Fachhochschul-Masterstudiengang
INFORMATION ENGINEERING & -MANAGEMENT
4232 Hagenberg, Austria

Using Genetic Data from Lineages and Experiments to Measure Co-Evolutionary Dynamics in Egalitarian Groups

Masterarbeit

zur Erlangung des akademischen Grades
Master of Science in Engineering

Eingereicht von

Nadine Strasser, BSc

Betreuer: Dr. Charles Ofria
Michigan State University

Begutachter: FH-Prof. PD DI Dr. Stephan Winkler
Fachhochschule Oberösterreich

September 2020

Eidesstattliche Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt. Die vorliegende Masterarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Hagenberg, am 2. September 2020

Acknowledgments

This work was carried out during my (partly virtual) research visit at the BEACON Center for the Study of Evolution in Action at Michigan State University. It would not have been possible without the financial support of the Austrian Marshall Plan Foundation, which I would like to thank for funding my visit with the Marshall Plan Scholarship (MPS). Additionally, Michigan State University, specifically the Institute of Cyber-Enabled Research, supported this work by providing the necessary computational resources.

I offer my sincerest gratitude to my supervisor at Michigan State University, Dr. Charles Ofria. Without his guidance, encouragement, overwhelming support and inspirational ideas this thesis would not have been possible. I am deeply grateful to my colleagues Alexander Lalejini and Dr. Anya Vostinar for introducing me to the field of artificial life, for providing practical guidance for my research activities, the intellectual contributions and for sharing their exuberant enthusiasm. Special thanks also go out to Clifford Bohm for introducing me to MABE and for the many insightful discussions. Moreover, I would like to very much thank all of those with whom I have had the pleasure to work with, especially all members of the DevoLab, for making me feel at home in a foreign country.

My sincere thanks also goes to my advisor at the University of Applied Sciences Upper Austria, Dr. Stephan Winkler, for his excellent guidance and constant support, and for making this research visit possible in the first place by connecting me with the BEACON Center.

Finally, I would like to express my very profound gratitude to my parents. Their never-ending love, guidance and support are with me in whatever I pursue. This accomplishment would not have been possible without them and their continuous encouragement throughout my whole life.

Abstract

Evolutionary algorithms (EAs) are a common technique used for solving computational problems via simulations. This thesis proposes an implementation of an EA for measuring genetic signatures of co-evolution with lineage-based data. Moreover, experiments are conducted to research the interaction between different levels of selection mechanisms. Both approaches have not only the EA-aspect in common, but also the focus on co-evolutionary dynamics and the overall population model. The organisms of a population used for the EA consist of an egalitarian group: Each higher-level organism is composed of two different lower-level cells; one from type A and one from type B.

The first part of this thesis concentrates on the detection of co-evolutionary signatures using genetic data from lineages and tightly co-evolving organisms. Specifically, the author measures the effect of co-evolution on mutational changes along lineages under idealized conditions to determine whether lineage-based genetic data can be used to identify such relationships. The patterns resulting from tracking the mutational changes are analyzed using different approaches. Two ways for detecting a genetic signature are proposed: The first approach uses experimental manipulation and analyzes the mutation count, whereas the second one detects genetic signatures in a far more restrictive setting, considering solely historical data. For this purpose, the Accumulated Mutations Metric, a newly developed and sophisticated way for detecting co-evolution from lineages, is introduced. As mentioned afore, each higher-level organism is composed of two individual cells, representing different lower-level types. These cells are tightly linked and jointly determine the fitness of the overall organisms. When the fitness contribution of one cell type is dependent on the state of the other cell type, the expectation is that strong selective pressures are existent for one cell type to change in response to the other. This thesis proposes a computational model that depicts various idealized scenarios in which it is expected that evolving organisms exhibit co-evolution. The author finds that lockstep-like pattern can successfully be detected using lineages, showing that the two different types of lower-level cells indeed form an egalitarian group.

The second part of the work addresses a broader range of symbiotic behavior by introducing migration and disentangling the two populations of lower-level cells. Multi-level selection is established by separating the fitness goals for the higher-level organism and the two lower-level cells. Then, the fitness contributions over time and the formation of subgroups are analyzed. Cells that have the possibility of reproducing individually can display either antagonistic interactions (*e.g.,* predators and prey or parasite and hosts), or mutualistic ones, as occurring in cooperating groups or after major evolutionary transitions. The author finds a priori unexpected dynamics within the multi-level population.

Kurzfassung

Evolutionäre Algorithmen (EAs) sind eine gängige Methode, Rechenprobleme via Simulationen zu lösen. Diese Arbeit präsentiert eine EA-Implementierung, die es ermöglicht, genetische Signaturen, die auf Koevolution hindeuten, anhand von Abstammungsdaten zu messen. Zusätzlich wird die Interaktion von Selektionsmechanismen auf verschiedenen Ebenen erforscht. Neben dem EA-Aspekt haben beide Ansätze den Fokus auf koevolutionäre Dynamiken und das Populationsmodell gemein. Die Organismen, die eine Population formen, bestehen aus einer sogenannten egalitären Gruppe: Jeder höhere Organismus besteht aus zwei niedrigeren Zellen; eine von Typ A und eine von Typ B.

Der erste Teil dieser Arbeit konzentriert sich auf die Erkennung von koevolutionären Signaturen innerhalb Organismen mit Hilfe von genetischen Daten. Koevolution wird anhand von Genommutationen entlang von Abstammungslinien gemessen. Mithilfe von idealisierten Szenarien soll herausgefunden werden, ob diese Art von genetischen Daten geeignet ist, um koevolutionäre Beziehungen zu identifizieren. Genetische Signaturen, also die Mutationsmuster über den Zeitverlauf, werden auf zwei Arten identifiziert: Der erste Ansatz basiert auf experimenteller Manipulation und analysiert die Mutationshäufigkeit; der zweite Ansatz erkennt die genetische Signatur in einem restriktiveren Umfeld, ohne experimentelle Manipulation mit Hilfe der Accumulated Mutations Metric. Diese eigens entwickelte Messgröße kann Koevolution auf Basis von Abstammungslinien identifizieren. Da jeder höhere Organismus aus einer A- und einer B-Zelle besteht, sind diese niedrigeren Zellen eng gekoppelt. Zusammen bestimmen sie die Fitness des gesamten Organismus. Dieses Setup, in welchem der Fitnessbeitrag der einen Zelle vom Zustand der anderen Zelle abhängig ist, führt zu einem starken Selektionsdruck, durch den sich die Zelltypen abhängig voneinander verändern. Ein Computermodell mit idealisierten Szenarien, von denen erwartet wird, dass die evolvierenden Organismen Koevolution aufweisen, wurde dazu entwickelt. Die Arbeit zeigt, dass sogenannte Gleichschritt-Muster erfolgreich mittels Abstammungslinien identifiziert werden können und, dass Typ-A und Typ-B Zellen zusammen tatsächlich eine egalitäre Gruppe formen.

Der zweite Teil der Arbeit thematisiert symbiotisches Verhalten im Allgemeinen. Um Multilevel-Selektion zu ermöglichen wird das Modell erweitert, indem die Fitnessziele des höheren Organismus und der niedrigeren A- und B-Zellen separiert werden. Die Fitnessbeiträge im Zeitverlauf und die Formation von Subgruppen werden analysiert. Dies geschieht unter der Annahme, dass Zellen, die die Möglichkeit haben, sich unabhängig voneinander zu reproduzieren entweder antagonistisches (*z.B.* Räuber-Beute- oder Parasit-Wirt-Beziehungen) oder mutualistisches (*z.B.* kooperierende Gruppen nach großen evolutionären Übergängen) Verhalten entwickeln. Die Autorin zeigt dabei zuvor unerwartete Dynamiken innerhalb der Multilevel-Population auf.

Contents

Chapter 1

Introduction

This introductory chapter gives an insight into the motivation for this work and a general overview of the the author's contribution, followed by a discussion of the scientific goals and research questions. One section is dedicated to the BEACON Center at Michigan State University since the work presented in this thesis is the result of the author's research stay at the BEACON Center. Finally, the outline of the thesis is presented.

1.1 Motivation and Overview

The fundament of science can be found in our desire to understand the world. From very early on, humans were interested in how life and all of its facets work. Thanks to the revolutionary invention of electronic computers and the world wide web, a sheer infinite seeming amount of information is nowadays accessible to a broad audience. However, computers are not only a source of information, but they advance and accelerate our expertise in predicting and understanding nature. One of the biggest accomplishments of this revolution will be the creation of artificial intelligence, accompanied by artificial life. AI and ALife (for detailed information about ALife see Section 2.2) will be accountable for a new form of intelligence, new species and potentially even life forms, all composed of computer programs.

The attempts to achieve this accomplishment can be traced back to the beginnings of the computer age: Well-known scientist, including Alan Turing and John von Neumann had visions of intelligent computer programs with the lifelike ability to self-replicate and behaviors, such as adapting to a changing environment. Back then, and even very much earlier, humankind looked at nature to find analogies for how our lives can be advanced. From early on, computers were not only used as calculating machines and military objects, but also to learn about humans and to simulate biological evolution. In the 1980s, computing activities motivated by biological models gained a whole new meaning, as subfields, today known as *e.g.,* machine learning and evolutionary computation were on the rise.

Life itself can be seen as an optimization to the surrounding environment fueled by evolution. Various subfields of computer science make use of this thinking by applying metaphors found in the biological world to computational problems. Evolutionary algorithms, with their most prominent example of genetic algorithms, are only one case

where biological knowledge is applied to computer science (for details see Section 2.1). But why is evolution so appealing as a source of inspiration for solving computational problems? Seeing it from the perspective of biological evolution being a constant adaptation to the surrounding environment, on a very abstract level, highly fit organisms have a higher chance of being able to reproduce among an enormous number of possible genetic sequences.

Those organisms then will be fit to survive and thrive in the given environment. From a slightly different angle, biological evolution is able to design ingenious solutions for complex problems. Both of this perspectives present striking similarities to what is needed for solving complex computational problems: effective and efficient use of parallelism, as the "solution space" of million of species/solution candidates is searched, and an intelligent approach for choosing the next genetic sequences/solution candidates, which will be evaluated.

Natural evolution has brought an extraordinary variety and complexity to today's ecosystem, by a remarkably simple proceeding: From a high-level perspective, evolutionary sequences of random variation, followed by natural selection in which the fittest tend to propagate their genetic material to future generations, have led to life as we know it. [37] At the beginning of time, life was somewhat simple compared to the biosphere we live in today. It is believed that natural evolution has undergone some major transitions, redefining what it means to be an individual. The development of more complex lifeforms, emerging from simpler individuals characterize some of the most profound events in the evolutionary history of nature (more details are given in Section 2.3.1). And symbiosis and co-evolution were not only vital concepts for the major evolutionary transitions, but are in general of utmost importance for biological and digital evolution (more information on these two concepts is provided in Section 2.3.2).

1.2 Purpose and Contributions

This work studies some of the underlying dynamics of co-evolution in egalitarian populations (*i.e.,* populations consisting of organisms with different cell types) based on genomic lineage data. Although, today's sequencing technology does not allow to perform this analysis in the biological world, all necessary components can be established in a digital system, by *e.g.,* implementing a genetic algorithm. A broader understanding will be established of how genomic lineage data could be analyzed as our ability to sequence this type of data advances (for more information on lineages see Section 2.3.3). Moreover, in the course of this research the author tries to better understand why organisms use cooperative mechanisms on the one hand and specialize on an individual basis on the other hand, by exploring multi-level selection (details of this concept are provided in Section 2.3.4).

The flexibility of artificial life systems makes them an ideal testbed for exploring useful metrics that help detecting the traits of tight evolutionary couplings, which will extend the understanding on how egalitarian transitions and symbiotic behavior arose throughout evolution. The aim of this work is to propose a simple metric that looks at mutation accumulation in tightly coupled lineages and to explore the interaction of organism-level and individual cell level selection pressures. The introduced approaches will be able to detect genetic signatures in idealized scenarios from egalitarian pop-

ulations. The author assumes that a genetic signature is present when genomes are being modified in some sort of systematic way. In a latter step, migration and selection with conflicting pressures are added to the setup to broaden the understanding of the interaction between high-level and low-level selective pressures.

This research demonstrates the power of digital evolution in action: High-level concepts, borrowed from biology help to improve the understanding of egalitarian populations, as well as of multi-level selection. This work elucidates how a genetic signature can be identified in egalitarian populations with the implementation of a genetic algorithm and how multi-level selection operates. With the findings of this work, the author intends to provide both, an insight into the evolutionary history of life and a new technique for lineage analysis, equally worthwhile for biologists and computer scientists.

1.3 Scientific Goals and Research Questions

There are many processes and behavioral ways in nature that humanity cannot explain or fully understand. Instead of assembling the possible evolutionary history of organisms by comparing current forms or using fossils, this work concentrates on the evolutionary mechanism itself. The author intends to demonstrate that co-evolutionary relationships within an egalitarian population will evolve if the surrounding environment offers the opportunity for this behavior to arise. Thereby, the author's focus lies on the following two research questions:

1. Is a genetic signature existing in egalitarian populations?
2. How are the two different types of selection mechanisms (*i.e.,* lower-level individual selection and higher-level group selection) interacting in a multi-level population?

More precisely, 1. will answer the question: Can the genetic signatures of an egalitarian transition in individuality be detected in the evolutionary histories of constituent species by looking at lineage-based data and without taking into account the specific genetic architectures of the participants? As transitioning species become increasingly coupled, it might be expected that genetic changes in one lineage drive genetic changes in the other. As a result, it might also be expected that the amount of mutation accumulation in one lineage at a given point in time will correspond with the amount of mutation accumulation in a tightly coupled lineage. The herein proposed metric looks at the variance of beneficial mutation accumulation along the genomes of two coupled lineages throughout an evolutionary process. In egalitarian populations three options of interaction between the mutations of the lower-level cells are possible:

- Simultaneous genetic changes between species occur. Thus, both sides need to be mutated in a coordinated fashion.
- Mutations in one partner are responded to by mutations in the other one. The aim of the mutations is better cooperation within the population.
- No patterns are observed. Each one is evolving in a similar genetic pattern as it would have evolved in non-cooperative circumstances.

The second question, as stated in 2. centers the different levels of selection itself: The selection mechanisms from the evolution of a multi-level population may act at each level. When a system with two levels exists and those levels are not perfectly aligned,

one level of selection will determine which higher-level organisms move on to the next generation, and a second level of selection will determine which lower-level individuals are used in the propagule[1] that forms the offspring group in that next generation. At the organism-level, cooperation for replicating the whole organism is going to be most important, whereas at the individual cell level, it will be most important that the cell's own genetic material is passed on to the next generation. [34, 46] Hence, individual interests as well as group interests need to be satisfied and the selection mechanisms play a crucial role in doing so. The author is particularly interested in the evolution of a multi-level population over time and the potential formation of niches[2].

Evolution being a steady process, experiments in natural systems take time. Digital evolution techniques, such as evolutionary algorithms make it possible to watch evolution in action and to understand why something evolved in a way that we can today watch in nature. Digital evolution technology has some advantages over experiments in natural systems, including time complexity, variety and stability. As of now, it can be challenging to apply the approaches proposed in this thesis to biological data since it is hard to collect sequence data for whole lineages at a useful resolution, as [44] and [71] have shown. However, as sequencing technology improves, it will get easier to collect detailed mutation accumulation data for natural systems. In the meantime, the proposed approaches give a valuable insight into how lineage data from biological systems could be processed in the near future, and provide an initial foray into a broad range of possible analyses.

1.4 BEACON - An NSF Center for the Study of Evolution in Action

> *The BEACON Center for the Study of Evolution in Action is an NSF Science and Technology Center founded with the mission of illuminating and harnessing the power of evolution in action to advance science and technology and benefit society.*
>
> Excerpt from BEACON Mission Statement [79].

The BEACON Center conducts multidisciplinary research on evolutionary processes and applies this knowledge to solve real-world problems. Research is carried out in three ways: in natural biological systems studied in the lab and field; with so-called digital organisms in computational systems; and in engineered systems. BEACON approaches evolution in an innovative way and brings together biologists, computer scientists and engineers to study evolution in action. The almost 600 members of the center conduct their research work in biological and digital realms, and use evolutionary computing. The systems are used to understand how evolution in general works on the one hand and, to apply evolution to solve a range of real-world problems, on the other. [78]

BEACON stands for Bio/computational Evolution in Action CONsortium. It is a science and technology center in the United States, which is sponsored by the National

[1] Broadly speaking, the propagule is the sum of organisms that a population is formed out of. It is the starting point or, technically speaking, the seed.

[2] In biology, an ecological niche describes a part of an environment occupied by a specific species [77].

Science Foundation. The BEACON consortium is headquartered at Michigan State University with partner institutions at North Carolina A&T State University, University of Idaho, University of Texas at Austin and University of Washington. The director of the center is Dr. Charles Ofria. [78, 86]

The author worked with the Digital Evolution Laboratory (DevoLab) at Michigan State University, which is part of the BEACON Center. It is led by Dr. Charles Ofria. The research conducted within DevoLab focuses on the interplay between computer science and Darwinian evolution, using the principles of both fields to enhance the overall understanding. The goal of DevoLab is to use computational systems in order to better understand how evolution works. This is both, to make an impact on evolutionary biology, as well as on the computational side. With digital evolution, experiments are conducted faster and with more detail than it is possible in nature. In that way, DevoLab tries to answer some fundamental evolutionary questions. Regarding the impact on the computational side, DevoLab treats evolution as an algorithm. And as evolution's powerful abilities in nature are understood, humankind can start applying those same principles to solve all sorts of real-world problems. In summary, the DevoLab performs experimental studies on digital organisms with the goals of improving the understanding of how natural evolution works and applies this knowledge to solve biological, computational and engineering real-world challenges. The tools that are developed in the Digital Evolution Laboratory include:

- The **Empirical Software Library** makes it easy to build scientific software.
- The **MABE** project (see Section 4.1.1), which is a Modular Agent-Based Evolution platform that facilitates digital evolution experiments.
- **Avida**, which is a scientific software platform for conducting and analyzing experiments with self-replicating and evolving computer programs.
- **Avida-ED**, the educational version of Avida, which is used in 100+ biology classes around the world to teach fundamental principles of how evolution works.

Current projects of DevoLab members focus on the evolution of suicidal altruism, division of labor, major transitions (see Section 2.3.1), evolutionary cancer detection, genetic architecture and sexual selection. [85, 86]

1.5 Thesis Outline

The thesis is organized as follows: First, the computational and biological background of this work is illuminated in Chapter 2 and terms, such as *genetic algorithm*, *egalitarian population*, *co-evolution* and *symbiosis* are explained. Subsequently, Chapters 3 and 4 give an insight into the general solution approach for detecting co-evolutionary dynamics and the implementation details of the evolutionary model, developed during the author's research visit at BEACON. Finally, the conducted experiments are described in Chapter 5, followed by a discussion of the results, conclusions and potentials for future work in Chapter 6.

The entire source code described in this thesis, as well as the analysis scripts, an overview of the random number seeds used for the experiments and all figures (*i.e.,* including those not shown in this thesis) are available at a specifically created GitHub repository, found at https://github.com/nstrasser/MasterThesisProject. [73]

Chapter 2

Background

This chapter describes the underlying concepts applied in this research, ranging from evolutionary algorithms via artificial life and digital evolution through to basic biological ideas. Genetic algorithms are used to conduct the experiments described in Chapter 5. Artificial life is the domain in which the overall project was realized. And the underlying biological models are portrayed to get a sense of why this research and its result described in Chapter 6 have an applied value and to better understand why the approach presented in this thesis is useful.

2.1 Evolutionary and Genetic Algorithms

In computer science, evolutionary computation is a subfield of artificial intelligence and serves as umbrella term for algorithms that are inspired by biological evolution. Evolutionary algorithms are a subset of evolutionary computation and are divided into three main techniques:

- Genetic algorithms [25]
- Evolution strategies [9]
- Genetic programming [29]

Evolutionary algorithms (EAs) offer a convenient way to study evolution in simulated environments: EAs imitate natural evolution with the objective of generating efficient solutions for computational problems. An EA is a metaheuristic optimization method with a population-based approach. The adaptation of the individuals within the population is considered as optimization process. [20]

EAs use populations of individuals[1] to test many feasible solutions, which are refined over generational time. The fitness function plays a crucial role within an EA, as it defines the evolutionary goal and the evaluation of the individuals is based on the fitness function. EAs make use of the evolutionary mechanisms selection and the variation operators mutation and crossover[2]. Whereas the variation operators create diversity[3] within a population, selection pressures are used to increase the fitness among the

[1] Individuals are also called solution candidates or phenotypes.

[2] Crossover is sometimes also called recombination.

[3] Genetic diversity within a population is essential for finding good solutions and being able to adapt to changing environments.

individuals of a population. The fitness value of an individual is determined by the fitness function. In that way, EAs decide which individuals of a population are suitable for crossover and/or mutation. Hence, individuals that outperform others are able to propagate their genomes[4] within the population. Crossover generates a new individual, a so-called offspring, from the characteristics of two or more parent individuals. If a population is asexual, no recombination takes place and an offspring is generated from single parents. Mutation alters an offspring's genome. [2, 4, 14]

For this herein proposed work, genetic algorithms (GAs) were utilized. They mimic Darwinian biological evolution and are based on the evolutionary ideas of natural selection and genetics. The structure of a simple GA is specified in Algorithm 2.1.

Algorithm 2.1: Structure of a simple genetic algorithm [2, 37].

1: Randomly generate initial population of individuals.
2: Evaluate each individual based on the fitness function.
3: **while** n offspring individuals are not generated **do**
4: Select parent individuals from the population based on their fitness scores.
5: Apply recombination operator with crossover probability p_c to generate offspring.
6: Mutate the offspring with mutation probability p_m.
7: Add offspring to the new population.
8: **end while**
9: Replace the current population with the newly formed population.
10: Increase the generation counter.
11: **if** number of maximum generations has not been reached **then**
12: Go to step 2.
13: **end if**

Each iteration of the process described in Algorithm 2.1 is called a generation. And an entire set of generations is called a run or replicate. [37] In GAs, individuals are typically represented either as bit strings (binary encoding) or as real value sets. Experiments described in Chapter 5 were conducted using a binary encoding for individuals (*i.e.,* bit strings), where a genome describes a single solution candidate, a gene being a single bit in the genome and two alleles (*i.e.,* 0 and 1) being available. Moreover, an asexual population was used, which means that no crossover is performed. Depending on the representation of individuals, certain mutation methods are common, such as bit flips in the genomes with a binary encoded individual or changes of the value itself with real-valued encodings. As mutation method, bit flips are used (*e.g.,* a genome of 0010011110 might be mutated to 1010011110). The position that is mutated in the bit string is randomly determined by a uniform distribution. A mutation can have a beneficial, deleterious or neutral effect on the individual, in terms of the fitness score. The selection method plays an important role as well, as it is used to increase the overall fitness within a population. Whether an individual gets selected depends on its fitness. There are several frequently used selection methods, such as proportional selection, lin-

[4] Genomes are abstract representations of individuals and are called chromosomes or genotypes, albeit small differences between those terms exist in biology. The genome is a set of all genes describing an individual. A gene describes a certain characteristic (*e.g.,* eye color) and alleles express the different possible settings for this certain gene (*e.g.,* green, brown, blue). [2]

ear rank and tournament selection. The author decided to use tournament selection "with replacement", due to its conceptual simplicity and quick execution. Local tournaments between k randomly chosen individuals are performed and the winner (*i.e.,* the individual with the highest fitness) is selected for variation. "With replacement" means that the same individuals can be selected more than once to become a parent. This process is repeated until a sufficient number of individuals has been selected for the next generation's population. [37, 69]

GAs represent a powerful tool for finding solutions to complex problems in a variety of application areas, such as economy, research, art and music by applying concepts of natural evolution [15].

2.2 Artificial Life

Life on Earth is incredibly sophisticated: Millions of species, made up from almost innumerable chemical compounds, interact in ways, obscure to the amateur eye. Due to life's long history and its many evolutionary paths, it is hardly possible to extract the fundamentals of life or distinguish pathways, potentially dead-ends in the future, from the general phenomena that formed today's rich ecology. The evolutionary clock cannot be turned back to the beginning of time and we have no other ecosystems that we could compare ours to, to perceive general properties of life. Thus, humankind has to take other actions to find out more about the essence of life. [82] British evolutionary biologist John Maynard Smith (1920-2004) already understood this in 1992:

> *So far, we have been able to study only one evolving system and we cannot wait for interstellar flight to provide us with a second. If we want to discover generalizations about evolving systems, we will have to look at artificial ones. [55]*

Artificial life (ALife) is an area of research that studies systems related to natural life. ALife attempts to understand the fundamental principles of life in a bottom-up manner. Therefor, researches investigate "life as it could be" by synthesizing intelligent, lifelike systems from scratch. [6] ALife is an interdisciplinary research field that overlaps with biology, chemistry, computer science, astrobiology, physics, evolutionary science, origins of life research, artificial intelligence and complex systems [90]. "Artificial life" does not mean that fake life is researched, but that "art" and "life" is combined to a life made by man rather than by nature [32].

The main goal of ALife is to understand what life is, where it came from, where it might go and what it means to be alive [32]. One could say that something that grows and is able to reproduce is alive, but that would mean that simulations in video games where animals grow and reproduce are alive, too, and that is certainly not true. Another point of view is that everything that has a DNA on Earth is alive, but the question of aliveness is not that easy to answer: As it is true that DNA is a trait for living beings on Earth, the function of DNA is the encoding of information that is passed on from parent to offspring. However, this function is not agnostic to DNA, but binary codes in computers encode information as well that can be passed on from parent to offspring in a simulation. Still, neither the computer nor the simulation is alive [90].

In ALife, researchers develop dynamical systems that mimic aspects of biological life, without agreeing on what "life" is [5]. Artificial life systems are useful for studying the Darwinian evolution and for finding methods that can be applied to solve real-world problems. The simulated lifelike processes often consist of highly simplified artificial organisms that allow to compare their behavior to certain behaviors observed in nature. In so doing, researchers aspire to grasp life's essential character. [82]

Biological research focuses on capturing the most important parameters of "life as we know it", whereas ALife tries to understand the most simple and general principles, which are underlying life and models those in simulations as "life as it could be". The simulation allows researchers to study and analyze new lifelike systems. [32]

2.2.1 The History of ALife

The term "Artificial Life" was first coined by the American computer scientist Christopher Langton back in 1987. He described this scientific field of research as follows:

> *Artificial Life is the study of man-made systems that exhibit behaviors characteristic of natural living systems. It complements the traditional biological sciences concerned with the analysis of living organisms by attempting to synthesize lifelike behaviors within computers and other artificial media. By extending the empirical foundation upon which biology is based beyond the cabon-chain life that has evolved on Earth, Artificial Life can contribute to theoretical biology by locating life-as-we-know-it within the larger picture of life-as-it-could-be. [31]*

Actually, the philosophy of ALife, which is creating imaginary beings and engineering lifelike artifacts, is much older than the 1980s, dating back to the earliest stone and clay figurines, through to hydraulic and pneumatic creations built as early as the third and second centuries BC. Philo of Byzantium (about 280-220 BC), a Greek engineer, invented the world's first robot: the Automate Therapaenis, a life-sized mechanical maid holding a wine jug and dispensing wine when a cup was placed in their hand. The wine then was mixed with water. [18]

The French inventor Jacques de Vaucanson (1709-1782) designed not only the first automatic loom, but also a duck with a fake digestive system. The duck was powered by a weight-mechanism consisting of over 1000 movable parts. The digestive system did not chemically breakdown its food, but the duck could eat a pellet, wriggle and a short time later a foul-smelling substance was emitted from its rear end. [32, 58]

John von Neumann (1903-1957), regarded as the foremost mathematician of his time, also researched the conditions for self-replication in cellular automatons[5] and the evolution of complexity. The Norwegian-Italian mathematician Nils Aall Barricelli ran the first ever evolutionary algorithms on computers in the early 1950s and is considered a pioneer in ALife research. And in the 1970s, British mathematician James Horton Conway invented a cellular automaton called "Game of Life" [23].

[5]Cellular automata are discrete models used in automata theory, the study of abstract machines. Moreover, they are used for computational problems that can be solved using such cellular automata by evolving a grid of "colored" cells according to a set of rules [33, 92].

2.2.2 ALife Today and Tomorrow

Today, the artificial life scientific community holds a yearly ALife conference and publishes its own journal called "Artificial Life" [74]. Today's ALife is divided into three subfields [3, 7]:

- Hard ALife; with a focus on hardware, mainly robotics.
- Soft ALife; which uses computational simulations to explore the processes and evolution of software models related to biological life.
- Wet ALife; which adopts approaches from biochemistry to study traditional biology by recreating aspects of biological phenomenas.

Moreover, ALife has always been tied to arts, which could be considered as a fourth subfield. Simulations of ALife are exhibited at ALife conferences and media galleries, to the point where an ALife-based android has conducted an opera [89].

Today, ALife creates simple models that reduce complexity of a biological phenomenon to understand how the real life example works in its core. Simple agents are evolved to learn complex tasks, which are necessary for achieving a goal that is challenging and not very well understood. From those models, researchers hope to gain understanding of the original, biological scenario and draw conclusions that can be incorporated into further models or help to understand traditional biology better. [7]

Especially soft ALife is interwoven with artificial intelligence (AI): Kenneth O. Stanley, research manager at OpenAI[6], addressed the challenge of open ended evolution[7] and created a new class of genetic algorithms[8], which was awarded the "2017 International Society for Artificial Life Award for Outstanding Paper of the Decade" [90].

Apart from AI, ALife is *i.a.,* concerned with swarm dynamics [27], which researches interactions between creatures, and has a big overlap with the origins of life research, despite ALife focusing on all the ways life could have arisen and origins of life trying to answer the question of how life historically arose, as living organisms are no longer just piles of molecules [90]. Another field is artificial chemistry, which focuses on the preconditions of the evolution of life. [18]

ALife today spans a wide range of disciplines, but all together ALife researchers try to understand the fundamental characteristics of living beings, how they emerged and how those characteristics can be reproduced from scratch in an artificial system [18]. Looking into the future, exciting happenings that would advance ALife in theory and practice include synthesizing OEE in an artificial system, the merging of AI and ALife or the discovery of extra-terrestrial life, which would permanently change the approach of researchers to the question of "What is life?" [90]. In the year 2000, a list with 14 open problems in ALife has been published, concerning questions of how life arose from the nonliving, potentials and limits of living systems (including OEE) and the relationship between mind, machines and culture. [8]

[6] OpenAI is a research laboratory founded by Elon Musk, Sam Altman and others. Its mission is to ensure that AI benefits humanity [84].

[7] Open ended evolution (OEE) is the idea that systems can get exponentially more complex with time and that this complexity never stops increasing. Life on Earth is considered such an OEE and the mechanisms of OEE are among the greatest mysteries of modern science. [18, 91]

[8] NeuroEvolution of Augmenting Topologies (NEAT) is used for neural network optimization and focuses on optimizing for diversity and not solely performance [57].

2.2.3 Soft ALife and Digital Evolution

Soft ALife uses the power of computers to better understand the fundamental processes of living systems. Complex models are programmed, which mimic behaviors of the biological world. [28] *E.g.,* bacteria cooperate in many different and interesting ways. Albeit biology has researched the large-scale patterns of bacterial colonies; it is hard to tell what exactly happens at the level of a single bacterium. This is a typical use case for soft ALife to run simulations and find out more about the dynamics of the individuals.

Soft ALife uses digital evolution to perform its experiments. Specifically, digital organisms are often the base for experiments conducted. Digital organisms are evolving and mutating computer programs that self-replicate. They are used to study Darwinian evolution and to test hypotheses or models of evolution. Therefore, digital organisms are closely entangled with ALife. [1, 41, 70]

For simulations, soft ALife frequently utilizes cellular automata, software agents, evolutionary algorithms (especially GAs [38]) and artificial neural networks. Well-known simulators for ALife and digital organisms include Polyworld [72] (1990-present), Tierra [50, 51] (1991-2004) and Avida [42] (1993-present). [6]

2.3 Underlying Biological Models

This section gives an overview of some biological concepts. High-level knowledge of those concepts is crucial to understand the work, presented in this thesis. In particular, this section elaborates on the concepts of major evolutionary transitions, co-evolution and symbiosis, evolutionary lineage data and multi-level selection.

2.3.1 Major Transitions in Evolution

Researchers have been interested in the origins of life for a long time. It is a very human desire, trying to understand where we came from and how life was formed over the Earth's evolutionary history. It is widely believed that planet Earth originated about 4.5 billion years ago and that life first has emerged somewhere between that time and 3.7 billion years ago [45]. Moreover, it is known that the early forms of life were simple microbial mats of coexisting bacteria and archaea. But how did life from back then become as complex as we know life today? Researchers believe that some major upheavals must have happened throughout evolution to transform single-celled life from around 3.7 billion years ago into today's tremendous biodiversity. [40] Although neither a theoretical reason is known nor evidence has been found why evolutionary lineages would increase their complexity over time, this increase of complexity is exactly what we can observe throughout evolution [61].

In their famous work "The Major Transitions in Evolution" [56] John Maynard Smith and Eörs Szathmáry described some of the upheavals that are believed to have led to life as we know it. Those major evolutionary transitions in individuality redefine what it means to be an individual. Such transitions occur when formerly distinct individuals unite to form a more complex lifeform capable of reproducing as a single, higher-level entity and often no longer being able to reproduce individually (see Figure 2.1). The more the history of life was researched the more it became clear that some major transitions in evolution had happened. After the origin of life itself, the evolution of

multicellularity, the transition from asexual clones to sexual populations, the formation of the eukaryotic cell, the upheaval from solitary individuals to colonies, known as the evolution of eusociality and finally, the shift from primate societies to human societies with language (sociocultural evolution) are some of those major transitions in evolution. [56, 60, 61] In general, those transitions in individuality fall into two categories: fraternal and egalitarian transitions [48]. Both will be described briefly in the following subsections, albeit the focus lies on egalitarian transitions, as they represent some of the initial motivation for this thesis.

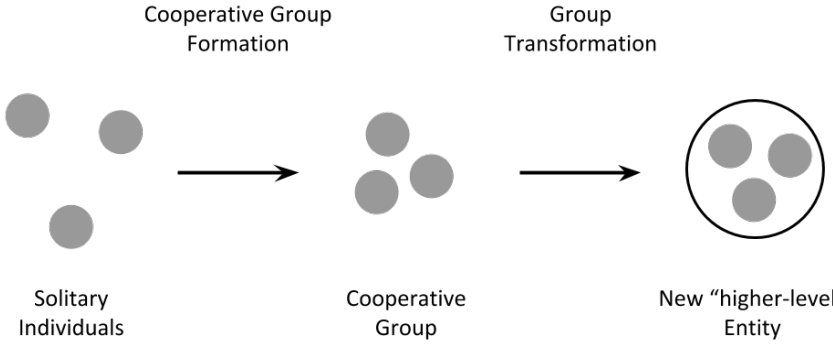

Figure 2.1: Schematic representation of a major evolutionary transition, which is accomplished in two steps: First, a cooperative group is formed and second, the group is transformed to a new, higher-level individual (adapted from [67]).

Fraternal Transitions

Fraternal transitions occur when genetically identical lower-level individuals stay together to form higher-level organisms that subsequently reproduce as one. Examples include *e.g.,* the evolution of multicellularity or of eusocial insect colonies. Multicellularity has evolved from single-celled protists, each of which could survive on its own. Today, those simple protists only exist as parts of larger organisms, such as animals, plants or fungi [36, 52, 61]. Eusocial organisms, such as ants, bees and wasps can survive only as part of a social group. They express complex behaviors, including group decision-making, cooperative care of juveniles and overlapping generations. Effectively, humans also show signs of such social groups. [47, 60, 61]

Egalitarian Transitions

Egalitarian transitions occur when different types of lower-level individuals come together as a higher-level organism to fulfill a united goal [56, 60]. A representative example for an egalitarian transition is the origin of the eukaryotic[9] cell. A simplified version of the so-called symbiogenesis (or endosymbiotic theory) is depicted in Figure 2.2. It

[9]Eukaryotes, together with bacteria and archaea form the domains of life. Eukaryotes are a higher form of organisms, whose cells have a nucleus. Plants, animals and we humans consist of such eukaryotic cells. More generally, everything that is alive and visible by eye consists of eukaryotic cells.

describes the origin of the eukaryotic cell out of prokaryotic[10] ones: A prokaryotic cell ingested another type of prokaryotic cell and the latter became a component of the first. Thus, a higher organism, which today is known as eukaryotic cell, emerged through evolution. Both cell types benefited from the arrangement: In the case of mitochondria[11], the host cell profited from the chemical energy produced by the mitochondrion and the mitochondrion came to appreciate the nutrient-rich and protected environment that the host cell provided [76]. As evolution proceeded, they replicated together and ended up depending on each other. Mitochondria is considered a bacterial endosymbiont[12] in eukaryotic cells, having its origin in symbiogenesis. [76, 35, 61] But how did it come that this conjunction was favored by evolution? Natural selection, as described by Charles Darwin, points out the answer to this question: On average, fitter organisms in terms of succeeding to survive, have a good chance of being selected to pass their genes on to the next generation and produce more offspring than less fit organisms. Survival and/or reproductive advantages can benefit species to outcompete other species or help them to avoid becoming extinct themselves. With eukaryotic cells, the assumption is that they gained advantages over their environment due to joining forces with mitochondria, a rich source of energy [76]. Hence, the ancestor of today's mitochondria was once a free-living prokaryot that formed a symbiotic union with another former separate prokaryotic cell. Today, mitochondria is only able to replicate within a host cell [61]. This whole transition is known as the transition from prokaryotes to eukaryotes and was an evolutionary milestone in the development of multicellular life, as we know it today. The tangible ideas of symbiogenesis as we define it today, have first been introduced by American biologist Lynn Margulis in 1967 [53].

Figure 2.2: Simplified illustration of symbiogenesis (adapted from [76]).

2.3.2 Co-Evolution and Symbiosis

Both, co-evolution and symbiosis are located within evolutionary biology and ecology. Evolutionary biology studies the changes that occur in living beings over time, looks at their generational history and tries to understand their origins. Ecology is a branch of biology focusing on interactions between organisms and with their physical environment. Symbiosis and co-evolution are important phenomenas, found at all levels of life

[10] Prokaryotic cells have a simpler genetic blueprint, without a nucleus. Opposed to eukaryotes, prokaryotes lack mitochondria and any other eukaryotic membrane-bound organelles. Bacteria and archaea are considered prokaryotes.

[11] Mitochondria is a so-called organelle of eukaryotic cells. Organelles are subunits of cells that have very specific functions. Mitochondria is *i.a.,* responsible for the energy production within the cell. They convert biochemical energy from nutrients into adenosine triphosphate (ATP), the energy currency of life. [35]

[12] An endosymbiont is any organism that lives within another organism.

and in almost every context that research has looked for them. Symbiosis describes a close and long-term relationship between two or more organisms from different species. The organisms are called symbionts. Co-evolution is present when two or more species reciprocally affect each other's evolution through the process of natural selection. [63, 68]

In general, three types of symbiosis are differentiated, as Figure 2.3 illustrates. Symbiosis occurs when different species have a long-term interaction, which is sometimes the case after a long time of co-evolution. A symbiosis can either be obligatory, which means that the symbionts depend on each other for survival, or facultative. In that case, symbionts are able to live independently. [80] Each species applies a selection pressure on the other one and thereby, affects each other's evolution. [26, 63]

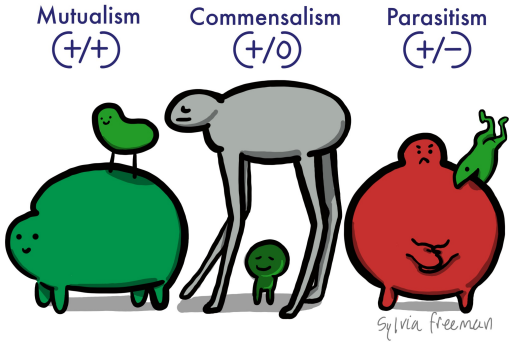

Figure 2.3: Visual representation of the three main types of symbiosis [81].

