Comparing different implementations of the classical gene regulatory network evolutionary model: the start of an exploration

Csenge Petak
csenge.petak@uvm.edu

1 Problem statement

Phenotypic variation is the fuel for evolution by natural selection. The way genotypic (and sometimes environmental) variation maps to variation in the phenotypes is dictated by a heritable "non-linear function" encoded by a suit of complex interactions between the molecules of cells in living organisms that result in different genes being switched on at different developmental times and in different cells types. These interactions between genes products that result in a specific phenotype can be thought of as a network, a gene regulatory network (GRN). The existence of complex, non-linear genotype to phenotype mappings in evolutionary systems is thought to be essential in adaptation to variable environmental conditions such that GRNs can keep a sort of "memory" of past selection pressures and "recall" good solutions later when the selection pressure reappears.

There are a number of different GRN evolution models published in the literature that aimed to simulate the way genotype to phenotype maps help adaptation in a fluctuating selection pressure scenario. These models have provided valuable insights into how biological systems as a whole could work in general, on an abstract level. However, I noticed that there is a lack of exploration in this field in terms of assumptions, parameters, and general implementations. Most of the studies copied their methods from a single original paper [17] and then from each other. Even when some models were implemented slightly differently, there was no explanation provided as of why that choice was better and likely it wasn't the result of an extended exploration of implementations. This Evolutionary Computation course has taught me that there are many ways to implement EAs and that the choice of selection and mutation methods can influence the kind of solutions we end up with, especially if there is epistasis between positions of our genome that results in a rugged fitness landscape, like it is the case in these kinds of models.

Thus, the **aim** of this project was to explore different implementations of the classic GRN evolution model with a fluctuating selection pressure. I compared how using different selection, mutation and population structure methods influence GRN evolution: the number of generations it takes to find a good fitness after repeated switching of the target and the kind of strategies that emerge. Also, one thing that hasn't really been investigated before using GRN models is the evolution and benefits of complexity. Thus, I also measured the complexity of evolved networks (in terms of number of updates to get to a phenotype) and compared it across my implementations, to see if complexity increased throughout the generations. Lastly, I was also curious to see what happens if I implement an island population structure model with migration, such that an island experiencing variable environmental conditions has an influx of solutions from islands that experience only one of those conditions.

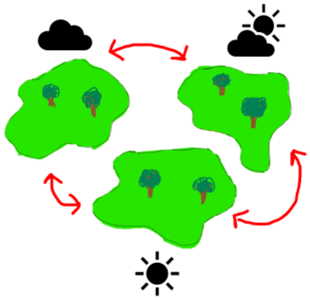


Figure 1. One island only ever experiences environment A, one island only ever experiences environment B, while a third island experiences A then B then A then B, etc. Migration rates are equal among all 3 islands in both directions.

2 Related Work

One of the first attempts to understand the evolution of regulatory networks was described in Wagner [17] followed by Wagner [18] which heavily inspired the implementation of my proposed model and most other studies in the field. Studies that specifically investigated the effect of environmental fluctuations on GRN evolution generally found that over time populations were able to adapt faster to the reappearing environments, as fewer mutation were needed for adaptation due to the evolved network structures. For instance, Crombach and Hogeweg [5] found that populations experiencing 2 alternating environments evolved GRNs that had high robustness, yet there were critical nodes (which they named “evolutionary sensors”) that when removed/added had highly beneficial effects and switched the expression pattern to be optimal in the new environment, enabling the populations to adapt faster when the environment changed. Draghi and Wagner [6] also found increased evolvability (in terms of faster re-adaptation to reoccurring change) over time in response to environmental fluctuation, as well as Lipson et al. [10], due to the evolution of modularity reducing the pleiotropic load such that mutations would affect genes involved in one environment but not the other. The consensus from these studies seems to be that the reason for increased evolvability is the evolution of specific GRN structures that bias the phenotypic variation available for selection such that it aligns with the direction of selection [9, 19].

Previous GRN evolution models have used fitness proportional selection without keeping any parents, with a couple of exceptions that have implemented tournament selection instead [2, 8, 16]. On the contrary, many studies have implemented crossover [1, 7, 15], but not adaptive mutation rates. However, even if some studies used uncommon approaches, they didn’t justify their choice of implementation or directly compare it to alternative methods. To the best of my knowledge, no previous studies have used the island population structure with migration in a GRN evolution model before.

