

A Computational Model of Developmental Exaptations

Alexa Renner¹, Émile Marois¹, Julian Fiorito¹, Jacob Ashworth¹, and Jason A. Yoder¹

¹Rose-Hulman Institute of Technology, USA
renneram@rose-hulman.edu

Abstract

We present a novel computational model aimed at exploring the concept of developmental exaptations, a previously hypothesized phenomenon wherein existing aspects of a developmental process are reused for a new, beneficial function. We leverage computational simulations to study this due to the challenges of observing developmental exaptations directly in nature. Employing a special fitness landscape, our model simulates the evolution of developmental strategies, allowing the study of how certain developmental instructions, which were initially evolved for a different function (or no function), can be repurposed to enhance fitness. The model utilizes indirect encoding to mimic biological processes, facilitating the evolution of exaptations. We demonstrate how crucial mutations contribute to achieving global fitness maxima, indicative of developmental exaptations for our special landscape. Our results not only provide computational evidence for the plausibility of developmental exaptations, but also open avenues for further research into detecting and understanding these phenomena in more complex and dynamic environments. This work establishes a new approach for modeling and analyzing developmental exaptations.

Introduction

The process of development is both remarkably adaptive and complex, growing organisms from single cells to adult forms. The study of the evolution of development (evo-devo) has revealed amazing discoveries, such as how *Hox* genes have been repeatedly utilized for different aspects of body plans across a wide range of species (Holland, 2013). During development, organisms must adapt as they grow but also may adjust how they behave and thus how they learn depending on their environment and stage of development. We consider the refinement or regulation of learning mechanisms to be one particularly interesting aspect of development to study. One known example of this is critical periods of development, in which organisms may become more or less sensitive to their environment, which in turn has a significant impact on how learning occurs (Knudsen, 2004). Because learning and development are intertwined processes, models that distinguish them may help to identify interesting interactions between them. In our model, we

utilize evolution to discover effective developmental strategies which regulate learning. We then strive to study specific phenomena during the evolution of these developmental strategies that might be difficult to study otherwise.

An especially interesting beneficial trait is an exaptation. Exaptations are traits that were “evolved for other usages (or for no function at all), and later ‘co opted’ for their current role” (Gould and Vrba, 1982). The genes encoding exaptations already exist and upon being utilized for a new, beneficial function may be selected and refined through subsequent mutations. An example of an exaptation is the antifreeze proteins in Antarctic notothenoid fish blood (Chen et al., 1997). These proteins are encoded by genes that arose from parts of the genes that produced a pancreatic compound being repurposed (Chen et al., 1997). Despite examples of exaptations in nature, the phenomenon is difficult to study and even harder to predict, not only due to evolution’s often slow pace for multicellular organisms but also because of the complexity of organisms’ biology.

A related concept within the field of evolutionary development, and the focus of our investigation in this paper, is the notion of a *developmental exaptation*. Developmental exaptations occur when a pre-existing aspect of development helps facilitate evolutionary changes in development (Chipman, 2021). Developmental exaptations are especially difficult to study, not only due to evolution’s long timescale but also the necessity to observe an organism’s development. A developmental exaptation could result when an evolutionarily malleable developmental program can use an existing characteristic, such as a “master” gene (e.g. *Hox*) which adaptively builds a leg when expressed. In such a case, if the gene was expressed at a different stage, it could be exapted to create a new structure (e.g. arm).

In this work, we present our efforts in modeling this difficult-to-study phenomenon, and our strategy for the detection of developmental exaptations in simulation.

Related Work

Computational modeling has often been used to study processes that are too difficult to observe directly. Interactions

between complex processes such as learning and evolution have been explored in past works (Todd et al., 2020; Hinton and Nowlan, 1987). Recently, there has been an increased interest in studying the evolution of development or evo-devo (Brakefield, 2011), which faces similar challenges for modeling given the scales of time and space involved for individuals and populations. While some species-specific models exist (Roth and Hartenstein, 2008), there are few published generalized models of development, although progress has been made recently in simulating the evolution of developmental learning strategies (Ashworth et al., 2022, 2023). We leverage this recent progress to model a hypothesized but previously unobserved phenomenon, developmental exaptations. In order to understand development (and thus developmental exaptations), one must also understand how it differs from learning. Learning allows an individual to improve over its lifespan using information gathered from past experiences, whereas development changes the methods the individual learns by, and changes the bodily structure to fit the individual's stage of life (Ashworth et al., 2023).

Chipman (2021) defines a developmental exaptation as “the process through which pre-existing characteristics of the developmental program facilitate evolutionary changes in development, by providing an internal environment that is conducive to a certain change.” In this context, we consider the *developmental program* as the sequence of developmental changes an individual undergoes in its lifetime. Developmental exaptations could happen in most developmental processes, from low-level processes such as gene regulation to higher-level processes, such as “relative timing of developmental events” (Chipman, 2021).