Mutualism is a specific form of symbiosis, where both symbiotic partners benefit from the relationship. Mutualistic relationships are present in many types of organisms, such as lichens, which are composite organisms consisting of algae or cyanobacteria and fungi species; the Buchnera–aphid symbiosis; and most plant-pollinator pairs. [49, 60] Mutualism is often a side-effect of an egalitarian transition (*e.g.,* in symbiogenesis).

The second type of symbiosis is called commensalism and describes a relationship, from which one partner benefits and the other one is unaffected. The unaffected partner might not even know that another organism is benefiting from them or simply does not care. Barnacles for instance benefit by finding food on whales, while the whale is completely unaffected. Other examples include eukaryotes that bear living cells of bacteria or eukaryotic microorganisms on their surfaces [19].

Parasitism is the third type of symbiosis and describes an interaction between species, where one benefits at the other's expense. Parasitic species usually live inside or on the body of their host and steal resources from them without immediately killing the host. Examples include mosquitoes that drink human blood, or ticks that live on pets. Parasitism seems to be a successful way of life since about 40 percent of all animal species live parasitic. [16]

Despite co-evolution and symbiosis being terms that can be tightly linked, this does not have to always be the case: Although a bee might be in a mutualistic relationship with a flower it does not necessarily has to have co-evolved with that flower. [26] And the three types of symbiosis do not have to be strictly dissociated. Studies have shown that the line between parasitism and mutualism is thin and often can be seen as a

continuum, depending on the environment. This means that the same relationship can be parasitic in one environment and mutualistic under different conditions. [35]

Coming back to Section 2.3.1 and how symbiosis and co-evolution is correlated with the research that this thesis presents, the author would like to introduce microbiomes. Microbiomes are ecological communities formed of microorganism[13] and are found in all multicellular organisms [19]. Among many other appearances, microorganisms are essential for humans, making up the human microbiome, which includes the gut flora [64]. The human microbiome acts pathogenic in many infectious diseases, but plays a vital role in the immune system. Its functions range from symbiosis to pathogenesis [21]. Microbiomes are considered an egalitarian transitions due to plants and animals living in close association with microbial organisms as a synergistic unit, either in a mutualistic, commensal or parasitic relationship [24, 35]. Although it can be hard to identify tightly coupled relationships, it is known that the human gut microbiome is important for the health of its host since it helps with the degradation of non-digestible polysaccharides [64]. This information was one of the reasons why the author decided to conduct experiments that look for tightly coupled relationships in egalitarian populations. Lots of co-evolution is everywhere on this planet Earth and it has a long history, dating back to several of the major transitions in evolution [55], but most of the time and especially in microbial organisms, it is difficult to tell whether long-term co-evolution is going on or if it is just a transitory relationship. Approaches to identify and characterize close microbial associates have already been discovered [43] and this thesis should therefore enhance the understanding of how tight co-evolutionary relationships can be identified in silico. As sequencing technology improves, it will become possible to identify co-evolution from actual biological lineages.

2.3.3 Evolutionary Lineage Data

In biology, a lineage is a continuous line of descent that depicts either a series of organisms, populations, genes or cells, which are connected by ancestor/descendant relationships. It is a subset of a phylogeny, which is mostly depicted as phylogenetic tree and describes the evolutionary relationship among organisms. [77]

The work proposed with this thesis uses evolutionary lineage data to track mutations occurring in individuals. With this mutation tracking, a possibly existing genetic signature should be identified. In traditional biology, a genetic signature describes a pattern of detectable nucleic acids[14] that specify a particular cell or disease [77]. The author slightly modified this definition: A genetic signature in the given ALife context is present, when genomes of individuals are being modified in some sort of systematic way. The software that is used allows to track lineages and hence, the mutations that occur along the line of descent of a single individual are analyzed, ranging from generation zero to the last generation, which has been run. Figure 2.4 depicts a small segment of such a phylogenetic tree, describing the evolutionary relationships within a population.

[13] Microorganisms are also called microbes and describe any form of microscopic organism.

[14] Nucleic acids are naturally occurring chemical compounds, which are the main information-carrying molecules of the cell. The two main classes of nucleic acids are deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). [88]

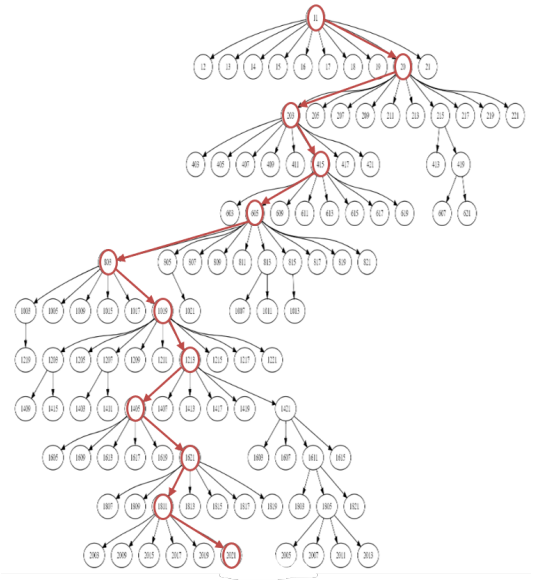

Figure 2.4: This phylogeny was constructed on the basis of experiments described in Chapter 5. A lineage is one path in this phylogeny, *e.g.,* the highlighted one. Each circle represents an individual, and each level one generation.

2.3.4 Multi-Level Selection

Multi-level selection is a controversial concept from biology, where group selection and the more conventional level of selection on the basis of the individual are combined [30]. The author made use of this concept for the second, modified project setup that studies the interaction of different types of selection mechanisms. With multi-level selection, natural selection acts not only on one level, but on both, the group and the individual level [34]. Hence, a selective pressure is present between groups and within groups. In this context, a group describes any subset of interacting individuals that interact more with each other than they would interact with partners selected randomly from the overall population. Such a group could *e.g.,* be a pair of friends that help each other or siblings. From a theoretical point of view, it makes sense that natural selection would favor such a behavior over a "group" that lets each other down. [87]

In the experiments described in Chapter 5, a group consists of two individual cells, arbitrarily named cells of type A and of type B, which together form an organism. Therefore, the within-group selection pressure is measured by looking at the fitness of the individual cells and the between-group pressure is determined by the fitness of the overall organism. The multi-level selection theory does not precisely tell how the selection pressures at various levels are weighted, which is one big point of criticism of this theory [22]. Since the author conducted the experiments in silico with digital evolution, this problem was circumvented by introducing the concept of migration. In general, migration is the movement of individuals between populations [77]. In this certain case, migration means that the individual cells forming the group are able to switch to another organism and thus, change their partner. This will be explained in-depth in the following chapters, starting with Section 3.2.

Chapter 3

Co-Evolutionary Dynamics

The author incorporated organisms into a simulated environment and observed how evolution forms the organisms' behavior over a long period of time. The organisms are implemented as groups of two individual cells, which allows them to evolve co-evolution and even a commensal or mutualistic behavior is possible. The evolutionary goal to which the organisms are exposed, provides the possibility to establish a tight coupling between the lower-level cells that form the higher-level organism. The forming of an egalitarian group, which means that the lower-level cells originate from different types, is forced through the definition of two types of cells: A- and B-cells. The model is set up in a way that A- and B-cells are not able to replicate, whilst they are not combined into a higher-level organism.

This chapter provides an high-level overview of the general solution approach for detecting co-evolutionary dynamics. An understanding of the concepts described in Chapter 2 is expected. In its first section, Section 3.1, this chapter presents the solution approach for answering the research question regarding the existence of genetic signatures in egalitarian populations. Section 3.2 focuses on the second research question, presenting a solution approach for how multi-level selection can be established in an egalitarian population.

3.1 Genetic Signatures of Co-Evolution

This research shows a way of studying genetic signatures of tightly co-evolving populations with lineage-based genetic data. The questions, which are asked in this context, include:

1. Is it possible to identify co-evolution from lineage-based genetic data?
2. Are types of cells linked in their evolution? If yes, is a genetic signature detectable?

And although it is hard to collect biological sequence data for whole lineages at a useful resolution as of now, this work should give an insight into how lineage data from biological systems could be processed in the future as sequencing technology improves. Therefore, useful lineage-based metrics are explored within an ALife environment.

The author differentiates between two ways of how lineage data for tracking mutations is available: First, in a laboratory setting, it can be possible to apply experimental manipulations to the organisms by varying the mutation rate. And second, it might not

be possible due to restrictions or due to lineage data already existing. Of course, historical data should not be a knock-out criterion for detecting a possibly existing genetic signature, as it will often be the case that experimental manipulation cannot be applied in a real-life setting due to various reasons.

The following section, Section 3.1.1, presents all aspects of the solution approach, which are universally utilized for the detection of co-evolutionary dynamics, regardless of whether experimental manipulation is possible. Following, Section 3.1.2 describes the possibilities for detecting genetic signatures of co-evolution when experimental manipulation can be applied. Namely, a fitness score comparison and a mutation count analysis will be presented. Lastly, Section 3.1.3 demonstrates ways for detecting co-evolution when either no experimental manipulation is possible or the lineage data, whose relationships should be identified, is already existing. This is a far more limited situation, but the author was able to cultivate a metric, which is able to detect co-evolution from just historical data. Hence, in that section a fitness score analysis, a mutation type analysis and the Accumulated Mutations Metric will be introduced.

3.1.1 Overall Approach

In this section, general approaches necessary for detecting genetic signatures of co-evolution are described. In particular, this section covers the formation of an egalitarian population, the evolutionary goal (*i.e.,* fitness function), how the genetic algorithm works in principle, what is meant by lineages and how they are constructed, how the presence of a genetic signature is detected, and finally, the idealized scenarios are presented.

Egalitarian Population

The author designed the population in a way that each higher-level organism consists of two different types of lower-level cells: an A-cell and a B-cell, each with their own binary genome consisting of zeros and ones (see Figure 3.1). At birth, an A-cell and a B-cell get linked into one organism. The cells are called the organism's endosymbionts and they are tightly-coupled until death. Since it might be expected that in biology, genetic changes in one partner drive genetic changes in the tightly coupled partner, the author decided to force this tight coupling between A- and B-cells from the beginning. In that way, it is assured that co-evolution between the cells can happen and symbiosis, potentially mutualism, could be established. The population is asexual, which means that no recombination is taking place when a new generation of organisms is drawn from the former one. In order to generate an offspring, only one parent is needed.

The "egalitarian" aspect is therefore ensured by the two different types of lower-level cells and the coming together of those two types into one, single higher-level organism. In contrast, in an fraternal environment all of the lower-levels that form the group would be identical. In this egalitarian environment, A- and B-cells are evolving independently and they both have their independent genetic representation, which means that one can change in a different way than the other. In this way, those cells form an egalitarian organism.

In contrast, a fraternal organism would consist of lower-level individuals that all encode the same, single genome. In nature, it can be thought of egalitarian and fraternal as follows: A eukaryotic cell that consists of the originally human cells and the

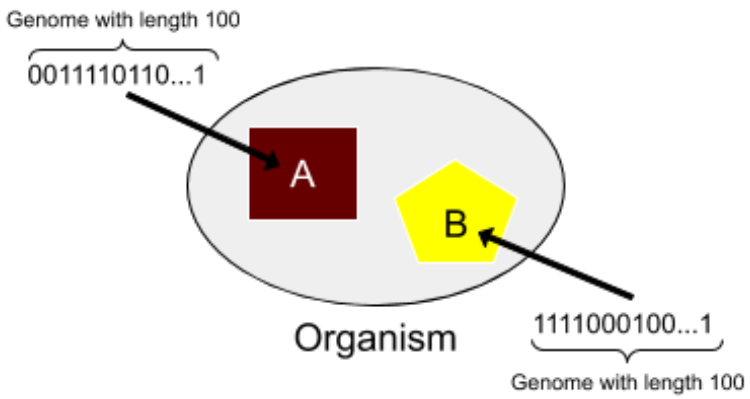

Figure 3.1: This figure shows an organism consisting of a cell from type A and one from type B with their sample genomes.

mitochondrial cells is called egalitarian. More generally, all organelles that live in some sort of higher-level organisms were part of an egalitarian transition in the past. And so, a single cell that consists of two or more different sub-types of cells forms an egalitarian group. When such single cells are combined to some even higher-level entity, this formation is called a fraternal group since all the single cells are equal.

Evolutionary Goal (Fitness Function)

This work was done with simple organisms. Their genomes consist of concatenations of zeros and ones. The evolutionary goal was designed, so that each genome can be subdivided into two parts: leading-ones and the tail-end. The leading-ones part contains all ones from the beginning of the genome until the first zero occurs. The tail-end comprises the rest of the genome. The greater the leading one part is, the better adapted the genome is to the evolutionary goal. The author named the problem that should be solved by evolution "Counting-Leading-Ones"-problem. The goal with this problem is to gain as many leading ones as possible within a genome. In terms of a genetic algorithm, this evolutionary goal is called fitness function.

Each one in the leading-ones part produces a fitness benefit, whereas each one in the tail-end (*i.e.,* after the first zero occurred in the genome) denotes a small-scale decrease in fitness. The higher the fitness of a genome is, the better it is adapted to the evolutionary goal and the higher is the chance that this genome is, in a mutated form, part of the next generation. The number of leading ones (*i.e.,* the number of ones that are in the genome before the first zero occurs) in each of the genomes determines the fitness contribution of that lower-level cell. Every leading one generates a fitness benefit of 1. Ones are only desirable at the beginning of the genomes but not after the first zero has occurred. Specifically, any ones after the first zero in the genome are penalized with a small fitness deduction (FD), which depend on the length of the genome and is

generally described with Equation 3.1.

$$\mathrm{FD} = \frac{b}{2 \cdot l(g)} \tag{3.1}$$

where:

b = benefit of leading one
$l(g)$ = length of genome g

Figure 3.2 shows a toy example to display how the fitness for an A- and a B-cell with genomes of size five is computed. A has two leading ones, whereas B has one. This gives A a initial fitness of two and B a fitness of one. After the first zero has occurred, A has a single one and B has two ones. Therefore, 0.1 and 0.2 is deducted of their scores, respectively, resulting in a fitness score of 1.9 for the A-cell and 0.8 for the B-cell.

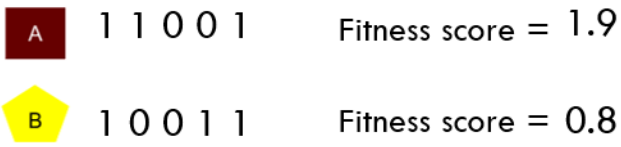

Figure 3.2: This figure shows the fitness evaluation for an toy organism consisting of an A- and a B-cell.

Genetic Algorithm

As described in Section 2.1, a genetic algorithm contains several elements. At the beginning, a population is formed out of organisms. Each higher-level organism consists of a lower-level A-cell and a lower-level B-cell. Each organism is then evaluated by how well it meets the given evolutionary fitness goal. The more adapted an organism is to that goal, the higher are the chances that this organism can propagate its genome to the next generation. After it is determined which organisms are the most fit ones, these organisms undergo the mutation process. During that step, some genomes are varied, in order to preserve genetic diversity. After mutation has happened, the chosen and mutated offspring move on to the next generation and build the new parent population. The evolutionary cycle then begins again.

The organisms that are allowed to propagate their genome to the next generation are determined by tournaments. This means that a bunch of organisms are randomly drawn from the population and they compete against each other. The one organism from that bunch that is best adapted to the evolutionary goal, is the one that wins the tournament and is allowed to propagate its genome to the next generation. Such tournaments are repeated until enough organisms were picked to form a new generation. Of course, it is possible that one organism from the current generation is multiple times the winner of such a tournament. This organism then is lucky since it is allowed to propagate its genome multiple times. On the other hand, this also means that the genetic diversity within the population is shrunk by such very well adapted organisms.

Before the generational step is completed, the genomes from the current population that were selected to be the origin of the next generation, have to undergo potential mutations. Mutations are necessary to (re)gain genetic diversity within the population and allow even better adaption to the evolutionary goal. In this context, a mutation means that a gene in the genome is altered, by changing its value from zero to one or from one to zero.

Lineages

The author researched how mutations in corresponding A- and B-cells occur along the line of descent. An A- and a B-cell are called corresponding, when they together form a higher-level organism. Therefore, such corresponding cells of type A and type B are tightly coupled within the overall population of higher-level organisms.

Questions that were asked in this lineage setup included: Are there any patterns in those mutational changes? Does co-evolution happen? How can this be measured? To answer those questions, single lineages from a type-A cell and its corresponding type-B cell are pulled out. The genotype of those lineages is then tracked and searched for any patterns that occurred between the A- and B-cell.

Along the lineage, every mutation that has occurred somewhere in the genome (*i.e.,* it does not matter whether the mutation occurred at the leading one part or the tail-end) is tracked. Mutations are visualized with the following symbols:

- '+' stands for a mutation that positively affected the fitness score, a so-called beneficial mutation.
- '-' stands for a so-called deleterious mutation, a mutation that negatively affected the fitness score.
- 's' and 'o' stand for mutations that had no effect on the fitness score.
    - 's' stands for a "silent" mutation, which is better known in the biological world as "neutral" mutation. This means that although the fitness score is the same, the genome is different. Therefore, a mutation must have occurred but it had no impact on the fitness score. This can for instance happen, when two mutations occur from one generation to the next, but they cancel out each other. *E.g.,* a one at the tail of the genome mutates to a zero *and* a zero at the tail mutates to a one in the same generation.
    - 'o' stands for no mutation. This means that the fitness score as well as the genome from one generation to the next did not change at all.

Figure 3.3 shows example mutations of A-cells that further illustrate how the different symbols can occur. For B-cells, the process of determining the symbols is identical and hence, not shown.

Presence of a Genetic Signature

For the purpose of answering the question regarding the existence of a genetic signature, patterns in the mutational changes between cell types and changes in the fitness are analyzed. If any signal can be identified from patterns in genetic changes along lineages,

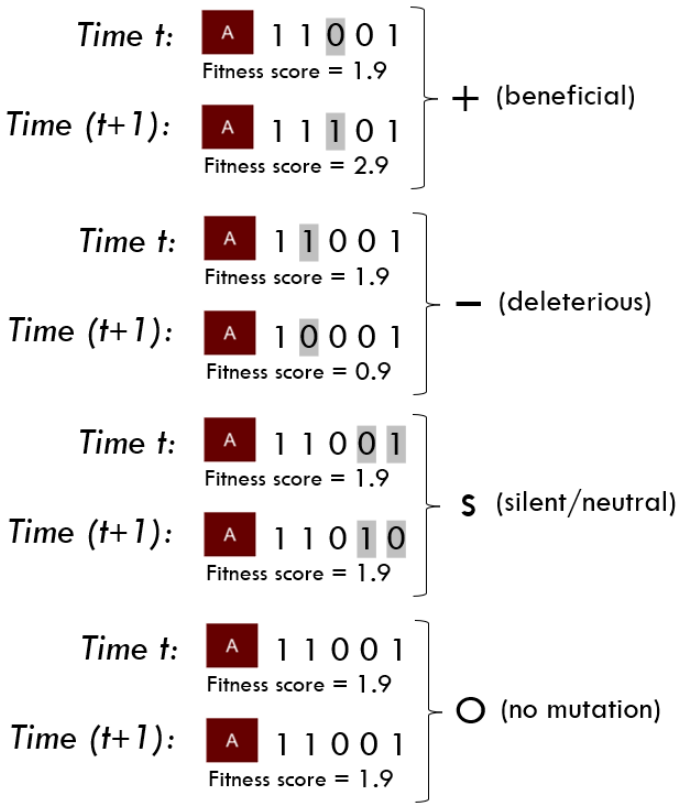

Figure 3.3: This figure shows examples of mutations that lead to the symbols +, -, s and o. The mutated positions in the genomes are highlighted.

a genetic signature is existing. In egalitarian populations three different patterns of genetic signatures are conceivable:

1. Simultaneous genetic changes between cell types. Thus, both sides are mutating in a lock-step coordinated fashion. This means that a mutation in an A-cell is immediately (*i.e,* in the same generation) responded to by a mutation in the coupled B-cell. Only if mutations occur in that manner, a simultaneous pattern is present.
2. Alternating mutations between cell types. In this case a mutation in an A-cell is responded to by a mutation in the coupled B-cell in one of the following generations. After that, this mutation in the B-cell is again responded to by a mutation in the A-cell in any following generational step. Generally, mutational changes in one partner trigger changes in the other partner in a subsequent generation.
3. No patterns are visible. This third possibility serves as a control in which no mutational patterns should be observed at all. In such a case, each lower-level cell is evolving in a similar genetic pattern as it would have evolved in non-cooperative circumstances.

This work focuses on simultaneous mutations and no patterns in genetic changes.

Idealized Scenarios

In order to see different genetic signatures, five idealized scenarios plus one scenario that is smoothly transitioning this "Genetic Signature"-phase into the "Multi-Level Selection"-phase were designed that foreordain how the fitness scores of the two lower-level cells must interact in order to be a successful higher-level organism. Since the selection mechanism is solely acting upon the higher-level organisms, it is crucial how the fitness scores of the cells interact with each other. Only those organisms that adapt and are considered reasonable in regards of the evolutionary goal will eventually prevail. The six scenarios are:

1. No Selection Pressure, where random drift is on both lower-level cells. The fitness for each cell is determined by the fitness function, but the organism's score is always a constant of 1. This means that no selective pressures arise and everything is driven by drift. A- and B-cells evolve completely independent. This scenario acts as control for all other scenarios.
2. Additive Evolution, where the fitness contributions of each of the cells are simply added. The cells' fitnesses are measured in regards of how well they reached the evolutionary goal.
3. Zero-Off Lockstep, where both lower-level cells must have the exact same number of leading ones in order to contribute to the overall fitness. This means that they need to have two mutations (*i.e.,* one in each cell) in the same generation for the mutations to be effective. If they have the same number of ones at the beginning, A's and B's individual fitness is added to get the fitness of the overall organism. If the number of leading ones differs, the organism's fitness is negative with a value of -2.
4. One-Off Lockstep, which is like Zero-Off Lockstep but the number of leading ones can be equal or differ by at most one for the fitness contributions to be added together. It is important to notice that this is a loosened version of the Zero-Off Lockstep. Only one of the two cell types is allowed to fall behind the other by one to preserve the simultaneous behavior. The mutations therefore must either occur synchronous or one is trailing the other, which is a weakened form of simultaneous.
5. One Follows, where one lower-level cell is allowed to be behind the second cell by an unlimited number of leading ones, but it has a strong selective pressure to never be ahead. If it gets ahead, the higher-level organism again gets a negative fitness score of -2 assigned. The author designed this scenario, so that the B-cell must always be following the A-cell in terms of leading ones.
6. Matching-Bits Lockstep (transitioning scenario), where the fitness of the pair is determined by the number of bits that match between the A-cell and the B-cell. The evolutionary goal as described in Section 3.1.1 is discontinued in this scenario.

The scenarios are designed in an idealized way to provide a baseline for what might be expected in more complex models. It is expected that No Selection Pressure shows no pattern in regards of a genetic signature. In Additive Evolution, selection pressures exist on both lower-level cells and the cells each work independently to fulfill the evolutionary goal. Since the A- and B-cells are still coupled, although very loosely, it could be that some slight form of co-evolution can be observed. Of course, it could also be

that no interaction at all is visible. The Zero-Off Lockstep and One-Off Lockstep scenarios should show simultaneous or one-off simultaneous mutations. In those scenarios the cells must work together in order to fulfill the evolutionary goal. Moreover, in the One-Off Lockstep scenario a strong selective pressure is present in the B-cell as soon as the A-cell is one ahead. This is due to the possibility of the A-cell to get another one ahead. In this scenario, that behavior would mean a negative fitness and consequently the non-fulfillment of the evolutionary goal would lead to a distinction of this A- and B-cell's genome. The One Follows theoretically is an Infinite-Off Lockstep since one cell type can fall behind the other by an infinite amount of ones. It is completely experimental whether some sort of genetic signature is visible in this scenario. The transitioning scenario called Matching-Bits Lockstep is also experimental and no expectations can be stated at this point.

As afore mentioned, the evolutionary goal for scenario Matching-Bits Lockstep is different from the one described in Section 3.1.1 under "Evolutionary Goal". In this transitioning scenario, the fitness score (FS) of an organism is computed according to Equation 3.2. The fitness of the individual cells is also the overall organism fitness since the individual fitness scores do not matter in this current phase of the project as selection is always done on the higher-level organism. Since A- and B-cells start off with all zeros the fitness function was designed this way. The intention is to make the cells as lockstep as possible and hence, a bias towards ones gives them an incentive to change. With this fitness evaluation, it is better to match than to not match, but if the bits are matching, it is better to match with a pair of ones than with a pair of zeros. This creates a lockstep-like effect, but not as harsh as with the leading-ones fitness function since the mutations can occur anywhere in the genome and not only on one exact spot to be beneficial in terms of fitness. It is worth mentioning that this scenario is designed to be sign epistatic. This is since two deleterious mutations combined become beneficial in this Matching-Bits Lockstep scenario: Imagine a pair of zeros is matching and gives a fitness benefit of ten. When the zero in the A-cell is changed to a one by a deleterious mutation, the fitness is decreased by nine (since a one has a fitness benefit of one, but the pair of zeros is destroyed). If the zero in the B-cell would have been mutated instead the fitness would have also been decreased by nine for the same reason. If those two mutations are combined, the expectation would be that the organism looses 18 fitness points, but instead it gains two points since a pair of matching zeros was mutated to a pair of matching ones. Therefore, this scenario has a behavior that is called sign epistasis as the individual mutation impact does not line-up.

This Matching-Bits Lockstep scenario is a smooth transitioning point to the multi-level selection phase where the selective pressure leads to rather selecting for A-cells with more zeros, B-cells with more ones and organisms whose A- and B-cell genomes match. The pathway for the higher-level organism fitness has already been set with this Matching-Bits Lockstep scenario.

$$\mathrm{FS} = k \cdot 10 + o \tag{3.2}$$

where:

k = number of matching bits in A and B
o = number of ones in A and B

Since the higher-level organism's fitness score is determined by the fitness scores of the two lower-level cells and the current scenario (except for scenario Matching-Bits Lockstep), the author would like to clarify what this means by further developing the example fitness computation of an organism, previously shown in Figure 3.2. Figure 3.4 shows the fitness scores that those genomes would achieve in the different scenarios, annotated with explanations.

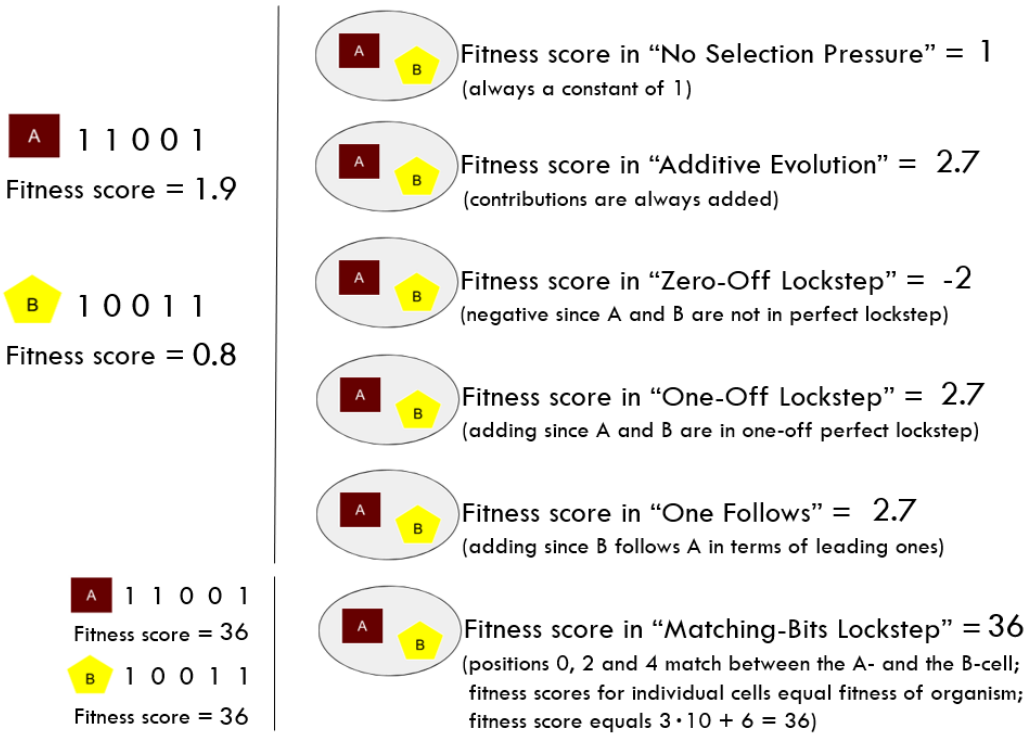

Figure 3.4: This figure shows how the fitness contributions of the A- and B-cell are combined in the different scenarios. Transitioning scenario Matching-Bits Lockstep constitutes an exception from the "Counting-Leading-Ones"-problem, as this figure clarifies.

The author chose to detect genetic signatures in idealized scenarios since it was not possible to take recourse to any preliminary work from other researchers, showing what might be expected. If no genetic signature is visible in idealized scenarios, where such a signature can be expected justifiably, it is for sure that a genetic signature cannot be observed in more complex and lifelike environments with the herein proposed setup.

3.1.2 Detecting Co-Evolution using Experimental Manipulation

Motivation

If it is possible to do experimental manipulation in a laboratory setting, a rather easy way exists for detecting co-evolution from lineages. By manipulating the mutation rates and looking at the mutation counts, as well as the fitness score changes, co-evolution can be identified. If the mutation rate of any particular type of cell (*e.g.*, the B-cell) in

the organism is raised, the expectation is that an increase in the number of accepted mutations can be observed in the other cell-type (*i.e.,* the A-cell in this setup) when the A-cell and the B-cell are truly co-evolving. By implication, this means that if an increase in the total number of accepted mutations is observed in the cell type that has no raised mutation rate, it is a strong signal for co-evolution between the A- and the B-cell. There is no other reason for such an increase than the tight coupling between the two types of cells in the organism.

In a wet lab, it sometimes is possible to do active experimental manipulations in such a manner that the mutation rates of cells that form an organism are different, but it is certainly not always doable. If it is possible, such signs of co-evolution can be highlighted in the real world by looking at the number of accepted mutations along lineages (of *e.g.,* microbiomes). A possible setup could look something like this: Compare 'A' from an environment where 'B' is unaltered to 'A' in an environment where 'B' is mutating faster. Are there differences in how 'A' evolves? If not, there is no evidence that symbiosis is going on. If, however, speeding 'B' up in terms of mutation rate causes 'A' to increase its number of accepted mutations, that is a strong signal that some sort of symbiosis is going on between the A-cell and the B-cell.

The differing mutations rates are tested with the scenarios described in Section 3.1.1 under "Idealized Scenarios". The expectation for the Zero-Off Lockstep and the One-Off Lockstep is that if the mutation rate for the B-cells is pushed up, the A-cells also collect more mutations and have a strong selective pressure to keep up with the B-cells since they are tightly coupled. Conversely, the expectation in No Selection Pressure is that if the mutation rate of B-cells is increased, the A-cells keep doing exactly what they have done before since no co-evolution should be going on in this scenario. In the Additive Evolution, the expectation is not clear. If, for instance, weak co-evolution is there, the expectation is to see some sort of, potentially a quite low, increase in accepted mutations.

With this setup, the author aims to answer the question of "Is the number of accepted mutations in the type-A cell affected by the higher mutation rate in the type-B cell?"

This setup is valuable due to its strong applied focus in biology. Such experiments would *e.g.,* be sensible with microbiomes, especially the engineering of microbiomes to fix gut problems.

Fitness Score Comparison

The fitness score is analyzed to show the fitness contributions of the A- and the B-cell. Those contributions are compared between differing and equal mutation rates and tell how well the organisms were able to adapt to the fitness goal in the course of evolution. The comparison between differing and equal mutation rates should give first signs of whether co-evolution might be expected.

Mutation Count Analysis

In those experiments, the mutation rate of one cell type is raised to see how the evolutionary process is affected. The overall mutations are counted (*i.e.,* the sum of beneficial, deleterious and neutral mutations).

3.1.3 Detecting Co-Evolution from Historical Data

Motivation

The author has already proposed a way to detect co-evolution when it is possible to manipulate the mutation rates by looking at the mutation counts. But what can be done to identify a genetic signature of co-evolution when the lineage data is already there and it is unclear what the relationships between those lineages could be? Since it is not always possible in systems to manipulate the mutation rates, the author searched for a possibility to detect a genetic signature using equal mutation rates. The subsequently proposed metric, the Accumulated Mutations Metric, allows to detect co-evolution with even more limited data from biological systems, where no experimental manipulation is possible.

The question that is answered with this setup reads as follows: "Is a genetic signature detectable when the data is more limited since no experimental manipulation is possible?"

As stated before, experimental manipulation is only possible in few systems since it could be the case that lineages are already there and biologists want to know the relationship between those or, especially in natural systems, it is almost never possible to apply manipulations since the experiments are of an observational approach and natural evolution should not be biased.

Fitness Score Analysis

The analysis of the fitness score shows important information on how well the organisms were able to adapt to the goal, determined by the fitness function, in the course of evolution. This gives an overview on the goal attainment.

Mutation Type Analysis

With that analysis, the author looked at the distribution of the three different types of possible mutations as described in Section 3.1.1 under "Lineages". It is counted how many beneficial, deleterious and neutral mutations accumulate over the lineage of the dominant organism that was selected for further investigation. This analysis acts as control mechanism to prove that evolution works as it is expected and as it can be observed in nature.

Accumulated Mutations Metric (AMM)

The AMM is the analysis, truly responsible for detecting co-evolution: The developed metric looks for simultaneous (or close to simultaneous) genetic changes in tightly coupled lineages. Therefore, the metric tracks mutation accumulation over time (based on [17]). The beneficial mutations are analyzed over time in both types of the lower-level cells that form one higher-level organism. The expectation is that the beneficial mutations occur closely aligned in the lockstep patterns. In principle, the metric measures how many beneficial mutations the A-cell has had more than the B-cell along their lineages. Computing the sample variance of those differentials is a good way to boil the metric down to one number and to identify whether or not a genetic signature is present.

To get even stronger results, the author decided to not look at every single time point but to look at chunks of a hundred time points each. Equation 3.3 shows how the AMM is computed.

$$\text{AMM} = s^2 \begin{pmatrix} a_1 - b_1 & a_2 - b_2 & a_3 - b_3 & \dots & a_n - b_n \end{pmatrix} \tag{3.3}$$

where:

s^2 = sample variance (for details see Equation 3.4)
a_i = number of beneficial mutations in chunk i of the A-cell
b_i = number of beneficial mutations in chunk i of the B-cell
n = total number of chunks

$$s^2 \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix} = \frac{\sum_{j=1}^{n} (x_j - \bar{x})^2}{n-1} \tag{3.4}$$

where:

$\begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix}$ = array with differentials in beneficial mutations
n = sample size (*i.e.,* total number of chunks)
x_j = value of the j^{th} element
$\bar{x}$ = sample mean

In order to be able to use this metric, mutations along a lineage are analyzed. Therefore, it is necessary that the lineage is processed to a "mutation-string", as described in Section 3.1.1 under "Lineages". Since the beneficial mutations of an A-cell should be compared to the beneficial mutations of a B-cell, that preprocessing step is crucial for being able to apply the AMM. A simplified example showing how the metric is computed is presented in Figure 3.5.

The value that results from the number of differences in beneficial mutations is believed to be capable of detecting co-evolution since the more the A- and B-cells differ in the positions of their beneficial mutations, the higher the AMM-value is due to no relationship being existent between the A and the B. So, if A and B are coordinated in the evolutionary process, the expectation is that they get approximately the same number of beneficial mutations at close-by generations, which results in a lower AMM-value.

Nevertheless, the AMM-value alone is not that expressive. On the contrary, only the correlation between the AMM-values for intra-run, inter-run and inter-treatment comparison can tell whether co-evolution is present. These three ways of comparing the mutations of A-cells and B-cells are the true signature feature of the AMM. All three of those comparisons must align with previously specified expectations in order to indeed show a signal that is evidence for the existence of a genetic signature of co-evolution. The three types of comparison, namely intra-run, inter-run and inter-treatment are defined as follows:

- In the intra-run comparison, both lower-level cells actually evolved together and originate from the same replicate. The expectation here is that the value for the Accumulated Mutations Metric is rather small, especially in contrast to the inter-run comparison. The cells that are compared in the intra-run originate from the same organism and hence, they are actually coupled partners and truly belong together.

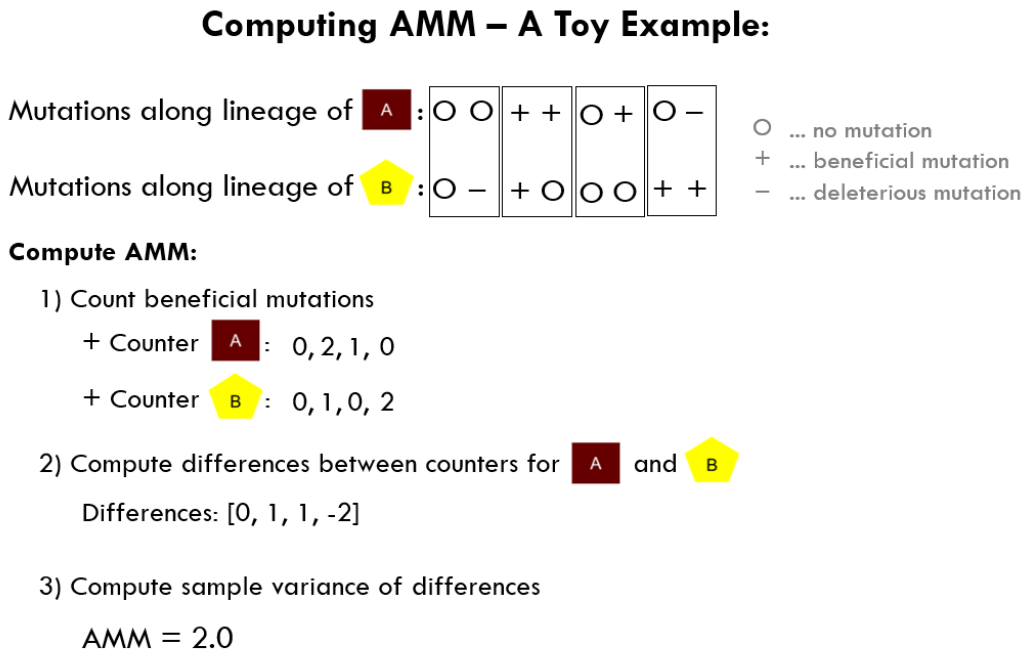

Figure 3.5: This figure shows how the Accumulated Mutations Metric is computed based on a toy example that was run for eight generations. Chunks of two instead of a hundred generations are analyzed, as shown with the black boxes. First, the number of beneficial mutations for each chunk is counted and then, the differences between those mutations in the A- and B-cell are computed. Last, the sample variance of the stored differences is calculated. In this toy example, the Accumulated Mutations Metric is 2.