I implemented an evolutionary algorithm (EA) with two target phenotypes that are switched periodically, and an encoding in which the input ("DNA") of the GP mapping is fixed but instead the mapping itself is evolved ("development") to get to the current optimal phenotype. I hypothesised, that over the generations a mapping to evolve that when mutated has a high probability of switching between the two phenotypes, and thus it will take less and less generations for populations to find the new target after a switch, based on previous studies mentioned above. I also hypothesised that having different mutation and selection implementation will influence the diversity of the populations and thus the rate at which they adapt to change in the short-term, as well as the kind of strategies that evolve in the long-term. Finally, I expect that islands with variable environmental conditions receiving solutions from other islands will be able to adapt faster after a change.

3 Methods

First, I implemented the most commonly used and simple model of GRN evolution to serve as a baseline against my various implementations. Since I implemented many kinds of modifications to this model, instead of running repeated experiments with all possible combinations of implementations, I compared each separate implementation to this baseline algorithm for this final project, as I was running low on time.

All code is available at:

https://github.com/Cpetak/EC_final_project

3.1 Basic algorithm:

A population with a 1000 individuals was generated at the beginning of each experiment. Each individual started with an adjacency matrix (w, representing a weighted, directed GRN) as a "genotype" with randomly generated weights. The input (representing gene products coming from the mother) was a vector ($\vec{P}$) of 0s of length equal the number of nodes in the GRN (number of rows or columns in w, $N = 50$, in all experiments) except for 1 at a fixed position. In order to generate phenotypes for each individual, the input vector was iteratively multiplied $N * 3$ times with their individual matrices as such:

$$P_i(t+\tau) = s\left(g_i(t)\right)$$
$$g_i(t) := \sum_{j=1}^{N} w_{ij} P_j(t)$$
$$s\left(x_i\right) = \frac{1}{1+e^{-\alpha(x_i)}}$$

Where τ is a time constant, g_i describes the “strength" of interaction of the product of gene j with gene i and s is a sigmoid function, where $\alpha = 10$ for all experiments. The input of $s()$ can be any rational number ($\mathbb{Q}$), but the output will be bounded between 0 and 1.

After this iterative matrix multiplication, the individuals’ phenotype was a fixed, randomly chosen subset of $\vec{P}$ of size n, which was compared to the current optimal $\vec{P}$ (that changes every G generations) to calculate fitness. The two target $\vec{P}$’s of size n were generated at the beginning of each experiment, target 1 was a random sequence of 0s and 1s and target 2 was 1 - target 1. The fitness of an individual was the sum of $1 - |phenotype_i - current.target_i|$. For individuals for which the iterative matrix multiplication didn’t converge by $N * 3$ steps the fitness was set to 0.

After the initial population and optimal $\vec{P}$s were generated, and each individual’s phenotype was calculated and their fitnesses were determined, individuals were sorted based on fitness and the top 20% were selected for survival and reproduction. Each selected individual was copied p times where $p = (popsize/popsize * 0.2) - 1$ to end up with a constant population size. Children were then mutated at $N * mutrate$ ($mutrate = 0.1$) positions by randomly drawing new weights from a Gaussian distribution with $\mu = 0$, $\sigma = 1$. At the end of a generation, individuals were replaced with a mutated copy of themselves if they have been in the population for more than 30 generations (i.e. max age). All experiments were repeated 5 times and run for 50,000 generations.

Variable parameters of the basic algorithm:

Max age	Season length (G)	Prop. genes for fitness (n/N)
0	50	0.2
30	100	0.5
inf	500	0.8
	1000	
	5000	

Measuring complexity: The number of iterative multiplications needed for the output to converge (i.e. further multiplication doesn't change $\vec{P}$ anymore).

Measuring diversity: Average of standard deviations along the columns of the matrices of a population.

3.2 Alternative selection implementations:

Tournament selection: After allocation of fitnesses, 100 individuals were selected at random without replacement, sorted based on fitness and the top 20% were selected for survival and reproduction (same proportion as the one used for truncation selection). This process was repeated 1000/20 = 50 times independently from one another. The number of children each selected individual had was the number of times it was picked as a winner minus 1, so that the population size remained constant over the generations.

Fitness proportional selection: After allocation of fitnesses, 1000 (= population size) individuals were selected at random with replacement for survival and reproduction. The probability of an individual being picked corresponded to its fitness / sum(fitnesses of individuals in the population). The number of children each selected individual had was the number of times it was picked minus 1, so that the population size remained constant over the generations.

Rank proportional selection: Same as above, expect the probability of an individual for being picked was its rank in the population (highest for the most fit individual) / sum(ranks).