Graham and Oppacher (2007) developed a computational model of a (non-developmental) exaptation by using a genetic algorithm with four different “niches” with populations of the same size in each niche. Each niche had a distinct fitness function, but shared the same genetic algorithm parameters. An individual migrating to a different niche constituted an exaptation event in this model. They found that out of six possible migration routes between the niches, three were successfully followed, thus demonstrating a model of evolutionary exaptations (Graham and Oppacher, 2007). While this model does not attempt to capture any aspect of development, it provides a precedent for simulated exaptations on specialty fitness landscapes.

Recently, computational models have been developed as an abstraction of the evolution of development (Ashworth et al., 2022, 2023). The model used in Ashworth et al. (2022) focuses on evolving developmental programs. In the model, a population is initialized with individuals (agents) on an NK fitness landscape with developmental programs (a set of steps encoding how the agent moves on the landscape) with a given lifespan, which defines the length of the agent's developmental period. The starting positions of the agents do not change; instead, the developmental programs which dic-

tate how the agents traverse the landscape are evolved. An agent can take two different types of actions on the landscape— the “look” action, which allows it to collect information about the fitness of adjacent locations, or the “walk” action, which moves it to a different location and a different fitness. Individuals with evolved developmental strategies were able to escape local optima on more complex NK landscapes (Ashworth et al., 2022).

Building on Ashworth et al. (2022), Ashworth et al. (2023) explored the effects of indirect encoding of the genotype of an individual on the evolution of developmental strategies. In their model, each individual was assigned a developmental program, which is a set of instructions that would move the individual on the fitness landscape. The developmental programs were sequences of blocks, and blocks were sequences of steps. In other words, the *developmental program* was indirectly encoded because each block could be expressed multiple times. Interestingly, the authors noted that “...higher program mutation rates are preferable to higher block mutation rates...”, and that this “...suggests ideas related to ‘teaching old genes new tricks.’” (Ashworth et al., 2023). This finding is promising for studying exaptations, as they involve repurposing an old trait for a new purpose. We build upon this model in our work.

This notion of altering developmental programming leads naturally to questioning the variability of evolutionary outcomes. One natural question to ask is, “would evolution take the same course if restarted from some point?” The answer to this question could be useful in determining the factors that affect the probability of traits developing through evolution (Ferguson and Ofria, 2023). Ferguson and Ofria (2023) investigated how the probability of a trait evolving changes throughout the lineage of a successful individual. To do this, they used a simulation in which the simulated organism evolved associative learning. They then re-ran the simulation from certain important points— when a new mutation was introduced, for example— and looked at the proportion of individuals in the population that evolved associative learning in those re-runs. They found that at certain points, the percentage of the re-runs that evolved associative learning, called potentiation, increased dramatically. This shows that potentiation can grant useful insight into the mutations necessary for certain events to occur or traits to evolve (Ferguson and Ofria, 2023). We use the idea of potentiation in our model to study the conditions that increase the probability of a *developmental exaptation*.

Incorporating ideas from potentiation (Ferguson and Ofria, 2023), evolving developmental programs (Ashworth et al., 2022, 2023), modeling exaptations (Graham and Oppacher, 2007), and the theory of developmental exaptations (Chipman, 2021), we have built a computational model to study developmental exaptations. Our model provides computational evidence for plausibility and enables further investigation through simulation experiments and analysis.

Model

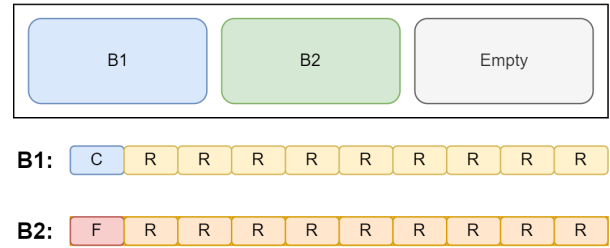
We aim to model developmental exaptations to establish computational plausibility and study potential underlying mechanisms. To do this, we devised a special fitness landscape to make evolving and studying developmental exaptations straight forward. In our model, evolution defines the *developmental strategy*, which dictates how the individual changes position— whether it moves to higher or lower fitnesses, and when those moves occur. This is best analogized to the developmental regulation of learning. The *developmental strategy* dictates how the agent moves, or “learns” a phenotype, akin to reinforcement learning.

Fitness Landscape In our model, a fitness landscape is a set of nodes connected to each other by edges in an undirected graph (see Figure 2). We say that a node is *adjacent* to another node if those two nodes share an edge. Each node has an associated fitness value. By varying the topology of the graph or by modifying its associated fitness values, we can regulate the difficulty of the fitness landscape. An agent can navigate through a landscape using a series of *steps*. A *step* is an instruction for an individual on a fitness landscape to move to an adjacent node or remain in its current location.

