate several factors which potentially lead to changes in genome structure and a reduced genome in simulated bacteria using the Aevol platform.

Problem Statement

Among the difficulties of studying reductive evolution with in vivo evolutionary experiments, one of the most difficult obstacles to overcome is the lack of a full ancestral record. This lack of a full phylogeny can make it difficult or impossible to tell exactly when and how a specific event occurred, or a trait evolved or was lost, as illustrated in Figure 1.1 below. In this example, two related organisms A and B are compared with an attempt to determine when and how a specific trait was gained or lost by one of the organisms. This may be useful, for example, when attempting to estimate the relative importance (due to conservation over many generations) of some

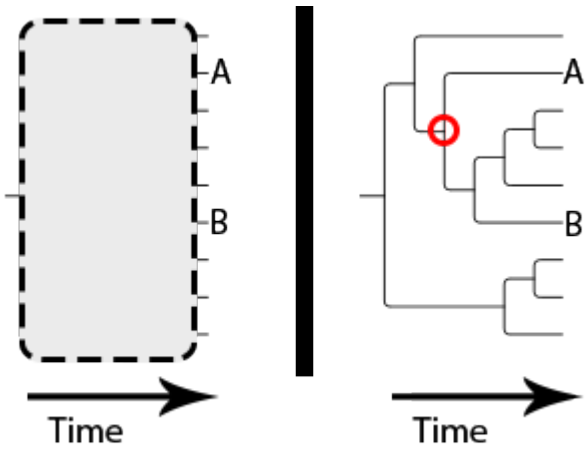

Figure 1.1: An illustration of unknown phylogeny. Since the phylogenetic information under the shaded box is typically not known, the point of divergence (red circle) can't be determined.

trait. Without the phylogenetic information (under the shaded box) it may not be possible to identify the point in their evolutionary history at which the two organisms diverged, making time estimates difficult or impossible.

Another major downside to in vivo evolutionary experiments is that they are slow. For example, the well-known E. coli Long-Term Evolution Experiment (LTEE) by Profesor Lenski at Michigan State University has been ongoing since February of 1988 and only passed generation 50,000 in 2010, 22 years later[1].

As an alternative to in vivo experiments, in silico evolutionary experiments are well-suited to the task of studying reductive evolution. Generations of organisms may be evolved within a very short time period, and a full "fossil record" of each lineage may be kept on disk for further analysis.

The in silico tool Aevol has a realistic artificial chemistry model which was developed specifically to study genome structure. It contains tools to analyze the robustness, fitness, and evolvability of digital organisms over time. In this thesis, Aevol is used to perform experiments in silico evolution in order to determine the effects of several different conditions on the genome of a simulated "wild type" bacteria.

[1] `http://myxo.css.msu.edu/ecoli/celebrate50K.html`

1.0.1 Genetic Variability

Of fundamental importance to this thesis is the question of how genome size evolves. Several models have been proposed over the years, and two which have proven to be particularly useful are the Moran and Lynch models. These two models relate to the question of genetic variability, i.e. the genetic differences

Michael Lynch, the famous geneticist currently at Arizona State University has a model for determining genetic variabiltiy which is still commonly in use now and is known as the Lynch model. In this model,

Report outline

This chapter serves as the introduction to the thesis and the research problem being faced. In Chapter 2, some necessary background information is provided on reductive evolution, in silico evolution in general, and Aevol in particular. Chapter 3 describes the experimental setup. Chapter 4 provides the results and analysis of the experiments of this thesis, and Chapter 5 presents the conclusions drawn from this work.

Chapter 2

Background

In this chapter, some of the theoretical background information required for this thesis is examined. An overview of experimental evolution is given before moving on to a discussion of Aevol, the specific tool that was used in this thesis. The chapter closes examining how Aevol can be used to study reductive evolution and by discussing the current state of the literature surrounding reductive evolution.

2.1 Experimental Evolution

As discussed in Section 1, in vivo experiments, although sometimes more realistic, have their own set of difficulties. Some examples of these difficulties include recreating challenging environmental conditions (e.g. simulating the open ocean in a laboratory) and identifying and/or simulating the multiple selection pressures acting on genomes in the real world [3]. These challenges add enormous difficulty and complexity to conducting proper experiments and isolating the specific factors which lead to particular outcomes.

In silico evolution simulates the evolution of organisms in software, thus allowing for far greater control and analysis of the environment and other experimental conditions. In contrast to in vivo experiments, a greater amount of control is also available with regard to the way organisms may interact, reproduce, and evolve. For example, a genome may be created completely from scratch or an existing genome may be fed into the simulation. Reproduction rates can depend on overall fitness, on relative fitness, or some other

criterion.

Factors such as the mutation rates or selection pressure are then parameters for the model and may be kept constant or allowed to vary over time. Given that these are parameters of the system, they may be tightly controlled, leading to a clearer picture of the factors influencing different outcomes. An underlying deterministic model can also allow for a reconstruction of the system from any given point, allowing one to easily create a record of events, including phylogenetic trees.

Although there are certainly limitations to using in silico evolution (see Section 5.3), overall in silico experimental evolution is a valuable tool for studying the underlying mechanisms of evolution when real-world experiments would be prohibitively time-consuming or expensive.

2.2 Aevol

Aevol follows a "sequence-of-nucleotides" model [3] in which organisms are simulated with a binary genome which can either be generated at random or input as a previously-generated sequence. Aevol essentially consists of three steps: 1) decoding the genome of these organisms to produce artificial proteins, 2) selecting the most fit individuals and 3) reproduction of these fittest individuals with possible variations (mutations, rearrangements, etc.). The sections below examine each of these steps in greater detail.

2.2.1 Why use Aevol

Many silico evolution tools have been used to test various aspects of evolution: Tierra [23] and Avida [21], in which the genetic units are computer programs fighting for CPU time, were some of the first; DOSE, an ecology-conscious method of checking for heterozygosity (variation within a population) [7] is a more recent example; and many more (see [20] for a more comprehensive review).

The in silico tool Aevol was developed to "study the evolution of the size and organization of bacterial genomes in various scenarios"[3]. The program has also been expanded upon and tested in a variety of scenarios over the years (e.g. [22], [19], [18]). Some examples of experiments in which Aevol was the primary mechanism of in silico experimental evolution include: test-

ing the predictability of evolution with high mutation rates as in viruses [4], determining whether selection is able to overcome evolution's drive towards more complex organisms [16], examining the role of mutators in reorganizing the genome in order to overcome mutational load [24], examining the effects of population shape on levels of cooperation [18], modeling regulatory networks [25] and more.

As an in silico experimental evolution tool, Aevol embodies several of the advantages of in silico evolution in general. There is a "fossil record" of each generation, experimental conditions are tightly controlled, and experiments are easily repeatable. The encoding/decoding strategy of Aevol follows a biologically realistic model, in the sense that there are many degrees of freedom between an organism's genome and its proteome. Many genes may encode for very simple proteins (e.g. if the genes contain the same or similar sequences), and by contrast, overlapping genes may code for complex proteomes.

In the following sections, the in silico experimental evolution tool Aevol will be examined in greater detail.

2.2.2 Aevol's Architecture

Aevol's three main steps—decoding the genome, selection, and reproduction—are illustrated in Figure 2.1 and are discussed in greater detail in the following sections.

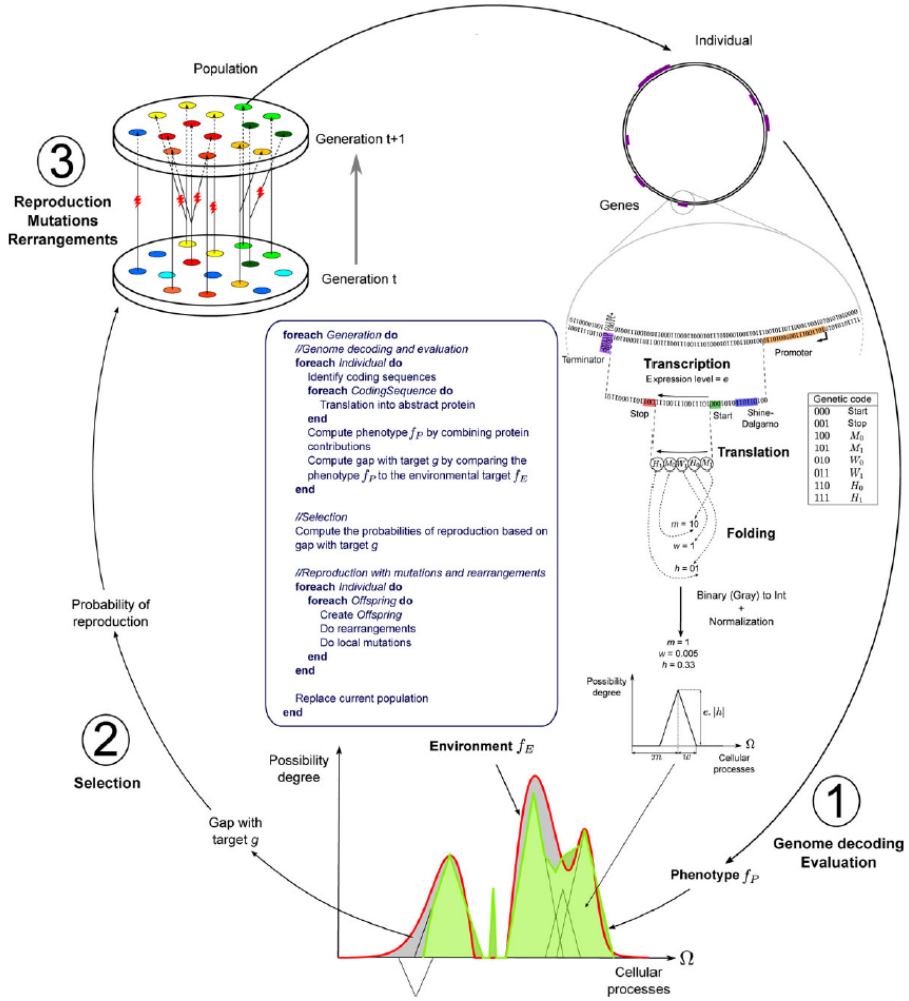

Figure 2.1: Overview of Aevol's architecture, from [3]. In step 1, the binary genome of the organisms is decoded and the resulting phenotype (bottom, in green) is matched up against an environmental target function. In step 2, the probability of reproduction is calculated for each organism, based on its fitness (gap between the environment f_E and its phenotype). In step 3, the next generation is created based on the probabilities from step 2, along with random mutations, rearrangements, etc.