- In the inter-run comparison, the cells were both evolved with the same scenario, but they are not from the same replicate. If a genetic signature really is present, the value for the metric in this type of comparison must be significantly higher than in the intra-run comparison. A higher value means that the mutation strings of the A-cells and B-cells did not match very well. In turn, this is evidence for a bigger difference at what time beneficial mutations occurred and less a sign for co-evolution and hence, a simultaneous genetic signature. This means that, if a genetic signature is present, these cells that are compared with inter-run should not show a sign of co-evolution since they are not linked at all. They can be thought of as unrelated community members that were undergoing the same selective pressures.
- The inter-treatment comparison takes one lower-level cell from a scenario-replicate and the other lower-level cell from the No Selection Pressure-scenario. If there indeed is a genetic signature, the value of the metric must be drastically higher in this comparison than in the two other comparisons. This comparison provides an upper-bound for what values can be expected.

In conclusion, if a signal is visible in the intra-run comparison, it is necessary to show that this signal is not visible in the inter-run comparison. This draws the conclusion

that the signal has to do with the replicate and not the environment. Furthermore, it is inevitable to show with the inter-treatment comparison that the signal that has been seen during intra-run does not match with another scenario. If all of this is true, a real signal was observed in the intra-run comparison, which means that a genetic signature of co-evolution has been identified. Since the scenario No Selection Pressure acts as control, the expectation is that all three comparisons look identical for this scenario.

The AMM is intended to be exploratory, but there is also a real-life application that could already work in the near future. Imagine, sequencing technology improved a lot and it is possible for biologists to pull out meaningful lineages from a microbiome. If this was the case, AMM would be a screening metric for interactions between lineages. The metric looks at the differences in two lineages and if the metric outputs a low value in comparison to an equivalent population that biologists already know that it is not coupled that would be a strong sign that the tested lineages are truly tightly coupled. The real control of this metric is the comparison of the given value with the value outputted by a population, which is known to not be coupled, when being run through the Accumulated Mutations Metric. If the given value is substantially lower than the value of the known population, a co-evolutionary dynamic in the population of the given value is present. In a laboratory setting, biologists would take a pair of lineages that might be coupled but cannot be known to be coupled a priori and a pair of any other random lineages from the same microbiome that they know cannot be tightly coupled as a control mechanism.

3.2 Multi-Level Selection

So far, selection has only happened on the higher-level organisms and not on the lower-level cells. Moreover, it was imposed that both cells in the organism would always be passed on to the offspring. Hence, mutualism was forced under those idealized conditions. The aim was to show that a genetic signature can be detected when beneficial mutations in both cell types are closely linked to each other. After looking at those preliminary baseline models to get a feeling for what can be expected, the author now starts to look at a broader range of symbiotic behavior where the relative strengths of higher-level and lower-level selective pressures can be adjusted. The major trigger for looking into that direction was the work described in [66], which looked into similar directions of symbiotic behavior but with a resource-stealing aspect and with real-valued organisms[1].

The conclusions from the first phase of the project, which dealt with the existence of genetic signatures of co-evolution, made it clear that a viable next step is to generalize these approaches to tightly coupled symbioses and thus, look at multi-level selection. The main question in this context is: “How do group-level and individual-level selection mechanisms interact?” The author wants to find out how different probabilities of migration affect the group- and individual-level selection pressures. Therefore, the A- and B-cell of an organism must have the possibility to replicate independently (see Section 3.2.1). This is achieved with the introduction of migration, as described in Section 3.2.3.

[1] Vostinar’s Symbulation, an agent-based modeling of symbiont ecology and evolution, allows organisms to lower their fitness in order to harm the fitness of the partner *or* to lower their fitness to help the fitness of the partner.

It is necessary to dis-entangle the organism's cells before this horizontal gene transfer mechanism, which is called migration, can be introduced.

The Accumulated Mutations Metric described in Section 3.1.3 under "Accumulated Mutations Metric (AMM)" has given the author a promising direction for looking more into lockstep-like patterns since it was possible to see co-evolution happen when analyzing the lineages. Therefore, this second phase of the project also uses evolutionary goals that trigger the lockstep pattern. The individuals that form the higher-level organism are disentangled and evaluated separately with different fitness goals (see Section 3.2.2).

3.2.1 Dis-Entangling Organisms

In the setup so far, an organism consists of two cells from independent populations. As soon as such an organism is formed, those two cells stay together forever. To allow multi-level selection in the first place, it is necessary to introduce migration, which requires to decouple the cells within an organism.

With migration it has to be possible that genomes are transferred horizontally (*i.e.,* within the same generation), as Figure 3.6 shows. The counterpart to horizontal gene transfer is vertical gene transfer or better known as mutation, which is an essential variation method in evolutionary processes. In this new setup, migration as well as mutation is enforced.

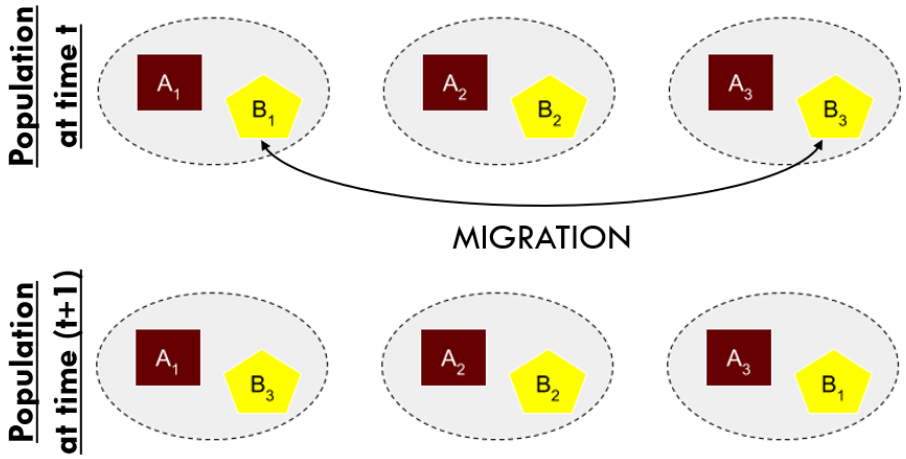

Figure 3.6: This figure shows how migration occurs within a population and what that population looks like at the next time step. In this toy example, the population consists of only three organisms and cells B_1 and B_3 are migrated.

3.2.2 Selection with Conflicting Pressures

As of yet, the two types of lower-level cells have been measured with the same fitness function and the fitness of the higher-level organism has been determined by how well the lower-level cells contribute to the current scenario. In this new setup, three distinct fitness functions or evolutionary goals, as they are called as well, are introduced and

allow multi-level selection with conflicting pressures. Each of the two lower-level populations have their own fitness function plus a different one for the higher-level population. To ensure egalitarian behavior, horizontal gene transfer is additionally allowed while keeping the vertical gene transfer. By varying the migration rate, the author hopes to identify interesting patterns in the interaction of the individual-level and group-level selection mechanism.

On the group level, the fitness is determined by the number of matching bits between the two lower-level cells. Therefore, this first evolutionary goal is also a manifestation of the lockstep pattern. The leading-ones component is no longer used since the results from the first phase of the project support the conclusion that this leading-ones component unnecessarily overcomplicates the evolutionary process. On the individual cell level, the fitness functions discourage matching bits by selecting for the number of zeros and the number of ones, respectively in the lower-level populations of A- and B-cells. Thus, conflicting selective pressures are established. Figure 3.7 shows in an exemplary way how the fitness scores for the individual cells, as well as for the overall organism are computed.

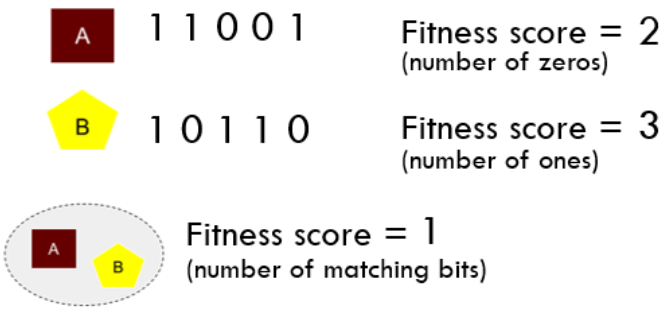

Figure 3.7: This figure shows a toy example for the computation of the three different fitness goals.

3.2.3 Introducing Migration

So far, no migration was possible since the two cells that originally formed the organism sticked together until the very end. In this new setup, migration will be introduced as follows: A migration rate of *e.g.,* ten percent means that ninety percent of the lower-level individual-pairs from the current generation stay together and move on to the next generation as pairs. The pairs that are best adapted to the group-level evolutionary goal (*i.e.,* matching bits between the A-cell and the B-cell) are selected to move on. This is what has happened with hundred percent of the organisms that were selected for the next generation in the former setup. The remaining ten percent of the next generation's population are picked as follows: Ten percent of the A-cell and the B-cell population are picked according to how well they independently meet their evolutionary goals (*i.e.,* number of zeros for A-cells and number of ones for B-cells). Those lower-level cells are then randomly paired, so that one from each of the two populations is a part of the higher-level group. Those, through migration newly-formed organisms move on

to the next generation as well. In that way, a trade-off between group-level fitness goals and individual-level fitness goals is produced and the migration rate can be varied. All selection is done based on how well the organism, the type-A cell and the type-B cell each meet their individual evolutionary goal.

The aim of this work is to identify loose and tight bindings between groups and individuals with migration rates ranging form zero to hundred percent. This work should improve the understanding of who outperforms whom (*i.e.,* organism, A-cell and B-cell) under conflicting selective pressures. It is not far to seek that a migration rate of zero percent should result in perfect organisms, whereas a migration rate of a hundred percent should generate perfect A-cells and B-cells. Nevertheless, the in-between migration rates are the truly interesting ones because in a setup like this, it has not been researched yet how conflicting pressures behave in such an environment. The expectation is that such migration rates will, in biological terms, yield a pretty stable co-existence between commensalism and mutualism. The lower-level cells will develop a commensalistic behavior (*i.e.,* one cell type benefits whereas the other cell type is unaffected since the cell types cannot steal resources from each other) and the higher-level organisms live in a mutualistic relationship since the overall organism benefits from the cooperation of its both cells.

Chapter 4

Implementation

This chapter yields a deeper insight into the technical implementation of the approach described in Chapter 3. It is not necessary to understand this chapter to be able to understand the following chapters regarding conducted experiments, results and conclusions. The source code for the implementation described with this chapter is available at the GitHub repository [73].

First, the overall methodology is described in Section 4.1, then Section 4.2 describes the implementation details for detecting genetic signatures of co-evolution, and lastly, Section 4.3 explains the details of the multi-level selection implementation.

4.1 Methodology

In order to study the co-evolutionary dynamics, the author used a genetic algorithm. GAs imitate natural evolution with the objective of generating efficient solutions for computational problems [4] (for more information see Section 2.1). To implement and evaluate the populations of evolving digital organisms, MABE (Modular Agent-Based Evolution platform) [10], which is described in Section 4.1.1, is used. Python [65] and R [62] are used for data preprocessing and analyses. Details are given in Sections 4.1.2 and 4.1.3, respectively. Finally, Section 4.1.4 gives an insight into High Performance Computing, necessary for conducting the experiments described in Chapter 5.

4.1.1 Modular Agent-Based Evolution Platform (MABE)

MABE is a modular and reconfigurable tool for evolving and analyzing life as it could be. It is implemented in C++ and runs on Windows, Mac and Linux. It is available as a GitHub repository[1] and is used for digital evolution research purposes. MABE creates and manages populations of digital organisms, which are evaluated in so-called worlds. The big advantage is its modularity: MABE provides an accessible framework that facilitates reuse by having implemented common concepts of digital evolution systems and leaving experimentally dependent details up to the user. For those non-common elements, MABE provides standardized interfaces to allow interchangeability. [10, 75, 11]

[1] https://github.com/Hintzelab/MABE

Concepts that are supported in MABE include [10]:

- File I/O
- Data management
- Parameters and configurations
- Population management
- Lineage tracking
- Genomes
- Brains
- Evaluation methods (called “Worlds”)
- Selection schemes

File I/O to lineage tracking are seen as fixed concepts that only need to be implemented once and are the core elements of MABE, whereas the other concepts are seen as fluid and dependent on the experiments. For those latter concepts, MABE provides standardized interfaces (so-called modules) and examples but the user has to implement the details on their own. MABE's generality is only possible because the tool does not make any attempt to define the modules, aside from how they are interacting with the core parts.

Within MABE there are different modules that plug together and let users easily build an artificial life world, where evolution happens in a very natural way. MABE's functional overview is depicted in Figure 4.1 [75]:

- Genomes: Heritable, mutable data. Predefined genomes include circular and polyploid genomes. The author used simple bit-based genomes.
- Brain: Data processors that receive inputs and produce outputs. Examples include Markov brains, constant value brains, genetic programming brains and LSTM brains. No brain was used in this herein presented work.
- World: Worlds evaluate organisms and the author implemented two of them, “DualWorld” and “MigrationWorld”.
- Optimizer: Optimizers take the results of an evaluation performed by a world and a population to select parents and generate a new population. The so-called simple optimizer was used that lets the selection and mutation scheme design up to the user.
- Archivist: Archivists are the centerpiece of MABE since they determine what data needs to be saved and how often. Archivists generate result files in csv-format at the end of a MABE run. All experiments were run with the LODwAP archivist, which saves population statistics and snapshots in regards of organisms that are on the line of descent. This archivist makes it possible to reconstruct the evolutionary history.

Moreover, MABE provides utilities and standalone Python-tools [10, 11]:

Utilities - File manager: keeps track of files that are used during a MABE run.

Utilities - DataMap: is a container that facilitates moving data between different modules. In addition, data that should be outputted at the end of a MABE run is stored in the DataMap.

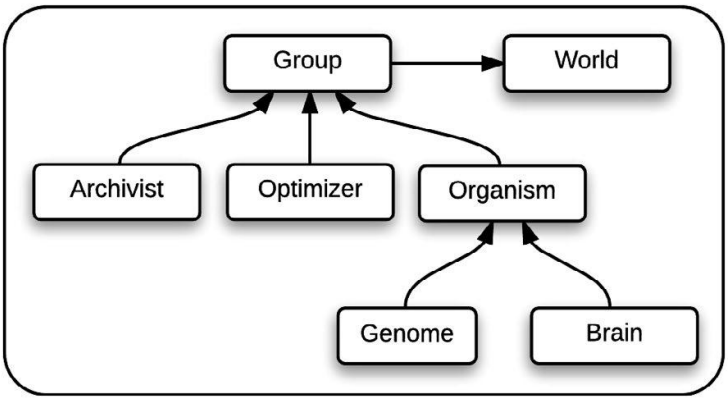

Figure 4.1: Functional overview of MABE. [75]

Utilities - Parameters: are a configuration system that enables the user to set parameters for their worlds, genomes, brains etc. and hence, influence how MABE will operate. Parameters can also be implemented by the user and the author did that with *e.g.,* the mutation rates, the used scenario and the migration rate. Parameters are set in settings files that are used to configure MABE runs. There are three different settings files, which contain all of the possible settings for MABE in the current compilation:

- settings.cfg: The main settings file that controls, among others configurations, the number of updates MABE runs for, which archivist it uses, how large a population is and the random number seed.
- settings_world.cfg: This settings file defines the world that is used plus all parameters that this world needs are set here. The author implemented own worlds and parameters for those worlds and used this file to set them accordingly.
- settings_organism.cfg: This file controls which brain and genome is being used. Since the author used no brain and the genomes are implemented within the worlds, this file is not of relevance for the proposed work.

Standalone - MBuild: automates the process of compiling MABE and creates project files for the user's editor of choice (*e.g.,* Visual Studio, X Code), as well as a MABE executable. The file "buildOptions.txt" specifies which parts of MABE are included in the compilation.

Standalone - MQ: facilitates running MABE many times with different configurations. MQ has also a "-d" flag that, when active, submits all jobs to Michigan State University's High Performance Computing Cluster (HPCC) and schedules SLURM jobs[2] instead of running it on the user's local computer. MQ relies on the file "mq_conditions.txt" that specifies the parameter variations for the runs.

[2]For more details see https://slurm.schedmd.com/documentation.html.

For the herein proposed work, the author made use of the standardized interface, overall helper utilities and both of MABE's standalone tools. The organisms, worlds, fitness functions, selection and mutation methods were designed and implemented by the author. The summarized workflow looked as follows:

1. Implement organisms, selection and mutation scheme.
2. Implement world including needed parameters and desired fitness evaluation.
3. Build MABE with those newly implemented components using MBuild.
4. Configure the file "mq_conditions.txt" accordingly and run several runs using MQ on the HPCC.
5. Pre-process data and generate graphs.
6. Draw conclusions.

4.1.2 Python

Python is a high-level, general-purpose programming language that was created by Guido van Rossum and first introduced in 1991. It supports multiple programming paradigms, such as structured, object-oriented and functional programming. Python is used by many developers around the globe to rapidly develop highly readable algorithms. Python is an intuitive scripting language that uses dynamic type checking and automatically manages memory. It is interpreted at run time. [59, 65]

Python is the author's tool of choice to process the huge amounts of data that are generated with the runs on the HPCC. A preprocessing step is done with the help of two Python-scripts. The first script condenses the raw data. And the second script generates several csv-files that contain the preprocessed data in an aggregated form to facilitate the following steps with R.

4.1.3 R

R is an interpreted programming language for statistical computing and graphics. It is supported by the R Foundation for Statistical Computing and was first introduced in 1992. R is widely used in the field of data analysis and by statisticians for developing statistical software. [62]

R-scrips are used to compute statistical values and tests, as well as to generate graphs. All visualizations that are included in Chapter 5 were drawn with R.

4.1.4 High Performance Computing

Working with digital organism is computationally intensive. Therefore, Michigan State University's ICER (Institute for Cyber-Enabled Research) provides the resources to not only do one of these digital evolution runs but to do many of them. In that way, it is possible to compare how evolution occurs across all of these different instances and, furthermore, develop general principles about digital evolution. In general, HPCs offer significantly greater speed and capacity than machines that are built for commercial use [54]. The ICER offers several High-Performance Computing Clusters (HPCC; see Figure 4.2 for overall layout). Jobs are submitted to the main queue and the scheduler assigns jobs automatically to an appropriate cluster. All clusters run on the Linux distribution

CentOS 7 and use SLURM as resource manager. In total, ICER provides nine clusters, 845 nodes, 22816 cores, 142.3 TB memory and 526 GPUs. More than 500 titles and 3000 different versions of software are accessible out of the box on the HPCC. [83]

Without the HPCC it would be hardly possible to provide the statistical power, which is the rigor behind these experiments, to draw tight conclusions. The HPCC allows researchers at Michigan State University to quickly and effectively run huge amounts of programs in parallel. Thus, the flow of research is not interrupted and researchers do not have to stop their work and wait months for results. The HPCC allowed the author to get results back in a couple of days instead of weeks.

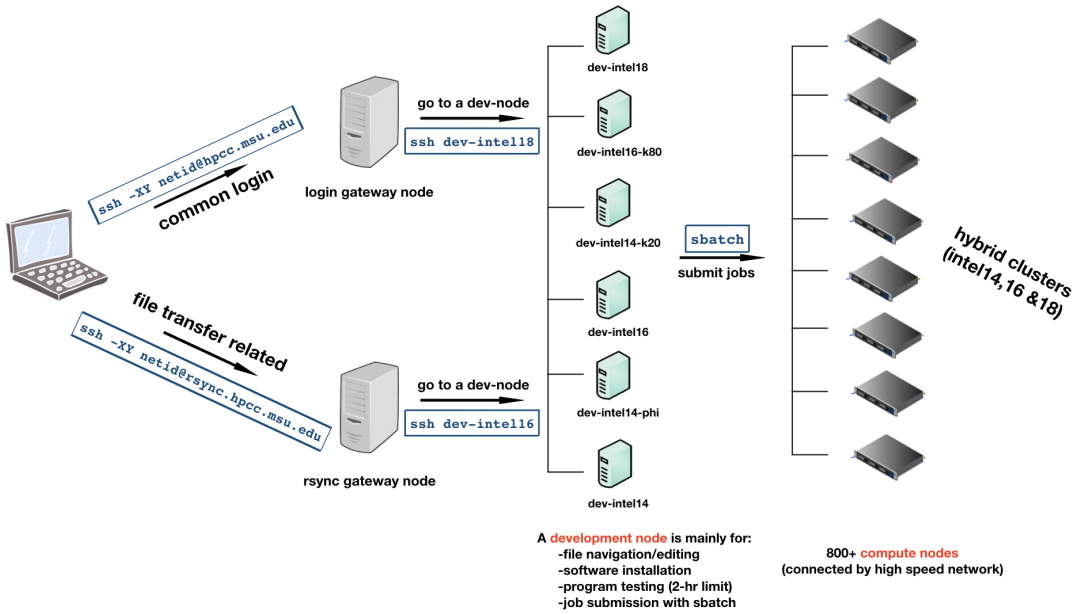

Figure 4.2: HPC layout at ICER. [83]

4.2 Genetic Signatures of Co-Evolution

The implementation details for studying genetic signatures of co-evolution with lineage-based data are described in this section. As in Chapter 3, first the universal implementation details are described in Section 4.2.1, followed by the details on when experimental manipulation is possible (see Section 4.2.2), and when it is not (see Section 4.2.3).

4.2.1 Overall Implementation

The MABE platform was used for the implementation of a genetic algorithm (for more details on MABE see Section 4.1.1). Therefore, a new world called “DualWorld” was designed and implemented.

Egalitarian Population

A higher-level organism always consists of two lower-level cells. One is always from type A and the other is from type B. Each cell has its own bit-based genome that is encoded as bit string, as previously shown in Chapter 3 with Figure 3.1. Initially, the bit strings start off with all zeros and a fixed genome size of 100. The reason why genomes start off with all zeros and not *e.g.,* random, is that all switches away from a zero are fitness changing - either in a positive or negative way. Each cell is separately evaluated by the fitness function.

In the MABE environment, two classes were implemented to allow egalitarian behavior. An instance of class "DualAgent" (see Program 4.1) is the equivalent of an higher-level organism and an instance of class "Agent" (see Program 4.2) is a lower-level cell. The global constant "tagSize" determines the genome's length, which is in all conducted experiments 100.

```
1 class DualAgent {
2   public:
3   std::shared_ptr<Agent> A; // cell of type A
4   std::shared_ptr<Agent> B; // cell of type B
5   double score = 0.0;       // organism score
6   DualAgent() {}
7   DualAgent(std::shared_ptr<Agent> A_, std::shared_ptr<Agent> B_) : A(A_), B(B_) {}
8 };
```

Program 4.1: Source code of class DualAgent (C++).

```
1 class Agent {
2 public:
3   std::shared_ptr<Organism> org;  // MABE organism
4   std::bitset<tagSize> genome;    // bit-based genome of length 'tagSize'
5   double score = 0.0;             // cell score
6   double dualScore = 0.0;         // score of corresponding organism
7   Agent() {}
8   Agent(std::shared_ptr<Organism> org_, std::bitset<tagSize> genome_) : org(org_),
      genome(genome_) {}
9 };
```

Program 4.2: Source code of class Agent (C++).

Moreover, three parameters that are set in a configuration file, were added to class "DualWorld". Those parameters are added to MABE's parameter map and can be accessed easily throughout the world (see Program 4.3).

```
1 static std::shared_ptr<ParameterLink<std::string>> scenarioPL;   // current scenario
2 static std::shared_ptr<ParameterLink<double>> aMutationRatePL;   // mutation rate for A
3 static std::shared_ptr<ParameterLink<double>> bMutationRatePL;   // mutation rate for B
```

Program 4.3: Parameters of DualWorld (C++).

Evolutionary Goal (Fitness Function)

The fitness function is programmed to fulfill the evolutionary goal described in Section 3.1.1 under "Evolutionary Goal (Fitness Function)". The author calls the fitness function "Counting-Leading-Ones"-problem since the consecutive ones from the beginning are counted and bring the most fitness benefit for an individual cell. As mentioned before, the fitness contribution of a lower-level cell is mainly determined by the number of leading ones in the genome. Additionally, ones after the first zero in the genome are penalized on a small-scale with a fitness deduction (see Equation 3.1 in Section 3.1.1, "Evolutionary Goal (Fitness Function)" for details). Since the genomes are all of length 100 and a leading one means a fitness benefit of 1, a fitness deduction of 0.005 is applied for every one occurring in the tail-end.

The author decided that the tail-end of the genome should not be neutral in terms of fitness but can decrease the fitness. Ones that occur in the tail-end are punished with small fitness decreases as described above. This is necessary to reduce noise in the genomes to an acceptable amount. Also, including the tail-end of the genome and not only looking at the leading-one part has the advantage of introducing a small tie-breaker in terms of fitness scores that has a great impact on which genomes are selected for the next generation. Since the GA uses tournament selection, it is guaranteed that very small differences in fitness scores matter a great deal.

In the fitness deduction formula (see Equation 3.1 in Section 3.1.1, "Evolutionary Goal (Fitness Function)"), the genome length is multiplied by two since each higher-level organism consists of two lower-level cells and a leading one that newly occurs in one of the cells should never be outperformed by its tail. The setup shown in Figure 4.3 clarifies this reasoning.

One leading one must always outweigh the tail with its ability to shift a negative organism's fitness score to a positive one. Would the genome length not be multiplied by two, a gained leading one could be outperformed by the tail, as shown in Figure 4.4. Since such a behavior is not desired, the genome length is always multiplied by two.

From the implementation perspective, the fitness function is shown in Program 4.4. A second function that does not take the tail-end into account was necessary as well for scenarios Zero-Off Lockstep, One-Off Lockstep and One Follows to determine whether A-cells and B-cells act according to the given scenario (see Program 4.5).

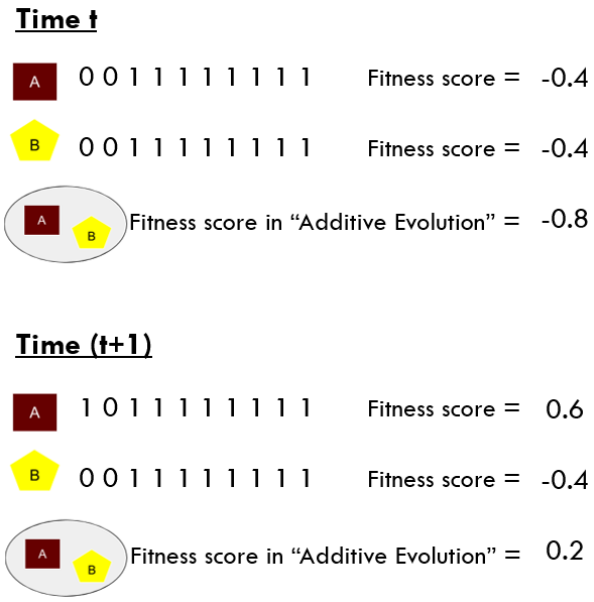

Figure 4.3: Correct and desired evaluation of the genomes. A tail-end one leads to a fitness decrease of $\frac{1}{2 \cdot 10} = 0.05$ in this toy example with a genome length of 10.

Figure 4.4: Unwanted evaluation of the genomes. A tail-end one leads to a fitness decrease of $\frac{1}{10} = 0.1$.

```
1 double DualWorld::countInitialOnes(std::bitset<tagSize>& testGenome)
2 {
3   bool allOnes = true;
4   double score = 0.0;
5   for (int i = tagSize - 1; i >= 0; i--)  // iterate over whole genome
6   {
7     if (testGenome[i] == 0)               // tail-part of the genome begins
8     {
9       allOnes = false;
10    }
11
12    if (allOnes && testGenome[i] == 1)    // leading-one is detected
13    {
14      score++;
15    }
16
17    if (!allOnes && testGenome[i] == 1)   // one at the tail-part is punished
18    {
19      score -= (1.0 / (tagSize * 2));
20    }
21  }
22  return score;
23 }
```

Program 4.4: Source code of fitness function (C++).

```
1 double DualWorld::countInitialOnesNeutral(std::bitset<tagSize>& testGenome)
2 {
3   double score = 0.0;
4   for (int i = tagSize - 1; i >= 0; i--)
5   {
6     if (testGenome[i] == 1) // leading-one
7     {
8       score++;
9     }
10    else
11    {
12      return score;         // tail-part begins; break as all leading ones have been found
13    }
14  }
15  return score;
16 }
```

Program 4.5: Source code of fitness function focusing solely on number of leading ones without considering the tail-end (C++).

Genetic Algorithm

The individuals are incorporated into a simulated environment and it is observed how evolution shapes the individual's genome over a long period of time. For the formation of an egalitarian organism, two distinct populations of individuals are used, such that the organism represents a tight link from one individual of the first population to one individual of the second population. These higher-level organisms each replicate, copying both lower-level cells with them. The developed evolutionary algorithm is more specifically a genetic algorithm and uses tournament selection to manage the evolutionary process and per-site mutation rates for the lower-level cells. Selection is always done on the higher-level organisms, whereas the mutations, which are simple bit flips, are performed on the lower-level cells. Algorithm 4.1 describes the high-level implementation.

Algorithm 4.1: Genetic algorithm used in DualWorld for the detection of genetic signatures (C++ style pseudo code).

```
1:  procedure evaluate(groups, updates)
        Input: groups, MABE parameter for populations of type-A and type-B cells;
        updates, number of generations to run.
2:      initializePopulation(popA, popB, popDual, groups, popSize) ▷ initialize with zero
3:      while update! = updates do          ▷ iterate while not all updates are finished
4:          for i ∈ popSize do
5:              scoreDual[i] ← evalDual(popDual[i])                  ▷ see Algorithm 4.2
6:          end for
7:          for i ∈ popSize do
8:              newDualAgent ← doSelection(popSize, scoreDual, popDual, groups, 7)
9:              mutateSelection(newDualAgent)                        ▷ see Algorithm 4.3
10:             popDual.push_back(newDualAgent)
11:             popA.push_back(newDualAgent.A)
12:             popB.push_back(newDualAgent.B)
13:         end for
14:         groups["A::"].archive()   ▷ MABE functionality needed for lineage tracking
15:         groups["B::"].archive()   ▷ MABE functionality needed for lineage tracking
16:         update + 1                                               ▷ update is finished
17:     end while
18: end procedure
```

The genetic algorithm uses tournament selection with a group size of seven. A tournament size of seven means an intermediate selection pressure of being a fit individual. The tournament size determines how strong this pressure is: A group of two would mean a pretty low pressure (*i.e.,* if a weak individual competes against an even weaker one, it still wins and moves on its genome to the next generation), whereas a tournament size that equals the overall population size would mean an immense selective pressure (*i.e.,* only the most fit individual from the whole population is allowed to move its genome on to the next generation). The size of the tournament is also important to retain genetic diversity amongst a population. With a too low tournament size, it is dangerous that

Algorithm 4.2: EvalDual function called from within the GA-implementation for the evaluation of a DualAgent (C++ style pseudo code). This algorithm shows examples for scenarios Additive Evolution, Zero-Off and One-Off Lockstep.

```
1:  function evalDual(dualAgent)                    ▷ evaluate DualAgent based on scenario
        Input: dualAgent, the organism that is evaluated.
2:      if scenarioPL = Additive Evolution then
3:          dualAgent.A.score ← countInitialOnes(dualAgent.A.genome)
4:          dualAgent.B.score ← countInitialOnes(dualAgent.B.genome)
5:      else if scenarioPL = Zero-Off Lockstep then
6:          initialOnesA ← countInitialOnesNeutral(dualAgent.A.genome)
7:          initialOnesB ← countInitialOnesNeutral(dualAgent.B.genome)
8:          if initialOnesA = initialOnesB then
9:              dualAgent.A.score ← countInitialOnes(dualAgent.A.genome)
10:             dualAgent.B.score ← countInitialOnes(dualAgent.B.genome)
11:         else
12:             dualAgent.A.score ← −1.0
13:             dualAgent.B.score ← −1.0
14:         end if
15:     else if scenarioPL = One-Off Lockstep then
16:         initialOnesA ← countInitialOnesNeutral(dualAgent.A.genome)
17:         initialOnesB ← countInitialOnesNeutral(dualAgent.B.genome)
18:         if initialOnesA = (initialOnesB + 1) ∨ initialOnesA = initialOnesB then
19:             dualAgent.A.score ← countInitialOnes(dualAgent.A.genome)
20:             dualAgent.B.score ← countInitialOnes(dualAgent.B.genome)
21:         else
22:             dualAgent.A.score ← −1.0
23:             dualAgent.B.score ← −1.0
24:         end if
25:     end if
26:     dualAgent.score ← dualAgent.A.score + dualAgent.B.score
27:     dualAgent.A.dualScore ← dualAgent.score
28:     dualAgent.B.dualScore ← dualAgent.score
29:     addToDataMap(dualAgent)    ▷ MABE functionality needed for lineage tracking
30:     return dualAgent.score
31: end function
```

the overall evolutionary goal cannot be fulfilled and a too high tournament size entails the risk that fixation of that specific genome (*i.e.,* this genome is the only one within the whole population that moves on to the next generation) arises. This leads to a loss of genetic diversity and can make it difficult to achieve the evolutionary goal. This is why the author decided to use an intermediate selective pressure of being fit. The goal of each individual is to replicate and the selection mechanism determines those agents that are truly allowed to do so.

No crossover is used since the aim of this project is to keep things as simple as possible to begin with. This is a first exploratory work into how lineage-based data can be used

Algorithm 4.3: MutateSelection mutates the offspring that was previously selected through tournaments (C++ style pseudo code). It is also called from within the GA-implementation.

```
1: function mutateSelection(newDualAgent)
       Input: newDualAgent, the newly selected organism.
2:     numberOfMutations = Random::getBinomial(tagSize, aMutationRatePL)
3:     for i ∈ numberOfMutations do
4:         newDualAgent.A.genome.flip(Random::getIndex(tagSize))
5:     end for
6:     numberOfMutations = Random::getBinomial(tagSize, bMutationRatePL)
7:     for i ∈ numberOfMutations do
8:         newDualAgent.B.genome.flip(Random::getIndex(tagSize))
9:     end for
10: end function
```

to detect relationships among different cells in an egalitarian population. Therefore, "is allowed to replicate" means that the organism is able to move on its genome to the next generation, either completely identical (*i.e.,* if no mutation occurred) or slightly modified by mutations.

The mutation rate is implemented as a per-site mutation rate. This means that the mutation rate states the probability for each bit (*i.e.,* per site) in the genome to be mutated. The alternative would be a general mutation rate that states how many percent of the whole population is mutated. In this herein presented experimental design, a per-site mutation rate of *e.g.,* 0.01 means that a mutation occurs on a specific site with a chance of one percent. Therefore, this mutation rate results in one mutation per lower-level cell and generation in average due to the fixed genome length of 100. To determine the exact number of mutations (*i.e.,* zero, one or more) for each agent, a binomial distribution is used. A mutation results in a single bit flip. This means that the bit at the mutated position is flipped from 0 to 1 or vice versa.

To demonstrate that the algorithm is robust in terms of population size and mutation rate, different combinations of per-site mutation rates (0.001, 0.003, 0.01, 0.03) and populations sizes (10, 100, 1000, 10000) were used for the experiments. A population size of 1000 and a mutation rate of 0.01 served as pivot. Each replicate ran for 5000 generations.

A mutation rate that is too high carries the risk of a melt-down within the population. In that case, no meaningful results are possible anymore since random and unregulated behaviors occur. However, a too low mutation rate also carries risks: When too little genetic diversity within the population is achieved, some genomes will fixate[3] within the population and no new solutions are possible, thus, potentially resulting in a non-attainment of the evolutionary goal. The mutation rate that is just right for a evolutionary problem, permits as many mutations as necessary but not more. If this is the case, a possibly existing pattern in mutations is best identifiable.

Same goes with the population size: If the population is too small, no reasonable

[3] In biological terms, fixation means only one allele remains in the population and every individual has the exact same value for this allele.

results can be achieved and if it is too high, the genetic algorithm has to run for more generations in order to achieve good results.

Lineages

For tracking the lineages, LODwAP (Line of Descent with Aggressive Pruning) is used. Line of descent describes kinship between an individual and its ancestors. In the case of a cell, the line of descent is a list of that cell's parent and their parents' parent and so on. The aggressive pruning aspect helps to erase unneeded data from memory: LODwAP periodically checks if there has been coalescence (*i.e.,* a single most recent common ancestor for the current generation). If this is the case, the algorithm can for sure tell that the chosen cell for the lineage is on the line of descent. [75]

With LODwAP, at the end of a run, one organism (*i.e.,* one cell of type A and one cell of type B) is chosen and its lineage is tracked back from the last generation up to generation zero. A run technically terminates after 5000 updates/generations, but it runs another 1000 updates to give the algorithm more time to find the single most recent common ancestor. If it can be found, coalescence is reached. It is necessary to run those additional 1000 updates to assure that at generation 5000 an organism that indeed is on the line of descent gets picked. MABE outputs one file per cell (*i.e.,* two files per replicate) that contain the lineage of one A-cell from the population and its corresponding B-cell. Such a file contains the following columns for each generation:

- Time to coalescence, which is the time until the most recent common ancestor is found.
- ID of the cell.
- An alive-flag that tells whether the cell was alive at this generation.
- The bit-based genome of the organism. This is necessary to be able to generate the "mutation-string" in a further preprocessing step. The genome makes it possible to track and reconstruct the mutational changes along the lineage.
- The genome score, which is the fitness score of the individual cell.
- The score, which is the fitness score of the overall organism.
- The time of birth, which states the generation at that the cell was born.

Presence of a Genetic Signature and Necessary Data-Preprocessing Steps

This work proposes several ways to detect co-evolutionary genetic signatures from lineages. Those are described in-depth in Sections 4.2.2 and 4.2.3.

All experiments were run on the HPCC. Therefore, it was necessary to process the raw data that resulted from the 16500 MABE replicates before the data could be moved from the HPC to the author's local machine for further analyses.

The replicates were run with the help of MQ, a MABE tool that allows to run MABE many times while varying parameters in a controlled way on the HPCC [75]. So, evolution was observed many times while trying different experimental conditions.

After getting the results from the single replicates, which were two files per replicate (*i.e.,* one with the line of descent of the selected A-cell and one with the LoD of the corresponding B-cell) the author ran two Python-scripts in succession that processed the data for the subsequent visualizations with R.

The first one compresses the data to one huge csv-file that is the base for all further steps. With those strings the mutational changes along the lineage are tracked with the symbols '+', '-', 'o' and 's' as described in Section 3.1.1 under "Lineages". The resulting csv-file contains the following information for all replicates:

- Experiment: This column contains the configuration details of the run (*i.e.,* population size, scenario, mutation rate of A-cell, mutation rate of B-cell).
- Replicate: In order to be able to rerun the experiments, the random number seed with which the replicate was conducted is saved in this column.
- Cell: This column tells whether it is the mutation string of the A- or the B-cell.
- Fitness-Score: This column contains the fitness of the individual cell after 5000 updates.
- Mutation-String: The mutation-string with a length of 5000 (since 5000 generations were run) is stored in this column and generated out of the line of descent. It consists of four different symbols as described in Section 3.1.1 under "Lineages" and tracks all mutations that have happened along the lineage.