Steps Agents use steps to determine how they should move to navigate a landscape. In our model, we utilize three steps: *climbing steps*, *falling steps*, and *repeating steps*. The *climbing step* moves the agent to the adjacent position with the highest fitness, or maintains its location if at a local maxima. In a biological context, this would represent an organism gathering information from the environment and then making the most beneficial small change in behavior. The *falling step* moves the agent to the adjacent position with the lowest fitness, or maintains its location, if at a local minima. This is analogous to an organism choosing a potentially risky behavior to explore the space and escape local optima. The *repeated step* repeats the step that was executed immediately previously, if such a step exists— if it does not, then it executes a *climbing step*. For example, if there was a *falling step* at position $n - 1$, and a *repeating step* at position n , the program would execute *falling steps* at both positions. This *repeated step* models a developmental process that is dependent on the other processes occurring before it.

Developmental Strategy In our simulation, each individual’s genome defines its *developmental strategy*, which is composed of a *developmental program* and a *block bank* (see Figure 1). This *developmental strategy* is what determines the agent’s actions and hence navigation of the landscape from a given starting point through its lifetime. A *block* is a sequence of some predefined number of steps. A *block bank* stores all the blocks that could be used in an agent’s *developmental program* and has a defined maximum number of blocks it can contain (see Figure 1). At the start of the sim-

Example Block Bank:



Example Developmental Program:



Figure 1: An illustration of an example *block bank* and *developmental program* for an agent. Each block is composed of a sequence of steps: *falling* (F), *climbing* (C), or *repeating* (R). Not all blocks must be used in the program and the *block bank* need not be full. Unused blocks can be thought of as analogous to non-coding genes. The *block bank* and *developmental program* are evolved as the agent’s genome.

ulation, some predefined number of *blocks* are instantiated and can be mutated, copied, and used in the *developmental program*. The *developmental program* uses blocks from the *block bank*. The *developmental program* need not use all of the blocks in the *block bank*, and can use the same block multiple times, thus creating a type of indirect encoding in which one genetic component (a block) may encode multiple characteristics (developmental steps at different stages of life). When the *developmental program* is executed, each block (with each step) is executed in sequence.

The indirect encoding of the *developmental strategy*, as originally defined by Ashworth et al. (2023) works well in modeling developmental exaptations for several reasons. For one, it makes it simpler for parts of the genome (*blocks*) that are initially used for one function to be repurposed. Indirect encoding is also more aligned with biology in that genes encode instructions for proteins which are then used downstream for a variety of purposes and numerous phenotypic factors (Chipman, 2021). For our purposes, it also provides a clear indication of when a feature is repurposed, which is helpful for identifying an exaptation.

Agents In our model, each agent has a phenotype and an evolved *developmental strategy* which defines its genome. An agent is mutated if a block in the agent’s *block bank* is mutated, the agent’s *developmental program* is mutated, or if a block in the *block bank* is copied. A block in the *block bank* is mutated if a step in that block is replaced with a different step. A program mutation replaces an existing block of the *developmental program* for some agent by a different

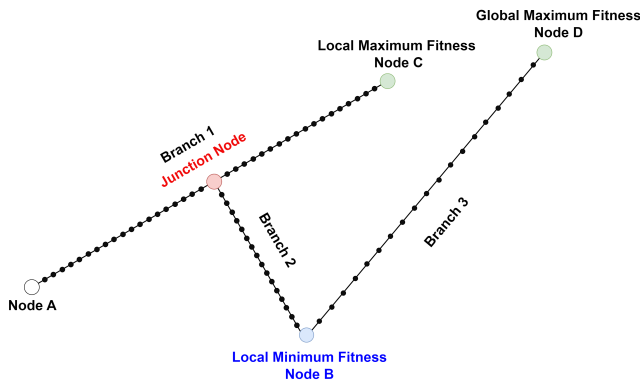


Figure 2: Specially-designed fitness landscape. Higher positions roughly indicate higher fitness values of nodes. Branch 1 includes Node C, but not Node A. Branch 2 include the Junction Node, but not Node B. Branch 3 includes Node D, but not Node B. All agents start at Node A. To reach the highest fitness in the landscape, agents must first climb up Branch 1 to the Junction node, fall down Branch 2 to Node B, and then climb up Branch 3 to Node D. We ensured that the difference between the fitnesses at two adjacent nodes on Branch 2 was greater than the difference between two adjacent nodes on Branch 1. This way, if an agent at the Junction node executes a *falling step*, the step will put the agent on Branch 2. While other mechanisms could allow finding new paths on other landscapes, the descending navigation is necessary due to how our landscape is designed. We denote the fitness values of Nodes A, B, C, D, and Junction as a, b, c, d and j respectively; the number of nodes on Branches 1, 2, and 3 as $x, y,$ and z respectively; and the position of the Junction Node on Branch 1 as m . Fitness for node i (ascending) on Branch 1 is $a + \frac{i}{x} \cdot (c - a)$. Fitness for the Junction Node is $j = a + \frac{m}{x} \cdot (c - a)$. Fitness for node i (ascending) on Branch 2 is $b + \frac{i}{y} \cdot (j - b)$. Fitness for node i (ascending) on Branch 3 is $b + \frac{i}{z} \cdot (d - b)$.

block from the *block bank* for that same agent. A phenotype is simply the node that an agent is currently located at and may change over its “lifetime.” For a given experiment, the *developmental program* is a fixed number of blocks, which in turn each contain their own steps (e.g. 6 blocks of 10 steps).