Decoding the Genome

In Aevol, a genome consists of a string of binary characters where 0 is complementary to 1. Each organism in the population has a double-stranded

circular genome which is either generated randomly or which was provided as input. To decode the genome and produce the phenotype, the sequence is searched for transcribed regions. Transcribed regions are denoted by promoter and terminator sequences. The promoter sequence is a sequence whose Hamming distance d is within $d_{max} = 4$ mismatches of the predefined consensus sequence 01010110011110010010110. Terminators are sequences which can form a stem-loop structure with a stem size of 4 bases and a loop length of 3 bases (i.e. $abcd***\overline{dcba}$ where a is complementary to $\overline{a}$, b is complementary to $\overline{b}$, etc.). Lastly, the initiation and termination signals are sought, which are simply Shine-Dalgarno-like signals (i.e. $011011****000$ to start and $011011****001$ to stop). Lastly, an expression level e is assigned to each coding region, following the formula $e = 1 - \frac{d}{d_{max}+1}$ where d is again the Hamming distance between the coding region and the consensus sequence given above and d_{max} is the maximum allowable distance (i.e. 4).

Once an initiation sequence is found, the following bases are read three at a time (codon by codon) until a stop codon (by default 001) is found. If a stop codon is not found, then no protein is produced. Since a transcribed region may have multiple initiation signals, operons are therefore allowed. The codons following an initiation signal encode for three parameters according to the genetic code given in Figure 2.1: m (mean), w (half-width), and h (height), which together define a triangle representing a "cellular process".

A cellular process is simply an abstract representation of some phenotypic function and is represented by the ordered set $\Omega = [a, b] \subset \mathbb{R}$. Together, these cellular processes make up the organism's proteome. To keep things simple, Ω is a one-dimensional space in the interval $[0, 1]$, i.e. a "cellular process" is simply a real number, and the genomic encoding of each cellular process determines the function $f(x) : \Omega \to [0, 1]$. The mean m gives us the specific cellular process in the range $[0, 1]$. The width w describes the "scope" of the process, i.e. the *pleiotropy* of the process, meaning the subset of the protein that is in the interval $(m - w, m + w) \subset \Omega$. The height determines the degree of possibility of the process, i.e. its relative strength.

The codons are read one after the other and their Gray codes[1] are used to compute the real numbers m, w, and h as follows. Each parameter (m, w,

[1] A binary encoding such that two successive values (e.g. 2, 3) only differ by at most one bit (e.g. 0011, 0010). See [9] for an overview.

h) is assigned two codons in the genetic code (see Figure 2.1), for example $w_0 = 010$ and $w_1 = 011$. Any w_0 codons become a 0 in the Gray code, and vice versa with 1s. So if, for example, when reading the coding sequence, the codons w_1, w_0, w_1, w_0 are read, the Gray code becomes 1010, which is 12 in decimal. This is done for m, w, and h, and the resulting values are then normalized to be in the proper range. w's range is specified in the parameter file (as `MAX_TRIANGLE_WIDTH`), h must be in the range $[-1, +1]$ (indicating that both activating and inhibiting processes are allowed) and m must be in the range $[0, 1]$ (the range of Ω).

Given the fact that there are likely multiple coding sequences in a genome, several triangles (cellular processes) are translated from the genome, each parameterized by its own m, w, and h. These triangles form the phenotypic function f_P. *Fuzzy logic* is used to find the overall contribution of each cellular process, using the Lukasiewicz fuzzy operators[2]. Roughly speaking, the activating proteins are added up, as are the inhibiting proteins, and the difference between these two totals represents the final function f_P. More formally, if f_i is the possibility distribution of the i-th activator protein and f_j is the possibility distribution of the j-th inhibitor protein, then the phenotype of the individual is defined as:

$$f_P = max\left(min\left(\sum_i f_i(x), 1\right) - min\left(\sum_j f_j(x), 1\right), 0\right)$$

Selection

After the genome is decoded, the organisms are tested for their fitness. Fitness in Aevol is related to the gap between the phenotype of a sequence f_P and the environmental target function f_E, as illustrated in Figure 2.1. This environmental target function f_E is a user-defined set of Gaussians which are specified in a parameter file, with each Gaussian being identified by three parameters: its height, its location along the axis, and its width. The difference between the phenotype (as calculated above) and the environmental function is the "metabolic error", labeled g in the figure and is more formally defined as: $g = \int_a^b f_E(x) - f_P(x)dx$. The idea is illustrated in Figure 2.2

[2]See `https://en.wikipedia.org/wiki/Lukasiewicz_logic` for an introduction.

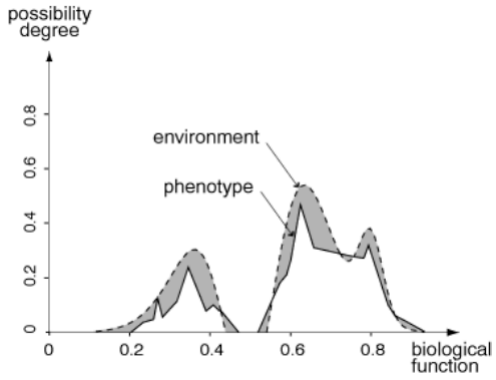

Figure 2.2: Overview of Aevol's conception of fitness, from [13]

Aevol contains several selection schemes, but here only the `fitness_proportionate` scheme will be considered, since this was the only selection scheme employed in these experiments. In this scheme, the probability of reproduction for each organism i is proportionate to its fitness, namely:

$$P(\text{reproduction}) = \frac{e^{-k*g}}{\sum_{i=1}^{N} e^{-k*g_i}}$$

where k is a user-definable parameter which determines the selection intensity and g is the metabolic error.

Reproduction

Once the fittest organisms in the population are found and their probabilities of reproducing are calculated as described in the previous Section, new organisms are produced. This is done for each potential parent organism by drawing from a multinomial distribution with the probability of reproduction given above. The population size N is kept constant and a record of each generation is kept so that the phylogenetic lineages can be recreated. Since the population size is held constant, this implies that a single organism with a high probability of reproduction may produce multiple offspring and an organism with low probability of reproduction may produce none.

When new organisms are created and their genomes are copied from their parent organisms, it is at this stage that some of the driving forces in evolution occur, namely the possibility for variation through mutation, indels, and frameshifts. Offspring will receive their parents' genome but their

genome may be subject to perturbations due to stochastic effects. Mutation rates are set in the parameter file and include point mutations, insertions and deletions (indels), and rearrangements (duplication, deletions, translocations, and inversions).

The mutation, indels, rearrangement, etc. events are carried out by first determining the number μ of such events which will occur, based on the mutation rate specified in the parameter file and drawing from a binomial distribution (e.g. $B(L, \mu_{\text{point}})$ for point mutations, $B(L, \mu_{\text{large deletions}})$ for large deletions, etc. where L is the size of the genome). Then a random point (or points, in the case of e.g. rearrangement) is chosen and the event is carried out, with the order of these events shuffled randomly.

2.3 Analyzing with Aevol

Once the experiments have completed, Aevol by default produces several statistics files which include information about genome size, the percentage of coding DNA, number of genes, average metabolic error, and many other statistics. It further includes a number of post-treatments that allow one to analyze specific individuals or the population at large, including tools for determining robustness, evolvability, coalescence, and the lineages.

One of the key features of Aevol is the ability to look back in time at the "archaeological record" of previous organisms, which is stored on disk, in order to perform various analyses. This is primarily done with a myriad of post-treatments", i.e. supplementary programs. These post-treatments generally require a `lineage` file, which is a file generated by Aevol that shows a record back in time of the line of descent for an individual. One may specify either the best-ranked individual (i.e. the fittest) or a specific individual by their unique identification number. The basic idea is illustrated below in Figure 2.3.

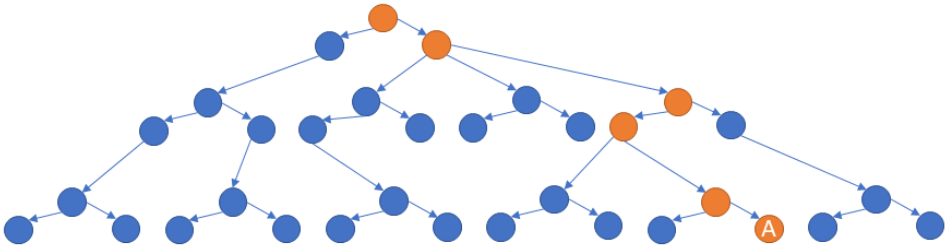

Figure 2.3: A basic illustration of a lineage. The ancestors of the individual (labeled 'A') can be traced back through the previous generations for all of its ancestors (all in orange).

In the following subsections, other statistics that Aevol calculates are examined, as these factors play a major role in reductive evolution.

2.3.1 Evolvability

Evolvability is usually defined as the ability of a system (in this case an organism) to evolve. In other words, a system has evolvability "if mutations in it can produce heritable phenotypic variation"[28]. However, of critical importance is that this is not simply having a large amount of genetic diversity, but rather *adaptive* diversity which provides some benefit.

2.3.2 Robustness and Antirobustness

Robustness is broadly defined as the ability of an organism to withstand disruptions or perturbations without affecting its phenotype. There are differing kinds of robustness, as well: one may describe robustness in terms of mutational robustness–which describes the extent to which an organism's phenotype is not affected by stochastic mutational events–or one may speak of environmental robustness, which describes the ability of an organism to maintain its phenotype in diverse environments with little or no loss in fitness.

Also important is the idea of antirobustness, wherein an organism may actually *thrive* on such perturbations. This has been suggested as a possible method of minimizing the effects of *Muller's ratchet*, wherein deleterious mutations accumulate in a population as a result of genetic drift[12]. A

large number of deleterious mutations may provide fodder for selection to fix more beneficial mutations in the population[24].