3.3 Alternative mutation implementations:[1]

Crossover: After mutating the children, crossover frequency times population size/2 number of pairs of individuals were select at random for crossover. The same individual couldn't be selected more than once per generation. A column of an adjacency matrix corresponded to a promoter region of a gene, or in other words, contained the weights of the incoming edges. Inspired by Draghi and Whitlock [7], weights associated with input edges of a gene were linked and weren't broken apart, instead crossover only rearranged columns. For **two-point crossover**, crossover happend between two randomly picked columns, such that between those 2 columns of the adjacency matrix of the other parent were placed, and vice-versa. For example, one of the two resulting offsprings could end up with the following matrix where the different colors represent different parent of origin :

$$w_{example} = \begin{pmatrix} 1 & 0.1 & 0.8 & 0 & -1.2 & 0 \\ 0 & 0 & 0 & 0.3 & -2.1 & 0 \\ 0 & 0 & 0 & 0 & 0.9 & 1.9 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

And the other resulting offspring was the inverse of the above: 2 blue columns, followed by 3 red ones, followed by 1 blue. For uniform crossover each column was picked from one parent over the other with a 50% chance. After crossover, each pair of individuals were replaced by the 2 crossed over "children".

Crossover frequencies tested: 0.3 and 0.8, for both two-point and uniform crossover.

Occasionally large mutations: With probability "large mutation rate" each individual of a population had each of their weights in a randomly selected row of their adjacency matrix mutated (redrawn from the same Gaussian distribution as before). This represented transcription factor coding region mutations, or to think about it another way, this was like mutating a node in a network such that all of its outgoing edges changed.

Large mutation rates tested: 0.1 and 0.5

Adaptive mutation rate (1/5th rule): Each generation, I calculated the number of children that had a higher fitness than the parent with the lowest fitness. If this number was higher than 1/5th of the total number of children, I increased the mutation rate by 0.01. Instead, if this number was lower than 1/5th of the total number of children, I decreased the mutation rate by 0.01.

3.4 The island model:

The population (990) was divided up into 3 equally sized subpopulations (330 each) in which the fitness function compared phenotypes to different targets based on the island they belonged to (one of the two static ones or the variable one), and truncation selection (selecting the top 20% / 3 in each population) and crossover was carried out independently in each island. After selection, mutation and possible crossover, individuals migrated from one island to another each generation based on the specified migration rate.

Migration rates tested: 0.1, 0.01, 0.001 and 0. Both with and without two-point crossover at frequency 0.3.

4 Results

4.1 The basic model:

As expected, the average fitness of the population dropped significantly after the first environmental switch, however, this drop became less and less severe and the highest average fitness right before each switch climbed higher and higher

[1] I didn't end up implementing one separately evolvable mutation rate for each weight in the GRN like I proposed to, due to time constrains.

over the generations. This trend could be observed in the first 1000 generations (= 10 switches, one every 100 generations). At this point, on average, the average population fitness decrease slightly (both pre- and post switch), then stabilised, see Figure 2. When looking at the best output over time, it is clear that while in the beginning the populations were trying to follow the moving fitness peaks, overtime individuals with in-between 0.5 values (instead of either 0 or 1) at about 56% of their output positions where selected for reproduction, see Figure 10A and 11A.

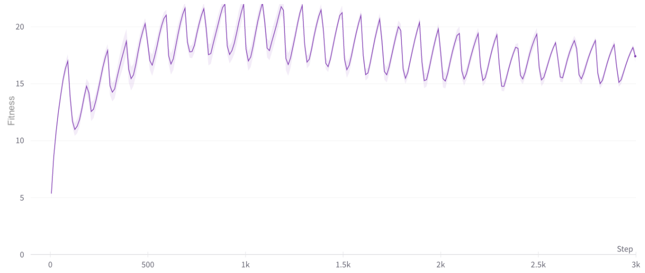


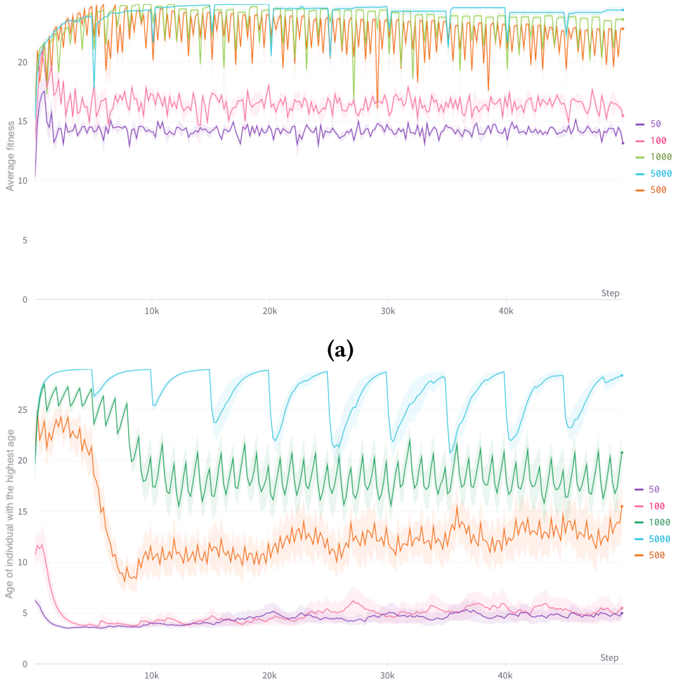
Figure 2. Average fitness of individuals in the population over time in the basic model. Max possible fitness: 25. Showing mean with standard error.