Population We create a population of agents to navigate a given fitness landscape. Each agent starts out with a *developmental strategy*. One type of mutation, *block copying mutations*, can only occur until all positions of the *block bank* are occupied. To evaluate agents, their *developmental programs* are executed, and the agent’s fitnesses are calculated based on their final positions in the fitness landscape. The agents with the highest fitness are cloned in the next generation (elitism). The remainder of the population is deter-

mined using truncation selection where agents in the top half are copied and stochastically mutated.

Experimental Setup In order to study *developmental exaptations* with this model, we decided to tailor the starting populations and fitness landscape to give a strong chance for an exaptation to occur. Our special fitness landscape designed for this purpose is shown in Figure 2. In the sections that follow we explain how, given our setup, it becomes necessary for an exaptation to occur in order for an agent to reach the global maximum fitness.

Special Fitness Landscape The *developmental program* of the agent determines where it travels on our landscape (see Figure 2). For all of our results, the special fitness landscape was created with 39 nodes on Branch 1, 16 nodes on Branch 2, and 20 nodes on Branch 3. On Branch 1, there were 19 nodes before the Junction node and 19 additional nodes after the Junction node (including Node C). Node A’s fitness is always 0, Node B’s fitness was set to 1, Node C’s fitness was set to a local maxima of 19, and Node D’s fitness was set to a global maximum of 40.

Special Population Initialization 50 agents were instantiated with a *block bank* of maximum capacity 3 and a single block consisting of one *climbing step* followed by nine *repeating steps*. The block was specifically designed to have a *climbing step* followed by many *repeating steps* to make it more likely for a single *functional block mutation* (which changes the function of the block) to cause a cascade of descending steps. Such a block could be re-introduced to the *developmental program* to allow the agent to reach node B, before ascending to the global maximum fitness.

The sum of the lengths of each block used in the *developmental program* was set to be at least the total number of edges the agent must travel to reach the global maximum fitness. We began experimentation with our special landscape using a program of size 6 so that the agent could travel over the 57 edges between node A and node D, and a runtime of 100 generations. At least two different blocks are needed for the agent to reach node D (a block with primarily *falling steps* and a block with primarily *climbing steps*), so the agent would need to copy a block to reach the highest fitness. There was a 10% chance of each block in the block bank being copied, and a 1% chance of a block being mutated so that block mutations were fairly rare but possible. The simulation used an elitism value of 10%, and a program mutation (where one block in the *developmental program* is replaced by another) chance of 15% .

Reaching Global Maxima We say an agent ascends or descends when it moves to a position with higher or lower fitness, respectively. To reach the global maximum, Node D on this landscape, an agent must ascend until it reaches or has passed the Junction node, descend until it reaches Node B,

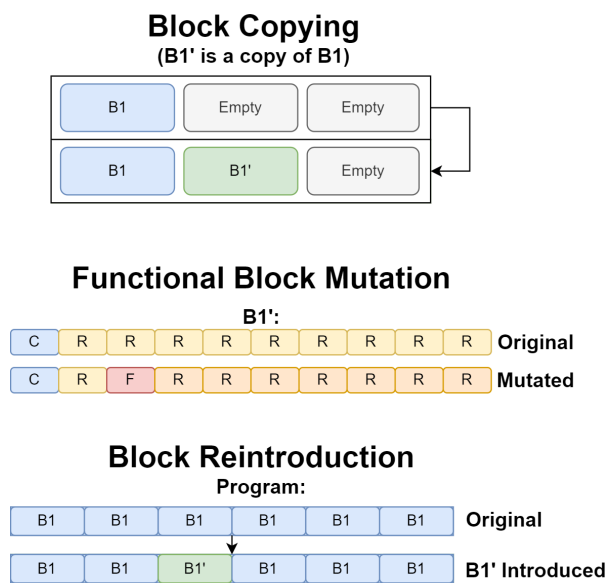


Figure 3: A diagram of the crucial mutation events necessary for an exaptation to occur, and in the case of our special landscape, reaching the global fitness.

and ascend to Node D (see Figure 2). To ascend or descend, it must use blocks with some number of *climbing steps* or *falling steps* respectively, which could be followed by *repeating steps*. We define three specific mutations that must happen for Node D to be reached on our model, and we call these *crucial mutations*.

Crucial Mutations The first *crucial mutation* is a *block copy mutation*, which is when an existing block is directly copied into an empty space in the *block bank*. The second is a *functional block mutation*: a copied block (or the original) must mutate such that it would change the path traveled by the agent (i.e. the function of the block changes). The third is a *block introduction mutation*: The program must be mutated such that some block that was previously not in the strategy is introduced to it and executed (See Figure 3).