There is, then, a seeming trade-off between robustness and evolvability. The more robust a system is, the less phenotypic variation is generated by random mutation events, and thus less evolvability. However, a key factor is to distinguish between genomic and phenotypic robustness; a strong phenotypic robustness promotes structural evolvability, as the likelihood that a mutation is deleterious is smaller in populations with more robust phenotypes. For a fuller discussion, see [28].

2.3.3 Fitness

In population genetics, "fitness" can relate to either the genotype or the phenotype and denotes the contribution of an individual to the gene pool of the next generation; it is usually discussed in terms of *absolute fitness* and *relative fitness* [8]. The absolute fitness W of a genotype is the level of proportional change in the abundance of that genotype from one generation to the next. So if, for example, a genotype in generation t exists with abundance $n(t)$, then $n(t+1) = W * n(t)$.

Relative fitness, on the other hand, describes the change in genotype *frequency*, i.e. the fraction of all chromosomes in the population that carry a particular allele. For a population of size N, if a genotype's frequency at time t is given by $p(t) = n(t)/N(t)$, then $p(t+1) = \frac{w}{\bar{w}} * p(t)$, where $\bar{w}$ is the mean relative fitness in the population. This implies that $\frac{w}{\bar{w}}$ is proportional to $\frac{W}{W}$.

2.4 Related Work

Much work has already been done in the field of reductive evolution, and this section will look at the current state of the literature.

Liard et al. showed in [16] that selection for fitness is not necessarily enough to overcome the tendency of organisms to become more and more complex. They describe the problem of a "complexity ratchet," that is, that the tendency of organisms becoming more and more complex as irreversible once the organisms had reached a certain complexity level. In their words:

> "Since gene deletion is obviously deleterious [in this scenario], the only available evolutionary path for already complex organisms is a headlong rush toward increasing complexity by acquiring new genes. Hence the ratchet clicks, further widening the fitness valley that separates the current genome from a simple one, soon making it so wide it is very unlikely to be crossed."

Echoing the findings of Knibbe et al. [14] they found, however, that limiting *robustness* can overcome the tendency of organisms to become more and more complex, because this places an upper bound on the amount of information that an organism can transmit in its genome. Increasing the mutation rate forced gene elimination despite the fitness loss because it lowered the information content of the genome. A mutation rate of even $\mu = 10^{-4}$ resulted in nearly 40% of their organisms developing a simpler genome.

Knibbe et al. also found, via in silico experimentation, that the accumulation of non-coding DNA strongly depends on the mutation rate. This in turn affects the selection tradeoff between reliably passing on the existing genome and having the mutational variability to adapt to new challenges. Under higher mutation rates, their organisms closely resembled viral genomes in that they had almost no non-coding sequences. When the selection strength was larger, genomes were larger.

Koskiniemi et al. [15] performed in vivo experiments on the bacterium *Salmonella enterica* in which the effects on fitness of random deletions was measured. Some 25% of the deletions actually caused an increase in fitness under some conditions, suggesting that there is a certain cost associated with having superfluous genes and thus gene loss may be selected for under certain conditions.

Batut et al. [2] found that in organisms with smaller effective population sizes, such as endosymbiotic bacteria like *Buchnera aphidicola*, gene loss may occur as a result of the smaller population sizes effectively clicking down on "Muller's Ratchet". In this situation, deleterious mutations accumulate in a population because the smaller population size does not allow for enough variability, so deleterious mutations become fixed in a population.

Liard et al. showed in their experiments with Aevol that complexity can arise in organisms in even the most simple of environments (a single triangular target function) and with the most basic beginnings (a single gene

and protein) [16]. They found that the complexity continues to increase, even when the resulting organisms are much less fit than the simpler ones, and that this complexity even overcomes selection. They also found that whether an organism would develop into a simple or complex organism was usually determined quite early in its evolution, around generation 10,000. Reasoning that genome compactness is a direct driver of mutational robustness, as Knibbe et al. [14] showed, and that more robust organisms can be selected over fitter organisms under heavy mutational stress as shown by Wilke et al. [30], they wondered if robustness could select for a more simplified genome. To test this, they changed the mutation rate to be exceptionally high (up to $\mu = 1e^{-4} * bp^{-1} * gen^{-1}$) and ran their simulations for another 100,000 generations, and the results confirmed their expectations: The higher the differential between the old and the new mutation rate, the larger percentage of organisms evolved a more reduced genome (up to 91% in the case of a change from $\mu_{\text{old}} = 1e^{-6}$ to $\mu_{\text{new}} = 1e^{-3}$).

Chapter 3

Methods

With an understanding of the basics from Chapter 2, in this chapter an overview of the contributions of this thesis and an outline of the methods used in the experiments are presented.

3.1 Contributions

3.2 Experimental Designs

To assist in determining which conditions might lead to reductive evolution, one must isolate individual variables and change just one thing at a time in order to see what effect, if any, it has on the final genome. To that end, a series of experiments were designed and conducted in which a "wild type" genome was allowed to evolve in differing conditions for 500,000 generations before analyzing the effects. To first create the wild type, a genome was generated randomly in Aevol which had at least one coding gene and which was allowed to evolve for 10 million generations in a non-varying environment. By the beginning of its use in these experiments, the wild type comprised 13,237 base pairs with 132 functional genes (i.e. genes which produce a gene product) and around 37% non-coding DNA.

In these experiments, separate (initially clonal) populations of the wild types were created, one for each condition (including one for the controls), which were then allowed to continue to evolve in a static environment in a total of 6 different conditions: with an increased/decreased selection

strength (k_+/k_-), an increased/decreased mutation rate (μ_+/μ_-), and an increased/decreased population size (N_+/N_-).

For all experiments here, the original target environmental function in which the wild types were generated was also used. This environmental target function, f_E, is comprised of three Gaussian functions $y_(x)$, each of which is characterized by a height h, mean m, and width w:

$$y(x) = h * e^{\left(\frac{-(x-m)^2}{2*w^2}\right)}$$

A visualization of the target function used in the experiments, f_E, can be seen below in Figure 3.1. Each Gaussian's parameters h, m, and w are given in the figure. The gray shaded area is the phenotypic target area.

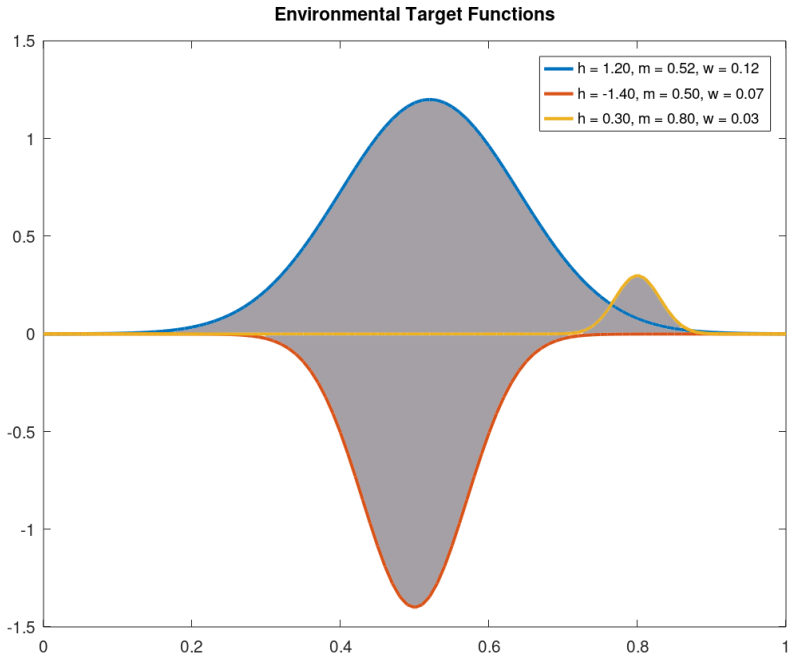

Figure 3.1: A visual representation of the target function f_E for the experiments.

In addition to the six tested changed conditions (i.e. $\langle\mu_+/\mu_-\rangle$, $\langle k_+/k_-\rangle$, and $\langle N_+/N_-\rangle$), a control condition experiment was performed in which the wild type was simply allowed to continue to evolve for the 500,000 generations with the same parameters as during the generation of the wild type.

To minimize bias, for each condition five runs were performed (that is, 5 rounds of mutation up, 5 rounds of mutation down, etc.), where each run had a differing random seed to control for the deterministic effects of the pseudorandom nature of Aevol's stochastic processes. This lead to a total of 35 experiments, all of which were carried out on a cluster from bwCloud[1].

The resulting data was processed using a combination of Python[2], Pandas[3], and Jupyter Notebooks[4].

3.2.1 Inputs

In Table 3.1, the parameter values for all input parameters may be found. Please note that for μ, this represents the mutation rates for point mutations, small insertions, and small deletions. The rearrangement rates were not changed under any condition and were always $1e-6$ for duplications, deletions, translocations, and inversions.

Condition	**Parameter**		
	μ	k	N
control	$1.00E{-}7$	1000	1024
mutation up (μ_+)	$4.00E{-}7$	1000	1024
mutation down (μ_-)	$2.50E{-}8$	1000	1024
selection up (k_+)	$1.00E{-}7$	4000	1024
selection down (k_-)	$1.00E{-}7$	250	1024
population up (N_+)	$1.00E{-}7$	1000	4096
population down (N_-)	$1.00E{-}7$	1000	256

Table 3.1: Table of input parameters. μ is the mutation rate, k is the selection strength, and N is the population size.

As can be seen in Table 3.1, only one parameter varied per condition in order to isolate potential influences on the outcome. A multiplier of 4 was chosen for the differing conditions relative to the control condition (e.g. $|N_+| = 4096 = 4 * |N_{\text{control}}| = 4 * 1024$ organisms). This choice of a 4x multiplier was chosen following the model of Carde et al.[6].

[1] `https://www.bw-cloud.org/`
[2] `https://www.python.org/`
[3] `https://pandas.pydata.org/`
[4] `https://jupyter.org/`

3.3 Evaluation Strategy

In this section, the criteria to be used for evaluating the results are examined. The primary criteria will be calculating the evolved genome's evolvability, robustness, and structure, with a statistical analysis of the changed conditions vs. the control condition.