In comparison the control, which was directly evolving the output without a GRN, had a much lower performance overall and wasn't able to increase in average fitness much before the next switch (t-test p=0.0007, average fitness for control: 11.8, for GRN model: 17.7).

The oldest individual rarely exceeded 10 and the average age of the individuals of the populations hovered around 0.5. The average complexity of networks in the population dramatically decreased in all experiments during the initial generations, from 65 to 3 within 4000 generations. Interestingly, it did not further decrease from there but stayed at 3 for the entirety of the simulations.

Changing the season length had dramatic consequences to almost all measured variables. With increasing season length, both the average fitness and the max fitness increased, as well as the average and max ages of individuals of that population, see Figure 3. The diversity of the population was the highest for the highest season length (one-way ANOVA across 5 groups p=7.9177e-06, and Tukey HSD p=0.001 between season length 5000 compared to all other groups). Intriguingly, when looking at the outputs of the highest fitness individuals over time, we can see that when the season length is 5000 instead of 100 there are no locations with 0.5 value, only 1 or 0, see Figure 10D and 11D.

Changing the proportion of genes considered for fitness didn't change much on the outcome of the simulations. As expected, increasing the proportion decreased the proportional average fitness of the populations. When only 10% of the genes were considered, the average fitness hovered around 90% of the maximum fitness which decreased to 70% and 65% for runs where 50% and 80% of the genes where considered, respectively.



(a)

(b)

Figure 3. Average fitness (a) and maximum age (b) of individuals in the population over time with 5 different season lengths. Max possible fitness: 25, max possible age: 30. Showing mean with standard error.

4.2 The effect of having max age:

After experimenting with the basic model, I tested whether having no parents surviving to the next generation or having no max age at all (thus leaving the possibility of someone born at the beginning to survive until the end of the simulation) would change the results. Unexpectedly, the highest maximum and average fitness was observed when all parents were replaced by children (i.e. max age = 0), in this case the populations were able to increase in fitness faster and find better solutions before the environment switched again, see Figure 4. When we look at the output of individuals with the highest fitness over time, we can see that certain loci became 0.5 instead of 0 or 1 much earlier during the simulation, and the final output has more loci with 0.5, if max age is 0 instead of 30, see Figure 10B and 11B, which could explain this increase in fitness.

To my even bigger surprise, the average age of the individuals was higher when the max age was set to 30 than when the individuals could live for infinitely long (p=0.029), see Figure 5.

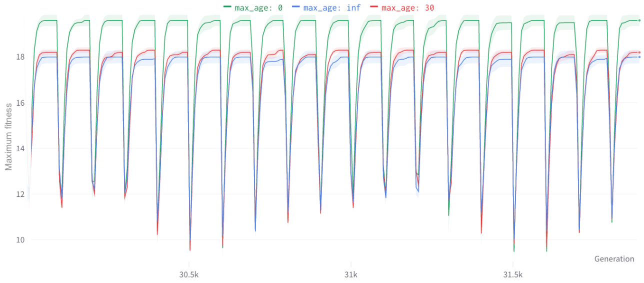


Figure 4. Fitness of the individual with the highest fitness in the population over the generations, between generation 30k and 32k. Max possible fitness: 25. Showing mean with standard error.

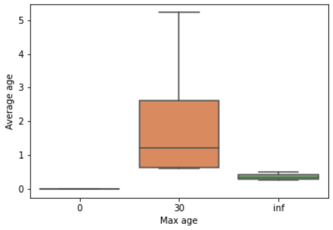


Figure 5. Boxplot of average age of individuals in populations where the maximum age an individual could live to was 0 (i.e. no individual made it to the next generation), 30, or infinite (i.e. no individual was replaced at a certain age).

4.3 The effect of different selection methods:

In terms of average fitness, tournament and truncation selection methods did the best, while rank proportional selection had a significantly lower performance and the fitness proportional selection methods seemingly got stuck at a low local optima in all 5 repetitions of the experiments (one-way ANOVA across 4 groups p<0.001, and Tukey HSD p<0.01 for all pairwise comparisons except for tournament and truncation that were not significantly different), see Figure 6.