In a second preprocessing step, this csv-file is split up in several smaller ones that are in turn used for detecting genetic signatures in Sections 4.2.2 and 4.2.3. Different methods to look for a genetic signature are described in those sections, dependent on whether experimental manipulation is applicable.

Idealized Scenarios

The idealized scenarios have already been described in depth in Section 3.1.1 under "Idealized Scenarios". Algorithm 4.2 describes the implementation details. And following, some additional information regarding the scenarios is provided:

- In scenario No Selection Pressure, the organism's score is always a constant of 1.0 to not provide any selective pressures during the tournament selection.
- In all scenarios except Matching-Bits Lockstep, a cell's fitness is measured with the "Counting-Leading-Ones" fitness function.
- In Zero-Off Lockstep, A and B must be in perfect sync, whereas in One-Off Lockstep they must either be in perfect sync or B is allowed to be one leading one behind A.
- In One Follows, B can be behind A by an infinite amount of leading ones, but it can never have more leading ones than A.
- In the Zero-Off Lockstep, One-Off Lockstep and One Follows scenarios, a negative fitness score means a fitness score of -2 for the higher-level organism. Each cell gets assigned a fitness score of -1 when they are not behaving in accordance with the scenario. Since the fitness values of both cells are always added up for the organism's fitness score, a misbehavior according to the scenario's evolutionary goal results in an organism's fitness score of -2.
- Matching-Bits Lockstep forms an exception from the otherwise consistent fitness goal, named "Counting-Leading-Ones"-problem.

4.2.2 Detecting Co-Evolution using Experimental Manipulation

This section gives an overview of the general motivation and subsequently presents the analyses conducted to detect co-evolution using experimental manipulation, namely the mutation count analysis and the fitness score analysis.

Motivation

If a speed-up of A-cell's mutation rate unleashes an increased amount of accepted mutations in the B-cell, although the mutation rate stayed the same, a signal of co-evolution is found. Besides the analysis regarding the number of accepted mutations, the fitness contributions of the A- and B-cell after 5000 generations are explored. Since "DualWorld" has two mutation rate parameters implemented (one for A and one for B; see Program 4.3) it was trivial to run experiments with different mutation rates.

Fitness Score Comparison

The fitness scores are analyzed to get a first idea of whether co-evolution might be present. Therefore, the population is analyzed after the last generation to see how A- and B-cells each contribute to the overall fitness. The general approach of this comparison is described in-depth in Section 5.1.1 under "Fitness Score Comparison - Overall Remarks".

Mutation Count Analysis

These analyses show the overall number of mutations that occurred along a lineage on the y-axis, split-up by the cell in which the mutations have occurred on the x-axis. Several replicates were run and the results can be seen in Chapter 5. The general approach of this comparison is described in-depth in Section 5.1.1 under "Mutation Count Analysis - Overall Remarks".

4.2.3 Detecting Co-Evolution from Historical Data

Again, the motivation for detecting co-evolution using historical data is described, followed by details on the fitness score analysis, the mutation type analysis and the implementation of the AMM.

Motivation

Since experimental mutation is mostly not possible, especially when lineage data already exists and the relationship between lineages should be detected, it is necessary to gain understanding of how genetic signatures can be detected with equal mutation rates for lower-level cells. In these experiments, the mutation rate parameters in "DualWorld" had the same value for the A-cell and the B-cell. To ensure that evolution is working correctly, first, the fitness score analysis and the mutation type analysis were performed. However, the main contribution of this work is the development of the Accumulated Mutations Metric.

Fitness Score Analysis

Basically, this analysis provides an insight into how well the organisms are adapted to their environment. Foremost, this analysis is of a descriptive nature and should show that evolution works as it is supposed to do. The general approach of this comparison is described in-depth in Section 5.1.2 under “Fitness Score Analysis - Overall Remarks”.

Mutation Type Analysis

The occurrences of the different mutation types (*i.e.,* beneficial, deleterious and neutral) are analyzed for each replicate that was run. This analysis should provide evidence that the artificial evolution presented with this research, works as it is observed from evolution in nature. The analysis was done for the overall number, as well as for the fractions to make the data better comparable throughout different configurations (*i.e.,* in terms of population size and mutation rate). Again, the general approach of this comparison is described in-depth in Section 5.1.2 under “Mutation Type Analysis - Overall Remarks”.

Accumulated Mutations Metric (AMM)

As Figure 3.5 from the previous Chapter 3 has illustrated, the metric basically counts the pluses from the mutation records for an A- and a B-cell at certain time points and subtracts those values. Therefore, the metric looks at the number of beneficial mutations that the A-cell has had more than the B-cell. To boil it down to one number, the variance of those differentials is computed. Program 4.6 shows the source code implementation for the AMM.

```
1 import statistics
2
3 def compute_amm(genome_a, genome_b):
4   chunk_size = 100
5   differentials = []
6   if len(genome_a) == len(genome_b):
7     for i in range(0, len(genome_a), chunk_size):
8       beneficial_mut_a = genome_a[i:i+chunk_size].count('+')
9       beneficial_mut_b = genome_b[i:i+chunk_size].count('+')
10      differentials.append(beneficial_mut_a - beneficial_mut_b)
11  return statistics.variance(differentials)
```

Program 4.6: Source code of Accumulated Mutations Metric (Python).

As mentioned in the solution approach, the metric does not look at every generational time step, but a sliding window was implemented that looks at a hundred generations at once. Each generation is only looked at one time since the window with a fixed size of 100 moves in independent steps, not in overlapping ones. This allows to look at the difference in beneficial mutations between the A-cell’s and B-cell’s lineage over hundred generations at a time. The advantage in comparison to an overlapping window is that one big event that happened at some point in A but not in B does not forever

keep the number of beneficial mutations apart. And the advantage to the previously tested window size of one, where the beneficial mutations were added over time, is that the beginning of the lineages (and especially the differences there) did not bake more into the variance than differences at the end of the lineages. This was in fact a risk for falsifying the results, but it could be circumvented with the final implementation of AMM, where chunks of hundred generations each are analyzed.

The different comparisons (*i.e.,* intra-run, inter-run and inter-treatment) have already been described in Section 3.1.3 under "Accumulated Mutations Metric (AMM)". Each replicate was run with a different random number seed to guarantee independence between all results. As already mentioned, the true expressiveness of the AMM lies less in the actual value and more in the arrangement of the AMM-values for intra-run, inter-run and inter-treatment comparison. The resulting heat maps were generated for the intra-run comparison to get a feeling for what can be expected. More importantly, the generated box plots show the AMM-value that measures whether co-evolution is occurring on the y-axis and the different types of comparisons (*i.e.,* intra-run, inter-run, inter-treatment) that describe the relationship between the A- and the B-cell on the x-axis. Additionally, the pairs in the inter-run and inter-treatment comparisons were switched to make sure that there is no bias in the random pairing process and looked at those graphs as well. The general approach of this comparison is described in-depth in Section 5.1.2 under "Accumulated Mutations Metric (AMM) - Overall Remarks".

4.3 Multi-Level Selection

The codebase remains the same except for some minor changes. "DualWorld" was reimplemented as "MigrationWorld", which has three different fitness functions for the A-cell, the B-cell and the overall organism. Also, the possibility of migration (*i.e.,* horizontal gene transfer), in addition to the already available vertical gene transfer (also called mutation), was added to the implementation of the GA.

The scenario-parameter was removed and a new parameter that determines the migration rate was added to class "MigrationWorld" (see Program 4.7).

```
1 static std::shared_ptr<ParameterLink<double>> aMutationRatePL; // mutation rate A
2 static std::shared_ptr<ParameterLink<double>> bMutationRatePL; // mutation rate B
3 static std::shared_ptr<ParameterLink<double>> migrationRatePL; // migration rate
```

Program 4.7: Parameters of MigrationWorld (C++).

4.3.1 Dis-Entangling Organisms

The cells are still encoded as bit strings with a fixed genome size of 100 and there are still two classes: Agent and DualAgent. The only modification in this overall organism's design is the introduction of a flag that shows whether the organism was moved on to the next generation through group-level selection or migration, which equals individual-level selection (see Program 4.8).

```
1 class DualAgent {
2   public:
3   std::shared_ptr<Agent> A;
4   std::shared_ptr<Agent> B;
5   double score = 0.0;
6   bool isFromGroup = true;       // new flag
7   DualAgent() {}
8   DualAgent(std::shared_ptr<Agent> A_, std::shared_ptr<Agent> B_) : A(A_), B(B_) {}
9 };
```

Program 4.8: Source code of class DualAgent, modified for MigrationWorld (C++).

As in the experiments regarding genetic signatures, the genomes again start off with all zeros. Therefore, A-cells and organisms start at maximum fitness. This allows to research, whether the selective pressures are strong enough to let them drop from maximum fitness in order to help the fitness goal prevalent with the current migration rate.

4.3.2 Selection with Conflicting Pressures

For the new evolutionary goals, it was important that a conflicting pressure between the group- and individual-level was deliberately provoked. This is achieved through three different fitness functions:

- Organism's fitness: The organism's fitness score equals the number of matching bits between its A-cell and its B-cell.
- A-cell fitness: An A-cell's fitness is measured by the number of zeros in its genome.
- B-cell fitness: A B-cell's fitness is measured by the number of ones in its genome.

It is important for the introduction of migration that the fitness of A-cells and B-cells is measured individually with independent fitness functions. The cells' fitness functions conflict with the organism's fitness evaluation since only two of the three fitness values can be high: High A-cell fitness implies either very poor organism's fitness (if the B-cell fitness is high as well), or very poor B-cell fitness (if the organism's fitness is high), or both (the organism's fitness and the B-cell fitness are low). This goes for all combinations of organism's, A-cell and B-cell fitness. Therefore, a conflict in regards of the selective pressures is inevitable, as little to no selective pressure is given that selects for mediocrity. Since genomes have a length of 100, the maximum fitness for an A-cell, a B-cell and the overall organism is 100.

The code-snippets used for the evaluation functions are shown with Programs 4.9, 4.10, 4.11 and 4.12.

```
1 std::tuple<double, double, double> MigrationWorld::evalDual(DualAgent& dualAgent) {
2   dualAgent.A->score = evalAgentA(dualAgent.A->genome); // evaluate A−cell
3   dualAgent.B->score = evalAgentB(dualAgent.B->genome); // evaluate B−cell
4   dualAgent.score = evalGroup(dualAgent);               // evaluate organism
5
6   dualAgent.A->dualScore = dualAgent.score;
7   dualAgent.B->dualScore = dualAgent.score;
8
9   addToDataMap(dualAgent);
10
11  return { dualAgent.A->score, dualAgent.B->score, dualAgent.score };
12 }
```

Program 4.9: Source code of fitness function for evaluating a DualAgent (C++).

```
1 double MigrationWorld::evalGroup(DualAgent& dualAgent)
2 {
3   double matchingBits = 0.0;
4   for (int i = tagSize - 1; i >= 0; i--)
5   {
6     if (dualAgent.A->genome[i] == dualAgent.B->genome[i])
7     {
8       matchingBits++;
9     }
10  }
11  return matchingBits;
12 }
```

Program 4.10: Source code of fitness function for evaluating the overall organism (C++).

```
1 double MigrationWorld::evalAgentA(std::bitset<tagSize>& testGenome)
2 {
3   return (tagSize - testGenome.count()) * 1.0;
4 }
```

Program 4.11: Source code of fitness function for evaluating an A-cell (C++).

```
1 double MigrationWorld::evalAgentB(std::bitset<tagSize>& testGenome)
2 {
3   return testGenome.count() * 1.0;
4 }
```

Program 4.12: Source code of fitness function for evaluating a B-cell (C++).

4.3.3 Introducing Migration

Migration is implemented by allowing horizontal gene transfer in the selection process. The migration rate determines how many organisms of the whole population are selected as groups (*i.e.,* high-level selection without migration) and how many are selected via migration (*i.e.,* low-level individual selection). Organisms that are selected via migration are not selected as organisms, but an individual A-cell and an individual B-cell is selected based on their cell fitness and they are then randomly mixed-up to an organism. The GA used for this multi-level selection is described with Algorithm 4.4.

Algorithm 4.4: Genetic algorithm used in MigrationWorld for the analysis of multi-level selection behavior (C++ style pseudo code).

```
1:  procedure evaluate(groups, updates)
        Input: groups, MABE parameter for populations of type-A and type-B cells;
        updates, number of generations to run.
2:      initializePopulation(popA, popB, popDual, groups, popSize)
3:      while update! = updates do          ▷ iterate while not all updates are finished
4:          for i ∈ popSize do                                          ▷ see Program 4.9
5:              [aScores[i], bScores[i], groupScores[i]] ← evalDual(popDual[i])
6:          end for
7:          for i ∈ popSize · (1 − migrationRatePL) do ▷ selection as in Algorithm 4.1
8:              newDualAgent ← doSelection(popSize, groupScores, popDual, groups, 7)
9:              mutateSelection(newDualAgent)                           ▷ see Algorithm 4.3
10:             newDualAgent.isFromGroup = true;
11:             popDual.push_back(newDualAgent)
12:             popA.push_back(newDualAgent.A)
13:             popB.push_back(newDualAgent.B)
14:         end for
15:         for i ∈ popSize · migrationRatePL do      ▷ low-level selection (migration)
16:             newA ← doSelectionIndividual(popSize, aScores, 7)
17:             newB ← doSelectionIndividual(popSize, bScores, 7)
18:             mutateSelection(newA, newB)          ▷ works similar to Algorithm 4.3
19:             newDualAgent = DualAgent(newA, newB)
20:             newDualAgent.isFromGroup = false;
21:             popDual.push_back(newDualAgent)
22:             popA.push_back(newDualAgent.A)
23:             popB.push_back(newDualAgent.B)
24:         end for
25:         groups["A::"].archive()   ▷ MABE functionality needed for lineage tracking
26:         groups["B::"].archive()   ▷ MABE functionality needed for lineage tracking
27:         update + 1                                                  ▷ update is finished
28:     end while
29: end procedure
```

Chapter 5

Experiments

This chapter describes the experiments that were conducted in order to detect genetic signatures and find out more about the interaction between group-level and individual-level selection mechanisms. It is divided into two sections. Section 5.1 focuses on experiments to detect genetic signatures of co-evolution and is split up into experiments to detect co-evolution when experimental manipulation is possible and when it is not. And the second section, Section 5.2, focuses on the interaction between selection mechanisms and analyzes the evolutionary behavior with different portions of selective pressures on the group- and individual-level. All analysis scripts and result figures shown in this chapter and beyond are available at the GitHub repository [73].

5.1 Genetic Signatures of Co-Evolution

All experiments regarding the existence of a genetic signature were conducted with lineage data. Since the fitness goal is in all scenarios (except Matching-Bits Lockstep) to reach as many leading ones as possible, the obvious choice would be to only look at mutations that affect this leading-one part of the genome. However, the author decided to look at all mutations (*i.e.,* leading-one mutations plus mutations at the tail-end) that occurred in the genome, to show how good the proposed metrics work, as they are able to detect the genetic signatures not only in the leading-one mutations but in the far more general, overall mutations as well.

The author decided to do the same analysis for the Matching-Bits Lockstep, as for the other five scenarios. Since the Matching-Bits Lockstep is the transitioning scenario into the "Multi-Level Selection"-phase, the results should indicate the direction in terms of mutation rate and population size for the experiments described in Section 5.2.

For the experiments done with different mutation rates, the author used a per-site mutation rate of 0.01 as basis and increased it to 0.03 (*i.e.,* a threefold effect) and to 0.1, which is a tenfold effect, during the experiments. The population size was constant at 1000 organisms. This resulted in four different configurations, as shown in Table 5.1. All four configurations were compared with configuration E3 from Table 5.2 since a mutation rate of 0.01 was the pivot for those analyses. Table 5.1 shows that configuration D2 and D4 are mirroring configurations D1 and D3. This was a control mechanism to assure that everything works as expected. Since configurations D1/D2 and D3/D4 show

the same results except for inverted high-low mutation rates between the A-cell and the B-cell, the figures and tables later in this chapter show only configurations D1 and D3.

Configuration	Mutation Rate A-Cell	Mutation Rate B-Cell
D[1]1	0.01	0.03
D2	0.03	0.01
D3	0.01	0.1
D4	0.1	0.01

Table 5.1: Configurations for experiments with different mutation rates.

For the experiments with equal mutation rates, seven different combinations of populations size and mutation rate (*i.e.,* configurations) were run for each of the six scenarios to show the robustness of the genetic algorithm. Runs were done with populations of 10, 100, 1000 and 10000 organisms. The per-site mutation rate was varied between 0.001, 0.003, 0.01 and 0.03. A population size of 1000 and a mutation rate of 0.01 served as pivot for the configurations, which are listed in Table 5.2.

Configuration	Mutation Rate A-Cell & B-Cell	Population Size
E[2]1	0.001	1000
E2	0.003	
E3	0.01	
E4	0.03	
E5	0.01	10
E6		100
E7		10000

Table 5.2: Configurations for experiments with equal mutation rates.

The general experiment setup includes three different comparisons, which are used to ensure that a genetic signature for a lockstep pattern really is there: intra-run, inter-run and inter-treatment comparison. For each comparison, 50 independent data points were generated and subsequently plotted. It was not necessary to use a statistical control as *e.g.,* jackknife since the author ran so many replicates with different random number seeds that every replicate is only used once: Either its A- or its B-cell is used but never both of them. This assures that all comparisons are independent. To generate 50 independent data points for each comparison, it was necessary to run 250 replicates per configuration and scenario:

- 50 scenario-replicates for the intra-run comparison, where the lineage of an A-cell is compared to its corresponding B-cell's lineage and the replicate therefore is compared with itself.

[1] "D" stands for Different.
[2] "E" stands for Equal.

- 100 scenario-replicates for the inter-run comparison, where an A-cell's lineage is compared to a B-cell's lineage from a different replicate. Since no replicate should be used in more than one of the 50 data points to assure independence, it is necessary to run twice as much replicates as data points needed. This assures that from each replicate either the A-cell's lineage *or* the B-cell's lineage is used.
- 50 scenario-replicates plus 50 replicates from the scenario No Selection Pressure for the inter-treatment comparison. As with the inter-run comparison, no A-cell's *and* B-cell's lineage from a single replicate is taken but only one of them. That is the reason why twice as many replicates as data points are needed for this comparison, as well.

This summarizes in 200 scenario-replicates plus 50 replicates from the No Selection Pressure scenario per configuration (*i.e.,* mutation rate and population size combination) and scenario. In the inter-run and inter-treatment comparisons, the lineages from the different replicates are mixed up randomly although it should not matter at all to begin with since the replicates were all run with different random seeds. This assures independence between the replicates per se. For inter-treatment comparisons, the pair of replicates (one from the scenario and one from No Selection Pressure) is randomly shuffled to not influence whether an A-cell was taken from the current scenario and the B-cell from No Selection Pressure or vice versa.

In total, five different analyses were conducted and they are summarized in Table 5.3 and described in the following Sections 5.1.1 and 5.1.2.

Name of Analysis	Mutation Rates?	Goal of Analysis
Fitness Score Comparison	Different	Identifying Signs of Co-Evolution
Mutation Count	Different	Detecting Co-Evolution
Fitness Score	Equal	Overview over Goal Attainment
Mutation Type	Equal	Controlling Mechanics of Evolution
AMM[3]	Equal	Detecting Co-Evolution

Table 5.3: Description of different analyses, used for detecting genetic signatures of co-evolution.

5.1.1 Detecting Co-Evolution using Experimental Manipulation

The goal of these experiments is to detect genetic signatures of co-evolution from lineage-based data, when experimental manipulation in terms of controlling mutation rates is possible. This is achieved with two kinds of analyses: the fitness score comparison and the mutation count analysis. This section first describes those analyses in general, followed by showing the results for each of the six scenarios. Lastly, conclusions are presented.

[3] "AMM" stands for Accumulated Mutations Metric.

Fitness Score Comparison - Overall Remarks

First of all, the fitness contributions of the A-cell and the B-cell in different scenarios were analyzed. The fitness score after 5000 generations was used as reference to assess how well evolution played out in terms of evolutionary fitness goal attainment for the A-cell and the B-cell. The aim of this analysis is to show that evolution indeed works as expected and to identify possibly existing signs of co-evolutionary behavior.

The contributions were compared to configuration E3 of detecting genetic signatures with equal mutation rates. The following tables show statistical summaries for configurations E3, D1 and D3. Moreover, graphical representations of the tables as box plot are available on the GitHub repository [73]. As mentioned before, E3 was also compared to D2 and D4 (inverted high-low mutation rates between A- and B-cells), but since the results were the same, they are not shown herein.

All statistics for this analysis were done with the data found in files "different_fitness_score.csv" and "equal_fitness_score.csv". The tables were generated using intra-run comparison data only, with the R-script "a_fitness_score_statistics.R". Data from equal mutation rates for A and B is compared to differing ones, looking at the fitness scores of the two cell types at generation 5000.

Mutation Count Analysis - Overall Remarks

The mutation count analysis was conducted with different mutation rates for the A- and B-cell. The expectation is to make co-evolution - if existing - visible by comparing the mutation count of equal mutation rates for A and B to the mutation count of different ones.

Figures or tables represent the results of this method. When tables are shown, the corresponding figures are available at the GitHub repository [73]. The figures each show one scenario. On the x-axis is the cell type (*i.e.,* A or B) annotated and the y-axis shows the accumulated number of mutations (*i.e.,* the sum of beneficial, deleterious and neutral mutations) along one lineage. The mutation count does not include no-mutations since they are representing generations in which no mutation has happened in the genome. If they would be added the count would always be 5000, which equals the number of generations that were run during the experiments.

There are three graphs in each figure. The first shows the behavior when A and B had the same mutation rate during the experiments (configuration E3), the second graph shows the results when A-cells were mutated with a per-site mutation rate of 0.01 (*i.e.,* unaltered; configuration D1) and B-cells with a three times higher mutation rate of 0.03. In the third graph, A-cells' mutation rate is unaltered again at 0.01 and B-cells' mutation rate is ten times higher at 0.1, in contrast to the A-cells' rate (configuration D3). The graphs show box plots, which are overlaid by scatter plots. Each graph consists of 50 data points per cell type from the data produced with intra-run comparison replicates.

The data for the plots and tables showing the results is found in files "different_mutation_count.csv" and "equal_mutation_count.csv". The plots and tables for detecting genetic signatures were implemented with the R-scripts "a_mutation_count_analyses.R" and "a_mutation_count_statistics.R". The plots compare the behavior between equal mutation rates and different ones, in the hope that interesting conclusions can be drawn from that.

No Selection Pressure

This scenario acts as a control when no selection pressure and therefore, no co-evolution is present. Table 5.4 shows statistical values for the achieved fitness scores with configurations E3, D1 and D3. As expected, the mutation rate hardly influences the outcome and A- and B-cells underperform due to evolution being completely driven by drift.

Regarding the mutation counts[4], Figure 5.1 shows that the number of mutations are balanced when the mutation rate is the same for A- and B-cells. Not surprisingly, the mutation count goes up when the mutation rate is increased. That is why the graph shows about the same amount of mutations for the A-cells in the graphs in the middle and on the right-hand side, compared to the graph on the left; whereas the mutation count is elevated for B-cells with a mutation rate of 0.03 and even more elevated with a mutation rate of 0.1 for B-cells. Those results show that evolution works as expected.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E3	1000	0.01	A	-0.27	0.75	4.83	0.88	50
			B	-0.28	0.73	5.76	0.86	50
D1	1000	0.01	A	-0.30	0.73	5.76	0.91	50
		0.03	B	-0.30	0.26	2.77	0.47	50
D3	1000	0.01	A	-0.29	0.74	5.78	1.03	50
		0.1	B	-0.30	0.26	5.74	0.68	50

Table 5.4: Summary statistics for fitness scores in scenario No Selection Pressure.

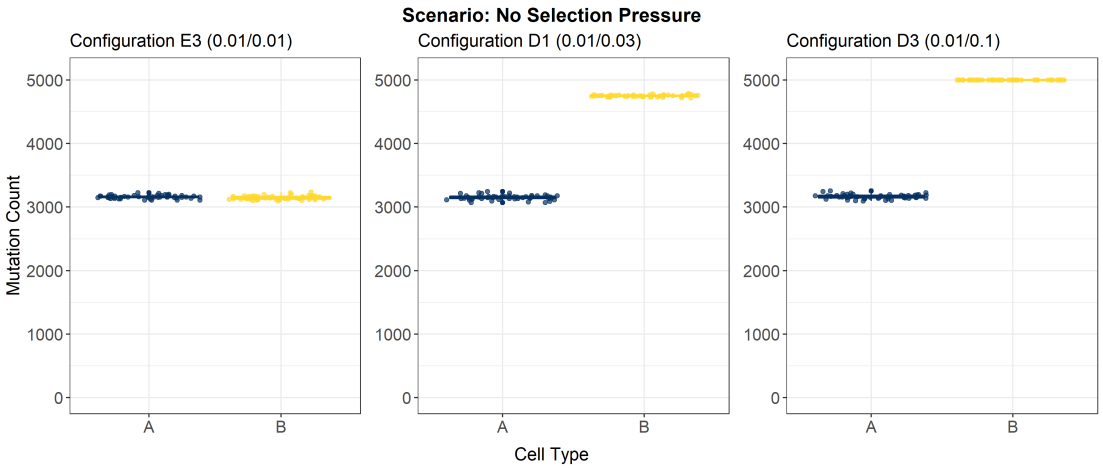

Figure 5.1: Result of mutation count analysis for scenario No Selection Pressure.

[4]At first glance, it might be strange that the author has this this sort of data, although nothing affects fitness in this scenario. This is, because individual scores for the cells are still calculated, but the selection happens on the organism-level and there, no selection pressure is present.

Additive Evolution

In this scenario, the fitness scores of A- and B-cells are always summed. Table 5.5 shows that A- and B-cells perform excellent in this scenario with a mutation rate of 0.01. D1 and D3 show that mutation rates of 0.03 and 0.1 are too high and lead to a populational meltdown. As expected, the organisms still perform better at a high mutation rate of 0.1 than in scenario No Selection Pressure.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E3	1000	0.01	A	99	100	100	99.92	50
			B	98	100	100	99.90	50
D1	1000	0.01	A	96	100	100	99.24	50
		0.03	B	25.82	35.86	41.90	36.32	50
D3	1000	0.01	A	99	100	100	99.98	50
		0.1	B	6.80	10.81	14.79	10.87	50

Table 5.5: Summary statistics for fitness scores in scenario Additive Evolution.

In terms of mutation count, the original expectation was that the cell types should be completely independent. As before, a higher mutation rate also means more mutations overall, which is just how evolution works. However, in terms of independence, Figure 5.2 tells a different story: It seems as if the cell types in this scenario are not as independent as assumed. The graph in the middle shows slight signs of A-cells being pulled along by the higher mutation rate for B-cells, but the graph on the right does not show this behavior.

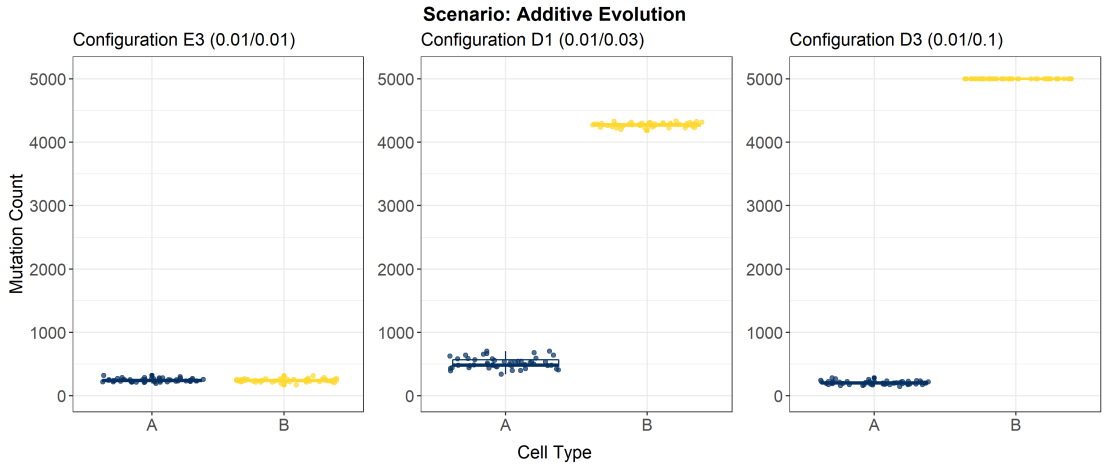

Figure 5.2: Result of mutation count analysis for scenario Additive Evolution.

The author's assumption for this is that A and B are part of a very loosely linked symbiosis: In this scenario, two independent cells form one organism and since they stay together as long as they are alive, they are still very loosely linked although being

independent individuals. B's mutation rate impacts A's lineage since their evolution is tied together. Therefore, changing B's dynamics still slightly affects A's dynamic in some way. And what drives the overall dynamic is that these two cell types must replicate together, although being independent.

Table 5.5 further verifies this assumption: The fitness contributions after 5000 updates show that the higher mutation rate knocks the fitness over. The author assumes that A in this scenario adapts quickly and so there is almost no variation left in the population of A-cells since A is already almost perfectly adapted. If this is the case, B drives the evolution although it performs very poorly because the population of B-cells is the only source of variation that is still available in the overall population. And because B-cells have a higher mutation rate, such a cell sometimes might get two good mutations and then the A-cell has the opportunity to develop some hitchhikers, which drives the mutation count of the A-cells up, as seen with configuration D1. However, B's mutation rate in configuration D3 seems to be too high to observe this behavior. In conclusion, some sort of very weak co-evolution seems to be present since an interesting dependency between A- and B-cells can be observed in which B drives evolution even though it has a terrible fitness score.

And although A's and B's fitness values are just summed together, they are very much contributing to each other's fitness. This shows that scenario Additive Evolution is not independent (as previously expected), but the evolution of one influences the fitness of the other at a very low rate.

Zero-Off Lockstep

In this scenario, A and B are linked and a selective pressure is present. Table 5.6 depicts the fitness score statistics. In configuration E3 with equal mutation rates, both cells do better. Configurations D1 and D3 clearly show that mutation rates of 0.03 or 0.1 are simply too high to perform good in this scenario and, more importantly, that the B-cell is holding back the A-cell. The assumption is that this behavior is due to the tight coupling of the two cell types.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E3	1000	0.01	A	84	88	93	88.16	50
			B	84	88	93	88.16	50
D1	1000	0.01	A	37.86	46.90	48.90	45.99	50
		0.03	B	37.92	46.89	48.87	45.98	50
D3	1000	0.01	A	13.84	16.84	16.91	16.49	50
		0.1	B	13.77	16.80	16.86	16.45	50

Table 5.6: Summary statistics for fitness scores in scenario Zero-Off Lockstep.

The analysis of the mutation count should further verify this assumption and it does indeed: When the mutation rate is equal for both cell types, the mutation count is also pretty similar. And, as before, a raise in the mutation rate of B also ups the number of mutations. However, here is the truly interesting behavior: Although A's mutation

rate stayed the same compared to the graph on the left-hand side, the mutation counts of the graphs in the middle and on the right-hand side are now higher. The number of mutations in A gets pulled along by the higher mutation rate of B in this scenario (see Figure 5.3). Therefore, we see evidence for co-evolution between A- and B-cells since there is no other reason than a tight link between A- and B-cells, for the A-cells to be influenced by the higher mutation rates in the evolution of B-cells.

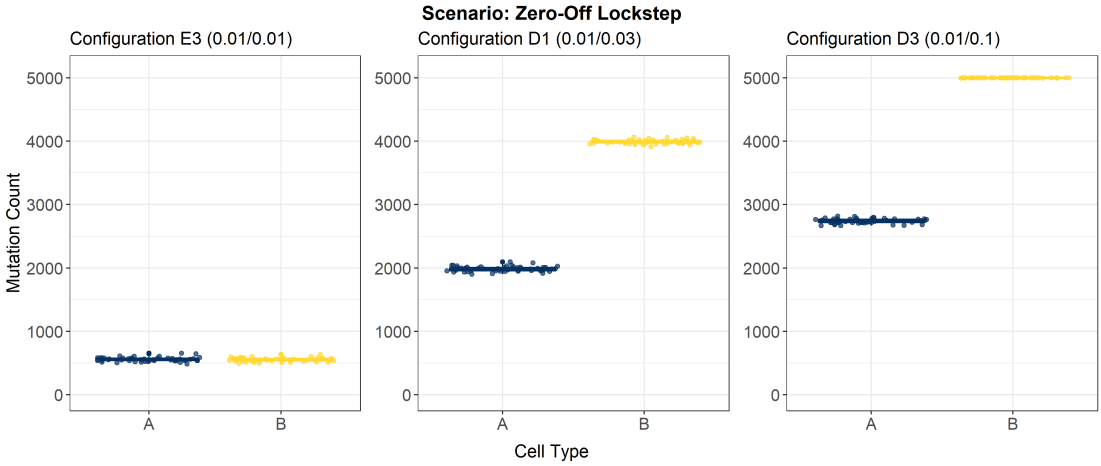

Figure 5.3: Result of mutation count analysis for scenario Zero-Off Lockstep.

The author has also looked at several genomes of A- and B-cells and she could see that the mutations in this scenario were perfectly synchronized, just as expected.

One-Off Lockstep

In One-Off Lockstep, the selective pressures on the A- and B-cells that form one organism, is to be synchronous or one-off synchronous in terms of leading ones. Table 5.7 shows that A- and B-cells overall perform a little bit better in configuration E3 than in the Zero-Off Lockstep scenario. In D1, there seems to be an outlier, but the mean fitness score is still at about 50. Again, in D3 the mutation rate of the B-cells causes a meltdown. However, again it is visible that the higher mutation rate in D1 and D3 holds back A-cells' fitness values as well.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E3	1000	0.01	A	79.97	93	100	92.77	50
			B	78.95	92.99	100	92.29	50
D1	1000	0.01	A	-1	48.39	50.90	47.13	50
		0.03	B	-1	47.91	49.90	46.72	50
D3	1000	0.01	A	13.83	18.78	19.77	18.08	50
		0.1	B	13.80	17.79	18.84	17.52	50

Table 5.7: Summary statistics for fitness scores in scenario One-Off Lockstep.

Table 5.8 supports the assumption of A and B being tightly linked and co-evolution being present. As in the previous scenario, an increase of B's mutation rate results in more accumulated mutations in this B-cell. As the author has hypothesized, the A-cell also accumulates more mutations, albeit the mutation rate was not raised. Therefore, the mutation count analysis shows strong evidence for the presence of co-evolution in this scenario.

Configuration	Pop. Size	Mut. Rate	Cell	Mean	Count
E3	1000	0.01	A	586.60	50
			B	602.48	50
D1	1000	0.01	A	2091.92	50
		0.03	B	4002.78	50
D3	1000	0.01	A	2796.80	50
		0.1	B	4998.66	50

Table 5.8: Mean values for mutation counts in scenario One-Off Lockstep.

The author again looked at the mutations in the genomes of A- and B-cells and they were indeed perfectly or one-off perfectly synchronized. This coincides precisely with the author's expectations of what should happen in this scenario.

One Follows

In One Follows, B-cells can fall behind A-cells by an unspecified number of leading ones. Table 5.9 shows a similar result than Table 5.5 from scenario Additive Evolution: In configuration E3, A- as well as B-cells have very high fitness scores. In D1 and D3, A-cells still have those high fitness scores, but B-cells perform rather poorly. The higher mutation rate of B-cells does not hold back A's fitness values and, therefore, no immediate indication for a co-evolutionary relationship is present.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E3	1000	0.01	A	99	100	100	99.96	50
			B	95.98	100	100	99.26	50
D1	1000	0.01	A	80.91	100	100	98.8	50
		0.03	B	18.76	36.35	41.89	35.67	50
D3	1000	0.01	A	100	100	100	100	50
		0.1	B	5.78	10.83	14.83	11.18	50

Table 5.9: Summary statistics for fitness scores in scenario One Follows.

The mutation count analysis shown in Table 5.10 shows a very slight effect of A's mutation counts being pulled along by B's mutation counts of higher mutation rates, if at all. This makes perfect sense since the analysis detects tight co-evolution and this

One Follows scenario implements a very loose form of co-evolution. Therefore, A- and B-cells are only linked very loosely and no signal of tight co-evolution is detectable.

Configuration	Pop. Size	Mut. Rate	Cell	Mean	Count
E3	1000	0.01	A	140.32	50
			B	348.86	50
D1	1000	0.01	A	321.22	50
		0.03	B	4333.12	50
D3	1000	0.01	A	179.82	50
		0.1	B	4999.48	50

Table 5.10: Mean values for mutation counts in scenario One Follows.

Matching-Bits Lockstep

In this scenario, the highest possible fitness score is 1200 (instead of previously 100) since the number of matching bits is multiplied by 10, which results in a maximum of 1000 and the overall number of ones in the A- and B-cell is added. Since genomes have a length of 100, the best fitness score is $100 * 10 + 100 * 2 = 1200$. Moreover, the overall organism fitness equals the individual A- and B-cell fitness. Therefore, it is no surprise that Table 5.11 shows the same values for cells of type A and type B.

Overall, there is an indication for co-evolution in this scenario since the higher mutation rate of B-cells holds back A's fitness values (see Table 5.11). This is not surprising, as this scenario is also a manifestation of simultaneous mutational changes.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E3	1000	0.01	A	1189	1200	1200	1198.66	50
			B	1189	1200	1200	1198.66	50
D1	1000	0.01	A	1025	1054.50	1090	1057.24	50
		0.03	B	1025	1054.50	1090	1057.24	50
D3	1000	0.01	A	843	890	948	890.92	50
		0.1	B	843	890	948	890.92	50

Table 5.11: Summary statistics for fitness scores in scenario Matching-Bits Lockstep.

The mutation count analysis (see Table 5.12) further confirms this assumption, as a raise of B's mutation rate in configurations D1 and D3 ups the numbers of mutations that the A-cells received on average, although their mutation rate stayed the same as in configuration E3. Therefore, the analysis signals tight co-evolution.

Configuration	Pop. Size	Mut. Rate	Cell	Mean	Count
E3	1000	0.01	A	259.02	50
			B	260.04	50
D1	1000	0.01	A	2002.8	50
		0.03	B	3970.18	50
D3	1000	0.01	A	2731.78	50
		0.1	B	4998.86	50

Table 5.12: Mean values for mutation counts in scenario Matching-Bits Lockstep.

Conclusions

The analyses presented in this section (*i.e.,* fitness score and mutation count analysis), are able to correctly detect tight co-evolutionary relationships between lineages from different types of cells. These methods are applicable when mutation rates can be manipulated in the experimental setup. Looking at the fitness scores gives a first indication for whether or not co-evolution is present. Analyzing the mutation count either verifies that first signal or, as it was the case with scenarios Additive Evolution and One Follows, detects weak co-evolutionary relationships between the A- and B-cells.