3.3.1 Statistical Analysis of the Conditions

First, it must be determined whether the results of the condition (e.g. mutation up, selection down, etc.) were significantly different from the control condition, statistically speaking. To do this, for some test (e.g. robustness, evolvability, etc.) the mean across all seeds is calculated for a condition (e.g. μ_+) and compared to the mean across all seeds of the control condition. For these types of comparison, the "Mann-Whitney U" test is used. The Mann-Whitney U test is similar to the Wilcoxon signed-rank test but it is used when the distribution of the two samples cannot be assumed to be normally distributed. The Mann-Whitney U test can also be used when the sample sizes are different, for example when the number of base pairs is different between two organisms.

The purpose of the Mann-Whitney U test is to check whether two independent samples were selected from populations having the same distribution. Similar to other rank-sum tests, the test consists of the following steps:

1. Assign numeric ranks to all observations;
2. Add up the ranks for the observations from the first sample, giving us R_1
3. The statistic U_1 for the first sample is given by:

$$U_1 = R_1 - \frac{n_1 * (n_1 + 1)}{2}$$

   where n_1 is the sample size for the first sample and R_1 is the sum of the ranks from the first sample.

The U statistic for the second sample (i.e. U_2) is computed analogously. Then, $U = U_1 + U_2$ which, for sample sizes greater than 20, is approximately

normally distributed. Then the standard score is given by:

$$z = \frac{U - m_U}{\sigma_U}$$

where m_U and σ_U are the mean and standard deviation of U.

$$m_U = \frac{n_1 * n_2}{2} \text{and } \sigma_U = \sqrt{\frac{n_1 * n_2 \left(n_1 + n_2 + 1\right)}{12}}$$

In these experiments, the Python library function `scipy.stats.mannwhitneyu` was used on Pandas DataFrames containing the observations in order to calculate the Mann-Whitney U statistic. This function also calculates the p-value. With the typical confidence level of $\alpha = 0.05$, if the p-value is below 0.05, the null hypothesis H_o must be rejected that the two samples (i.e. the control and the condition) are from the same distribution; otherwise it must be accepted.

3.3.2 Fitness

Fitness in Aevol is closely tied in to the "metabolic error". This error, g, is calculated as the gap between the environmental function f_E and the phenotype of the organism, f_P. Once g is determined as described above in Section 2.2.2, it may be used to calculate the actual fitness of the organism according to the equation:

$$\text{fitness} = exp(-k * g)$$

where k is the *selection coefficient* variable set by the user in a parameter file. Aevol provides fitness statistics both for the fittest individual and for the population at large.

3.3.3 Robustness

Aevol calculates statistics for both mutational robustness as well as antirobustness. Robustness in Aevol is calculated similarly to evolvability (see Section 3.3.4): a `lineage` file (see Section 2.3) for an individual is fed to the post-treatment `misc_ancestor_robustness`, which generates a user-specified large number of offspring whose fitness is then measured. The percentage of these offspring which are neutral (i.e. whose phenotype is not affected by the mutations) is the robustness, and the percentage of positive

offspring (i.e. those whose fitness is greater than their parent's) determines the antirobustness.

In these experiments, at the end of the run of 500,000 generations a lineage file for the best individual at generation 500,000, i.e. the individual whose metabolic error was smallest, is generated. This lineage file is then fed in to the post-treatment `aevol_misc_ancestor_robustness` and robustness statistics are generated for each generation in the lineage file. The statistics produced by this post-treatment are summarized in Table 3.2 below.

Statistic
Fraction of positive offspring
Fraction of neutral offspring (aka reproductive robustness)
Fraction of neutral mutants (aka mutational robustness)
Fraction of negative offspring
Cumulative delta-gaps of positive offspring
Cumulative delta-gaps of negative offspring
Delta-gap for the best offspring
Delta-gap for the worst offspring
Cumulative delta-fitness of positive offspring
Cumulative delta-fitness of negative offspring
Delta-fitness for the best offspring
Delta-fitness for the worst offspring

Table 3.2: Table of robustness statistics calculated by Aevol for the best individual with the provided lineage.

These statistics for both the control and the specific desired condition (e.g. mutation up) may then be compared. Because this data is somewhat noisy (owing to the fact that the fitness may change rapidly), occasionally box and whisker plots will be used to show the overall spread rather than plotting the robustness generation by generation.

3.3.4 Evolvability

In Aevol, evolvability is calculated by generating a large set of offspring for a specific individual (one whose lineage was generated using the post-treatment `aevol_misc_lineage`) at regular periods along their lineage and

then determining the number of "positive offspring". Positive offspring are defined as those whose fitness is greater than their parent's. The evolvability of an individual is then the sum total of all improvement of all of the beneficial offspring, i.e.:

$$\text{evolvability} = \frac{|\text{positive offspring of } i|}{|\text{total offspring of } i|} * \sum \Delta_{fitness}^{\text{positive offspring}}$$

where $\sum \Delta_{fitness}^{\text{positive offspring}}$ is the cumulative sum of the fitness increase for the positive offspring. Thus, evolvability in Aevol accounts for both the likelihood of a positive mutation and the average improvement provided by said mutation. Practically speaking, to find an organisms evolvability in Aevol one must give the post-treatment `misc_ancestor_robustness` a lineage file and then multiply the fraction of the number of positive offspring (column 2) by the cumulative total of the fitness gap g of the positive offspring (column 10).

3.3.5 Structure

Another strength of Aevol is its ability to analyze changes in the structure of DNA and RNA. As with fitness, Aevol produces statistics about individuals and the population at large for many aspects of genome structure, including: the number of coding vs. non-coding bases (i.e. they respectively do or do not code for at least one protein), the average size of coding and non-coding DNAs, the number of genes, the number of "essential" base pairs (i.e. those that are part of a functional coding sequence), etc. Tables 3.3 and Table 3.4 summarize the different statistics and where they are to be found Aevol's output files.

As mentioned in Section 2.4, Koskiniemi et al. [15] tested the hypothesis that reductive evolution might occur partly as a result of selection, where fitness is gained by selecting for a

Non-coding bases determine regulation

Stat File	
`stat_genes_`⟨`best/glob`⟩	`stat_bp_`⟨`best/glob`⟩
number of coding RNAs	number of bp not in any CDS
number of non-coding RNAs	number of bp not included in any functional CDS
average size of coding RNAs	number of bp not included in any non-functional CDS
average size of non-coding RNAs	number of bp not included in any RNA
number of functional genes	number of bp not included in any coding RNA
number of non-functional genes	number of bp not included in any non-coding RNA
average size of functional genes	number of non-essential bp
average size of non-functional genes	number of non-essential bp including non-functional genes

Table 3.3: Statistics found in `stat_genes` and `stat_bp` output files from Aevol. ⟨best/glob⟩ indicates that statistics are available for both the best individual and the average across the whole population.

"Essential" base pairs are those whose mutation would change the phenotype of the organism.

Stat File	
stat_fitness_⟨best/glob⟩	**stat_mutation_⟨best/glob⟩**
population size	number of local mutations
fitness	number of chromosomic rearrangements
genome size	number of switches
metabolic error	number of indels
parent's metabolic error	number of duplications
metabolic fitness	number of deletions
secretion error	number of translocations
parent's secretion error	number of inversions
secretion fitness	
amount of compound present in grid-cell	

Table 3.4: Statistics found in `stat_fitness` and `stat_mutation` files from Aevol. ⟨best/glob⟩ indicates that statistics are available for both the best individual and the average across the whole population.

3.4 Expected Results

Given the state of the literature and other experiments, as partially described in Section 2.4, the table below summarizes the expected results.

Experiment Predictions						
Effect on:	**Condition**					
	μ_+	μ_-	k_+	k_-	N_+	N_-
Genome Size	$-^{[5,\ 17]}$	$+^{[5,\ 10]}$	$+^{[3]}$	$-^{[3]}$	$-^{[2]}$	$+^{[2]}$
Fitness	$+^{[1,\ 27]}$	$-^{[27]}$	$+^{[2]}$	$-^{[2]}$	$+^{[8,\ 27]}$	$-^{[8,\ 27]}$
Amount of non-coding DNA	$-^{[14]}$	$+^{[14]}$	$+^{[3,\ 14]}$	$-^{[3,\ 14]}$	$-^{[3]}$	$+^{[3]}$
Number of genes	$-^{[14]}$	$+^{[14]}$	$+^{[14]}$	$-^{[14]}$	$-^{[2]}$	$+^{[2]}$
Average size of genes	$-^{[16]}$	$+^{[16]}$	$-^{[3]}$	$+^{[3]}$	$-^{[2]}$	$+^{[2]}$
Robustness	$-^{[14]}$	$+^{[14]}$	$-^{[3,\ 14]}$	$+^{[3,\ 14]}$	$-^{[11]}$	$+^{[11]}$
Evolvability	$+^{[14]}$	$-^{[14]}$	$+^{[3]}$	$-^{[3]}$	$-^{[29]}$	$+^{[29]}$

Table 3.5: Predictions for the experiments. μ is the mutation rate, k is the selection rate, and N is the population size. μ_+ indicates an increased mutation rate, μ_- a decreased rate, etc. A $+$ in the main grid space indicates an expected increase (over the control condition) for that condition, and a $-$ indicates an expected decrease for that condition.

Chapter 4

Results and Discussion

This chapter presents the results of the experiments as performed according to the plan described in Chapter 3. Section 4.1 provides the data and these results are discussed in Section 4.2.

4.1 Results

In this section, the results of the experiments are presented. For each examined aspect, both a graphical representation of the data and some of the statistics are given, beginning with the conditions' impact on genome size. It should be noted that in many of the figures throughout this chapter, a rolling window of the mean was used to smooth the values, resulting in many of the plots only showing data starting after a few thousand generations (typically 10,000). It should be noted, however, that this is simply an artifact of the graphing process and organisms were continuously evolved for 500,000 generations.

4.1.1 Genome Size

In this section the results of the different conditions are examined, focusing on which, if any, lead to a reduced genome. Figure 4.1 presents the main findings regarding genome size. In the figure, the blue line shows the control condition and the other colors show the changed conditions: μ_+/μ_- (mutation up/down), k_+/k_- (selection up/down), and N_+/N_- (population up/down). As can be seen from the figure, the expectations from Section 3.4

did not always hold up, as several conditions actually *increased* over their original size.