4.4 The effect of crossover:

As expected, diversity of the population was the highest when the crossover frequency was the highest, uniform crossover being higher than two-point (one-way ANOVA p=0.017), see Figure 7. The average fitness also increased with crossover, however, whether it was two-point or uniform didn't matter much, and neither did the crossover frequency, see Figure 8.

4.5 The effect of adaptive mutation rates:

Implementing the 1/5th rule resulted in an increase in mutation rates quickly during the first 2000 generation of the experiments (from 0.1 to 0.3, meaning that instead of 10%, 30% of the genes underwent mutations at each generation). For the rest of the 48k generations the mutation rate stayed around this number. In terms of average fitness over the generations, having a mutation rate of 0.3 lead to a much higher fitness (t-test p=0.0058, average of 21 instead of 17.7).

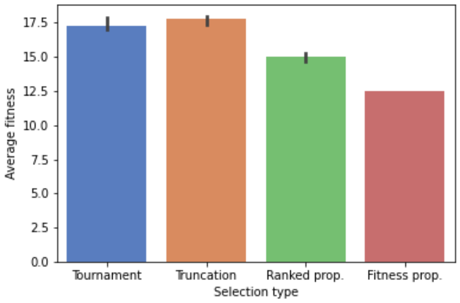


Figure 6. Barplot with 95% confidence intervals of average fitness of individuals in populations where individuals were selected based on tournament, truncation, rank-proportional and fitness-proportional selection.

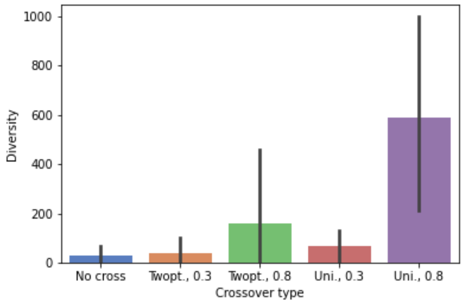


Figure 7. Barplot with 95% confidence intervals of diversity estimates of populations with no crossover, two-point crossover and uniform crossover with 30% and 80% of individuals undergoing crossover at each generation.

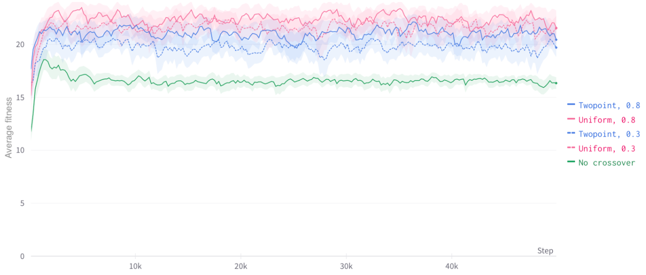


Figure 8. Average fitness of individuals in the population with no crossover, two-point crossover and uniform crossover with 30% and 80% of individuals undergoing crossover at each generation. Max possible fitness: 25. Showing mean with standard error.

Also, when looking at the output of the best fitness individual over time we can see that less loci take on the value 0.5 compared to the experiments with the static 0.1 mutation rate, and at the end of the simulations only 28% of the loci are 0.5 compared to 56% before, see Figure 10C and 11C.

4.6 The effect of occasional large mutations:

Having an occasional large mutation (i.e. mutating the whole row of the adjacency matrix, representing a transcription factor coding sequence mutation) had a positive effect on the average fitness over time, especially when the frequency of the large mutation was set to 0.5 meaning the every generation had a 50% chance of having the large mutation compared to when this frequency was only 0.1 (one-way ANOVA with 3 groups p<0.001, Tukey test for all p<0.01, except for control vs large mut rate 0.1). Having large mutations didn't affect any of the other measured variables.

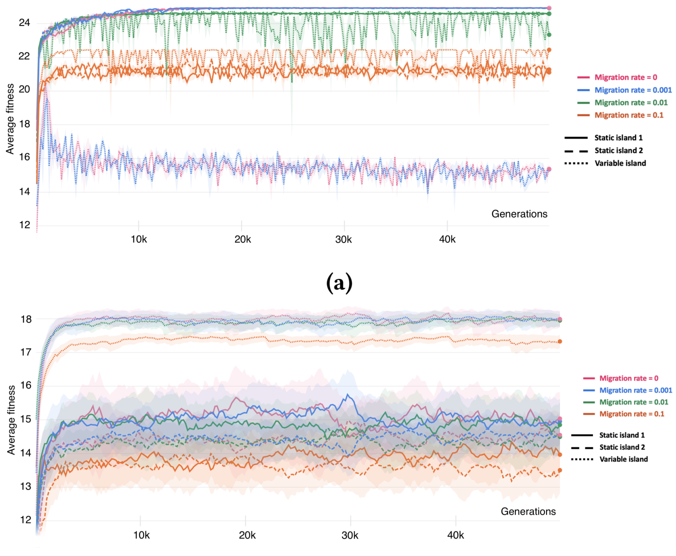
4.7 The island model:

Decreasing the migration rate between the 3 islands, 2 with opposite static targets, 1 switching periodically between the two static targets, increased the average fitness over time in the 2 static islands. Similarly, for the island with the variable environment, decreasing the migration rate from 0.1 to 0.01 increased the average fitness, however, further decrease in migration resulted in a significant reduction in fitness, see Figure 9a.

When I added **two-point crossover** (crossover only occurred between individuals of the same island) the islands experiencing static selection pressure had a significantly reduced average fitness over time, just like the variable island at migration rates 0.1 and 0.01. Instead, variable islands with low migration rates (0.001 and 0) benefited from crossover, see Figure 9b.

5 Discussion

Evolvability can be defined as the ability to evolve, and it is closely related to the ability of creating heritable and adaptive phenotypic variation, since phenotypic variation is the ultimate fuel for evolution by natural selection. There are different ways biological systems can influence variation in phenotype space, including changes in DNA copying and error correcting molecular machinery to increase or decrease mutation rates and thus exploration of the genotype space, or evolving a genotype to phenotype map such that changes in the genotype space have smaller or larger consequences on phenotype [13], or such that the distribution of phenotypes produced by uniform random mutations is biased toward certain directions due to past selection [19]. In a variable environment, evolvability can be increased in order to re-adapt to the reoccurring environment faster by the above mentioned methods as well as 1) evolving phenotypic plasticity if the environmental change is tied to a reliable cue or 2) bet hedging strategies which are "blind" to the changes in the environment [3].



(a)

(b)

Figure 9. Average fitness of individuals in the 3 islands over time with (b) and without (a) crossover. Max possible fitness: 25. Showing mean with standard error.

Bet hedging is the ability to reduce variation in fitness while keeping the same mean fitness over the generations. There are two main kinds of bet hedging strategies: conservative and diversifying. Conservative bet hedgers adopt a generalist strategy - their fitness is worse than the specialists' in any given environment, however, when the environment switches they have less (if any) reduction in fitness and temporarily out-compete the specialist that experience a severe reduction in fitness [12]. Diversifying bet hedgers instead create offspring of which some proportion are very fit in one environment while the rest are very fit in the other environment. Thus, no matter what the environment is like at any given generation, at least some of the offspring will survive and reproduce [4].

In this final project, I wanted to investigate what kind of strategies evolve in a fluctuating selective pressure scenario when only the genotype to phenotype mapping can evolve without any sensing of the environment. Based on previous similar GRN studies, I expected a diversifying bet hedging strategy to evolve, where a GRN structure would evolve such that mutating a few connections in the network would mean a switch between the two optimal outputs hence increasing the speed at which populations can re-adapt to reoccurring targets [5]. Instead, what I found was that in the basic model by the end of the simulations 56% of the output loci had a value of 0.5 instead of either 0 or 1, meaning that they adopted a conservative bet hedging strategy (i.e. a generalist phenotype, see Figure 11). The remaining 44% of the loci