In scenarios where tight co-evolution is present (*i.e.,* Zero-Off Lockstep, One-Off Lockstep and Matching-Bits Lockstep) the above mentioned methods make the genetic signature visible: The fitness scores of A-cells diminish by the higher mutation rate of B-cells in configurations D1 and D3, and the mutation counts increase significantly in comparison to the control configuration E3, although A's mutation rate stayed the same. The number of mutations in A-cells gets pulled along by the higher mutation rate of B-cells, which is a clear sign of co-evolution. In the control scenario No Selection Pressure, the analysis correctly showed no signs of co-evolution.

For scenarios Additive Evolution and One Follows, the methods showed slightly surprising results: One Follows basically is an Infinite-Off Lockstep scenario, in which no tight co-evolution could be detected, but very weak forms of co-evolution got visible through the mutation count analysis. The author assumed that Additive Evolution is completely independent and absolutely no signs of co-evolution are there, but the mutation count analysis detected a weak signal, which makes perfect sense in retrospective: If one cell (*i.e.,* the A-cell in this setup) is adapting meaningfully faster to the evolutionary goal than its partner (*i.e.,* the B-cell in this case), that partner will be pushed away from the no-mutation case and thus, increase the total number of mutations encountered since this partner is the only one that still can provide variation within the population. Therefore, the conclusion is that truly independent evolution is not possible, as long as cells are not reproducing independently. And independent reproduction is only possible, when migration is introduced.

5.1.2 Detecting Co-Evolution from Historical Data

The goal of these experiments is to detect genetic signatures of co-evolution from lineage-based data, when experimental manipulation is not possible. In biological systems, it is often the case that mutation rates cannot be modified and the hereafter shown methods propose a way of identifying co-evolution even in such challenging circumstances. As before, when tables are shown instead of figures, the latter are provided on the GitHub repository [73]. In this section, three kinds of analyses are presented: the fitness score analysis, the mutation type analysis and the Accumulated Mutations Metric. As before, this section first describes those analyses in general, followed by showing the results for each of the six scenarios. At the end of this section, again, conclusions are presented.

Fitness Score Analysis - Overall Remarks

This analysis should provide an insight into how well organisms have performed under different circumstances, in terms of population size and mutation rate. Thus, the fitness contributions of the A- and B-cell after 5000 generations were analyzed. The aim is to show that the mechanics of evolution work as supposed. The following tables show summary statistics for the seven configurations described earlier with Table 5.2, and were conducted with the data found in file "equal_fitness_score.csv" and generated using the intra-run comparison data only, with the R-script "a_fitness_score_statistics.R".

Mutation Type Analysis - Overall Remarks

The tables and figures presented for this analysis show the number of beneficial, deleterious and neutral mutations that occurred along a lineage as well as the fraction of those numbers to get a better feeling for what mutation types are how frequent in different circumstances. For the mutation type analysis, only intra-run comparison was adduced since both, inter-run and inter-treatment comparisons make little sense here. This analysis should show that evolution works as expected, by *e.g.,* showing that less beneficial and more deleterious mutations occur, as the individuals get better and it becomes more challenging that a random mutation has a positive impact on them. The data used can be found in file "equal_mutation_type.csv", the plots were drawn with the R-script "a_mutation_type_analyses.R" and the tables with "a_mutation_type_statistics.R".

Accumulated Mutations Metric (AMM) - Overall Remarks

AMM is capable of detecting close co-evolutionary relationships between different types of cells (*i.e.,* type A and B) when no experimental manipulation is possible. To make potential co-evolution visible, two different types of analyses were performed:

- Heat map: One heat map per scenario and configuration is generated with the data found in file "equal_amm_heatmaps.csv" and the code from R-script "a_amm_heatmaps.R". Heat maps visualize the intra-run comparison data and give a first feeling for whether a genetic signature can be expected. The assumption is that if a diagonal line from the lower-left corner to the upper-right is visible in the 50x50 matrix, there is indication for a genetic signature. A visible diagonal

line means that the intra-run AMM is smaller than the inter-run AMM, which in turn means that there is evidence to suggest that a genetic signature in the form of simultaneous mutations is present. Since the data is not fully independent in the heat maps, statistically correct conclusions must not be drawn from them but they are informative and give a first intuition for what can be expected from the second type of analysis. The data is not fully independent since one replicate is used for several squares in the matrix. This could not have been done in another way since the whole point of the heat maps is to compare each replicate with all of the other replicates and that makes them dependent.

- AMM box plot: All three comparison types (*i.e.,* intra-run, inter-run and inter-treatment) are visualized in one AMM box plot using the data from file "equal_amm.csv" and the R-code found in "a_amm_boxplots.R". Such a box plot is provided per configuration and for each scenario and overlaid by a scatter plot. If a genetic signature is present, the intra-run box plot has much lower values than the inter-run, which in turn has significantly lower values than the inter-treatment comparison. The inter-treatment comparison takes As and Bs from different scenarios and so the AMM-value should be very high. The intra-run comparison uses As and Bs from the same scenario, which were truly coupled during evolution. Therefore, the AMM-value should be very small if co-evolution is present. And the inter-run comparison takes As and Bs that were not coupled during evolution but exposed to the same evolutionary pressures. Therefore, the AMM-value should lie somewhere in between the AMM-values from intra-run and inter-treatment. If this gradation is visible in the AMM box plots, a co-evolutionary relationship between the A- and B-cells is existing.

To assure independence in the mixing up for inter-run and inter-treatment comparisons, the author analyzed whether it matters if the A- or the B-cell is taken from a replicate. Therefore, each of the pairs was looked at backwards, which means that A and B were switched. As expected, the result was clear across all experiments: There was no difference between the original mix-up and the switched one. Hence, the random pairing process does not bias the outcome. Therefore, this thesis does not talk about the outcomes from the switched ones since they only generated redundant results and do not provide any additional findings. This is in fact positive and further emphasizes the independence of the single replicates.

Additionally, statistical tests were run to prove numerically what can be seen in the visualizations. Kruskal-Wallis and Wilcoxon rank-sum test were utilized for the statistical analysis: On a very high level, Kruskal-Wallis first tests whether there is a difference and if there is one, Wilcoxon rank-sum test finds out what is different.

Heat maps and AMM box plots are shown for scenarios No Selection Pressure, Zero-Off Lockstep and One-Off Lockstep. The results of the remaining three scenarios are described with the values of the statistical tests. Figures for all scenarios, as well as the exact p-values of the statistical tests are accessible at the GitHub repository [73].

No Selection Pressure

Table 5.13 shows pretty similar data for all seven configurations. This is not surprising as selection happens on the organism-level and the fitness value there is a constant of 1.0.

So, those are the baseline results of a scenario where everything is driven by drift.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E1	1000	0.001	A	-0.32	-0.20	10.80	0.81	50
			B	-0.31	-0.20	6.76	0.69	50
E2	1000	0.003	A	-0.28	0.74	4.80	0.72	50
			B	-0.30	-0.21	5.74	0.7	50
E3	1000	0.01	A	-0.27	0.75	4.83	0.88	50
			B	-0.28	0.73	5.76	0.86	50
E4	1000	0.03	A	-0.30	0.26	6.82	0.86	50
			B	-0.30	0.72	5.74	0.77	50
E5	10	0.01	A	-0.32	-0.21	5.80	0.66	50
			B	-0.30	0.74	5.74	0.76	50
E6	100	0.01	A	-0.28	0.74	4.80	0.88	50
			B	-0.28	0.73	4.76	0.77	50
E7	10000	0.01	A	-0.31	-0.22	4.72	0.39	50
			B	-0.28	0.72	5.76	0.87	50

Table 5.13: Summary statistics for fitness scores in scenario No Selection Pressure.

Figure 5.4 shows the mutation counts for beneficial, deleterious and neutral mutations for configurations E1 to E7. As long as the mutation rate is steady (at 0.01), the amount of mutations roughly stays the same across the four different population sizes. As expected, at a constant population size of 1000, higher mutation rates mean more mutations and lower mutation rates lead to less mutations. This figure shows that the mutation part of evolution works exactly as assumed.

Figure 5.5 shows the same data as Figure 5.4 but with percentages instead of absolute mutation counts. This figure additionally shows that lower mutation rates slightly favor beneficial mutations over deleterious ones, whereas at higher mutation rates the ratio is balanced. The higher the mutation rate, the more neutral mutations are common. Those behaviors are not surprising with lineage-based data: At lower mutation rates, mutations in general are less common, as visible in Figure 5.4. Therefore, organisms with beneficial mutations have a high probability of moving on to the next generation and organisms with neutral mutations, where the genome changed but the fitness score stayed the same, are less likely selected for the next generation.

Figure 5.6 shows the heat map visualizations for this scenario. No diagonal line is visible in any of the heat maps and, therefore, the expectation is to identify no co-evolution in the box plots. Those are depicted with Figure 5.7 and indeed show no signs of co-evolution. This is positive since the author designed the current scenario in a way that no tight coupling is existing and evolution is solely driven by drift. All configurations in Figure 5.7 show similar distributed box plots for the intra-run, inter-run and inter-treatment comparison and that is why no genetic signature of co-evolution is identifiable. Krukal-Wallis was not significant for any of the configurations.

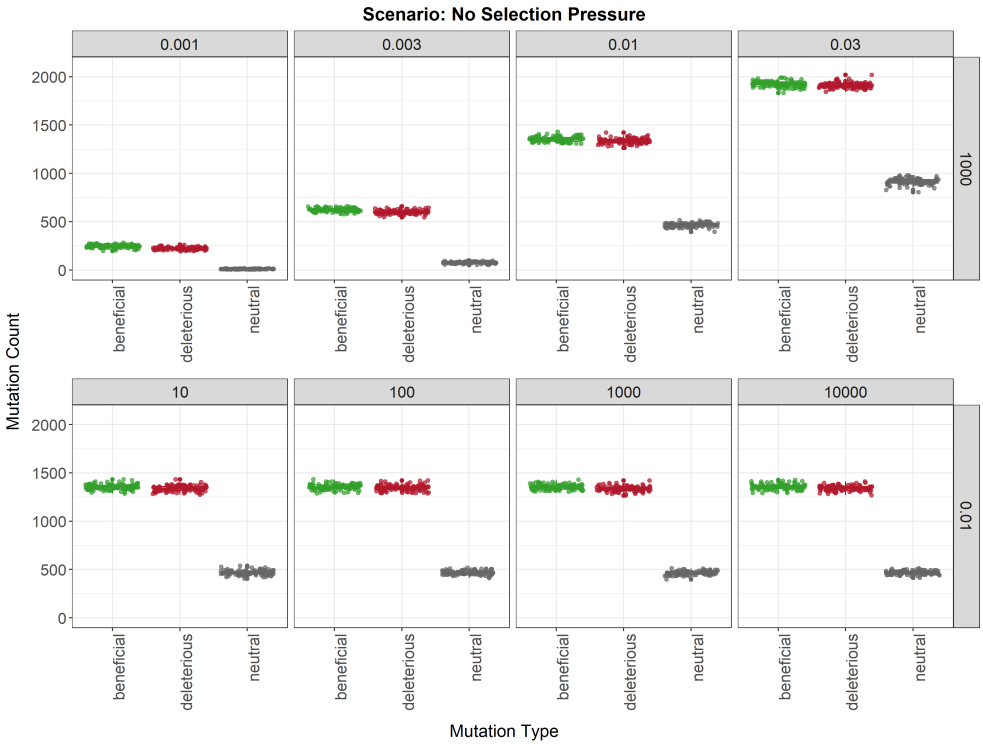

Figure 5.4: Result figure for mutation count analysis in scenario No Selection Pressure.

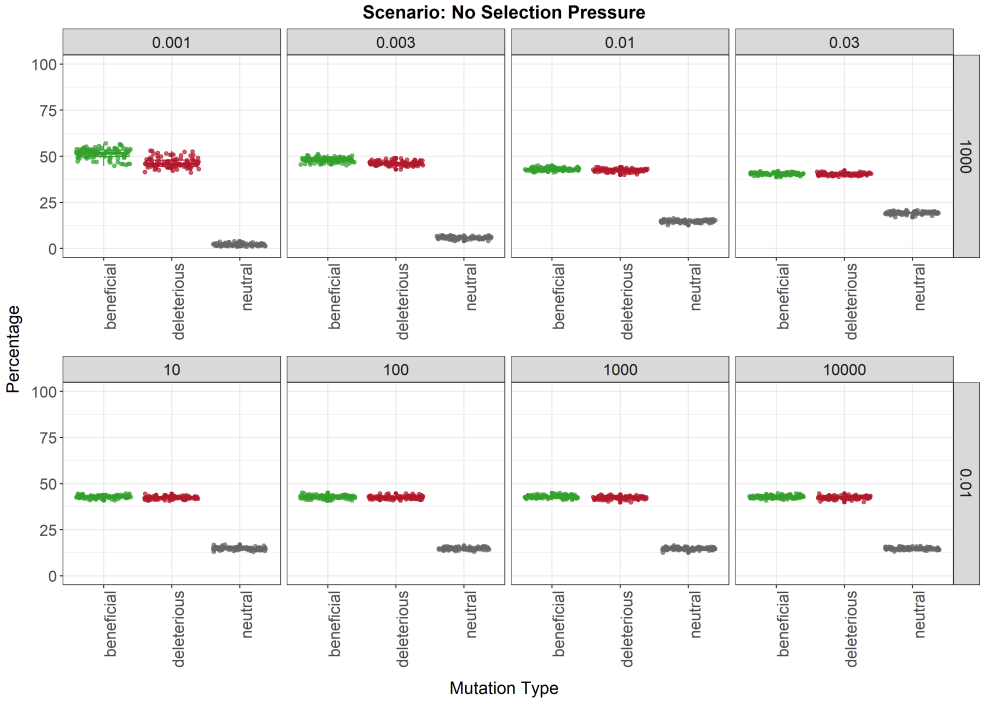

Figure 5.5: Result figure for mutation count percentages in scenario No Selection Pressure.

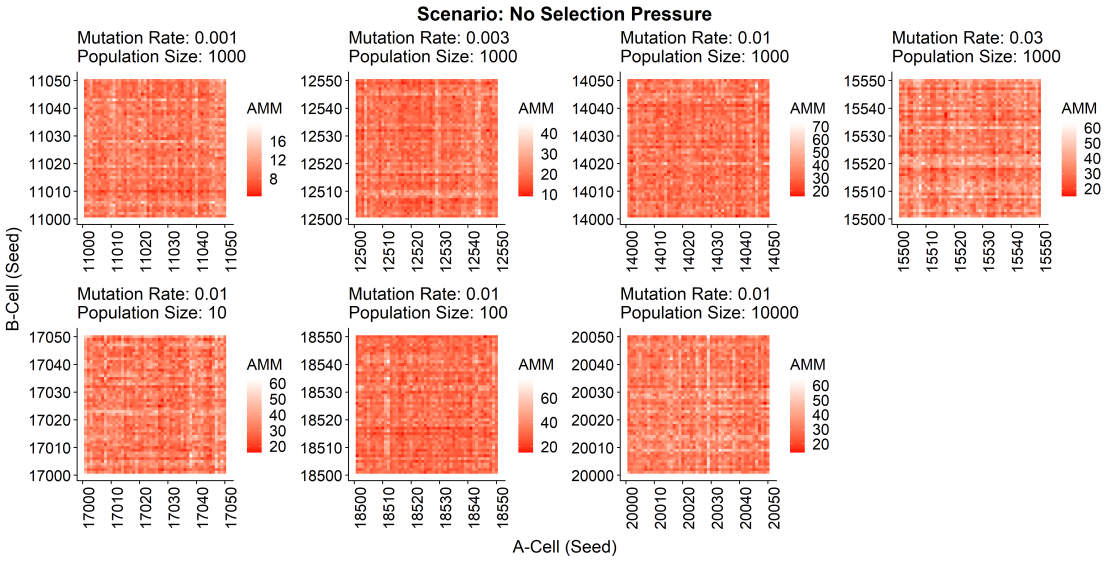

Figure 5.6: Heat maps with AMM-values for scenario No Selection Pressure.

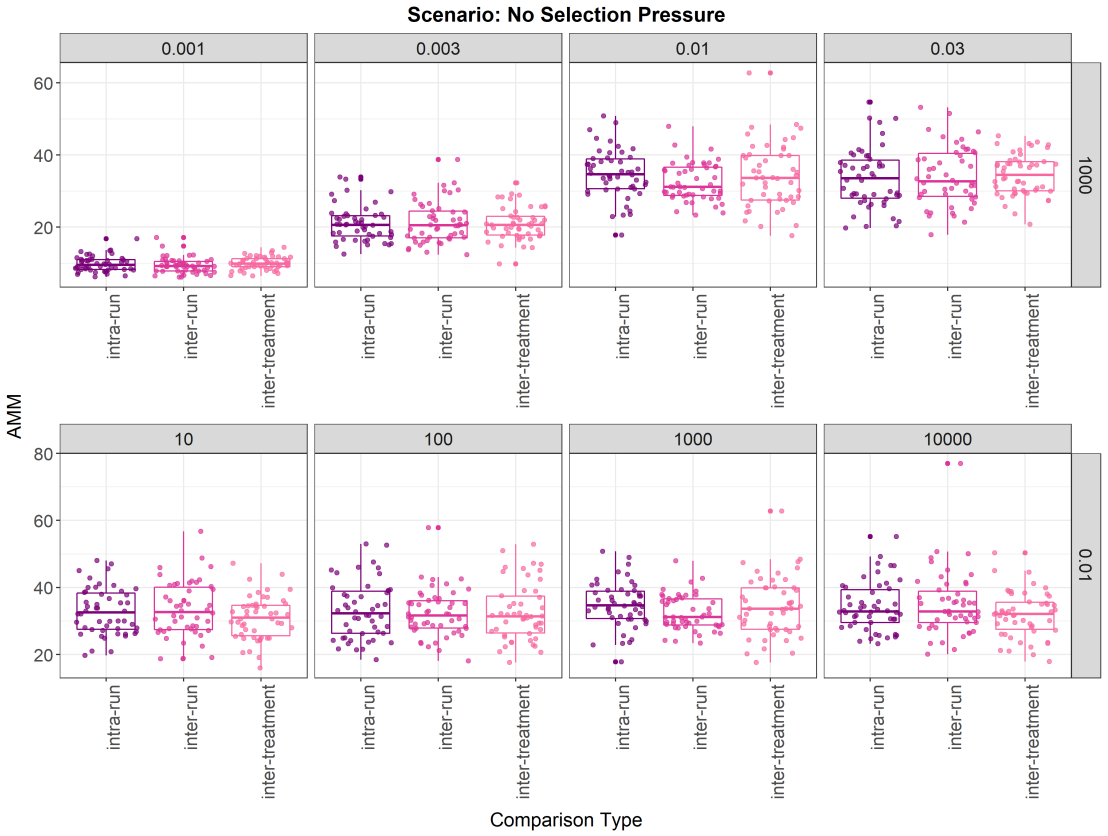

Figure 5.7: Box plots with AMM-values for scenario No Selection Pressure.

Additive Evolution

In this scenario the fitness values of the A- and B-cell are added to get the organism's fitness value, on which selective pressure is then present. Table 5.14 shows that the cells perform fantastic in configurations E1, E2, E3 and E7. In E4, the mutation rate is too high, and in E5, the population size too small. E6 represents a configuration, where the population size is still too small to perform really good, but big enough to not get terrible results.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E1	1000	0.001	A	100	100	100	100	50
			B	100	100	100	100	50
E2	1000	0.003	A	100	100	100	100	50
			B	100	100	100	100	50
E3	1000	0.01	A	99	100	100	99.92	50
			B	98	100	100	99.9	50
E4	1000	0.03	A	0.79	34.87	61.91	35.26	50
			B	11.79	37.38	69.92	36.81	50
E5	10	0.01	A	6.88	43.91	86.97	45.52	50
			B	9.84	47.93	81.94	47.31	50
E6	100	0.01	A	74.97	88.48	100	88.62	50
			B	75.96	90.48	100	90.52	50
E7	10000	0.01	A	99	100	100	99.98	50
			B	100	100	100	100	50

Table 5.14: Summary statistics for fitness scores in scenario Additive Evolution.

Table 5.15 shows the mean numbers and percentages of beneficial, deleterious and neutral mutations that have occurred in this scenario. Again, evolution works as intended and a higher mutation rate means higher mean values. The gap between beneficial and deleterious mutations in terms of percentages is bigger than in scenario No Selection Pressure.

In scenario Additive Evolution, the heat maps showed no diagonal lines and in the AMM box plots only the inter-treatment values were statistically significant. However, this is not enough to speak of co-evolution and, therefore, no couplings were detected in this scenario with the metric. The author would like to mention that intra-run versus inter-run was significant in configurations E1, E2, E4, E5 and E6, but the plots showed that they were significant in a bad way since the AMM-values for the intra-run comparison were more diverse than the ones for the inter-run comparison.

Config.	Cell	Ben.	Del.	Neu.	% Ben.	% Del.	% Neu.	Count
E1	A	100.24	8.64	0.02	92.09	7.89	0.02	50
	B	99.8	7.72	0.02	92.87	7.11	0.02	50
E2	A	91.6	24.04	0.72	78.72	20.67	0.61	50
	B	91.3	25.38	0.92	77.67	21.57	0.77	50
E3	A	133.46	104.36	7.62	54.39	42.54	3.08	50
	B	130.78	102.66	7.76	54.24	42.56	3.2	50
E4	A	1786.62	1786.02	702.22	41.81	41.79	16.4	50
	B	1794.5	1789.78	704.1	41.87	41.75	16.38	50
E5	A	984.24	731.3	183.34	52.16	38.53	9.32	50
	B	971.62	717.8	175.48	52.47	38.51	9.01	50
E6	A	431.56	297.02	22.32	57.73	39.36	2.91	50
	B	435.66	296.02	23.82	57.88	39.03	3.09	50
E7	A	88.76	84.22	6.12	49.58	47.05	3.37	50
	B	90.44	85.1	5.74	49.85	47	3.15	50

Table 5.15: Summary statistics for mutation types in scenario Additive Evolution. All values represent the mean values for beneficial, deleterious or neutral mutations. The values are provided as absolute numbers and as percentages.

Zero-Off Lockstep

Zero-Off Lockstep seems to be a difficult scenario for the setup since the organisms have a hard time adapting to the evolutionary fitness goal, as seen in Table 5.16. Only in E7, the organisms perform well. Apart from that, the table shows exactly what the author was hoping for: A- and B-cells differ by at most some decimal places but never by more than one whole number. This is due to this scenario's setup that does select for organisms, whose A- and B-cells lie level with each other in terms of leading ones.

The mutation type analysis for Zero-Off Lockstep shows similar results and confirms that evolution is working as expected, too. Figure 5.8 shows that a higher mutation rate causes more mutations. This is as it should be, but it is surprising that the lift from 0.01 to 0.03 has about a six-fold effect and not a threefold one, as expected. Moreover, this figure pictures that the smaller the population size gets the more mutations occur although the mutation rate stayed the same. This is because a smaller population has much less ability to filter out certain mutations. It is hard to purge deleterious mutations in small populations and so, they hitchhike along. In Figure 5.9 the ratio of deleterious and beneficial at a population size of 10 is almost equal, whereas at populations of 100, 1000 and 10000 it is not. This shows that evolution is not able to optimize to the target very quickly in smaller populations. This figure also depicts that at a mutation rate of 0.001 almost all mutations are beneficial. The explanation for that is that it is easier to avoid deleterious mutations at a lower mutation rate, as they occur all by themselves and are wiped out quickly since mutations are so rare.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E1	1000	0.001	A	1	6	10	5.46	50
			B	1	6	10	5.46	50
E2	1000	0.003	A	24	38	52	38.36	50
			B	24	38	52	38.36	50
E3	1000	0.01	A	84	88	93	88.16	50
			B	84	88	93	88.16	50
E4	1000	0.03	A	23.79	30.37	31.88	30.19	50
			B	23.84	30.36	31.88	30.19	50
E5	10	0.01	A	-0.05	3.94	8.98	3.86	50
			B	-0.04	3.94	8.95	3.85	50
E6	100	0.01	A	24.99	32.99	41.99	32.69	50
			B	24.98	32.99	41.99	32.69	50
E7	10000	0.01	A	100	100	100	100	50
			B	100	100	100	100	50

Table 5.16: Summary statistics for fitness scores in scenario Zero-Off Lockstep.

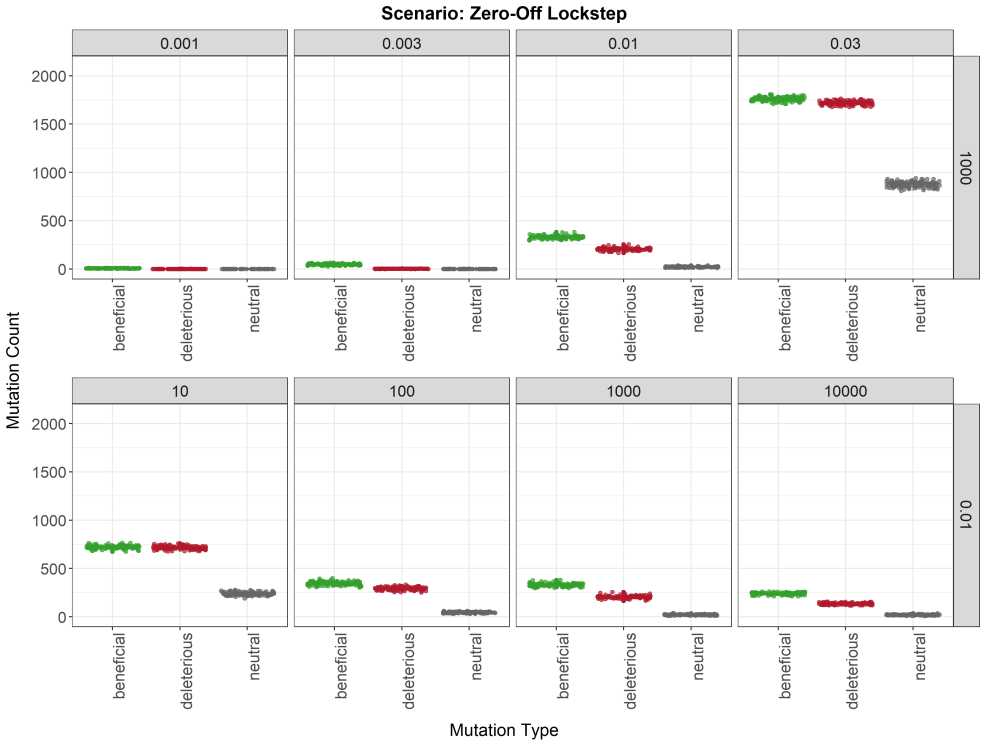

Figure 5.8: Result figure for mutation count analysis in scenario Zero-Off Lockstep.

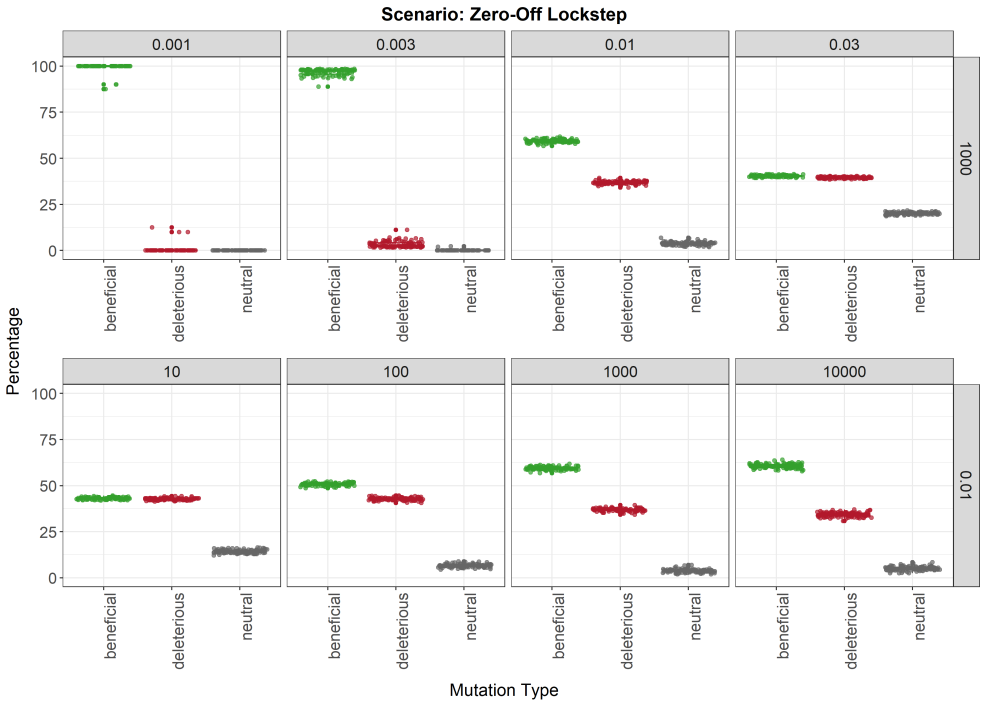

Figure 5.9: Result figure for mutation count percentages in scenario Zero-Off Lockstep.

As Figure 5.10 shows, diagonal lines are visible in five of seven configurations in this scenario. Therefore, the heat maps show indications for a co-evolutionary behavior in configurations E1, E2, E3 and E6. In the other configurations, either the mutation rate was too large (E4), too small (E7) or the population size was too small (E5).

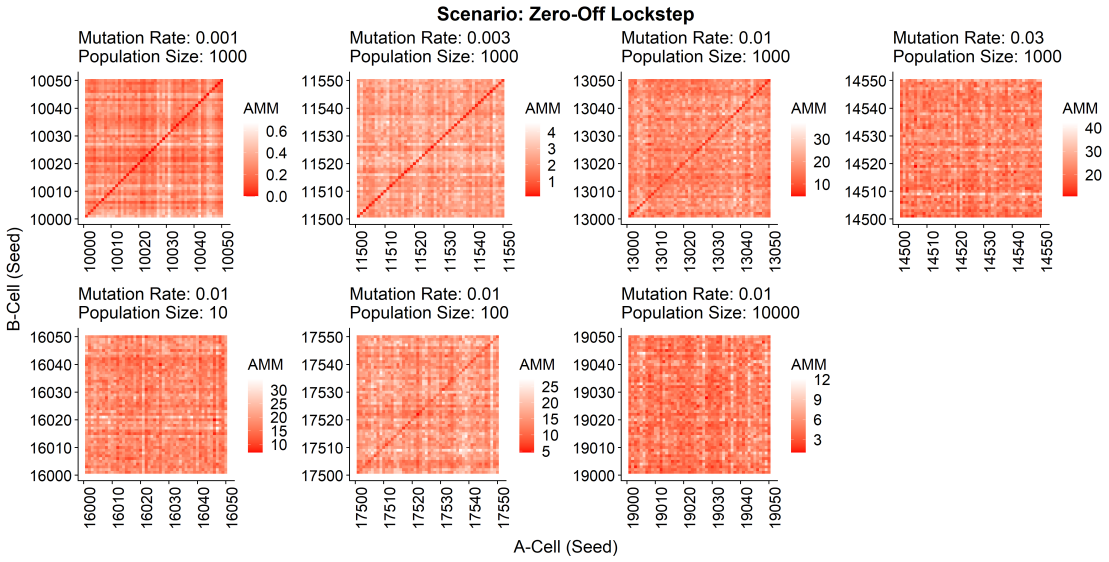

Figure 5.10: Heat maps with AMM-values for scenario Zero-Off Lockstep.

Figure 5.11 confirms the indications from the heat maps: Configurations E1, E2, E3 and E6 are significant with p-values of $< 2e^{-16}$, $< 2e^{-16}$, $< 2e^{-16}$, $9.10e^{-16}$ (intra-run versus inter-run) and $< 2e^{-16}$, $< 2e^{-16}$, $< 2e^{-16}$, $2.70e^{-14}$ (inter-run versus inter-treatment), using Wilcoxon rank-sum tests.

Configurations E4 and E5 are truly not showing signs of co-evolution, either due their p-value (E4: 0.63; Wilcoxon rank-sum) or from Figure 5.11, where the intra-run values of E5 are higher and more diverse than the inter-run ones. Configuration E7 is significant, both in terms of p-values (0.00067 for intra-run versus inter-run) and in terms of the figure, where intra-run AMM-values are smaller than inter-run ones.

Therefore, the metric has successfully detected co-evolution in configurations E1, E2, E3, E6 and E7.

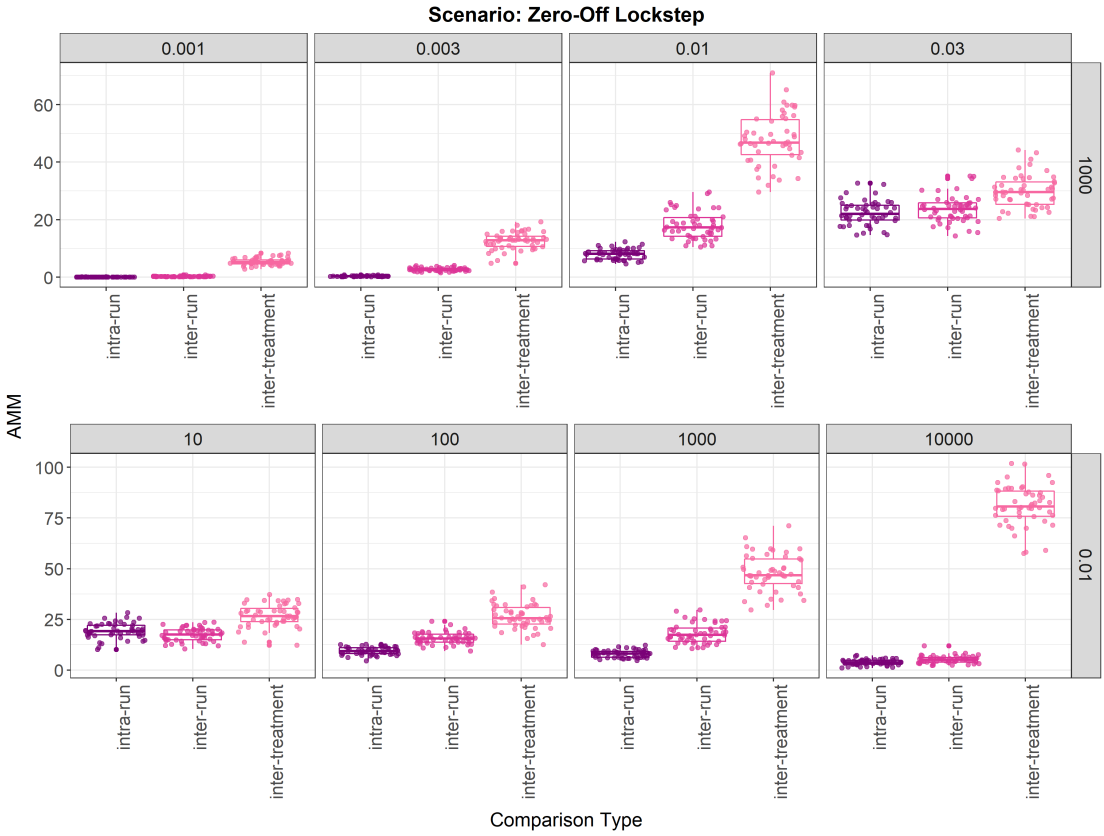

Figure 5.11: Box plots with AMM-values for scenario Zero-Off Lockstep.

One-Off Lockstep

One-Off Lockstep is still a difficult scenario, but overall the organisms perform much better than in Zero-Off Lockstep. Table 5.17 shows that organisms perform well in E1, E2 and E7. And configuration E3 works quite good too. Again, the table shows exactly what the author was hoping for: A- and B-cells differ by at most one and some decimal places but never by more than that. One-Off Lockstep selects for organisms, whose A-cells and B-cells are apart by one leading one at most.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E1	1000	0.001	A	100	100	100	100	50
			B	100	100	100	100	50
E2	1000	0.003	A	100	100	100	100	50
			B	100	100	100	100	50
E3	1000	0.01	A	79.97	93	100	92.77	50
			B	78.95	92.99	100	92.29	50
E4	1000	0.03	A	26.84	32.81	33.84	32.14	50
			B	25.84	31.84	32.87	31.66	50
E5	10	0.01	A	-1	22.93	43.95	21.89	50
			B	-1	21.94	43.92	21.37	50
E6	100	0.01	A	31.91	72.97	86	68.71	50
			B	31.94	72.47	85.98	68.17	50
E7	10000	0.01	A	100	100	100	100	50
			B	100	100	100	100	50

Table 5.17: Summary statistics for fitness scores in scenario One-Off Lockstep.

Table 5.18 shows the results for the mutation type analysis for scenario One-Off Lockstep. Overall, the results look similar to the ones from Zero-Off Lockstep. Only the difference is bigger, but this is plausible since in Zero-Off Lockstep not much evolution is happening due to its strict setup. The scenario is too challenging as that good final results could be achieved. In the One-Off Lockstep, evolution works great and almost moves the lockstep pattern trough the whole genomes of the A- and B-cells. One-Off Lockstep gives evolution the freedom of one generation spare time for mutations in the B-cell to catch up with corresponding ones in the A-cell. This is an important and necessary relaxation for evolution to perform good in these lockstep-like scenarios.

In One-Off Lockstep, intra-run versus inter-run and inter-run versus inter-treatment p-values from Wilcoxon rank-sum are significant in configurations E1, E3, E4, E6. The heat maps shown in Figure 5.12, conducted using the AMM-values, show indications of co-evolution for configurations E3, E5 and E6. And from the box plots of Figure 5.13, it is visible with the naked eye that co-evolution is present in configuration E3. To summarize this knowledge, the Accumulated Mutations Metric detects co-evolutionary behavior in E1, E3 and E4.

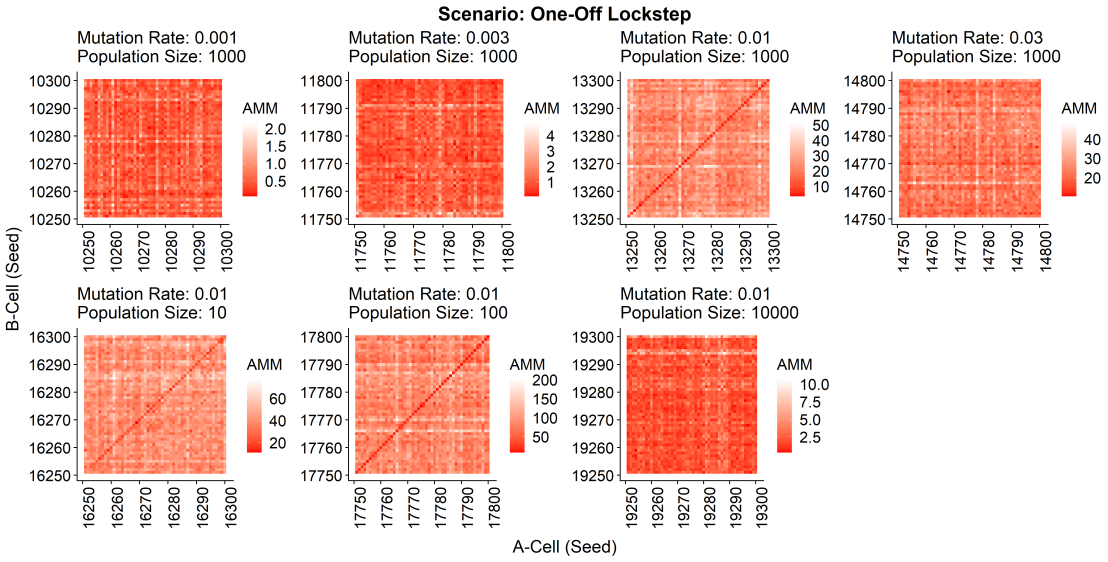

Figure 5.12: Heat maps with AMM-values for scenario One-Off Lockstep.