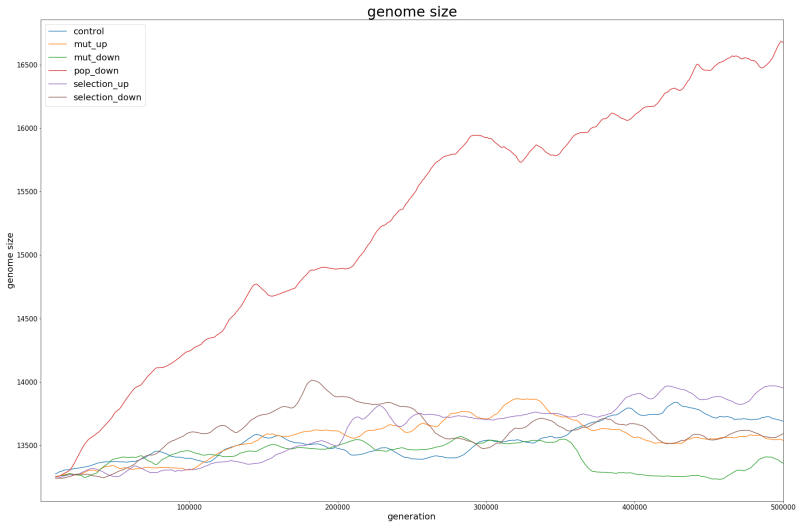

Figure 4.1: Genome size in number of bases of all conditions. Average taken across all five seeds for each condition.

In fact, the N_- condition had a runaway increase in the number of bases, reaching over 16,500 bases at one time, a 25% increase over the original wild type's roughly 13,200 bp. Surprisingly, even after 500,000 generations it seems that the upper limit may still not have been reached. The statistical results are given below in Tables 4.2 and 4.1, which show the mean and standard deviation among all five seeds over all generations.

Genome Size - Mean & Std. Dev. (in bp)			
	mean	**standard deviation**	**mean's % change from control**
control	13537.031513	144.351138	—
μ_+	13558.374160	155.077847	0.157661
μ_-	13410.743690	101.376104	-0.932906
k_+	13621.901104	229.635708	0.626944
k_-	13614.663599	172.675042	0.573479
N_-	15275.149392	950.266243	12.839727

Table 4.1: Average genome size and standard deviation for all seeds and all conditions.

Genome Size - Rank Sum & P-Values		
	rank sum U	**p-value**
μ_+	110508011469.50	0.00000000
μ_-	71008638349.50	0.00000000
N_-	16477791900.50	0.00000000
k_+	100151680984.50	0.00000000
k_-	88533681875.00	0.00000000

Table 4.2: Genome size statistics. Each condition is compared with the *control* condition.

The next most obvious observation is that of the remaining conditions, all but the k_+ condition (which had an an increase of 11% over the control condition) ended up with fewer base pairs than the control condition, though all increased slightly over their starting point. Also noteworthy is that the mutation down condition appears to have had a steady increase in the number of base pairs until a maximum of just over 14,000 around generation 350,000 before having a fairly sharp decline back to nearly the original size.

Examining the percent changed from the *control* condition shown in Figure 4.2 it can be seen that at points, the μ_- condition was nearly 5% smaller than the *control* condition.

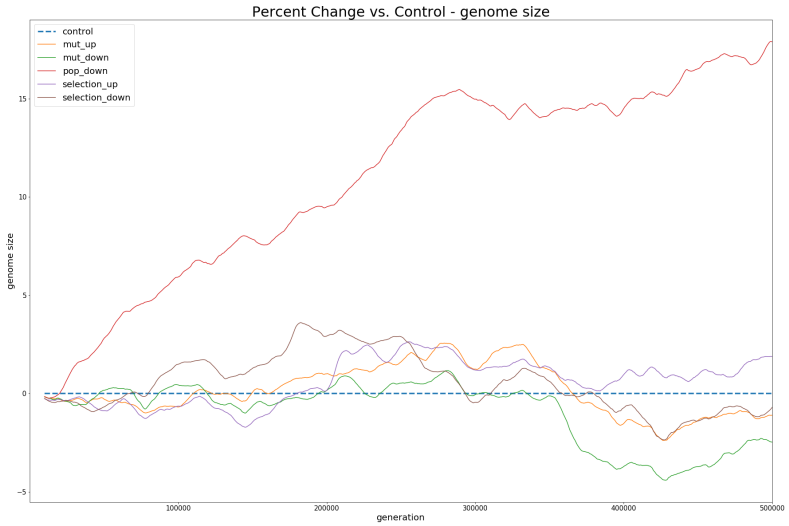

Figure 4.2: Genome size's percent change from the *control* condition.

4.1.2 Metabolic Error and Fitness

Figure 4.3 shows the mean fitness of the population for the control, mutation up/down, and population down conditions. Selection up/down were excluded from this graph because they were significantly outside of the range of the other conditions and made the results impossible to graph, but they are included in Figure 4.4, a histogram of the results.

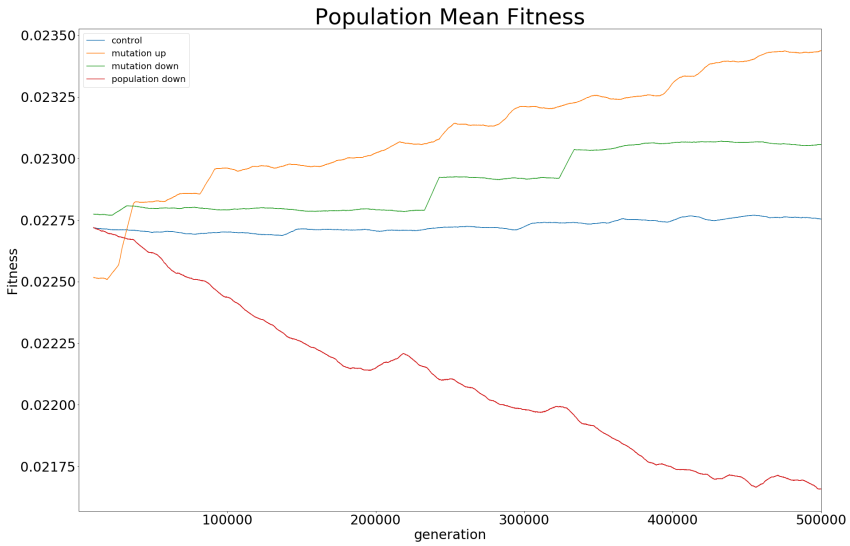

Figure 4.3: Plot of the mean fitness over time for the control, mutation up/down, and population down conditions.

The figure shows that the fitness of the control condition (in blue) consistently stayed at the same level, likely owing to the fact that the wild genotype had already been allowed to evolve for 10 million generations in this environment, so the phenotype closely lines up with the target before the simulations even began. Small fluctuations occurred due to mutations, insertions, etc. but since beginning with a clonal population of the best organism after 10 million generations in this same environment, the average fitness remained steady.

More interesting is the population down condition, where the average fitness in the whole population sharply declined, sinking to 4.5% below the control condition at generation 500,000. It seems that with the smaller population size, genetic drift may be more strongly at work in continually increasing the gap between the phenotype and environmental function, as the lack of variety inherent in a smaller population causes a cascade of increasingly deleterious consequences.

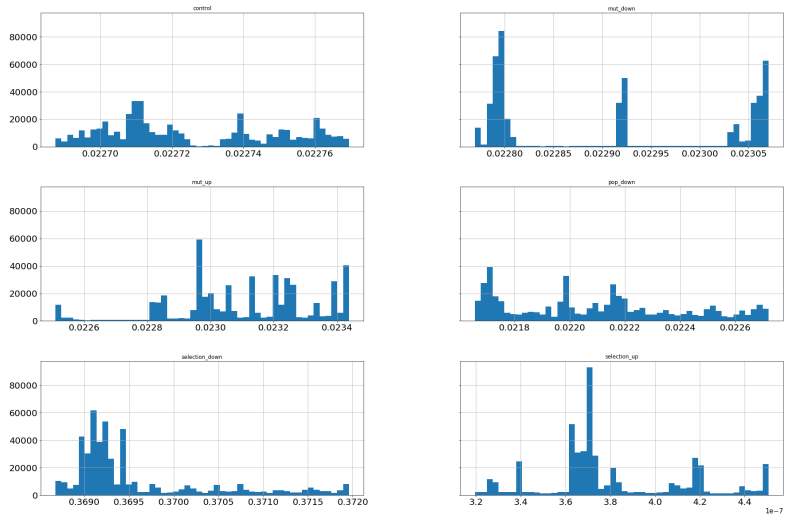

Figure 4.4: Histogram of population fitness at generation 500,000 for all conditions. Note that for the selection up condition, the scale is 1e-7.

Figure 4.4 shows a histogram of all fitness values for all conditions. Although the N_{-} condition ended the simulations in the most similar position to the *control* condition, Figure 4.4 makes it clear that the *control* condition remained much steadier throughout the simulation, regularly jumping from value to value, whereas the *mutation down* condition, as expected, tended to hover at certain locations in between rarer mutation events.

Figure 4.5 shows the mean metabolic error across all seeds for the whole population over time with respect to each condition.

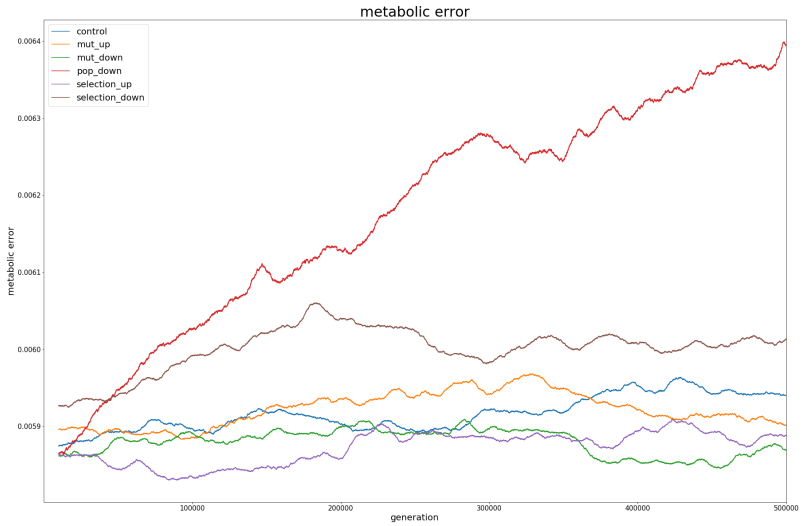

Figure 4.5: Plot of the metabolic error over time for all conditions, average of all seeds

Table 4.3 below gives the mean fitness scores for differing conditions. The selection up/down conditions seem to be somewhat anomalous, as they were several orders of magnitude smaller than all other conditions, despite having comparable genome sizes, numbers of genes, etc.