Necessity of Crucial Mutations The *crucial mutations* are necessary for reaching the global maximum in our model. To understand this, recall that in the initial population, each agent has a single block. The original block allowed the agent to climb consistently to reach Node C (see Figure 2). In order to reach the global maximum, an agent must have a somewhat consistent sequence of ascending, descending, and then ascending steps from node A to the junction node, then to node B, and finally to node D. This requires at least two different blocks being executed in a *developmental program*. For two different blocks to exist, the single block in the *block bank* must be copied through a *copy mutation*. One of the now-multiple blocks must undergo a

functional block mutation such that one of the blocks now has a cascade of falling steps, which could be achieved by a block mutation that results in a *falling step* being placed in it (ideally before a number of *repeating steps*). Finally, the mutated block must be re-introduced to the developmental program. For example, a block with a cascade of *falling steps* can send the agent to Node B so blocks with *climbing steps* can be used to reach Node D (see Figure 2).

Crucial Mutations and Developmental Exaptation

Reaching the global maximum guarantees all three *crucial mutations* have occurred; however, if this happens, we can also state that a developmental exaptation has occurred. To understand this, recall that the initial agents have a single block consisting of one *climbing step* followed by many repeating steps. That initial block corresponds to a preexisting trait in the *developmental program* in Chipman (2021)'s definition of a developmental exaptation. In particular, the many *repeating steps* defines a type of “behavioral stubbornness” regulating consistent behavior. This block allowed the agent to climb consistently to reach Node C (see Figure 2). The combination of the *block copy mutation*, *functional block mutation*, and *block introduction mutation* align with the notion of an “internal environment” that facilitates change. The *block copying mutation* allows different blocks to evolve from the same block structure, the *functional block mutation* results in a change in behavior of the subsequent *repeating steps* in the mutated block, and the *block introduction mutation* results in the *developmental program* being changed. This demonstrates repurposing of a preexisting feature. The population was initialized with a block which was functioned as a “climbing block.” Through the *crucial mutations*, this block can be repurposed to serve as a “falling block.” More specifically, the block’s feature of “behavioral stubbornness” in the developmental program is co-opted for consistent descent rather than ascent.

As such, the *developmental program* has facilitated changes in itself by providing an environment in which the *crucial mutations* are possible and can create these changes. Since these are each required to reach global fitness, we know that by our definition, reaching the global fitness thus requires an exaptation given the starting conditions of the population. In any case where the three *crucial mutations* occur, a block has been repurposed and re-introduced to the developmental program, so the order in which the *functional block mutations* and *block introduction mutations* occur is irrelevant. However, a *block copy mutation* must happen first in order for there to be a new block to introduce.

These *crucial mutations* are necessary (but not sufficient) for both reaching a global maximum and demonstrating an exaptation, given that each agent in the population is initialized with only one block. By recording when these mutations occur, we can attempt to gain insight into when developmental exaptations are likely to occur.

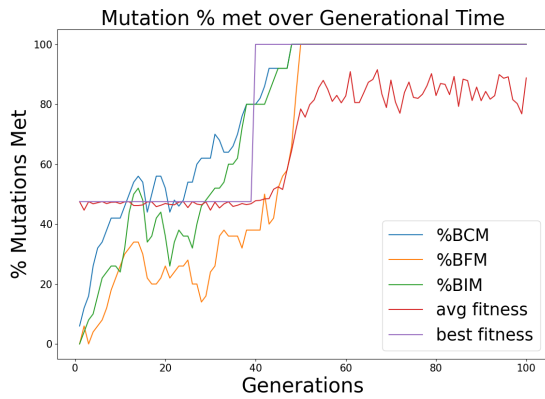


Figure 4: Population statistics of one evolutionary run. Fitness of a population’s average (avg) and best fitness, as well as the percentage of the population that has met the *crucial mutations* necessary for achieving global fitness. BCM = Block Copy Mutation, BFM = Block Function Mutation, BIM = Block Introduction Mutation. Potentiation is the percentage of simulations in which a global fitness is reached within 20 generations. (Seed: 329874)

Potentiation For reproducibility and analysis, we seed pseudo-random number generators for each experiment. We can measure the likelihood that a population will have at least one agent that reaches the maximum fitness after a set number of generations. We call this likelihood the potentiation of the population, which imitates the technique of “replaying” lineages (Ferguson and Ofria, 2023). To get the potentiation of any generation in our simulation, we “re-run” (using a different seed) the population from that generation for a predefined number of generations, repeating this for some number of re-runs, and record what percentage of those re-runs result in at least one agent reaching the maximum fitness. Because the maximum fitness in our model requires an exaptation to be reached, higher rates of potentiation may indicate a higher likelihood of an exaptation as well. As such, sudden increases in potentiation would indicate mutations beneficial to exaptation. However, exaptations can occur without an agent reaching the global maximum fitness, so a lack of increase in potentiation does not necessarily indicate that an exaptation has not occurred. Because our model is relatively computationally inexpensive, we are able to examine potentiation at every generation.