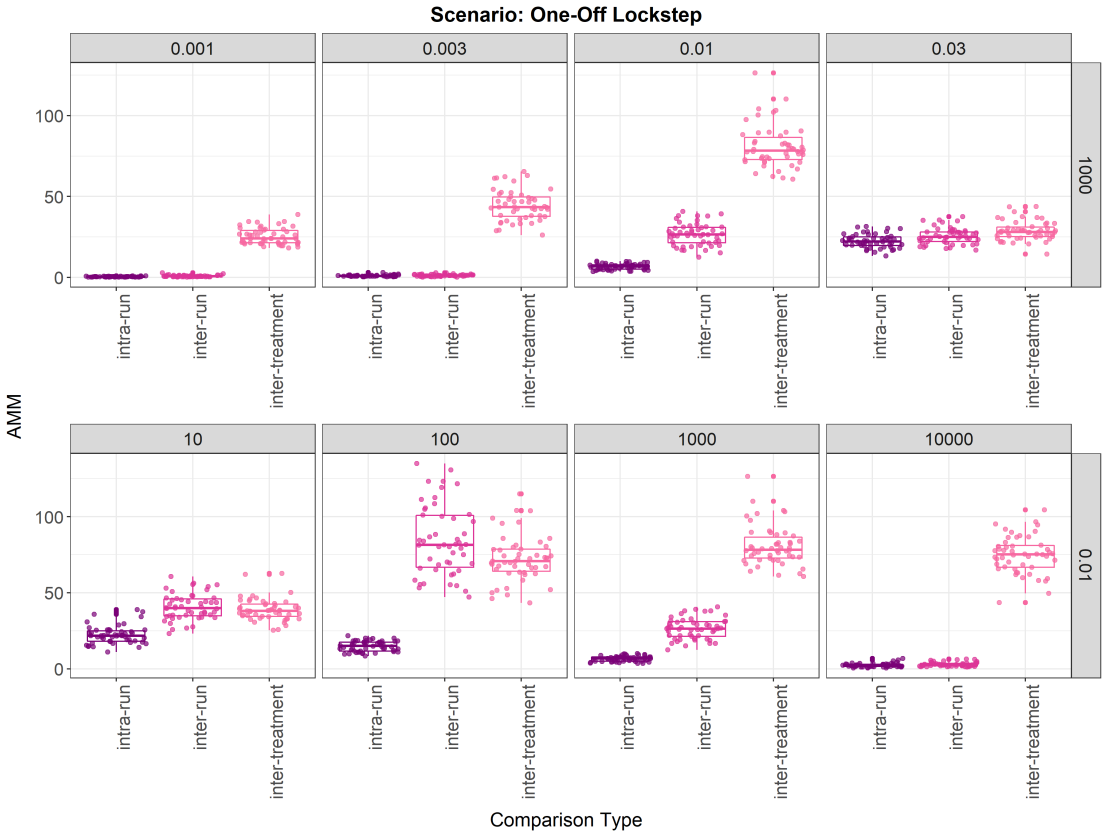

Figure 5.13: Box plots with AMM-values for scenario One-Off Lockstep.

Config.	Cell	Ben.	Del.	Neu.	% Ben.	% Del.	% Neu.	Count
E1	A	105.86	7.42	0.02	93.51	6.47	0.02	50
	B	104.56	6.66	0.02	94.05	5.93	0.02	50
E2	A	125.42	23.84	1	83.56	15.79	0.65	50
	B	125.12	23.52	1.18	83.64	15.57	0.79	50
E3	A	386.24	178.94	21.42	65.87	30.49	3.65	50
	B	393.4	186.52	22.56	65.3	30.95	3.75	50
E4	A	1784.86	1729.52	841.42	40.98	39.71	19.32	50
	B	1786	1731.94	844.9	40.94	39.7	19.37	50
E5	A	929.36	730.08	235.78	49.04	38.53	12.43	50
	B	932.72	739.24	239.88	48.79	38.67	12.55	50
E6	A	742.92	422.74	67.66	60.22	34.3	5.47	50
	B	751.64	429.58	68.04	60.16	34.41	5.43	50
E7	A	157.58	83.32	15.1	61.55	32.56	5.88	50
	B	162.16	87.48	13.16	61.73	33.26	5.01	50

Table 5.18: Summary statistics for mutation types in scenario One-Off Lockstep. All values represent the mean values for beneficial, deleterious or neutral mutations. The values are provided as absolute numbers and as percentages.

One Follows

As Table 5.19 shows, organisms perform mostly fantastic in this scenario. Only in E4, the mutation rate is too high to perform good and in E5, the population size is too small. Those results are reasonable since One Follows is an Infinite-Off Lockstep that therefore resembles Additive Evolution. Since the B-cells follow the A-cells in terms of leading ones, B's fitness trails behind A's.

In One Follows, Table 5.20 shows that evolution works correctly. Figure 5.14 shows additional information that is hidden in the table: The figure displays with its scatter plots a higher and a lower area across all seven configurations. One area represents A-cells, whereas the other one represents the B-cells. This is due to this scenario's setup, in which B-cells must be behind A-cells in terms of leading ones. And they can be behind them by an arbitrary number of leading ones. This setup explains the different mutation counts that become visible through a split into this two distinct areas.

In the One Follows scenario, no difference between intra- and inter-run lineage pairs was detected. This is reasonable since one of the lower-level individuals can fall behind the other by many leading ones, so the lineages are less tightly linked, allowing "unanswered" mutations to accumulate in one of the lineages. Hence, AMM was not able to detect any tight coupled co-evolutionary relationships. The heat maps show no diagonal lines and in the box plots, the intra-run AMM-values are in none of the configurations smaller or less diverse than the inter-run values.

It makes perfect sense that the metric does not work in this scenario, although only in retrospective. This scenario can be thought of as an "Independent-Off Lockstep",

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E1	1000	0.001	A	100	100	100	100	50
			B	100	100	100	100	50
E2	1000	0.003	A	100	100	100	100	50
			B	100	100	100	100	50
E3	1000	0.01	A	99	100	100	99.96	50
			B	95.98	100	100	99.26	50
E4	1000	0.03	A	40.83	52.41	69.93	53.27	50
			B	2.72	19.31	36.84	19.08	50
E5	10	0.01	A	45.92	64.94	98	65.21	50
			B	3.84	28.93	45.93	28.29	50
E6	100	0.01	A	86.97	96	100	95.57	50
			B	75.94	82.98	91.99	83.49	50
E7	10000	0.01	A	100	100	100	100	50
			B	92.97	100	100	99.86	50

Table 5.19: Summary statistics for fitness scores in scenario One Follows.

Config.	Cell	Ben.	Del.	Neu.	% Ben.	% Del.	% Neu.	Count
E1	A	99.64	7.28	0.04	93.21	6.75	0.04	50
	B	101.38	8.6	0.06	92.19	7.76	0.05	50
E2	A	89.12	23.06	0.5	79.1	20.45	0.45	50
	B	94.3	26.68	0.74	77.54	21.86	0.6	50
E3	A	80.02	54.7	5.6	57.01	39.07	3.92	50
	B	190.32	148.62	9.92	54.55	42.61	2.84	50
E4	A	1679.96	1668.2	594.3	42.62	42.32	15.07	50
	B	1888.62	1883.98	800.5	41.3	41.2	17.5	50
E5	A	801.28	555.92	101.12	55.07	38.07	6.86	50
	B	1150.7	873.66	262.46	50.34	38.2	11.45	50
E6	A	292.4	176.16	11.98	60.87	36.64	2.48	50
	B	585.62	418.62	34.98	56.35	40.29	3.36	50
E7	A	55.32	46.8	4.62	51.82	43.94	4.24	50
	B	109.3	106.58	7.36	48.96	47.75	3.29	50

Table 5.20: Summary statistics for mutation types in scenario One Follows. All values represent the mean values for beneficial, deleterious or neutral mutations. The values are provided as absolute numbers and as percentages.

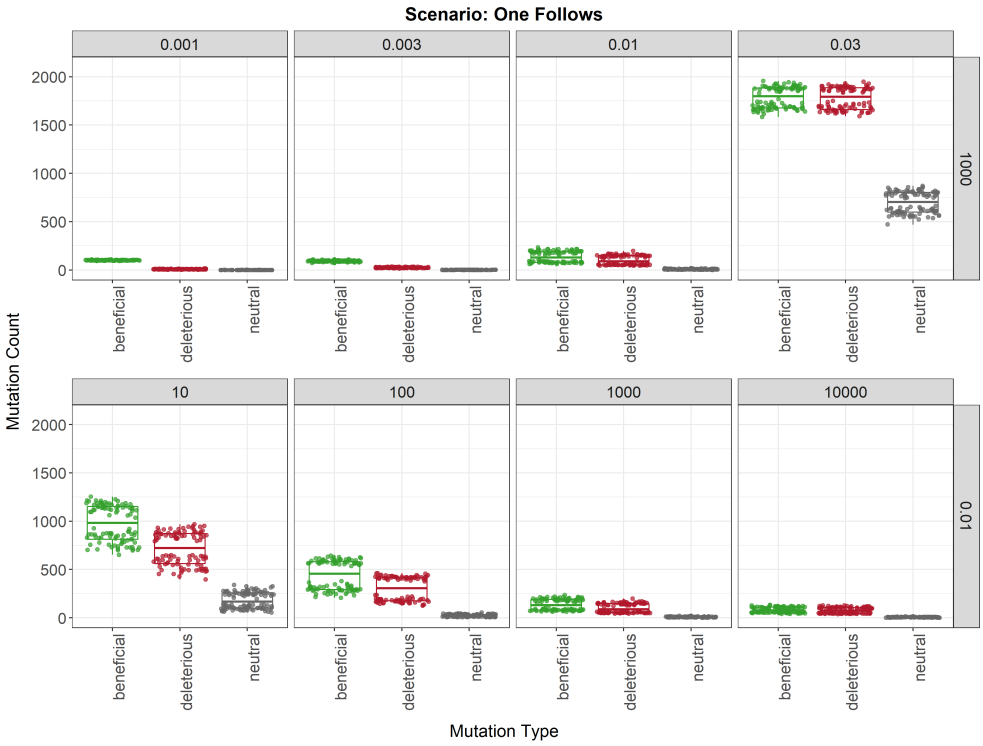

Figure 5.14: Result figure for mutation count percentages in scenario One Follows.

where the author initially thought that it could be possible to see a signal. It turns out that this was near-sighted. Since the B-cell can fall behind the A-cell in terms of leading ones by an "infinite" number, the AMM is no longer able to detect this signal. The cells are not tightly coupled and co-evolution is unidentifiable with this metric as it looks for clear, precise lockstep patterns in tightly coupled organisms. The mutation strings between intra-run and inter-run replicates are simply to similar to identify a signal.

Matching-Bits Lockstep

This scenario is subject to a different fitness function but still requires the lockstep-like pattern for solving the problem. The number of matching bits is counted, multiplied by ten and the overall number of ones in A and B is added. The fitness score for the overall organism equals the ones for the A- and B-cell. Table 5.21 shows that this scenario was easier to solve since the organisms almost every time got to perfect scores in configurations E1, E2, E3, E6 and E7. And there was neither a populational meltdown with high mutation rates (E4), nor was a quite small population a big obstacle (E5) for performing good in this scenario.

Table 5.22 shows the results of the mutation count analysis for scenario Matching-Bits Lockstep. The table looks similar to the counts from scenarios Zero-Off Lockstep (see Figures 5.8 and 5.9) and One-Off Lockstep (see Table 5.18). Therefore, this table is evidence that evolution works as expected in terms of different mutation rates and population sizes and nothing unexpected happened.

Config.	Pop. Size	Mut. Rate	Cell	Min.	Median	Max.	Mean	Count
E1	1000	0.001	A	1184	1194	1200	1194.44	50
			B	1184	1194	1200	1194.44	50
E2	1000	0.003	A	1200	1200	1200	1200	50
			B	1200	1200	1200	1200	50
E3	1000	0.01	A	1189	1200	1200	1198.66	50
			B	1189	1200	1200	1198.66	50
E4	1000	0.03	A	940	973.50	1018	973.46	50
			B	940	973.50	1018	973.46	50
E5	10	0.01	A	1016	1066	1112	1063.34	50
			B	1016	1066	1112	1063.34	50
E6	100	0.01	A	1151	1165	1184	1164.76	50
			B	1151	1165	1184	1164.76	50
E7	10000	0.01	A	1189	1200	1200	1199.56	50
			B	1189	1200	1200	1199.56	50

Table 5.21: Summary statistics for fitness scores in scenario Matching-Bits Lockstep.

Config.	Cell	Ben.	Del.	Neu.	% Ben.	% Del.	% Neu.	Count
E1	A	97.22	0	0	100	0	0	50
	B	97.22	0	0	100	0	0	50
E2	A	99.46	0.52	0	99.48	0.52	0	50
	B	99.24	0.7	0	99.3	0.7	0	50
E3	A	137.86	113.9	7.26	53.23	43.97	2.8	50
	B	138.4	113.98	7.66	53.24	43.84	2.92	50
E4	A	1977.24	1982.66	397.96	45.37	45.5	9.13	50
	B	1979.88	1978.66	399.86	45.43	45.4	9.17	50
E5	A	696.6	700.14	276.1	41.65	41.85	16.51	50
	B	691.02	692.26	274.52	41.68	41.76	16.56	50
E6	A	275.92	261.28	55.5	46.56	44.07	9.37	50
	B	279	260.06	55.6	46.93	43.73	9.34	50
E7	A	119.62	96.16	4.92	54.2	43.59	2.21	50
	B	121.18	95.42	5.12	54.7	43	2.29	50

Table 5.22: Summary statistics for mutation types in scenario Matching-Bits Lockstep. All values represent the mean values for beneficial, deleterious or neutral mutations. The values are provided as absolute numbers and as percentages.

The AMM heat maps indicate co-evolution in scenarios E1, E2 and E4. The AMM box plots detect co-evolutionary relationships between the A- and B-cells in configurations E1, E2, E4 and probably E6[5]. Kruskal-Wallis and Wilcoxon rank-sum tests approve these findings: Those four configurations are all significant in a good way. It is reasonable that the AMM does not detect co-evolution in configurations with a small population size (E5) or a low mutation rate in relation to the population size (E7).

Still, it was very surprising for the author that no co-evolution could be detected in configuration E3, although co-evolution was detected with higher (E4) and lower (E1, E2) mutation rates. The author looked further into it to find out what the cause of this behavior is. By looking at the raw genome data, the author saw that a per-site mutation rate of 0.01 is the optimal mutation rate since both, A- and B-cells, make it to all ones in their genomes very quickly. And if both genomes make it to mostly ones in most of the replicates that means that there is no difference between intra-run and inter-run data since genomes of all ones are compared to each other. All of the other mutation rates were significant since the genomes have more trouble building up all ones and thus, there are differences in the number of ones when compared with the AMM. With a mutation rate of 0.01, the AMM computes the sample variance of a lot of zeros and logically, then no signal is visible.

With this new knowledge, the conclusion for the Accumulated Mutations Metric is that this metric is useful when evolution is going on *and* when the gene that should be measured with the metric is not fixed within the population. Since the genomes get to perfect scores quickly in configuration E3 in this Matching-Bits Lockstep scenario, the genomes are fixed from early generations onwards. That is why the AMM cannot measure any difference between the intra-run and inter-run comparison. When genomes are nearly perfect from early generations onwards, there is no difference that could have been found between intra-run and inter-run data. This insight is important and valuable when it comes to the limitations of the AMM.

Conclusions

The fitness score analysis shows how good the organisms were able to adapt to the fitness goal. It seems, the "Counting-Leading-Ones"-problem is hard to master, especially in lockstep-like scenarios. By contrast, the "Matching-Bits"-problem is far easier to solve.

The mutation type analysis showed across all configurations and scenarios that evolution works exactly as it is supposed to. This is good and an important prerequisite for being able to truly test for co-evolutionary relationships with the AMM.

With the AMM, the author intended to provide a metric to test for tight co-evolution between different types of cells. AMM does not require any experimental manipulation and can therefore be utilized in various laboratory circumstances. AMM successfully detects simultaneous genetic changes between species. Specifically, a genetic signature is visible in the lockstep-like scenarios: Zero-Off Lockstep, One-Off Lockstep and Matching-Bits Lockstep. In those scenarios, the simultaneous genetic changes were detected by the AMM since the intra-run, inter-run and inter-treatment comparisons were significantly different. Matching-Bits Lockstep revealed an important limitation of the AMM when it comes to optimal mutation rates that result in perfect genomes.

[5] According figures can be found at the GitHub repository. [73]

5.2 Multi-Level Selection

So far, the work only focused on the high-level selective pressures, mandating that both cells in the higher-level organism would always be passed on to the offspring. Under these idealized conditions, where mutualisms were forced, a genetic signature could be detected, albeit only when beneficial mutations in one cell type were closely linked to beneficial mutations in the other cell type.

In this setup, three fitness functions are present and based on the degree of target fulfillment for every single one of them over time, conclusions can be drawn as to how the lower-level selection mechanism interacts with the higher-level one. This is tested with migration rates ranging from zero to hundred percent. All experiments are conducted with a population size of 1000, a per-site mutation rate of 0.01 and 5000 generations. These parameters are fixed and the only variable one is the migration rate. In the first generation, genomes are initialized with all zeros, which means that A's- and the organism's fitness start off perfect, whereas B's fitness is terrible. The author chose to initialize genomes with all zeros to minimize noise in the data and to see how A plays its advantage off over B.

The expectation is that individuals (*i.e.,* A- and B-cells) dominate and the organism's fitness is rather low at high migration rates, whereas the opposite behavior is observed at low migration rates. The truly interesting conclusions will be drawn form intermediate migration rates since it is not known yet how the cells and the organism react. At a migration rate of ten percent, it could be that ten percent of the population ends up emphasizing individual fitness and 90 percent organism fitness or it could also be that the whole population splits the difference. With no migration, "you better work with your partnered cell" since it is there for the rest of your life. As soon as migration is introduced, it is no longer guaranteed that this is the case.

Moreover, the author analyzed the co-existence of different sub-groups with a varying migration rate in this context: Are there any clusters of high and low fitness over time? Therefore, A's fitness, B's fitness and the organism's fitness are plotted at an early, middle and late generation. It could be possible to see some sort of speciation-event, where some organisms are specialized to be mutualist and always win on the group-level front and some are specialized to work independently and only win at that front. If something like this happens, the author expects that those "species" stay separated.

All experiments regarding multi-level selection were done with the data of whole populations. This is due to technical limitations of MABE to track lineages from migrated organisms. In a multi-level population, the lineage is not linear but a tree. The whole populations are analyzed in three different ways:

1. Averaged fitness score analysis: How do A-cells, B-cells and organisms evolve over time in terms of fitness?
   (see Section 5.2.1, datafile "migration_line_chart_pop.csv" and R-script "b_linechart.R")
2. Frequency of fitness scores: How are the fitness scores of A-cells, B-cells and organisms distributed within a population over time?
   (see Section 5.2.2, datafile "migration_facet_snapshots.csv" and R-script "b_facets.R")

3. Subgroup analysis: Are different subgroups existing within a population? If yes, what do the different niches represent?
   (see Section 5.2.3, datafile "migration_scat_hist_snapshots.csv" and R-script "b_scatter_hist.R")

All three analyses were conducted with eleven different migration rates (*i.e.,* from zero percent to 100 by tens), resulting in 660 replicates, which were run on Michigan State University's High Performance Computing Cluster. Since showing all figures would go beyond the constraints of this thesis, the author decided to show a subset of all migration rates. For the sake of completeness, all figures can be found on the GitHub repository [73]. The following sections, describing the individual analyses, are all structured equally: First, the analysis is described, then the results of the experiments with selected migration rates are shown and discussed, and finally, conclusions are drawn.

5.2.1 Fitness Score Analysis over Time

With this analysis, the fitness score improvement over 5000 generations is analyzed. Specifically, the influence of the migration rate is studied. For this purpose, the author constructed line charts that each show the fitness scores over time for one migration rate, split into the scores for A-cells, B-cells and the overall organism. The y-axis shows the generations and the x-axis depicts the mean fitness score. "Mean" because the author ran 20 replicates and averaged their fitness scores per generation. To make the line charts more descriptive, minimum and maximum scores over all 20 replicates are drawn into the plots with a shaded area.

The expectation is that lower migration rates benefit the organism's fitness, whereas the cells achieve high fitness scores with high migration rates.

Experiments with Different Migration Rates

Figure 5.15 shows that A-cells and the overall organisms always start with a perfect fitness score of 100. In the experiments, the genomes of the cells are initialized with all zeros. Since A-cells aim for zeros, their fitness scores are perfect. And the organisms' fitness is perfect as well, since A's and B's genomes match at the beginning due to the initialization. Moreover, the figure shows that the spread between maximum and minimum fitness scores becomes smaller as the migration rate increases.

At a migration rate of zero percent, the organism performs well and the individual cells level off at a medium fitness score of about 50. A migration rate of 20 percent means that 80 percent of every new generation's organisms were formed via group-level selection and 20 percent arose from individual-level selection, where well adapted A-cells and B-cells were selected and randomly matched up as organisms. At this rate the overall organisms do not perform as good as before, but still better as the individual cells. At a rate of 40 percent, a turnaround is visible and the individual-level consisting of A- and B-cells attain better fitness scores than the group-level organisms. This trend intensifies as the migration rate is raised and at a rate of 100 percent, the organisms' fitness is at rock bottom, whereas the fitness scores of A- and B-cells are near perfection.

Another observation from Figure 5.15 is that at a migration rate of zero percent, where only groups are moved on, the individual cells are still able to perform mediocre, while at a migration rate of 100 percent, where no groups but only individuals are

selected, the organism fitness score is extremely bad. Additionally, the figure shows that the higher the migration rate is, the quicker the fitness scores settle. At lower migration rates, the individuals require more generational time before they stagnate.

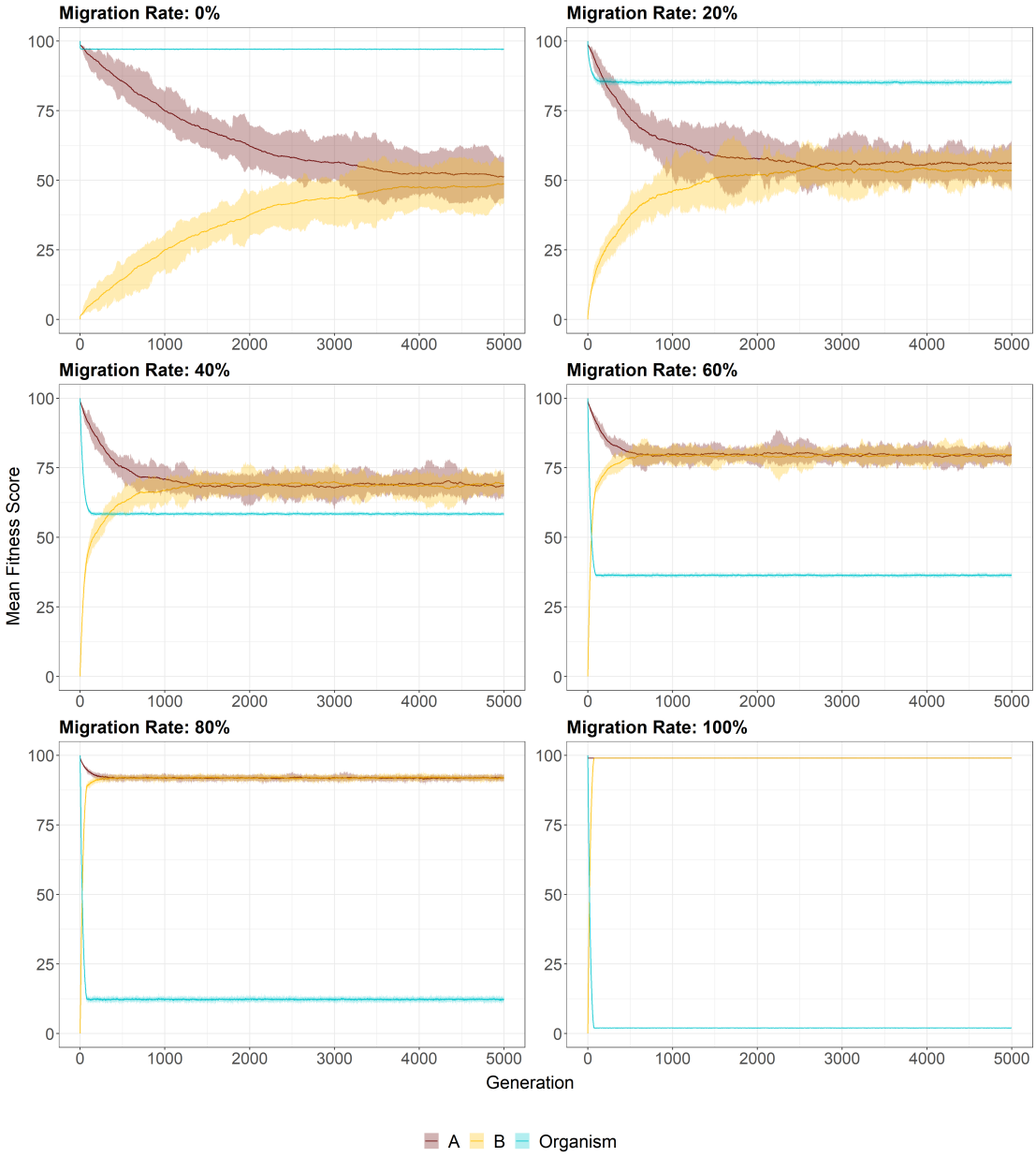

Figure 5.15: Results for migration rates of zero, 20, 40, 60, 80 and 100 percent.

Conclusions

The results for the different migration rates are very close to the author's expectations: The migration rate plays an important role in whether the fitness score is high on the

organism-level or on the level of the individual cells. At first glance, it might be confusing why cells still perform mediocre at zero percent migration and organisms perform so poorly at a migration rate of 100 percent. In hindsight, this makes perfect sense:

When organisms perform very well, this means that the genomes of A- and B-cells match. A's aim for zeros and B's for ones. At zero migration, they both reach fitness scores of about 50. This means that they match perfectly and their genomes, regardless of whether it is a cell of type A or B, consist of about 50 zeros and 50 ones. Therefore, they match perfectly and still perform moderate in regards of their individual goals. This supports the conclusion that the selective pressure for A's and B's individual goals is about equal. And this is indeed true since there is zero pressure on the individual-level selection because it does not exist.

And when the individual-level selection pressure is very high but the group-level one is non-existent, as it is the case with a migration rate of 100 percent, A's genomes consist mostly of zeros and B's mostly of ones. Therefore, almost nothing matches between the two cells that form one organism and the organism fitness score is near zero.

The author was surprised that the turning point for organism fitness versus individual fitness is between migration rates of 20 and 40 percent and not between 40 and 60 percent. Here, the behavior differs from what the author had expected.

5.2.2 Fitness Score Frequency over Time

The motivation for this analysis was that averaging fitness values across different replicates could hide behaviors within a single population. Therefore, the author decided to also analyze the frequency of certain fitness scores in individual replicates over time. Again, the fitness score was split up into the ones for A-cells, B-cells and the overall organism. The aim is to illustrate how the fitness scores are distributed within a population over time. For this purpose, the scores ranging from zero to one hundred were pooled into five distinct bins.

Each figure shows the results for one specific migration rate. The fitness scores are split up into the ones for A-cells, B-cells and the overall organism. Moreover, each of the twenty replicates run in total is depicted on its own. The x-axis shows the generations and the y-axis depicts the count of individuals with a certain score. The different colors show the scores, condensed into frames of 20 fitness points each.

The expectations are similar to the ones from the fitness score analysis described prior, although these experiments are exploratory and the results for the behaviors within a single population are solely of a descriptive nature.

Experiments with Different Migration Rates

Figures 5.16, 5.17, 5.18, 5.19, 5.20 and 5.21 show the distribution of fitness values across all 1000 organisms for all 5000 generations per replicate for a certain migration rate, split into A-cells', B-cells' and organisms' fitness. Overall the patterns look quite like what could have been expected from the previous averaged fitness score analysis.

In Figure 5.16, replicates 40445 and 40453 show completely different patterns for A-cells and B-cells. Individuals with fitness scores between 20 and 40 are far more frequent in later generations than in most of the other replicates.

In Figure 5.17 replicate 40491 is the most unique one. Starting at around generation

4000, some A-cells suddenly have fitness scores between 20 to 40 and some B-cells between 40 to 80. The overall organisms show very distinct and constant patterns over time. While some organisms perform very good (*i.e.,* fitness values of 80 to 100), a small amount performs mediocre. The assumption is that the organisms with scores between 20 and 60 were formed via individual-level selection and, therefore, perform less good in the organism fitness function, whereas the ones with scores between 80 and 100 were formed via group-level selection.

Figure 5.18 seems to show very clear results for all of the organisms that moved on to the next generation via group-level fitness selection. However, the ones that were formed via individual-level selection show a number of different phenotypes for A-cell's and B-cell's fitness scores.

Figures 5.19 and 5.20 show that no hidden dynamics are present when the organisms were selected on the group-level. On the individual-level selection, there are still patterns visible that vary from replicate to replicate. Regarding the organism fitness, again, no patterns are visible apart from that organisms that were selected on the group-level perform very good and the ones that were selected on the individual-level perform poorly.

In Figure 5.21, all organisms for the next generation were selected on basis of their individual cell fitness. Not surprisingly, A- and B-cells achieve very good fitness scores and organisms do extremely bad. Underlying dynamics are not visible anymore.

Conclusions

The figures make it clear that solely averaging across different replicates as done with the fitness score analysis over time is not a sufficient way to describe the interaction of group-level and individual-level selection mechanisms. It for sure provides a high-level overview for the mechanics, but it might be that underlying dynamics remain hidden, as some replicates are going extreme in one direction and others in another one, and averaging then destroys those dynamics. The twenty replicates for each migration rate revealed several distinct phenotypes across a population, which are lost when using averages.

To conclude, the figures disclose the variety of possible phenotypes in different replicates. Therefore, averaging across all of them might hide some dynamics within populations and it is of utmost importance to identify and analyze all of the possible dynamics within populations to gain a better understanding of the interaction between group-level and individual-level selection mechanisms.

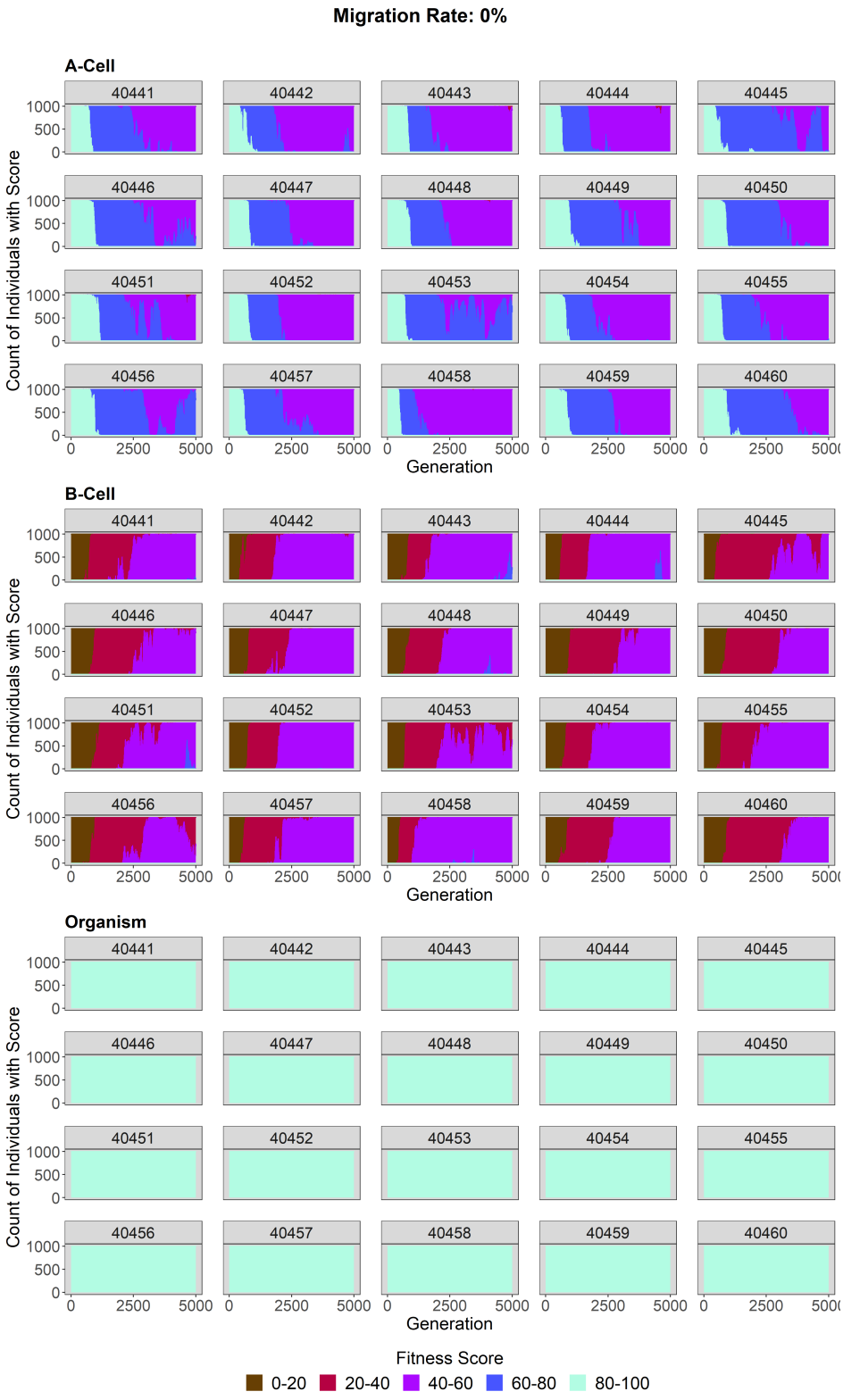

Figure 5.16: Fitness score frequency for a migration rate of zero percent.

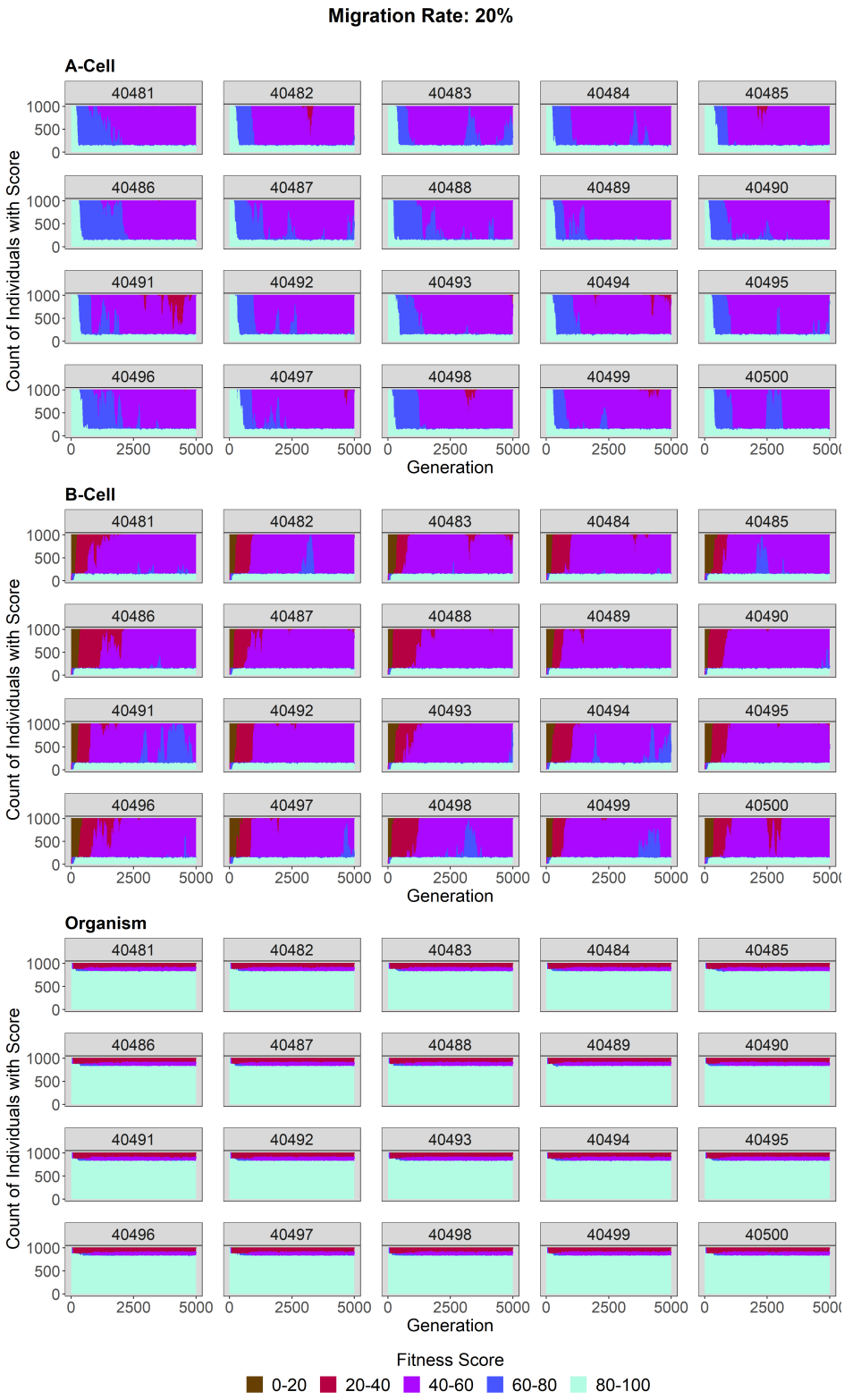

Figure 5.17: Fitness score frequency for a migration rate of 20 percent.