Fitness

	mean	**standard deviation**	**mean's % change from control**
control	0.022725	0.000023	—
μ_+	0.023107	0.000219	1.679131
μ_-	0.022909	0.000118	0.806895
k_+	0.000000	0.000000	-99.998326
k_-	0.369594	0.000810	1526.344327
N_-	0.022097	0.000310	-2.764272

Table 4.3: Fitness means and standard deviations across all seeds.

4.1.3 Genome Structure

In this section, the effects of the differing conditions on the structure of the genome as measured by the criteria in Tables 3.3 and 3.4 is examined.

Non-coding DNA

One important factor in genome structure is the amount of non-coding DNA, i.e. the number of bases which are part of a genome but which do not encode protein sequences. Aevol gives the number of bases which are not in any coding sequence, as well as the total genome size, so one may easily calculate this percentage. The results from the experiments are shown in Figure 4.6. Recalling genome size in Figure 4.1, it seems that as the genome size of the population down condition rapidly expanded, most of those expansions were non-coding.

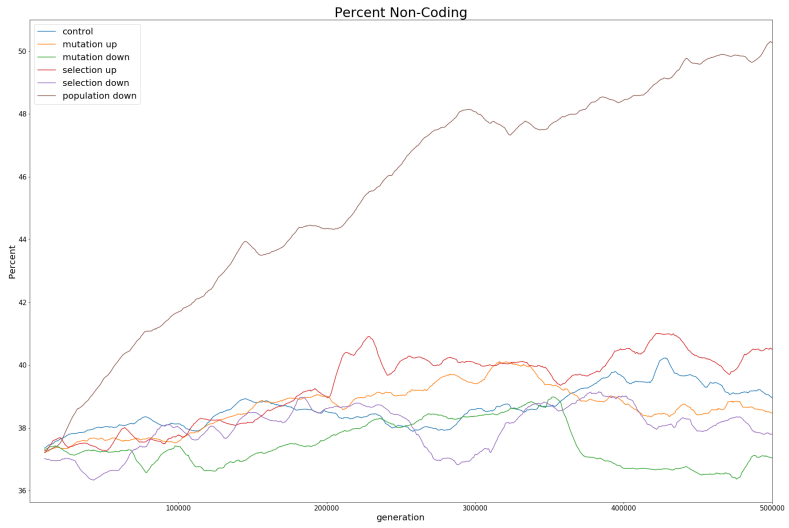

Figure 4.6: Plot of non-coding DNA over time for all conditions, average of all seeds.

By comparison, despite the k_+ condition's rapid decline in fitness, it did not acquire more than 1% non-coding DNA.

The average percentages of non-coding DNA are given in Table 4.4 below.

% Non-coding DNA			
	mean	**standard deviation**	**mean's % change from control**
control	38.622711	0.601258	—
μ_+	38.661764	0.707173	0.101114
μ_-	37.446414	0.694349	-3.045608
k_+	39.329164	1.138623	1.829114
k_-	38.004029	0.704953	-1.601860
N_-	45.462209	3.545778	17.708489

Table 4.4: Mean and standard deviation of non-coding percentages of DNA across all seeds.

Number of Functional Genes

Recall from Section 3.2 that a "functional gene" is a gene which produces a gene product. In this case, this means a triangle of mean m, height h, and half-width w. Figure 4.7 below illustrates the mean number of functional genes across the population for each condition.

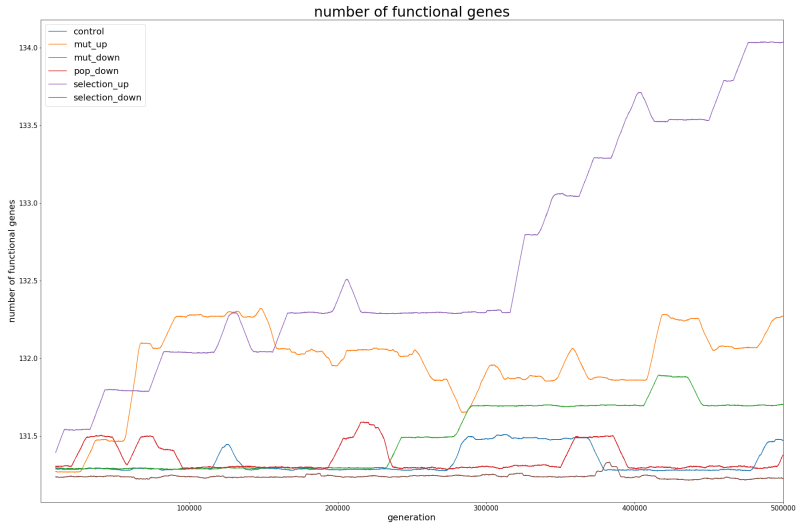

Figure 4.7: Plot showing the mean number of functional genes over time across all seeds.

As seen in the figure, the *selection up* condition showed the greatest increase in the number of functional genes in the whole population, reaching a maximum increase of about 2.2% (131 to 134). This is in agreement with the predictions (from Table 3.5) and the findings of Knibbe et al. [14]. Whereas most of the other curves are fairly flat, the mutation up condition fluctuates relatively rapidly, owing to the quick increase and decrease in the number of base pairs.

The mean and standard deviation of each condition is given in Table 4.5 below.

Number of Functional Genes

	mean	**standard deviation**
Control	131.33289822159796	0.08150846235734634
μ_+	131.97477775907822	0.25246331309927117
μ_-	131.50054072148615	0.20757038951512172
k_+	132.60656804679095	0.7195302117186394
k_-	131.2389093250132	0.013973734803973407
N_+		
N_-	131.35051857775917	0.08328014006877373

Table 4.5: The mean and standard deviation for the number of functional genes across all seeds and for all conditions.

Number of Non-Functional Genes

The average number of non-functional genes across all seeds for each condition is illustrated in Figure 4.8 below, and Table 4.6 provides the mean and standard deviation for all seeds and all conditions.

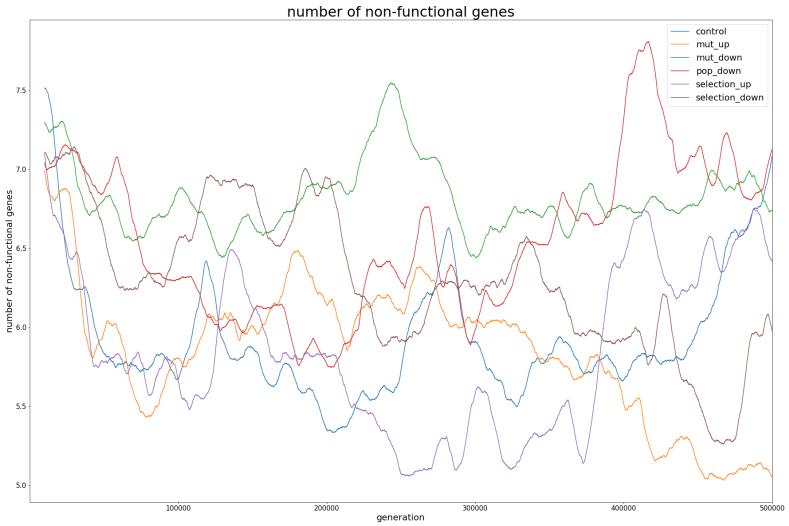

Figure 4.8: The mean number of non-functional genes across all seeds for each condition.

Number of Non-Functional Genes - Mean & Std. Dev.		
	mean	**standard deviation**
Control	5.927067481209257	0.3857900832994517
μ_+	5.856448948104222	0.42847313257452385
μ_-	6.821347895210327	0.22159072539984662
k_+	5.844289488031533	0.5069222102358901
k_-	6.297804225025876	0.45415175059248414
N_+		
N_-	6.555471025284081	0.4836308269184799

Table 4.6: The mean and standard deviation for the number of non-functional genes, all seeds, all conditions.

Average Size of Functional Genes

Figure 4.9 shows the average number of base pairs for each functional gene for the best individual, mapped out over the 500,000 generations. The pop-

ulation down condition continues to be the outlier in terms of genome structure, with the average size of the functional genes remaining noticeably lower than for any other condition. It is worth pointing out, however, that the difference between the smallest average and the largest average is only 1 base pair.

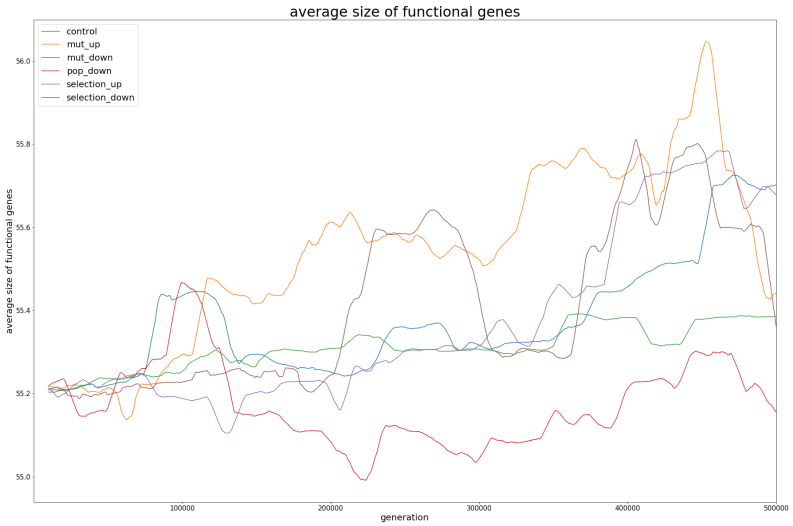

Figure 4.9: Plot showing the average size of functional genes over time for all seeds.