Lineage Analysis To record the lineage of an individual agent in a simulation, we choose the highest fitness agent at the end of the simulation and record its lineage. Importantly, our simulation has no crossover, so every child only has a single parent. This allows us to track a direct lineage from child to parent all the way to the 1st generation. For each agent in the lineage, we track their blocks, fitness, and

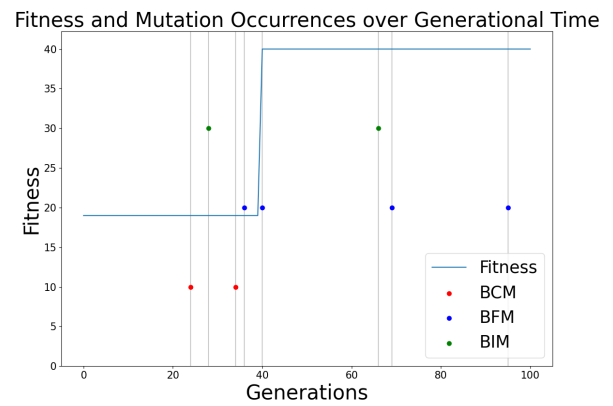


Figure 5: A graph showcasing the fitness of an agent and its ancestors (lineage) over generations. The dots represent occurrences of *crucial mutations* at certain generations, which are only set to specific y-values for visual clarity, and are not associated with any fitness values. (Seed: 329874)

strategy, and have methods to detect any *crucial mutations* that may occur. By reviewing this data, we can determine where an exaptation occurred.

Results

Running the experiment 100 times resulted in all runs having at least one agent reach the global maximum. This simply shows our setup made this result very likely. As a sanity check, we did verify that all three *crucial mutations* for an exaptation were met in each case, by programmatically reviewing the lineage of the successful individuals in each run and verifying that each of the *crucial mutations* had occurred. To better understand the mechanisms influencing the evolution of these populations, we will review an example evolutionary run. The code for conducting the experiments described in this paper can be found at https://github.com/rhit-easy-lab/2023-2024_EvoDevoResearch/tree/ALife_2024.

Figure 4 shows data collected from an example run, giving insight into the population’s statistics. As you can see, the average and best fitness plateau before a sudden spike in the best fitness. Plotting the independent percentages of the population that has experienced each of the three *crucial mutations* (but not necessarily more than one) shows that the frequency of those mutations in individuals goes up and down, but trends upward. Likewise, the potentiation goes up and down leading up to the fitness spike. Because of elitism, potentiation remains at 100% once a single individual reaches the global maximum. In this example experiment, the population was able to reach Node D, thus an exaptation must have occurred. By examining the lineages of simulations’ best agents, we observed that there were sharp changes in fitness hinting at beneficial mutations. By

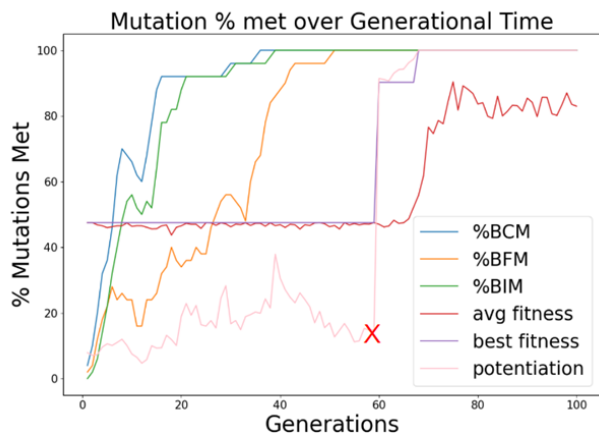


Figure 6: A graph showing the potentiation of a population, with rates of other factors, such as block copying. As we can see, when the best fitness increased– not to the highest possible fitness, but to a higher fitness– the potentiation (pink) increased drastically, thus indicating that an important beneficial mutation must have occurred. Notice that there is a downward trend in potentiation before the spike at generation 59, marked with an “X.” (Seed: 350)

tracking the occurrences of each *crucial mutation*, we found that our result confirmed that at least one occurrence of each *crucial mutation* could be found in an agent’s lineage before it reached the global maximum (Figure 5).

To explore runs when an exaptation might have occurred without reaching the global maximum, we sought an example of a potentiation spike that did not immediately result in reaching the best global maximum fitness (see Figure 6). We found an example run exhibiting this using potentiation. To calculate the potentiation, at each generation, that generation’s population was run for 20 additional generations. This potentiation run for 20 additional generations was repeated 1000 times. Because of elitism, a permanent increase in best fitness would be expected for any generation that had at least one agent that reached the maximum fitness. However, the fact that a spike in potentiation happened before reaching the maximum fitness (around generation 40) suggests that there could be a mutation leading to the eventual formation of a developmental exaptation (see Figure 6).

Potentiation steadily increased on average after the largest spike (at generation 60) until it reached 100%. This is indicative of an exaptation that did not achieve maximum fitness, as an agent must move onto Branch 3 to have a fitness higher than the local maximum. There is also a spike in potentiation shortly before generation 40 which decreases shortly after, indicating that some mutation which is beneficial to potentiation occurred shortly before generation 40 but was mutated out of the population shortly after.