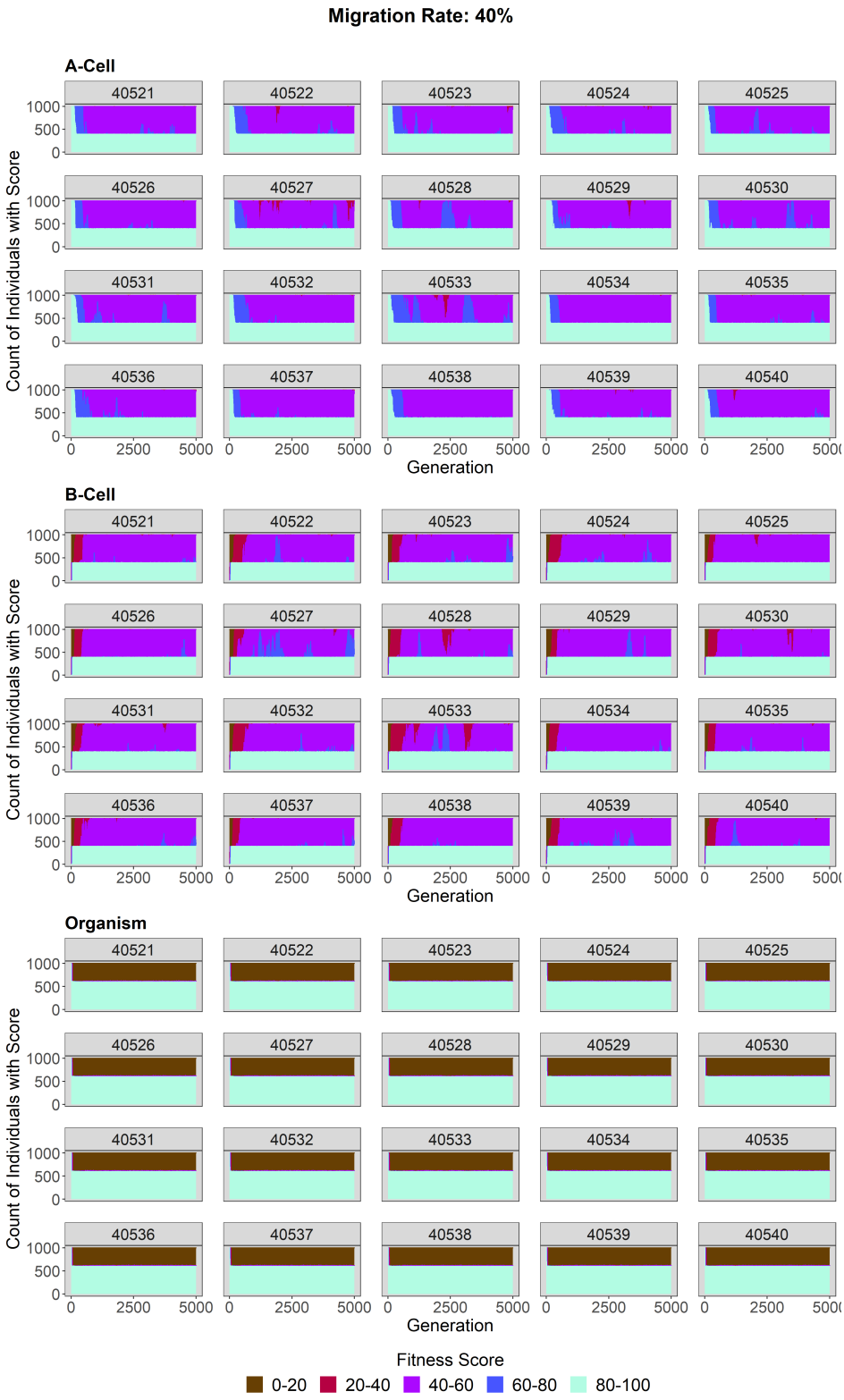

Figure 5.18: Fitness score frequency for a migration rate of 40 percent.

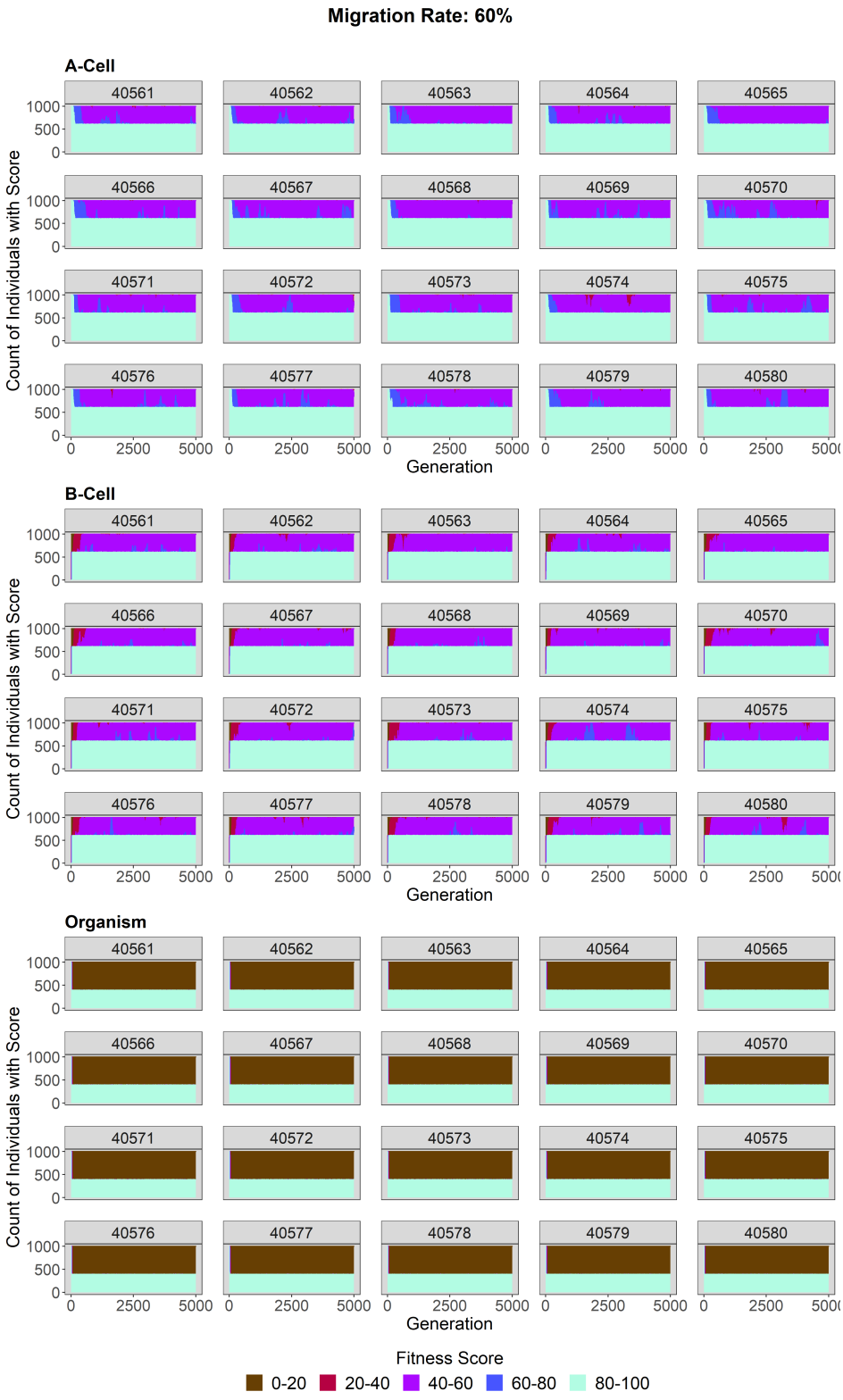

Figure 5.19: Fitness score frequency for a migration rate of 60 percent.

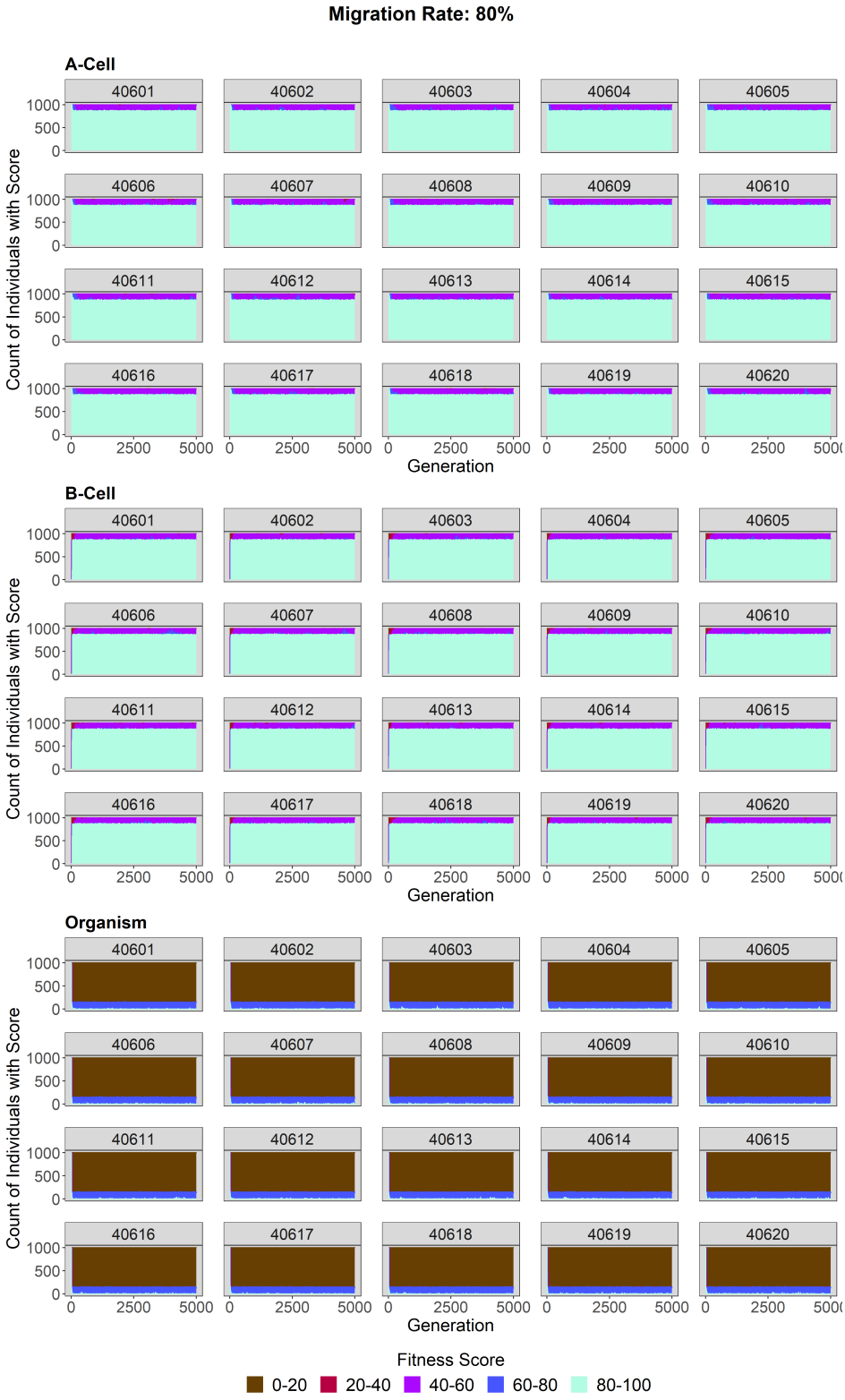

Figure 5.20: Fitness score frequency for a migration rate of 80 percent.

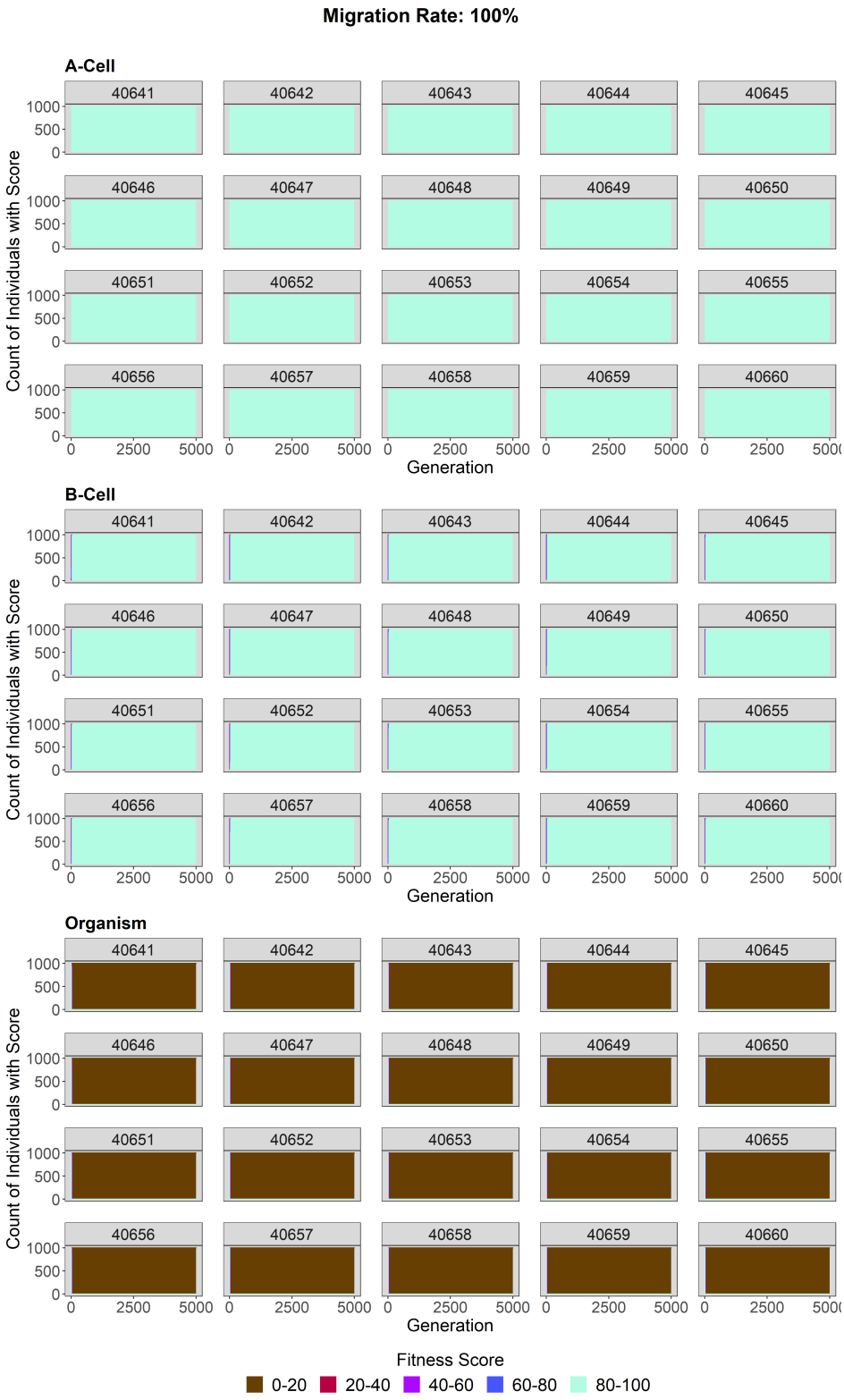

Figure 5.21: Fitness score frequency for a migration rate of 100 percent.

5.2.3 Existence of Subgroups

The previous analysis showed strong signals for the existence of various phenotypes within a population, and this analysis looks for subgroups in a population. The expectation is that subgroups might exist since migration leads to specialists on the organism-level and on the individual-level. Little is known about how much migration is enough for subgroups to arise and at what rates they vanish. The author decided to look at single replicates and compare those by eye in order to find representative ones, which are included in this thesis, as opposed to averaging the values over several replicates, as this has shown to potentially cover important dynamics.

The figures for this analysis represent the result for one replicate at a specific migration rate. Result plots are shown for an early, a middle and a late generation (*i.e.,* generations 10, 2500 and 5000). The late generation shows the endpoint, which is most interesting, and the early and middle ones give an insight into whether what can be seen at the endpoint was happening all along or, if not, how the organisms changed their behavior over time. All plots show the fitness of A-cells on the x-axis and the fitness of B-cells on the y-axis. The three plots on the left-hand side color-code the boolean value "isFromGroup" from the source code. This flag tells whether an organism was formed via group-level selection (*i.e.,* not migrated), where an organisms as a whole replicated or via individual-level selection (*i.e.,* migrated), where individual cells replicated. The other three plots on the right-hand side color-code the overall organism fitness. In essence, those organism fitness plots on the right are a sanity check to double-check whether the results on the left are reasonable. For instance, dots that are situated in the middle of the scatter plot have mediocre individual scores and should therefore have high organism fitness since they are organism-specialists.

Each of the six plots shown within one figure consists of a scatter plot with appertaining, marginal histograms. The histograms make it easier to identify at first glance how many of the 1000 organisms of a population are located at what part of the scatter plot since the dots in the scatter plots might highly overlap and that makes it difficult to conceive the actual number of individuals in a certain area.

Experiments with Different Migration Rates

The author tried to pick representative replicates for the following figures. The result figures of all other replicates, that have been run (*i.e.,* 20 replicates for each of the eleven migration rates), can be found on the GitHub repository [73]. The following figures show replicates with migration rates of zero, ten, 20, 50, 80 and 100 percent. The author chose those since they show the most unique and interesting results. Showing all migration rates would have pushed the boundaries of this thesis.

Figure 5.23 shows the results for a migration rate of zero percent. Not surprisingly, all organisms within the population were formed via group-level selection and they have mediocre A- and B-cell fitness but very high organism fitness. This observation conforms to the results that the author showed with the two previous analyses.

At a migration rate of ten percent, not much of the overall behavior changed, as Figure 5.24 illustrates. The only difference is that ten percent of the overall population was formed via migration. In regards of fitness scores, not much difference is visible, apart from some dots having slightly lower organism fitness, while the A- and B-cell

fitness raises. At a migration rate of ten percent, the population is not able to establish any individuality, although ten percent of the population are being selected for their individual fitness. Surprisingly, those organisms keep getting pulled back to the middle of the scatter plot, as it seems that ten percent is not enough to establish a separate niche.

Starting at a migration rate of 20 percent (see Figure 5.25), evolution creates different subgroups within the population. The particular area in the scatter plots shows the niche that certain individuals specialized in during evolution. In total, four different subgroups are visible. Before seeing those results, the author would have expected two separate niches: The niche in the middle with the not migrated dots represents organisms that were formed via group-level selection. The upper right-hand corner niche with migrated individuals is what is expected from individual-level selection that always selects for individualists. Not surprising, for those organisms the A- and B-cell fitness is high, whereas the organism fitness is rather low.

Furthermore, Figure 5.25 depicts two (to three) additional niches the author has not expected a priori: The two off-diagonal niches and the one in the middle of the box plot consisting of organisms formed via migration. So, how do these niches arise and why are they perfectly reasonable? All of them show organisms that were formed via migration, which means that the A- and B-cell was selected due to its individual fitness. Now, if the A- or B-cell previously came from an organism-specialist that was formed via no migration and the other cell originates from an individual-specialist, previously formed via migration, the two off-diagonal niches are created. And the third niche, overlapping with the organisms created via group-level selection represents a combination of the effect that causes the off-diagonal niches. Due to the theory of probabilities, it makes perfect sense that a small amount of the 1000 organisms within a population are settling in those off-diagonal niches. The niche with the dots on the upper-left represents organisms whose B-cells were individualists and whose A-cells were previously organism-specialists. For the niche on the lower right-hand corner, it was the opposite way around. And the dots in the middle of the scatter plot, representing organisms which were formed via migration are caused by former organism-specialists that got very lucky, as they were competing in tournaments selecting for individual-level fitness, in which only other organism-specialists were present. Therefore, they are marked as formed via migration, although they perform very good at the organism-level and rather poorly in regards of the individual fitness goals. Figure 5.22 summarizes the niches that were formed via migration.

Probability calculation explains why organism-specialists sometimes win individual rounds, which causes that creation of niches, a priori not expected. The general formula for estimating the number of organism-specialists that win individual-selection tournaments (FOWI; standing for Former Organism-specialists Win Individual rounds) is described with Equation 5.1.

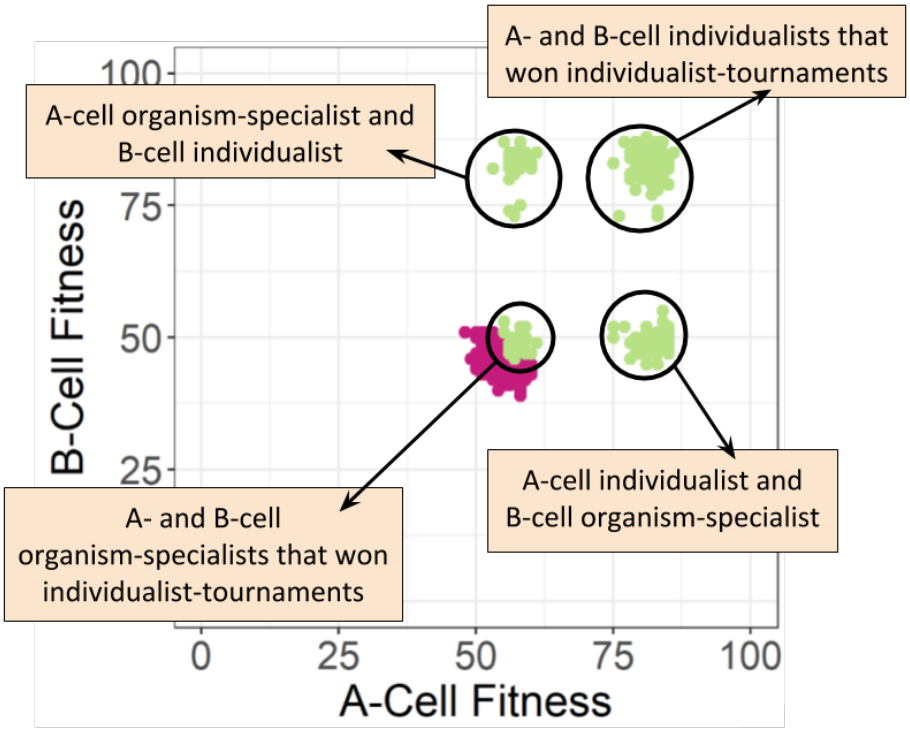

Figure 5.22: Subgroups formed via migration at a migration rate of 20 percent. The not encircled dots represent A- and B-cell organism-specialists that won organism-level tournaments and are formed via group-level selection.

$$\text{FOWI} = ((1 - \frac{r}{100})^t)^2 \cdot (p \cdot (1 - \frac{r}{100})) \tag{5.1}$$

where:

r = migration rate
t = tournament size
p = population size

Hence, the expression $p \cdot (1 - \frac{r}{100})$ describes the number of organisms effectively selected via group-level selection.

The estimated number for the reversed case of individualists winning group-level selection tournaments is described in Equation 5.2, where FIWO stands for Former Individualists Win Organism rounds. Here, the expression $p \cdot \frac{r}{100}$ describes the number of organisms effectively selected via migration (*i.e.,* individual-level selection). FIWO uses the same variables as FOWI.

$$\text{FIWO} = ((\frac{r}{100})^t)^2 \cdot (p \cdot \frac{r}{100}) \tag{5.2}$$

It is of utmost importance to realize that the only reason why *e.g.,* former organism-specialists win individualist-tournaments is because they got lucky and only competed against other organism-specialists. Also, those niches represent a small amount of the overall organisms selected via migration. The same goes for former individualists winning organism-specialists tournaments. Moreover, the numbers computed with FOWI and FIWO are the *estimated* numbers of the versatile evolution-process. The numbers will only be approximately correct since the "wrong" type (organism-specialist/individualist) can still win a tournament, leading to a "misclassification" in the next generation. Hence, the numbers will be very close, and the best approximations possible. Last, the author would like to clarify that the population is not actually forming four niches. There are always at most two, which are stably formed. The off-diagonal ones are not especially fit but keep getting re-formed from the ones that are doing well (*i.e.,* upper-right corner and middle). If one of those well-performing niches is too small, it becomes unlikely for an off-diagonal niche to be accidentally formed.

In Figure 5.25, 80 percent of all organisms are organism-specialists since the migration rate is 20 percent. Organisms for the next generation are determined with tournament selection and a tournament size of seven. This means that about 20 percent of the time (*i.e.,* $0.8^7 = 0.2097$), there will only be organism-specialists in a tournament. So, it does not matter whether the algorithm is selecting for individual- or group-fitness, there will only be organisms who won the last round as organism-specialists in the tournament, anyways. If that is happening about 20 percent of the time on average, it is no surprise that the figure shows off-diagonal niches, as well as this mixture in the middle of the scatter plot. The migrated dots over the not migrated ones in the middle occur, when it happens twice that there are solely organism-specialists in the tournament. This happens about 4 percent of the time (*i.e.,* $0.2097^2 = 0.044$). The population size in the experiments is 1000 and the population that is formed via group-level selection, is effectively of size 800. Therefore, theoretically around 35 individuals should be present in the middle subgroup, although they were selected via migration (*i.e.,* $0.044 * 800 = 35.2$; $FOWI \approx 35$).

Figure 5.25 also shows that at generation 10, the organisms start at the bottom right-hand corner since they are initialized with all zeros. The organism-specialists are pulled towards the middle and the individual-specialists are primarily pulled straight up. At generation 2500, the true individual-specialists are slightly pushed to the left and away from a perfect score of 100 due to the mutation-selection balance[6]. A mutation rate of 0.01 that results in one mutation on average due to a genome length of 100, is hefty to fight against for selection. Hence, the seen results make perfect sense.

With higher migration rates, as seen in Figures 5.26 and 5.27, it becomes less and less likely for organisms formed via migration to establish any other niche than the obvious one in the upper-right corner[7]. Due to fewer organisms being selected via group-level

[6] A mutation-selection balance is present when the ratio of deleterious alleles being created by mutation equals the ratio of deleterious alleles being eliminated by selection. The number of deleterious alleles within a population then are in equilibrium [12, 13, 39].

[7] Important clarification: With migration rates of 50 and 80 percent, it looks like there are more organisms concentrated in the middle niche than in the upper-right corner. To not be tricked by the scatter plot visualization, the marginal histograms tell the true story. Those show that there are about equally as many/far more organisms in the corner than in the middle. 500/200 organisms were selected based on their organism fitness, so it is reasonable to still see a bunch of organisms in the middle.

selection, fewer group-specialists are available to be selected from. The niche of organisms formed via migration in the middle of the scatter plot is eradicated, as both cells would have to be group-specialists. And this becomes less likely with higher migration rates (*e.g.,* for a migration rate of 50 percent, $FOWI = FIWO \approx 0.031$.).

Figure 5.27 shows that the off-diagonal niches vanish at a migration rate of 80 percent. There are no longer individuals present that are selected for their group effects in individualist-rounds ($FOWI \approx 0$). Only when a whole tournament consists solely of such organisms, one of them becomes the winner and moves its genome on to the next generation. With a higher migration rate, the possibility for this diminishes and the off-diagonal niches are no longer existent. In hindsight, this again is exactly what could have been expected. With a migration rate of 80 percent, suddenly some of the organisms selected on the group-level overlap with the niche created by organisms selected via migration. This is the reversed effect of what was visible at a migration rate of 20 percent. Again it is reasonable ($FIWO \approx 35$).

When all organisms are formed via migration, as shown in Figure 5.28, not surprisingly, all organisms are located in the upper-right corner of the scatter plot. They have excellent A- and B-cell fitness but abysmal organism fitness. This observation again conforms to the results that the author showed with the two previous analyses.

Conclusions

The expectations for this analysis were modest, as the author thought that these results are going to be unexciting in a sense of: "Seventy percent are selected as organisms and thirty as individual cells and that is exactly the division in the population." To a first approximation, it looks like this is exactly what is going on but by taking a step back from that, the results are decent since additional results were found.

Surprisingly, this analysis showed that not only two niches are formed, as expected but sometimes even four. This is due to organism-specialists taking over whole tournaments and being selected as winner during rounds, in which organisms are chosen based on their individual-level fitness (*i.e.,* via migration) or vice versa. As mentioned before, those niches are formed out of pure luck and solely the two expected ones are being formed stably, at most.

Moreover, the analysis illustrated that there is a little bit of noise around, to the point where ten percent migration was not enough to establish a separate niche. With a migration rate of ten percent, evolution is not able to establish any sort of individuality. This gives promising indication to look further into this type of research. Albeit, the deviation from the expectation makes perfect sense, it is still unknown how extreme it truly is.

To conclude, mutualism is present throughout all migration rates (except for 100 percent migration) since there were still organisms located in the center of the scatter plot. Apart from that, commensalistic relationships are observed since the organisms cannot actively harm each other. Hence, the overall organisms are doing what they can for themselves and the individual cells are either helping themselves or help the overall organisms.

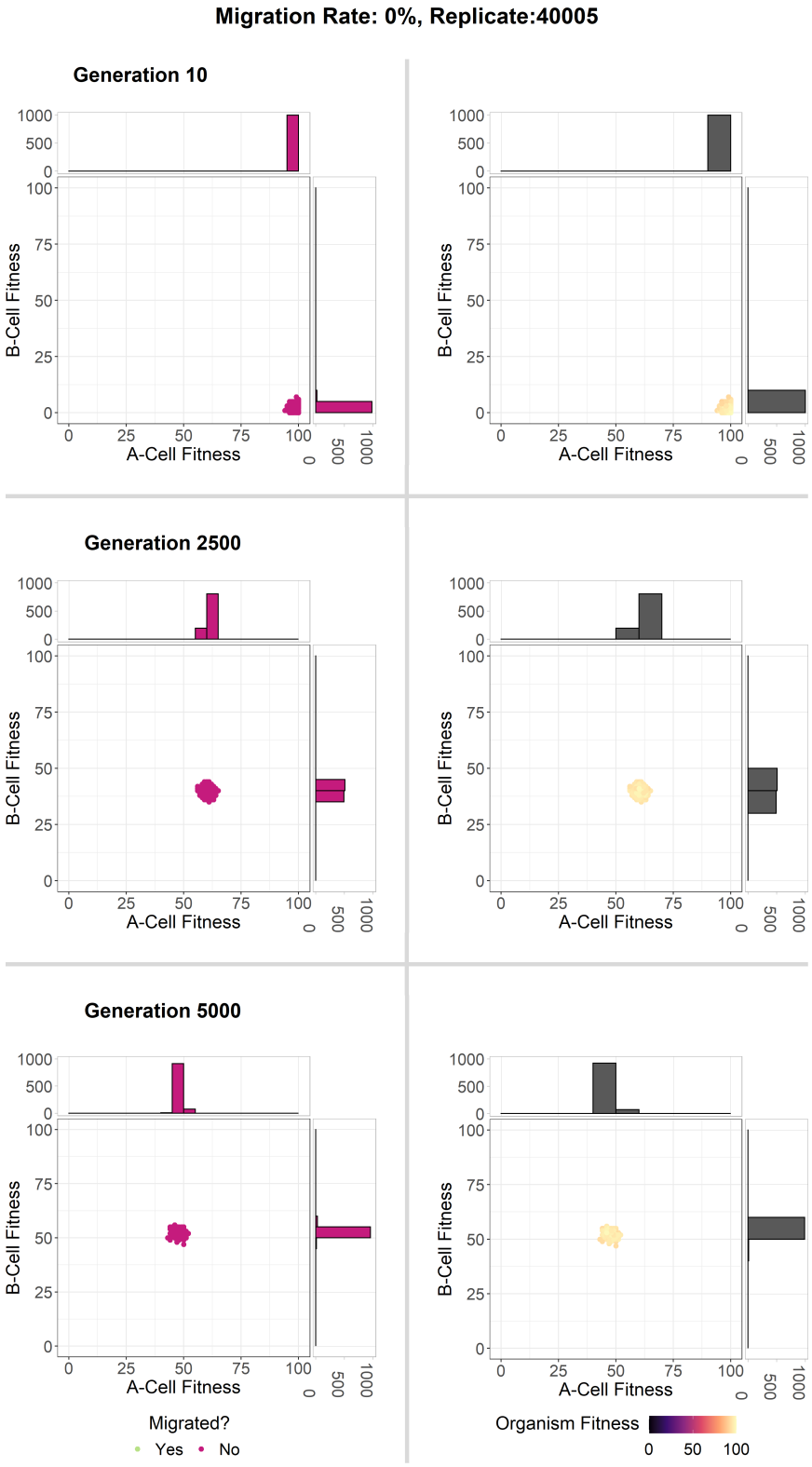

Figure 5.23: Analysis of subgroups for a migration rate of zero percent.

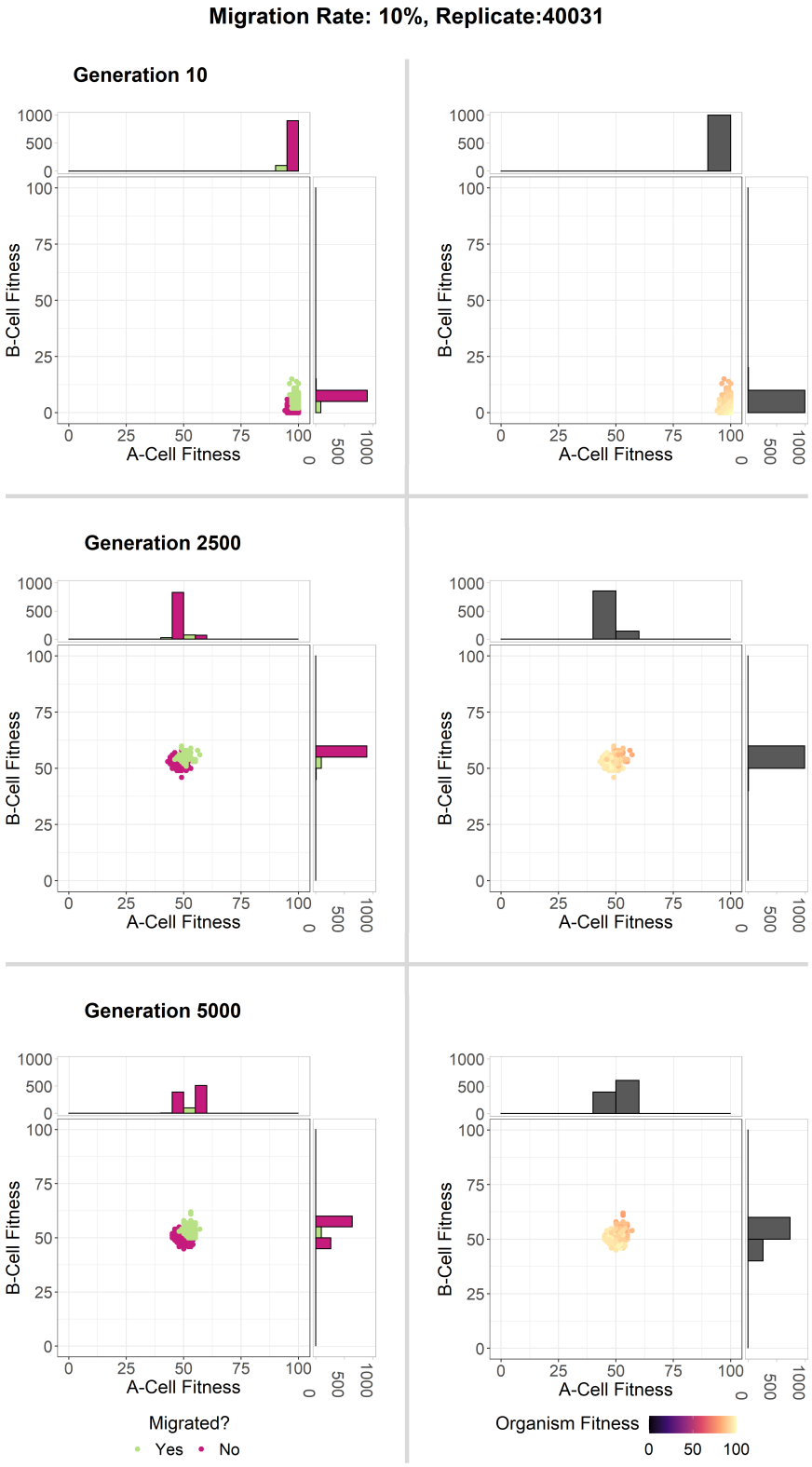

Figure 5.24: Analysis of subgroups for a migration rate of ten percent.

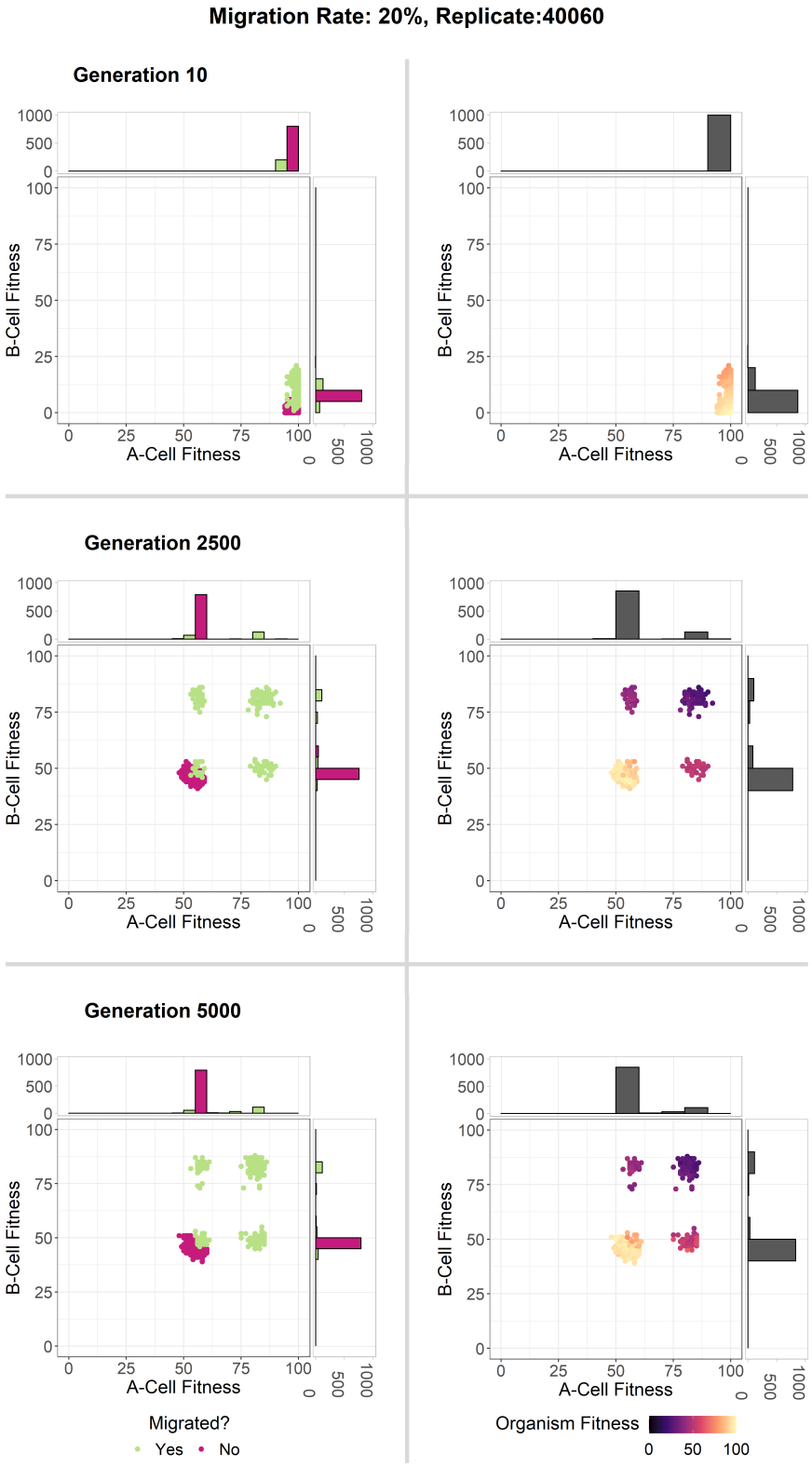

Figure 5.25: Analysis of subgroups for a migration rate of 20 percent.