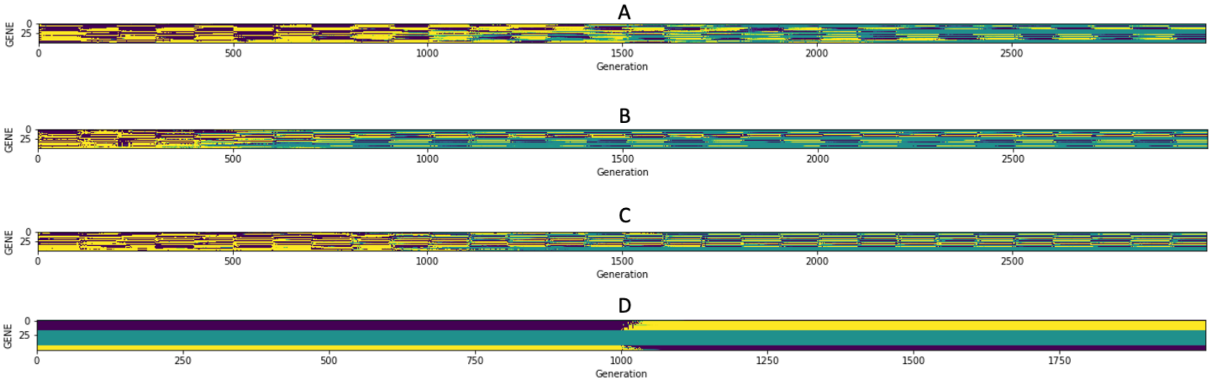


Figure 10. Output (i.e. phenotype that is the outcome of the iterative multiplication with adjacency matrix representing GRN) of the highest fitness individual at each generation. X axis showing generations, y axis showing the 50 genes in the network. Note: in these experiments only 25 genes were considered in the fitness function. Yellow: 0, Purple: 1, Blue: 0.5. All subfigures are showing the first 3000 generations of one run, except for D which is showing 44k – 46k. A = Basic experiment, B = Max age 0, C = Adaptive mutation rate, D = Season length 5000.

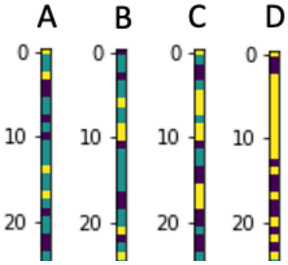


Figure 11. Output of the highest fitness individual at the last generation (generation 49999). Y axis showing the 25 genes considered in the fitness function. Yellow: 0, Purple: 1, Blue: 0.5. A = Basic experiment, B = Max age 0, C = Adaptive mutation rate, D = Season length 5000.

still matched the target of either 0 or 1, and it could be that the GRN structure evolved such that those loci were easier to "flip" like in previous studies, however, further investigation is needed to answer this question. Nevertheless, this is an interesting result that was pretty much universal in my different implementations. This strategy makes sense since lineages that had loci with 0.5 had non 0 fitnesses when the environment switched compared to lineages where all loci perfectly matched the previous target (0 or 1). This could explain the slight reduction in average fitness over time at the beginning of my experiments, followed by the plateau, and also why there was a reduced drop in fitness after some generations after the change in targets, see Figure 2.

Increasing the diversity of GRNs in the population either by 1) introduction or increasing the rate of crossover, 2) having a higher mutation rate due to the 1/5th rule, 3) occasionally mutating the entire row of the adjacency matrices representing the GRNs, or by 4) replacing all of the parents with mutated children such that no individual could survive to the next generation (i.e. max age = 0) all increased the average fitness of the populations over time as they were able to adapt to the changed environment faster with more variation available for selection to choose from. However, increasing variation is a double edged sword because while it is advantageous in the short-term adaptation to the new environment, increased variation can mean to loss of strategies that could have benefited the population or linage long-term when the environment switches [14]. In the most extreme case, if I set the mutation rate to 0 after adaptation to one of the targets, when the environment switched back to this same state in the future, the population would have had perfect fitness immediately and thus perfect "recall" to the previous state. Thus, more investigation is needed on the optimal rate of mutations.

In a similar vein, while **increasing season length** increased the average fitness over time, the longer the season length the less likely it is that the evolvability of the population will increase over time as specialists will out-compete all other strategies by the time the environment switches again if the season length is high [14]. Indeed, the only scenario where I found no generalist strategy to evolve during my exploration of different parameters and implementations was when the season length was set to 5000 generations, see Figure 11. Theoretically, decreasing the proportion of genes considered in the fitness function should enable a freer exploration of the genotype space and help the populations find strategies better suited to deal with the variability of the environment. However, I found that this proportion didn't have much effect probably because the half-generalist strategy described earlier was discovered either way, though more research is needed on this.

Another interesting finding of this study was that the **complexity of the evolved networks** decreased dramatically in the first few thousand generation of the experiments throughout all parameters and implementations. I am unaware of previous GRN evolution studies that looked at how quickly the iterative matrix multiplications converged, to the best of my knowledge they usually just implemented (like I did) a cut off point above which they don't keep updating the initial state. It is intriguing to think why would networks that quickly converge be selected over more complex ones and whether GRN evolution researchers have been wasting a lot of time and computational resources like I did (iteratively multiplying 150 times for each individual each generation when 3 would have been enough), however, I will investigate this topic separately in a future study.

I have also implemented an **island model** with different migration rates between 3 populations, 2 experiencing opposite static targets and 1 experiencing both of those targets in an alternating manner. Intuitively, it makes sense that the static islands had a reduction in fitness with increasing migration rate as more individuals adapted to the opposite target "arrived" to the island. On the other hand, the island experiencing a variable environment greatly benefited from high migration rates as it provided a constant source of good solutions for either of the alternating targets and thus it took the population less time to adapt compared to only relying on mutations as a source of variation. This has interesting implications on the study system I am working with for my PhD, as some populations of purple sea urchins experience very high rates of environmental change but there is also a very high rate of migration from populations experiencing static environments due to their larval life stage [11].