To gain a greater understanding of the events that caused

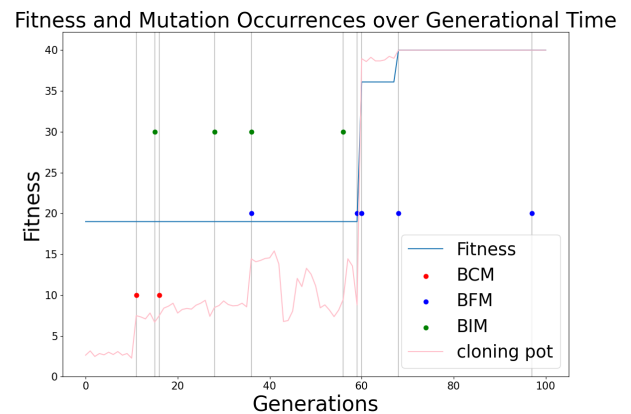


Figure 7: Analysis of a specific lineage of an agent that reached the global maximum fitness in the final generation. The pertinent events are shown to display the impact of lineage *cloning potentiation*. This indicates that if the individual of a particular generation were cloned into a population, the percentage of the time an individual from that population would reach a global maximum within 20 generations. (Seed: 350)

the increases and decreases in potentiation, we elected to plot the occurrence of various conditions being achieved for a particular individual’s lineage from the final generation (one that reached the global maximum). We utilized a technique we call *cloning potentiation* (see Figure 7) so that we could directly assess the impact of the individual’s mutation events on its potential evolutionary trajectory. We define *cloning potentiation* to be a potentiation technique wherein an agent is cloned to create a population consisting exclusively of clones of that specific agent, and *cloning potentiation* is calculated by using potentiation on that population of clones. We used *cloning potentiation* on each agent in the lineage of the individual. A spike in *cloning potentiation* in generations close to a spike in potentiation of the original population that originally occurred with all conditions being met prior to that spike could indicate that an exaptation did occur. Figure 7 shows how nearly every crucial mutation results in a positive increase in *cloning potentiation*.

To further investigate the lineage that produced the agent that reached the global maximal fitness, we created a visualization of the *developmental program* of an individual at three points through its evolution (see Figure 8). Recall that in the initial population, all individuals are initialized with a single block which is composed of a single climbing step followed by nine repeating steps. As can be seen in generation 0 (top) in Figure 8, this program leads the individual to climb for 40 steps before getting stuck at the local optima. By generation 60 (middle), we have seen the copying, mutation, and re-insertion of the block, and have achieved a new

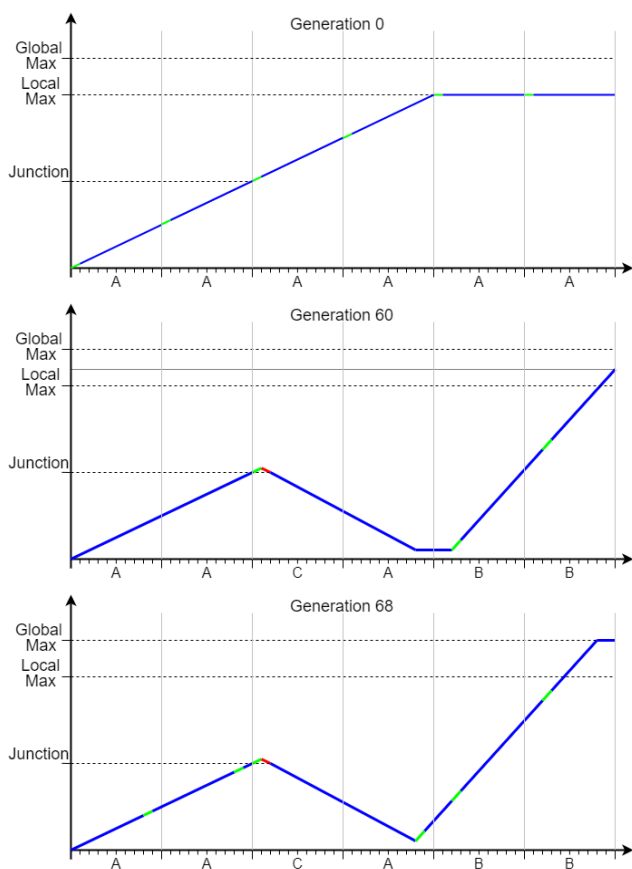


Figure 8: X-axis is time, Y-axis is fitness. Major points on the landscape such as the junction, local max (Node C), and global max (Node D) are labeled. Blocks are listed along the x-axis. The steps are color-coded on the line with *climbing step* being green, *falling step* being red, and *repeating step* being blue. Showing generations 0, 60, and 68. (Seed: 350)

fitness indicative of a successful exaptation. This program does not however reach maximal fitness, as the first *climbing step* does not occur until two steps too late to reach global maximum fitness. Finally, in generation 68 (bottom) a mutation occurs in block A to insert a *climbing step* early enough to reach maximal fitness but late enough to still reach the local minimum and find Branch 3 (see Figure 2). Not only does this help explain why the global maximum was not immediately found but it also provides a useful insight: potentiation based only on finding the global maxima may not be able to provide as much insight as previously thought.