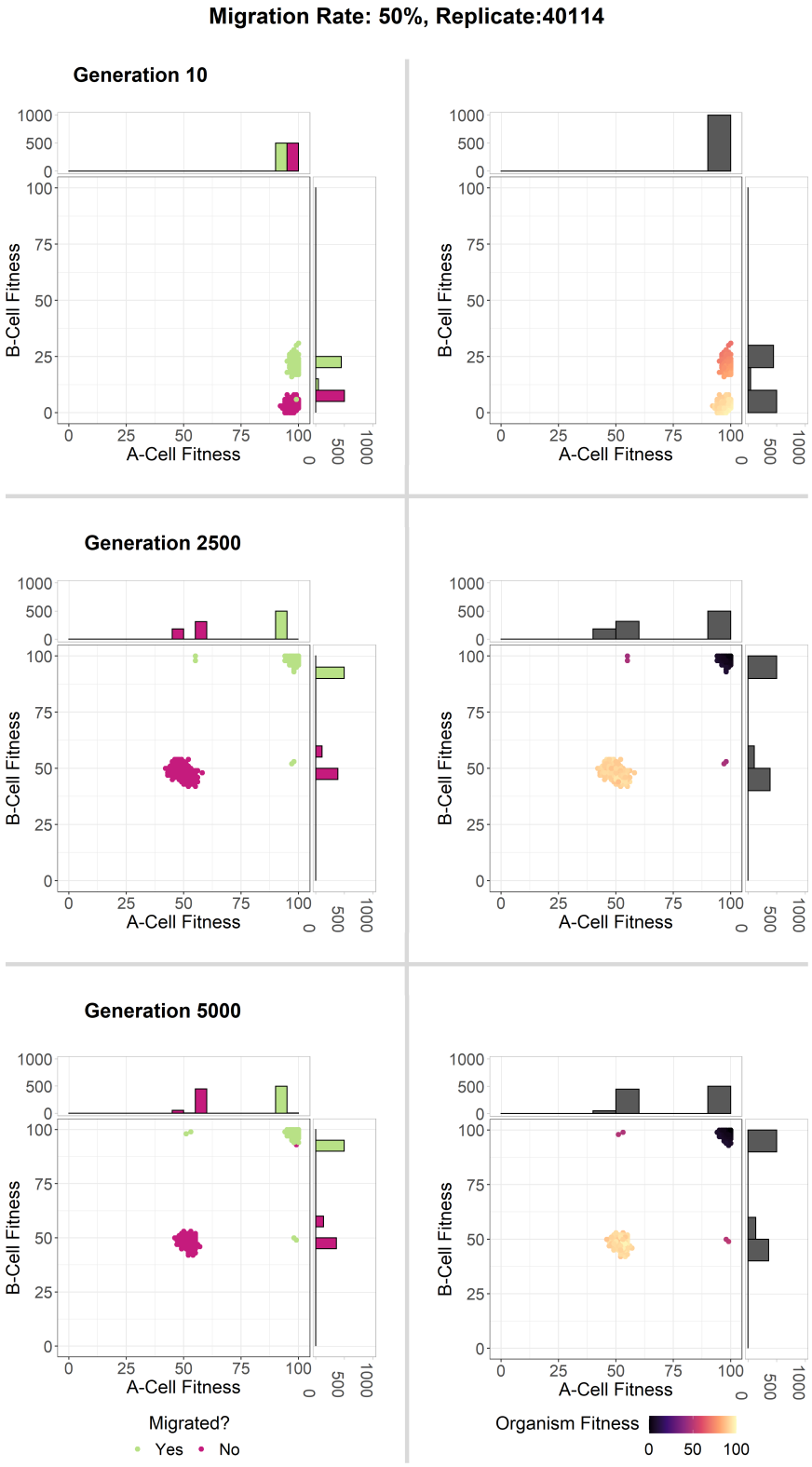

Figure 5.26: Analysis of subgroups for a migration rate of 50 percent.

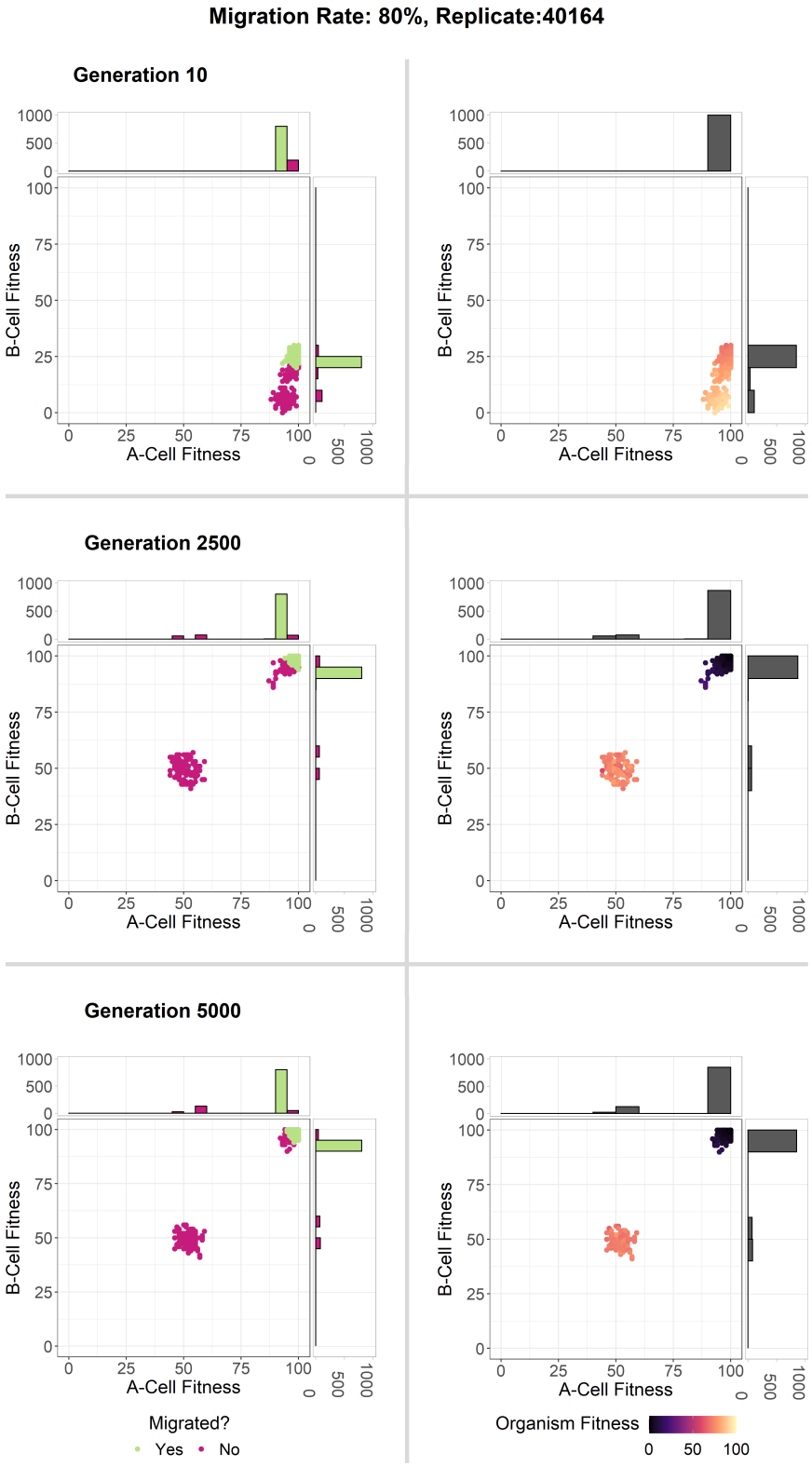

Figure 5.27: Analysis of subgroups for a migration rate of 80 percent.

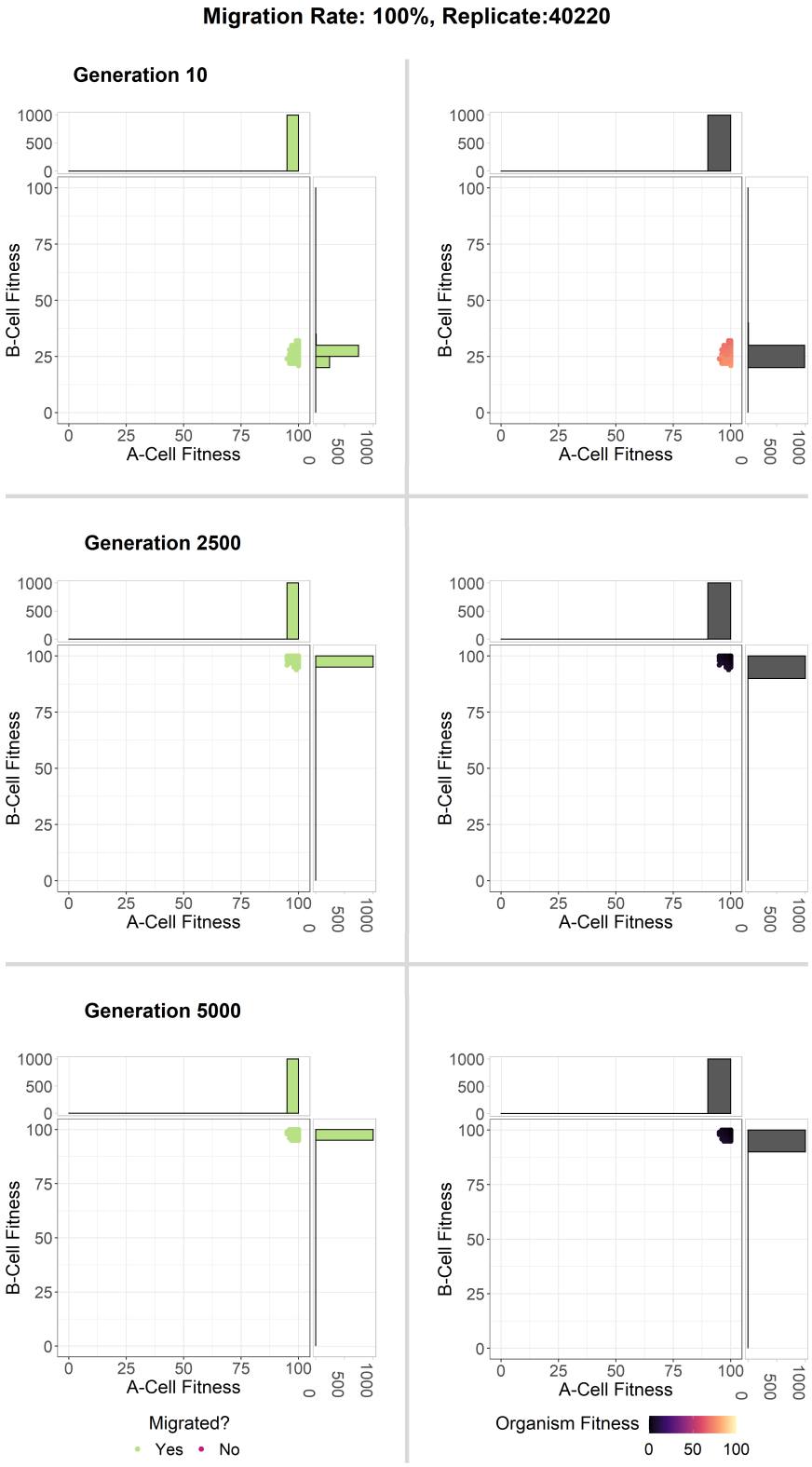

Figure 5.28: Analysis of subgroups for a migration rate of 100 percent.

Chapter 6

Conclusion and Future Work

This final chapter is divided into three sections: Section 6.1 summarizes the results of the experiments conducted, Section 6.2 states conclusions and the final Section 6.3 talks about possibilities to continue this herein presented work.

6.1 Results

As illustrated in Chapter 5, the author found promising ways to detect genetic signatures of co-evolution, in circumstances where experimental manipulation is and is not possible. Moreover, a first insight into the interaction of different levels of selection mechanisms were provided. Following, a concluding summary of those results is given in Sections 6.1.1 (Genetic Signatures) and 6.1.2 (Multi-Level Selection). The presented work yields promising results for in-depth analysis of biological data with lineage-based metrics.

6.1.1 Genetic Signatures of Co-Evolution

This thesis proposes two ways of detecting lockstep-like genetic signatures of co-evolutionary dynamics with lineage data. The first one is applicable when experimental manipulation is feasible and the second one works with just historical data, when no manipulation is possible. The results from Sections 5.1.1 and 5.1.2 are consistent in regards of different mutation rates and/or population sizes. Only when the mutation rate is very high (*i.e.,* configuration E4 with a mutation rate of 0.03) or the population size very small (*i.e.,* configuration E5 with a population size of 10), there is no evidence of a lockstep-like co-evolution between the two lower-level individuals that form the higher-level organism. The assumption is that such a high mutation rate/low population size leads to a meltdown in the population, which removes any actual difference.

Figures 6.1 and 6.2 show the quintessence of the ways for detecting co-evolution, proposed with this thesis. Scenarios No Selection Pressure and Zero-Off Lockstep were chosen for this summarized depiction since they are the most extreme ones: In the Zero-Off Lockstep, A's and B's beneficial mutations had to be in perfect sync and No Selection Pressure acts as control scenario.

The analysis of the mutation count (see Figure 6.1) gives indication for whether or not different types of cells are co-evolving. Therefore, mutation counts from configurations E3 (equal rates for A- and B-cells) and from D1 (differing rates) are compared.

If co-evolution is present, the mutation count from the cell with the non-manipulated mutation rate is elevated as well. In Figure 6.1, the patterns in the scenario No Selection Pressure act according to their mutation rates, but the pattern of the A-cell in scenario Zero-Off Lockstep in configuration D1 gets pulled along by the raise of the mutation rate of the B-cell, in comparison to configuration E3. Therefore, strong signals for no co-evolution in scenario No Selection Pressure and for co-evolution in scenario Zero-Off Lockstep are present. In conclusion, the A- and B-cells are indeed tightly-coupled in Zero-Off Lockstep and this analysis of the mutation counts, for which experimental manipulation in form of varying mutation rates is necessary, detected this genetic signature of co-evolution.

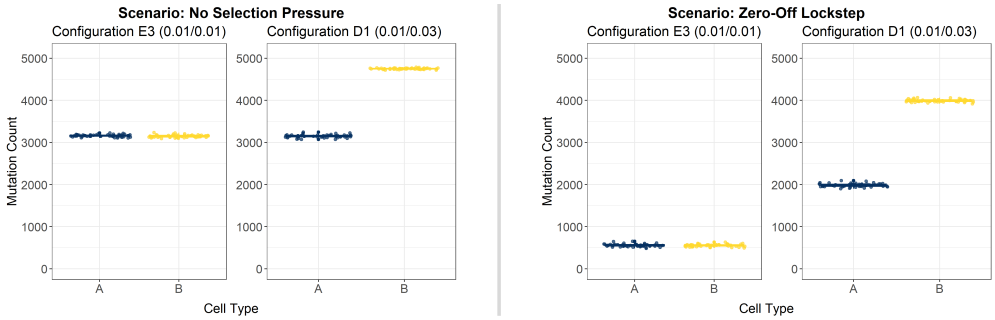

Figure 6.1: Summarized form of mutation count analysis for scenarios No Selection Pressure and Zero-Off Lockstep.

When no experimental manipulation is possible or the genetic signature of already available data should be analyzed, the Accumulated Mutations Metric (AMM) comes into play. AMM is able to detect possibly existing co-evolution from lineages with just historical data by analyzing the occurrences of beneficial mutations. A genetic signature is present, when the AMM-values differ between the intra-run, inter-run and inter-treatment comparison in ascending order. The most crucial difference is the one between the intra-run and the inter-run, where the intra-run values must be significantly lower than the inter-run values for co-evolution to be present. So, the AMM is capable of showing co-evolutionary dynamics in idealized scenarios and tightly-coupled species are detected. Figure 6.2 elucidates the detection of co-evolution: In scenario No Selection Pressure, the AMM-values are almost the same across the intra-run and inter-run comparison. As expected, no co-evolution is detected, as there is nothing in this scenario that could create co-evolution after all. In Zero-Off Lockstep, however, the metric outputs significantly lower values for the intra-run comparison than for the inter-run comparison. Therefore, co-evolution is present and AMM has detected the genetic signature of it. AMM looks for clear and precise lockstep-like patterns in tightly-coupled species. One potential weakness of the metric is that genomes must not be able to get to perfect scores early on in the evolutionary process, as AMM is not able to detect co-evolution in such circumstances (as seen in scenario Matching-Bits Lockstep). One way to eradicate this unwanted behavior is to make the fitness goal dynamic, by introducing a genome string that should be matched for perfect cell fitness scores, but this genome string slightly changes from one generation to the next. Therefore, there would never be one optimal solution and consistent pressure would be there. Another approach would be to elon-

gate the genomes to make it harder to perfectly match. In all presented experiments, genomes were of length 100. Although, those are possible approaches for eliminating this shortcoming of the metric, it might not be that relevant with biological sequence data since in nature there is no such thing as "perfect adaptation to a fitness goal".

Figure 6.2: Summarized form of AMM for scenarios No Selection Pressure and Zero-Off Lockstep.

6.1.2 Multi-Level Selection

The thesis made first steps into analyzing the interaction between group-level and individual-level selection mechanisms in egalitarian populations by introducing different fitness goals for A-cells, B-cells and the overall organism. The parameter "migration rate" was added to control the number of organisms that are picked via group-level and via individual-level selection to form a new generation. Three descriptive analyses were conducted to describe how different migration rates affect the evolutionary process along 5000 generations.

First, the fitness scores for A, B and the organism were analyzed over time by averaging the scores across 20 distinct replicates. This analysis provides a first insight into how well A-cells, B-cells and organisms adapt to their fitness goals with different migration rates. The results mostly matched the author's intuition for what should have happened. Truly surprising was the turning point, at which individual cell fitness took over overall organism fitness and in this way, commensalism took over mutualism. The author expected it at a migration rate of about 50 percent, but the visualizations showed that the turnaround happened earlier, somewhere between 20 and 40 percent migration.

The next analysis got rid of the averaging approach and looked at the frequency of certain fitness scores across twenty replicates, separately. The author found that populations have different phenotypes and, therefore, different underlying dynamics are present in replicates that were all produced with the same migration rate. Other than that, this analysis was of a descriptive nature to visualize entire populations. Overall, this was a more in-depth investigation of the fitness score analysis that has been conducted previously.

The third analysis addresses the existence of potential subgroups. It looks at individual replicates at an early, middle and late generation and analyzes the ratio of A-cell fitness to B-cell fitness. Surprisingly, the author found that a migration rate of ten percent is not enough to establish individuality within the population. Moreover, the experiments have shown that a maximum of four niches is established, in contrast to the expected two subgroups. Again, all results are explainable and hence, reasonable.

6.2 Conclusions

To conclude, preliminary work was done for identifying genetic signatures between different species and for better understanding the interaction of different levels of selection mechanisms. In this thesis, the underlying theoretical background, solution approaches for detecting co-evolutionary dynamics and implementation details, as well as conducted experiments were described and analyzed. Coming back to the research questions initially formulated in Chapter 1, genetic signatures are existing in egalitarian populations and can be identified with metrics; and the interaction of a lower-level selection mechanism and a higher-level one is interrelated with the migration rate.

While a metric for detecting genetic signatures of co-evolution in idealized scenarios when experimental manipulation is possible, was found straightforwardly, detecting it when no experimental manipulation can be done and just historical data is available, was far more challenging. With the Accumulated Mutations Metric (AMM), the author introduced a new measurement that screens for tightly-linked co-evolutionary dynamics between species in egalitarian populations based on genomic lineage data.

In regards of the second research question, the author found that different migration rates create various dynamics, as of how an individual-level selection mechanisms interacts with the one on the organism-level. Moreover, even the same migration rate results in a variety of phenotypes within a population, ranging from different populational patterns in the fitness score frequency to the appearance of niches. Those observations suggest that the population is diversified and not monolithic, as previously assumed. There are multiple strategies within a population, as replicates with the same migration rate showed slightly different results. The most exciting finding concerning this research question, was that a migration rate of ten percent across twenty independent replicates was not enough to establish individuality.

6.3 Future Work

From the very beginning, this research was meant to be exploratory to better understand the interaction of different levels of selection mechanisms and to provide a possibility for detecting a certain type of co-evolution from lineage-based data. There are several ways how this work could be continued:

- Development of a metric toolbox: The AMM, proposed with this work, provides an initial foray into a broad range of possible metrics to detect all sorts of co-evolutionary relationships between species. The ultimate goal is to detect that symbiosis is occurring, as well as its type. This can be achieved by developing a toolbox of synergistic approaches that can be layered on top of one another depending on

what manipulations can be performed and what data is accessible. This would allow to make predictions about lineage interactions in general. Accordingly, the lineages would run through all the applicable metrics from the toolbox and based on the ones where the lineages test positive, the relationship could be estimated. If *e.g.,* the AMM comes up positive while screening, this would mean that the lineages are super tightly-linked.

- More complex scenarios: Another possibility is to model more complex scenarios instead of the herein described idealized ones, to see how the AMM works there and what adaptations must be made to be able to observe the same genetic signatures there, as well. More complex scenarios could be progressively approaching more lifelike circumstances and eventually, experiments in wet labs to detect symbiosis could be conducted.
- Increment propagule size: So far, all experiments were conducted with a propagule size of two since an organism consists of an A-cell and a B-cell. Trying different propagule sizes would also be an interesting topic to look more into. An organism could look as shown in Figure 6.3.

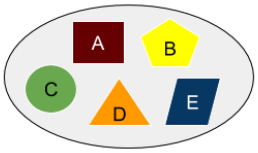

Figure 6.3: An organism with a propagule size of five.

- Finding metrics that work with multi-level selection *or* finding a way to track lineages in such an environment: This would mean enhancing the AMM to detect co-evolutionary relationships with lineages in a multi-level selection environment where migration is present. The other option would be to apply AMM as it is and to find a way to get meaningful lineage data out of a multi-level selection environment. The author faced the problem of not being able to track lineages as migration leads to a tree instead of a single lineage.
- Introducing antagonistic flavors: This could be achieved by building an environment where A- and B-cells can steal resources from each other, as [66] has already shown in a different model setup, to see how the metrics perform under such flavors. In the current model, an evolutionary arms race could *e.g.,* be introduced by favoring A-cells with zeros from the beginning of the genome and B-cells with ones from the end of the genome. This would result into both selective pressures performing well at earlier generations, but it would be interesting to see who outperforms whom as soon as they get in the way of each other. Other possibilities include A-cells selecting for leading ones and B-cells stealing fitness from A-cells with the number of bits that match between their genomes or modeling hosts (number of differences in the genomes) and parasites (number of similarities).

Apparently, there are lots of possibilities to carry this work forward. The findings discovered during this research project are first steps into a broad field of analyzing biological lineage-based genomic data to identify relationships between different species.

References

Literature

[1] Christoph Adami and Claus O. Wilke. "Experiments in Digital Evolution (Editors' Introduction to the Special Issue)". *Artificial Life* 10 (2004), pp. 117–122 (cit. on p. 11).

[2] Michael Affenzeller et al. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Chapman and Hall/CRC, New York, 2009 (cit. on p. 7).

[3] Wendy Aguilar et al. "The past, present, and future of artificial life". *Frontiers in Robotics and AI* 1.8 (2014) (cit. on p. 10).

[4] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996 (cit. on pp. 7, 34).

[5] Wolfgang Banzhaf and Barry McMullin. "Handbook of Natural Computing". In: Springer, 2012. Chap. 53 Artificial Life, pp. 1805–1835 (cit. on p. 9).

[6] Mark A. Bedau. "Artificial life: organization, adaptation and complexity from the bottom up". *TRENDS in Cognitive Sciences* 7.11 (2003), pp. 505–512 (cit. on pp. 8, 11).

[7] Mark A. Bedau. "Philosophy of Biology". In: North Holland, 2007. Chap. Artificial Life, pp. 585–603 (cit. on p. 10).

[8] Mark A. Bedau et al. "Open Problems in Artificial Life". *Artificial Life* 6 (2000), pp. 363–376 (cit. on p. 10).

[9] Hans-Georg Beyer and Hans-Paul Schwefel. "Evolution strategies – A comprehensive introduction". *Natural Computing* 1 (2002), pp. 3–52 (cit. on p. 6).

[10] Clifford Bohm, Nitash C G, and Arend Hintze. "MABE (Modular Agent Based Evolver): A Framework for Digital Evolution Research". In: *Artificial Life Conference Proceedings 14*. MIT Press, 2017, pp. 76–83 (cit. on pp. 34, 35).

[11] Clifford Bohm et al. "MABE 2.0 - and introduction to MABE and a road map for the future of MABE development". In: *GECCO '19: Proceedings of the Genetic and Evolutionary Computation Conference*. Association for Computing Machinery (ACM), New York, 2019, pp. 1349–1356 (cit. on pp. 34, 35).

[12] Reinhard Bürger and Josef Hofbauer. "Mutation load and mutation-selection-balance in quantitative genetic traits". *Journal of Mathematical Biology* 32 (1994), pp. 193–218 (cit. on p. 96).

[13] Brian Charlesworth. “Mutation-selection balance and the evolutionary advantage of sex and recombination”. *Genetical Research* 55.3 (1990), pp. 199–221 (cit. on p. 96).

[14] Carlos A. Coello, Gary B. Lamont, and David A. Van Veldhuizen. “Evolutionary Algorithms for Solving Multi-Objective Problems”. In: Springer, 2007. Chap. EA Basics, pp. 24–29 (cit. on p. 7).

[15] Cecilia Di Chio et al. *Applications of Evolutionary Computation: EvoApplications2012*. Springer, 2012 (cit. on p. 8).

[16] Andy Dobson et al. “In the Light of Evolution: Volume II: Biodiversity and Extinction.” In: National Academies Press, 2008. Chap. 4 Homage to Linnaeus: How Many Parasites? How Many Hosts? (Cit. on p. 14).

[17] Emily Dolson et al. “Interpreting the Tape of Life: Ancestry-Based Analyses Provide Insights and Intuition about Evolutionary Dynamics”. *Artificial Life* 26 (2020), pp. 58–79 (cit. on p. 27).

[18] Alan Dorin. *Biological Bits: A brief guide to the ideas and artefacts of computational artificial life*. Animaland, 2014 (cit. on pp. 9, 10).

[19] Angela E. Douglas. “Symbiosis as a General Principle in Eukaryotic Evolution”. *Cold Spring Harbor Perspectives in Biology* 6.2 (2014) (cit. on pp. 14, 15).

[20] Agoston E. Eiben and James E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003 (cit. on p. 6).

[21] Emiley A. Eloe-Fadrosh and David A. Rasko. “The Human Microbiome: From Symbiosis to Pathogenesis”. *Annual Review of Medicine* 64.1 (2013), pp. 145–163 (cit. on p. 15).

[22] Andy Gardner. “The genetical theory of multilevel selection”. *Journal of Evolutionary Biology* 28 (2015), pp. 305–319 (cit. on p. 16).

[23] Martin Gardner. “The Fantastic Combinations of John Conway’s New Solitaire Game Life”. *Scientific American* 223.4 (1970), pp. 120–123 (cit. on p. 9).

[24] Ricardo Guerrero, Lynn Margulis, and Mercedes Berlanga. “Symbiogenesis: the holobiont as a unit of evolution”. *International Microbiology* 16 (2013), pp. 133–143 (cit. on p. 15).

[25] John H. Holland. *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, 1992 (cit. on p. 6).

[26] Daniel H. Janzen. “When is it Coevolution?” *Evolution* 34.3 (1980), pp. 611–612 (cit. on p. 14).

[27] Kyung-Joong Kim and Sung-Bae Cho. “A Comprehensive Overview of the Applications of Artificial Life”. *Artificial Life* 12 (2006), pp. 153–182 (cit. on p. 10).

[28] Maciej Komosinski and Andrew Adamatzky, eds. *Artificial Life Models in Software*. Springer, 2009 (cit. on p. 11).

[29] John R. Koza. “Genetic Programming. On the Programming of Computers by Means of Natural Selection”. In: Bradford Books, 1992. Chap. 5 Overview of Genetic Programming, pp. 73–78 (cit. on p. 6).

[30] Jos Kramer and Joel Meunier. “Kin and multilevel selection in social evolution: a never-ending controversy?” *F1000Research* 5.F1000 Faculty Rev-776 (2016) (cit. on p. 16).

[31] Christopher G. Langton. “Artificial Life”. In: *Proceedings Of An Interdisciplinary Workshop On The Synthesis And Simulation Of Living Systems*. Wetview Press, 1989, pp. 1–47 (cit. on p. 9).

[32] Christopher G. Langton. “Artificial Life” (1992) (cit. on pp. 8, 9).

[33] Christopher G. Langton. “Studying artificial life with cellular automata”. *Physica D: Nonlinear Phenomena* 22 (1986), pp. 120–149 (cit. on p. 9).

[34] Egbert G. Leigh Jr. “How does selection reconcile individual advantage with the good of the group?” *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* 74.10 (1977), pp. 4542–4546 (cit. on pp. 4, 16).

[35] Purificacion Lopez-Garcia, Laura Eme, and David Moreira. “Symbiosis in eukaryotic evolution”. *Journal of Theoretical Biology* 434 (2017), pp. 20–33 (cit. on pp. 13, 15).

[36] Richard E. Michod. “Evolution of individuality during the transition from unicellular to multicellular life”. *National Academy of Sciences* 104 (2007), pp. 8613–8618 (cit. on p. 12).

[37] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1998 (cit. on pp. 2, 7, 8).

[38] Melanie Mitchell and Stephanie Forrest. “Genetic Algorithms and Artificial Life”. *Artificial Life* 1 (1994), pp. 267–289 (cit. on p. 11).

[39] Kimura Motoo. “A stochastic model concerning the maintenance of genetic variability in quantitative characters”. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* 54.3 (1965), pp. 731–736 (cit. on p. 96).

[40] E.G. Nisbet and Fowler C. M. R. “Archaean metabolic evolution of microbial mats”. *Proceedings of the Royal Society B: Biological Sciences* 266.1436 (1999), pp. 2375–2382 (cit. on p. 11).

[41] Bill O’Neill. “Digital Evolution”. *PLoS Biology* 1.1 (2033), pp. 11–14 (cit. on p. 11).

[42] Charles Ofria and Claus O. Wilke. “Avida: A Software Platform for Research in Computational Evolutionary Biology”. *Artificial Life* 10 (2004), pp. 191–229 (cit. on p. 11).

[43] Victoria J. Orphan. “Methods for unveiling cryptic microbial partnerships in nature”. *Current Opinion in Microbiology* 12.3 (2009), pp. 231–237 (cit. on p. 15).

[44] Elizabeth A. Ostrowski et al. “Genomic Signatures of Cooperation and Conflict in the Social Amoeba”. *Current Biology* 25 (2015), pp. 1661–1665 (cit. on p. 4).

[45] Ben K.D. Pearce et al. “Constraining the Time Interval for the Origin of Life on Earth”. *Astrobiology* 18.3 (2018), pp. 343–364 (cit. on p. 11).

[46] Joel R. Peck. “Group selection, individual selection, and the evolution of genetic drift”. *Journal of Theoretical Biology* 159.2 (1992), pp. 163–187 (cit. on p. 4).

[47] Nicola Plowes. “An Introduction to Eusociality”. *Nature Education Knowledge* 3.10 (2010), p. 7 (cit. on p. 12).

[48] David C. Queller. “Cooperators Since Life Began”. *The Quarterly Review of Biology* 72.2 (1997), pp. 184–188 (cit. on p. 12).

[49] David C. Queller and Joan E. Strassmann. “Beyond society: the evolution of organismality”. *Philosophical Transactions of the Royal Society B* 364 (2009), pp. 3143–3155 (cit. on p. 14).

[50] Thomas S Ray. “An Evolutionary Approach to Synthetic Biology: Zen and the Art of Creating Life”. *Artificial Life* 1 (1994), pp. 179–209 (cit. on p. 11).

[51] Thomas S Ray. *Evolution, Ecology and Optimization of Digital Organisms*. Tech. rep. Technical Report 92-08-042, Santa Fe Institute, Santa Fe, NM, 1992 (cit. on p. 11).

[52] Inaki Ruiz-Trillo and Aurora M. Nedelcu, eds. *Evolutionary Transitions to Multicellular Life. Principles and mechanisms*. Springer, 2015 (cit. on p. 12).

[53] Lynn Sagan. “On the Origin of Mitosing Cells”. *Journal of Theoretical Biology* 14.3 (1967), pp. 225–274 (cit. on p. 13).

[54] Charles Severance and Kevin Dowd. *High Performance Computing*. Connexions, Rice University Houston, Texas, 2012 (cit. on p. 37).

[55] John Maynard Smith. “Byte-sized evolution”. *Nature* 355 (1992), pp. 772–773 (cit. on pp. 8, 15).

[56] John Maynard Smith and Eors Szathmary. *The Major Transitions in Evolution*. Oxford University Press, New York, 1995 (cit. on pp. 11, 12).

[57] Kenneth O. Stanley and Risto Miikkulainen. “Evolving Neural Networks through Augmenting Topologies”. *Evolutionary Computation* 10.2 (2002), pp. 99–127 (cit. on p. 10).

[58] Sigvard Strandh. *The history of the machine*. Dorset Press, 1979 (cit. on p. 9).

[59] Armin C. Stross-Radschinski. *python - a programming language changes the world*. Brochure. URL: https://brochure.getpython.info/ (visited on 07/14/2020) (cit. on p. 37).

[60] Eors Szathmary. “Toward major evolutionary transitions theory 2.0”. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* 112.33 (2015), pp. 10104–10111 (cit. on pp. 12, 14).

[61] Eors Szathmary and John Maynard Smith. “The major evolutionary transitions”. *Nature* 374 (1995), pp. 227–232 (cit. on pp. 11–13).

[62] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2017. URL: https://www.R-project.org/ (cit. on pp. 34, 37).

[63] John H. Thompson. *Interaction and Coevolution*. University of Chicago Press, 1982 (cit. on p. 14).

[64] Elizabeth Thursby and Nathalie Juge. "Introduction to the human gut microbiota". *Biochemical Journal* 474 (2017), pp. 1823–1836 (cit. on p. 15).

[65] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009 (cit. on pp. 34, 37).

[66] Anya E. Vostinar and Charles Ofria. "Spatial Structure Can Decrease Symbiotic Cooperation". *Artificial Life* 24.4 (2018), pp. 229–249 (cit. on pp. 30, 108).

[67] Stuart A. West et al. "Major evolutionary transitions in individuality". *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* 112.33 (2015), pp. 10112–10119 (cit. on p. 12).

[68] Stuart West, Ashleigh Griffin, and Andy Gardner. "Social Semantics: Altruism, Cooperation, Mutualism, Strong Reciprocity and Group Selection". *Journal of Evolutionary Biology* 20 (2007), pp. 415–432 (cit. on p. 14).

[69] Darrell Whitley. "A genetic algorithm tutorial". *Statistics and Computing* 4 (1994), pp. 65–85 (cit. on p. 8).

[70] Claus O. Wilke and Christoph Adami. "The biology of digital organisms". *TRENDS in Ecology & Evolution* 17.11 (2002), pp. 528–532 (cit. on p. 11).

[71] Alex C. C. Wilson and Rebecca P. Duncan. "Signatures of host/symbiont genome coevolution in insect nutritional endosymbioses". *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* 112.33 (2015), pp. 10255–10261 (cit. on p. 4).

[72] Larry Yaeger. "Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or Polyworld: Life in a New Context". In: *Proceedings of the Artificial Life III Conference*. Addison-Wesley, Reading, MA, 1994, pp. 263–298 (cit. on p. 11).

Software

[73] Nadine Strasser. *nstrasser/MasterThesisProject: Supplemental Thesis Material Release*. Version 1.0. 2020. URL: https://doi.org/10.5281/zenodo.4009396 (cit. on pp. 5, 34, 54, 57, 65, 66, 81, 83, 93).

Online Sources

[74] The International Society for Artificial Life. *About ISAL*. URL: https://alife.org/about-isal/ (visited on 07/28/2020) (cit. on p. 10).

[75] Clifford Bohm and Jory Schossau. *MABE Project - Wiki*. URL: https://github.com/Hintzelab/MABE/wiki (visited on 07/12/2020) (cit. on pp. 34–36, 46).

[76] University of California Museum of Paleontology. *It Takes Teamwork: How Endosymbiosis Changed Life on Earth*. URL: https://evolution.berkeley.edu/evolibrary/article/_0_0/endosymbiosis_01 (visited on 08/04/2020) (cit. on p. 13).

[77] University of California Museum of Paleontology. *Understanding Evolution, Glossary*. URL: https://evolution.berkeley.edu/evolibrary/glossary/glossary.php (visited on 07/24/2020) (cit. on pp. 4, 15, 16).

[78] MSU – BEACON Center. *BEACON - An NSF Center for the Study of Evolution in Action*. URL: https://beacon-center.org/ (visited on 06/28/2020) (cit. on pp. 4, 5).

[79] MSU – BEACON Center. *BEACON Mission*. URL: https://www3.beacon-center.org/welcome/beacon-mission/ (visited on 06/28/2020) (cit. on p. 4).

[80] The Editors of Encyclopaedia Britannica. *Symbiosis*. URL: https://www.britannica.com/science/symbiosis (visited on 08/04/2020) (cit. on p. 14).

[81] Sylvia Freeman. *Symbiosis*. URL: https://www.expii.com/t/symbiosis-definition-types-10297 (visited on 07/28/2020) (cit. on p. 14).

[82] Steve Grand. *Artificial life*. URL: https://www.britannica.com/technology/artificial-life (visited on 08/05/2020) (cit. on pp. 8, 9).

[83] MSU – ICER. *High Performance Computing at ICER*. URL: https://wiki.hpcc.msu.edu/display/ITH/High+Performance+Computing+at+ICER (visited on 07/05/2020) (cit. on p. 38).

[84] OpenAI Inc. *OpenAI*. URL: https://openai.com/about/ (visited on 08/05/2020) (cit. on p. 10).

[85] Digital Evolution Laboratory. *DevoLab - Digital Evolution Lab at Michigan State University*. URL: https://devolab.org/ (visited on 06/28/2020) (cit. on p. 5).

[86] Charles Ofria. *Charles Ofria, PhD - Digital Evolution*. URL: https://ofria.com/ (visited on 06/28/2020) (cit. on p. 5).

[87] Steven Pinker et al. *The False Allure of Group Selection. An EDGE Original Essay (incl. Reality Club Discussion)*. URL: http://edge.org/conversation/the-false-allure-of-group-selection (visited on 08/06/2020) (cit. on p. 16).

[88] Richard J. Roberts. *Nucleic acid*. URL: https://www.britannica.com/science/nucleic-acid (visited on 07/04/2020) (cit. on p. 15).

[89] Keiichiro Shibuya. *Scary Beauty*. URL: https://scarybeauty.com/ (visited on 08/16/2020) (cit. on p. 10).

[90] Lana Sinapayen. *Introduction to Artificial Life for People who Like AI*. URL: https://thegradient.pub/an-introduction-to-artificial-life-for-people-who-like-ai/ (visited on 08/02/2020) (cit. on pp. 8, 10).

[91] Kenneth O. Stanley, Joel Lehman, and Lisa Soros. *Open-endedness: The last grand challenge you've never heard of*. URL: https://www.oreilly.com/radar/open-endedness-the-last-grand-challenge-youve-never-heard-of/ (visited on 08/02/2020) (cit. on p. 10).

[92] Eric W. Weisstein. *Cellular Automaton*. URL: https://mathworld.wolfram.com/CellularAutomaton.html (visited on 08/05/2020) (cit. on p. 9).