Table 4.7 below gives the mean and standard deviation of the results, and Table 4.8 gives the p-values for comparing the control condition to every other condition.

Average size of functional genes		
	mean	**standard deviation**
Control	55.375429751247275	0.13745740540215895
μ_+	55.5426245000319	0.20716437222446815
μ_-	55.309915451000805	0.05059383390579744
k_+	55.37046527382346	0.2024427915314053
k_-	55.414464890880595	0.1954246818788886
N_-	55.17428963311534	0.09886004440916518

Table 4.7: Mean functional gene size and standard deviation, all seeds, all conditions.

Average size of functional genes - rank sum & p-value		
	rank sum	**p-value**
μ_+	68252491056.0	0.00000000000000
μ_-	96067062201.5	0.00000000000000
N_-	28963878467.0	0.00000000000000
k_+	103481641646.0	0.00000000000000
k_-	124890321462.0	0.223664707704009

Table 4.8: Average size of functional genes - rank sum and p-value for all conditions and all seeds.

Since the k_- condition's p-value was ≥ 0.05, it must be concluded that these results are not significant.

4.1.4 Evolvability

In Figure 4.10 below, the results of the experiments on evolvability for the best individual's (at generation 500,000) lineage are shown.

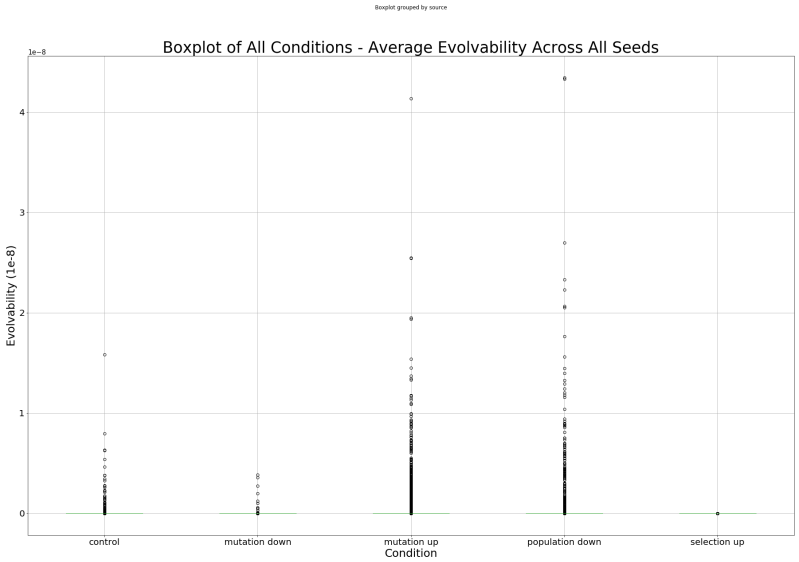

Figure 4.10: A box plot showing the mean evolvability spread of all seeds and all conditions. Higher numbers are more evolvable (keep in mind the scale is $1e^{-8}$).

The figure illustrates that, for each condition, overall the best individual still had quite low evolvability. For the selection up condition, however, the deviation from zero was even smaller, as illustrated in Table 4.10 below, which provides the mean and standard deviation of the evolvability of the best individual in each condition.

The evolvability rank sum and p-value information is given in Table 4.9.

Evolvability rank sum and p-value

	rank sum	**p-value**
μ_+	36203620.0	0.000000000309416
μ_-	7025283.5	0.010212448990316
k_+	13181851.5	0.000000000000074
k_-	13779001.5	0.000000084666842
N_+		
N_-	19403134.5	0.00000000000000

Table 4.9: Rank sum and p-values for evolvability, all seeds, all conditions.

Evolvability - Mean & Std. Dev.		
	mean	**standard deviation**
control	2.035523813975702e-11	3.2787490312081233e-10
μ_+	9.97655150597127e-11	8.41749150634588e-10
μ_-	6.064697990806935e-12	1.2471674281540128e-10
k_+	1.9498939718794653e-15	6.394659146405824e-14
k_-	4.3065723639721706e-10	5.20232907351617e-09
N_+		
N_-	8.837632116681012e-11	1.0799303682398726e-09

Table 4.10: Table illustrating the mean and standard deviation of the evolvability for each condition. μ is the mutation rate, k is the selection rate, and N is the population size.

4.1.5 Robustness

Recall from Section 2.3.2 that robustness is measured by the fraction of neutral offspring of an individual. In the following figure, a bar plot showing the spread of neutral offspring for the best individual is seen for the control condition as well as the six variations.

Robustness - Rank Sum & p-values		
	rank sum	**p-value**
μ_+	16756710.50	0.00000000
μ_-	5741758.00	0.00000000
N_-	12493750.00	0.00000000
k_+	13580845.50	0.00135870
k_-	13925884.00	0.00127325

Table 4.11: Mean robustness rank sum and p-values for all conditions and all seeds.

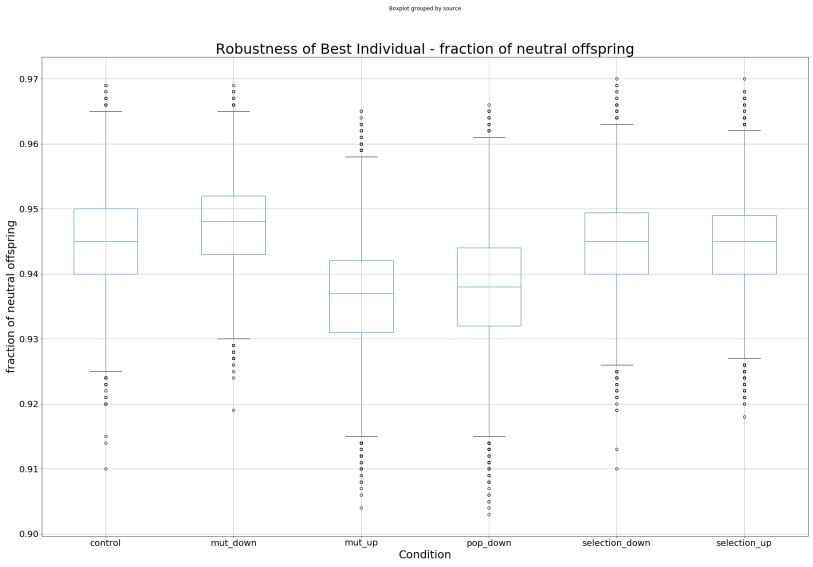

Figure 4.11: A box and whisker plot showing the spread of neutral offspring for the best individual at generation 500,000, all conditions. Higher numbers are more robust.

The mutation up condition clearly had the largest mean percentage of neutral mutants, at 0.5%. The full statistics are given in the tables below:

Robustness - means & standard deviations			
	mean	**std. dev.**	**mean's % change from control**
control	0.944925	0.007334	—
μ_+	0.936677	0.007907	-0.872870
μ_-	0.947543	0.007066	0.276989
k_+	0.944466	0.007372	-0.048598
k_-	0.944516	0.007424	-0.043350
N_-	0.937409	0.008846	-0.795408

Table 4.12: Robustness means and standard deviations for all conditions and all seeds.

4.2 Discussion

Analogously to Table 3.5, the results of the experiments in Table 4.13 are summarized below. For each cell, green indicates that the prediction was confirmed in these experiments and red indicates that the prediction must be rejected. The predictions for each condition are duplicated here for convenience, with a + indicating a predicted increase and − indicating a predicted decrease over the control condition.

Experiment Results Summary						
Effect On:	**Condition**					
	μ_+	μ_-	k_+	k_-	N_+	N_-
Genome Size	$-$[5, 17]	$+$[5, 10]	$+$[3]	$-$[3]	$-$[2]	$+$[2]
Fitness	$+$[1, 27]	$-$[27]	$+$[2]	$-$[2]	$+$[8, 27]	$-$[8, 27]
Amount of non-coding DNA	$-$[14]	$+$[14]	$+$[3, 14]	$-$[3, 14]	$-$[3]	$+$[3]
Number of genes	$-$[14]	$+$[14]	$+$[14]	$-$[14]	$-$[2]	$+$[2]
Average size of genes	$-$[16]	$+$[16]	$-$[3]	$+$[3]	$-$[2]	$+$[2]
Robustness	$-$[14]	$+$[14]	$-$[3, 14]	$+$[3, 14]	$-$[11]	$+$[11]
Evolvability	$+$[14]	$-$[14]	$+$[3]	$-$[3]	$-$[29]	$+$[29]

Table 4.13: A summary of whether the experiment results confirmed or denied the hypotheses of Table 3.5. were confirmed (green) or rejected (red), along with the predicted results of whether the given result would increase (+) or decrease (-) over the control condition.

According to the predictions based on a survey of the literature (summarized in Table 3.5), the μ_+, k_-, and N_+ conditions should have lead to a reduced genome. As shown in Figure 4.1 however, none of these predictions held up, and in fact the genome size increased in all conditions, including those that were predicted to be reduced. One possible explanation is that, because the wild types had already been evolving for 10 million generations, and the environment was not varied further for these experiments, perhaps the wild types had reached a sort of stasis for which another half-million generations were not sufficient to fully demonstrate the effects of any changes. This may not explain everything, however, as some conditions (e.g. μ_+, N_-) did see a rather rapid change in fitness, genome structure, etc. even in the (comparatively) short time.

Also of interest is that, along with the genome size, overall the fitness of the organisms did not change much with only one exception (k_+). It has been postulated that perhaps gene loss may be selected for in order to increase the overall fitness of an organism by removing the added cost of the superfluous genes [15]. These results do not fully agree with this hypothesis, as the results show that, in addition to the general accumulation of non-coding DNA across all conditions (Figure 4.6), which would theoretically have a higher fitness cost, the number of genes actually *increased* slightly for each condition, except for k_-. Likewise, fitness overall went *up* as the number of genes increased (except for N_-). Lastly, decreasing the selection pressure resulted in the highest overall mean fitness by a wide margin (Table 4.3). In Aevol, non-coding DNA is not really associated with a fitness cost, so this explanation may likely also not fully apply.