We designed experiments to easily detect exaptations, but have conducted additional tests with increased initial population randomization. When randomizing the initial block (but still instantiating agents with only one block), we still did see agents reach the global maximum, again necessitating an exaptation event, but it did take significantly longer. The model can also be used to allow multiple ran-

dom blocks, but doing so could allow for the global maximum to be achieved without an exaptation occurring. Thus, more extensive analysis will be required.

Conclusion

While this work provides evidence for the computational plausibility of developmental exaptations, there is room to improve the applicability of our methods and the biological feasibility of our model. Our model was designed to strongly reward developmental exaptations, but it is unclear how much of a fitness benefit they might typically afford in biology. A range of reward pressures should be tested in future models to examine the effect on the frequency of developmental exaptations.

One avenue for future work is to use exaptation conditions and potentiation shifts to detect an exaptation on a more complex landscape, such as NK fitness landscapes (Kauffman and Levin, 1987). Identifying and studying developmental exaptations on NK landscape models would lend biological feasibility to our findings. In addition to using NK models, incorporating dynamic fitness landscapes that change over either developmental or evolutionary timescales could provide greater insight into the environmental conditions likely to incentivize exaptation.

A different direction for future work would be to improve our detection criteria. We would like to create a method to detect which blocks are viable to be repurposed as exaptations. Additionally, we want to adapt our detection methods to be more generally applicable, as they presently are only able to function with our specifically defined steps. Finally, we would like to refine our detection criteria to improve their accuracy by using some form of potentiation data to detect any increase in average fitness, as opposed to just detecting when the global maximum fitness is reached.

Additional opportunities exist to expand the model to diversify the type of steps in order to utilize additional context dependence (actions based on local environment or time/location history). This could introduce additional complexity in a sequence of steps increases the feasibility of including higher-level components (i.e. at the block level) into the system, which in turn could more rigorously establish the model's connection of developmental exaptations.

Overall, this work has mapped the concept of a developmental exaptation to a computational model, allowing us to explore the ideas intrinsic to such phenomena. Furthermore, through analysis of our model, we have begun to generate insight into *crucial mutations* and their impact on the likelihood of the evolution of developmental exaptations.

Acknowledgements

We would like to thank the ALIFE 2024 reviewers for helpful feedback, which improved the final version of this paper. Also, special thanks to Lindsay Stolting for feedback on an early draft and Mitchel J Daniel for helpful discussions.

References

- Ashworth, J., Fiorito, J., and Yoder, J. (2023). Modeling Evolutionary Development with Indirect Encodings on Dynamic NK Fitness Landscapes. In *The 2023 Conference on Artificial Life*. MIT Press.
- Ashworth, J., Lee, Y., Shen, J., Kim, E., Decker, Z., and Yoder, J. (2022). Evolution of Developmental Strategies in NK Fitness Landscapes. In *The 2022 Conference on Artificial Life*, Cambridge, MA. MIT Press.
- Brakefield, P. M. (2011). Evo-devo and accounting for Darwin's endless forms. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1574):2069–2075.
- Chen, L., DeVries, A. L., and Cheng, C. C. (1997). Evolution of antifreeze glycoprotein gene from a trypsinogen gene in Antarctic notothenioidfish. *Proceedings of the National Academy of Sciences*, 94(8):3811–3816.
- Chipman, A. D. (2021). Developmental Exaptation. In *Evolutionary Developmental Biology*, pages 29–38. Springer International Publishing, Cham.
- Ferguson, A. J. and Ofria, C. (2023). Potentiating Mutations Facilitate the Evolution of Associative Learning in Digital Organisms. In *The 2023 Conference on Artificial Life*. MIT Press.
- Gould, S. J. and Vrba, E. S. (1982). Exaptation-A Missing Term in the Science of Form. Technical Report 1.
- Graham, L. and Oppacher, F. (2007). A multiple-function toy model of exaptation in a genetic algorithm. In *2007 IEEE Congress on Evolutionary Computation*, pages 4591–4598. IEEE.
- Hinton, G. and Nowlan, S. (1987). How learning can guide evolution. *Complex Systems*, 1(3):495–502.
- Holland, P. W. H. (2013). Evolution of homeobox genes. *WIREs Developmental Biology*, 2(1):31–45.
- Kauffman, S. and Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128(1):11–45.
- Knudsen, E. I. (2004). Sensitive Periods in the Development of the Brain and Behavior. *Journal of Cognitive Neuroscience*, 16(8):1412–1425.
- Roth, S. and Hartenstein, V. (2008). Development of *Tribolium castaneum*. *Development Genes and Evolution*, 218(3-4):115–118.
- Todd, G., Candadai, M., and Izquierdo, E. J. (2020). Interaction Between Evolution and Learning in NK Fitness Landscapes. In *The 2020 Conference on Artificial Life*, pages 761–767, Cambridge, MA. MIT Press.