Regarding the N_- condition, comparing this with the results of Batut et al. and their work with *Buchnera aphidicola* (see Section 2.4), it may be possible that gene loss is occurring because of the smaller effective population size clicking down Muller's Ratchet. This does not seem to be what is happening in these experiments either, however. Though mean fitness of the N_- condition did greatly decline (see Figure 4.3), the N_- condition also saw the largest accumulation of non-coding DNA (see Figure 4.6) and the second-highest level of evolvability (see Table 4.10), which should theoretically have lead to an easier time of overcoming the ratchet. On the other hand, the number of functional genes stayed more or less the same as all of

the other conditions, and the accumulated number of non-functional genes (pseudogenes) of the N_- condition was the largest (see Figure 4.8). These two factors together suggest that there almost certainly was some loss of fitness through pseudogenization; perhaps with more time, the pseudogenes would also have been lost.

Recall the work of Liard et al.[16] as discussed in Section 2.4, in which the proposed "complexity ratchet" was overcome by increasing the mutation rate. Their work was quite successful in overcoming this ratchet, but in the experiments of this thesis, both the number of genes and the average size of the genes increased. One possible explanation for this is that, in their work, their increased mutation rate μ_+ was set to $1e^{-3}$. By contrast, even the elevated mutation rate used here, μ_+, was just $4e^{-7}$, which is $\frac{1}{4} * e^4 \approx 13.65$ times lower than their highest mutation rate. Perhaps an even higher mutation rate would be sufficient to overcome the power of selection and reduce the genome further.

The robustness results indicate that the stable environment is not requiring much differentiation between the different conditions. The average difference in robustness between any of the conditions and the control condition was only 0.002803, i.e. only 0.28%. Robustness does not seem to be selected for in any of the conditions.

Chapter 5

Conclusions and Future Work

In this chapter, the overall conclusions drawn from the work are summarized. Additionally, some limitations of the experiments are discussed, and a few possibilities for further research are highlighted.

5.1 Conclusions

At generation 500,000 there were three conditions in which the genome size was lower than the control condition (see Figure 4.1): μ_+, k_-, and surprisingly m_-.

5.2 Relation to Real-World Organisms

This thesis was begun with the intention of examining the factors that may drive reductive evolution, with the overall intention of relating this to real-world examples. The marine cyanobacteria *Prochlorococcus* has large effective population sizes and high adaptability, which has resulted in a large number of ecotypes which exist at differing depths in the oligotrophic oceans.

5.3 Limitations and Future work

5.3.1 Parameters

Although it was the main design of these experiments, one limitation to consider is that only one parameter varied per condition. It may be possible

that it is only under a combination of conditions that reductive evolution more reliably occurs. Since both an increased mutation rate and decreased selection pressure lead to a reduced genome in the experiments, it may be worth testing whether a combination of the two would lead to an even greater reduction.

In a similar vein, only one value for each changed condition was tested. For example, only a mutation rate of $\mu = 4e^{-7}$ was tested for the μ_+ condition. Perhaps an even higher mutation rate, such as $\mu_+ = 1e^{-3}$, might be enough to reduce the number of genes, as predicted by Knibbe et al. [14] and [16]. Their mutation rates were $2.1e^{-5}$ and $1e^{-3}$, roughly 3.9 and 13.65 times higher than the rate found here. The testing of more extreme values for each parameter may provide a fuller picture of the balance between selection, mutation rates, population sizes, and reductive genomes.

5.3.2 Environments

Another limitation is that the environments did not vary in these experiments. Aevol includes the ability to vary the environment natively, so future experiments could vary the environment over time to see if doing so would increase or decrease robustness or evolvability, which are strong drivers of reductive evolution [3].

5.3.3 Limitations of the Aevol Model

Aevol as a modeling software is limited in that, like all models, it relies on several simplifications. For example, the population sizes tested here, even in the population up condition, are still much smaller than would be found in real world populations. It could be that in order to properly judge the effects of Muller's Ratchet under differing conditions, much larger population sizes are needed. However, the length of time required for running such a simulation in Aevol even on a decent cluster might make this intractable, as quadrupling the population size also quadrupled the run-time of these experiments.

Aevol also relies on the simplification that its space of "biological functions" are a one-dimensional function, when the real-world space is obviously much more complicated. But this simplification effectively disallows the use

of gene networks, which

Bibliography

[1] Thomas Bataillon. Estimation of spontaneous genome-wide mutation rate parameters: whither beneficial mutations? *Heredity*, 84(5):497–501, 2000.

[2] Bérénice Batut, Carole Knibbe, Gabriel Marais, and Vincent Daubin. Reductive genome evolution at both ends of the bacterial population size spectrum. *Nature reviews. Microbiology*, 12(12):841–850, 2014.

[3] Bérénice Batut, David P. Parsons, Stephan Fischer, Guillaume Beslon, and Carole Knibbe. In silico experimental evolution: a tool to test evolutionary scenarios. *BMC bioinformatics*, 14 Suppl 15:S11, 2013.

[4] Guillaume Beslon, Vincent F Liard, and Santiago F Elena. Testing evolution predictability using the aevol software. 16th international meeting of the European Society of Evolutionary Biology (ESEB 2017), August 2017. Poster.

[5] Katie Bradwell, Marine Combe, Pilar Domingo-Calap, and Rafael Sanjuán. Correlation between mutation rate and genome size in riboviruses: mutation rate of bacteriophage qβ. *Genetics*, 195(1):243–251, 2013.

[6] Quentin Carde, Marco Foley, Carole Knibbe, David P Parsons, Jonathan Rouzaud-Cornabas, and Guillaume Beslon. How to reduce a genome? alife as a tool to teach the scientific method to school pupils. In *The 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International*

Conference on the Synthesis and Simulation of Living Systems (ALIFE), pages 497–504. MIT Press, 2019.

[7] Clarence FG Castillo and Maurice HT Ling. Digital organism simulation environment (dose): A library for ecologically-based in silico experimental evolution. *Advances in Computer Science : an International Journal*, 3(1):44–50, 2014.

[8] A.D. Cutter. *A Primer of Molecular Population Genetics*. Oxford University Press, 2019.

[9] Robert W Doran. The gray code. *J. UCS*, 13(11):1573–1597, 2007.

[10] John W Drake. A constant rate of spontaneous mutation in dna-based microbes. *Proceedings of the National Academy of Sciences*, 88(16):7160–7164, 1991.

[11] Santiago F Elena, Claus O Wilke, Charles Ofria, and Richard E Lenski. Effects of population size and mutation rate on the evolution of mutational robustness. *Evolution*, 61(3):666–674, 2007.

[12] Isabel Gordo and Brian Charlesworth. On the speed of muller's ratchet. *Genetics*, 156(4):2137–2140, 2000.

[13] Carole Knibbe. *Evolution of genome structure by indirect selection of the mutational variability: a computational approach*. Theses, INSA de Lyon, December 2006.

[14] Carole Knibbe, Antoine Coulon, Olivier Mazet, Jean-Michel Fayard, and Guillaume Beslon. A long-term evolutionary pressure on the amount of noncoding dna. *Molecular Biology and Evolution*, 24(10):2344–2353, 08 2007.

[15] Sanna Koskiniemi, Song Sun, Otto G Berg, and Dan I Andersson. Selection-driven gene loss in bacteria. *PLoS genetics*, 8(6), 2012.

[16] Vincent Liard, David Parsons, Jonathan Rouzaud-Cornabas, and Guillaume Beslon. The complexity ratchet: Stronger than selection, weaker than robustness. *Artificial Life Conference Proceedings*, 1(30):250–257, 2018.

[17] Gabriel AB Marais, Alexandra Calteau, and Olivier Tenaillon. Mutation rate and genome reduction in endosymbiotic and free-living bacteria. *Genetica*, 134(2):205–210, 2008.

[18] Octavio Miramontes, Franois Taddei, Ariel B. Lindner, Antoine Frenoy, and Dusan Misevic. Shape matters in cooperation. *Artificial Life Conference Proceedings*, 1(28):340–341, 2016.

[19] Dusan Misevic, Antoine Frenoy, David P Parsons, and Francois Taddei. Effects of public good properties on the evolution of cooperation. In *Artificial Life Conference Proceedings 12*, pages 218–225. MIT Press, 2012.

[20] Vadim Mozhayskiy and Ilias Tagkopoulos. Microbial evolution in vivo and in silico: methods and applications. *Integrative Biology*, 5(2):262–277, 10 2012.

[21] C. Ofria and C. O. Wilke. Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229, 2004.

[22] David Parsons. *S 'e indirect election in é Darwinian evolution: m 'e mechanisms and implications*. PhD thesis, Lyon, INSA, 2011.

[23] T. S. Ray and J. Hart. Evolution of differentiated multi-threaded digital organisms. In *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*, volume 1, pages 1–10 vol.1, 1999.

[24] Jacob Rutten, Paulien Hogeweg, and Guillaume Beslon. Adapting the engine to the fuel: mutator populations can reduce the mutational load by reorganizing their genome structure. *BMC Evolutionary Biology*, 19, 12 2019.

[25] Yolanda Sanchez-Dehesa, Loïc Cerf, Jose Maria Pena, Jean-François Boulicaut, and Guillaume Beslon. Artificial Regulatory Networks Evolution. In *Proc 1st Int Workshop on Machine Learning for Systems Biology MLSB 07*, pages 47–52, Evry, France, September 2007.

[26] Zhiyi Sun and Jeffrey Blanchard. Strong genome-wide selection early in the evolution of prochlorococcus resulted in a reduced genome through the loss of a large number of small effect genes. *PloS one*, 9:e88837, 03 2014.

[27] Ali R Vahdati, Kathleen Sprouffske, and Andreas Wagner. Effect of population size and mutation rate on the evolution of rna sequences on an adaptive landscape determined by rna folding. *International journal of biological sciences*, 13(9):1138, 2017.

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

[29] Tanita Wein and Tal Dagan. The effect of population bottleneck size and selective regime on genetic diversity and evolvability in bacteria. *Genome biology and evolution*, 11(11):3283–3290, 2019.

[30] Claus O Wilke, Jia Lan Wang, Charles Ofria, Richard E Lenski, and Christoph Adami. Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412(6844):331–333, 2001.