I repeated this experiment with crossover and surprisingly I found that it had a negative effect on average fitness over time in all 3 islands with a high and intermediate migration rate. I originally hypothesised that crossover would help bring together different parts of the GRNs that are adapted to different static targets and thus enhance the ability of the populations to find a solution that can switch between targets more easily. It seems like this assumption of mine was wrong and crossover just ended up scrambling both of the solutions for the two targets.

Finally, I also implemented **4 kinds of selection methods**: truncation (default), tournament, rank-proportional and fitness-proportional selection. While my overall results and conclusions weren't much affected by the type of selection I used (in my limited exploration), importantly, fitness-proportional selection was the worst at re-adaptation and got stuck in local optima the easiest. Ranked-proportional selection did much better and truncation and tournament selection were the best, tournament selection having slightly higher population diversity (although not statistically significant due to high variation between repetitions). This is most likely because as individuals of a population get closer to a local optima, their fitnesses become more similar to one another and so fitness-proportional selection becomes random selection as each individual have similar probabilities for being selected for reproduction. On the other hand, tournament selection doesn't care about the magnitude of differences between fitnesses but introduces a different kind of stochasticity in the selection process due to the randomness of selecting the competing individuals in any tournament. Most of the previous studies investigating GRN evolution have used fitness-proportional selection, including the classic original paper from Wagner [18], and only a few have used tournament based selection. This result highlights the need to rethink some of the ways we usually implement GRN evolution models and agree on a standardised model on which we test different hypotheses.

6 Future work

The aim of this project was to compare the kind of networks and strategies that evolve in a variable environment with different implementations and parameters. Here, I just scratched the surface of this exploration as I didn't have enough time to run more combination of parameters or analyse the evolved network structures (e.g. in- and out-degree distribution, modularity, etc.). Specifically, comparing selection strategies relied on a single set of parameters. Also, all the different parameters and implementations could be tested in combinations, instead of comparing everything to the basic model, for which the parameters were somewhat arbitrarily chosen. For example, what if I tested the effect of maximum age on all 5 different season lengths? Is there an interaction between the kind of selection method I choose and the number of genes considered for fitness?

Nevertheless, by doing this Evolutionary Computation final project 1) I learned a lot about all the things one has to think about when designing an evolutionary algorithm (so many parameters and decisions about exact implementation one has to make!!), 2) developed a very fast code that takes advantage of the GPU so I'll be able to run a lot of repetitions for large population sizes for many generations and combination of parameters, 3) found some interesting results that I didn't foresee but retrospectively make sense, namely the emergence of the half-generalist phenotypes, 4) and opened up many exciting directions I could take in expanding this project into a chapter of my PhD dissertation. Some of these include:

- In the literature and in my project, for individuals that the iterative matrix multiplication don't converge by $N * 3$ steps the fitness was set to 0. Is this a good idea? What happens if we don't do that?
- How do things change if I play with α, the steepness of the sigmoid function applied to the output of the matrix multiplication? The higher this value the more robust the network becomes to mutations.

- What are ways to encourage diversifying bet hedging strategies to evolve? For example, would having a non-predictable periodicity for the environmental change help?

References

[1] Daniel Aguilar-Hidalgo, María C Lemos, and Antonio Córdoba. 2015. Evolutionary dynamics in gene networks and inference algorithms. *Computation* 3, 1 (2015), 99–113.

[2] Josh Bongard. 2002. Evolving modular genetic regulatory networks. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, Vol. 2. IEEE, 1872–1877.

[3] Carlos A Botero, Franz J Weissing, Jonathan Wright, and Dustin R Rubenstein. 2015. Evolutionary tipping points in the capacity to adapt to environmental change. *Proceedings of the National Academy of Sciences* 112, 1 (2015), 184–189.

[4] Oana Carja, Uri Liberman, and Marcus W Feldman. 2014. Evolution in changing environments: Modifiers of mutation, recombination, and migration. *Proceedings of the National Academy of Sciences* 111, 50 (2014), 17935–17940.

[5] Anton Crombach and Paulien Hogeweg. 2008. Evolution of evolvability in gene regulatory networks. *PLoS Comput Biol* 4, 7 (2008), e1000112.

[6] Jeremy Draghi and Gunter P Wagner. 2009. The evolutionary dynamics of evolvability in a gene network model. *Journal of evolutionary biology* 22, 3 (2009), 599–611.

[7] Jeremy A Draghi and Michael C Whitlock. 2012. Phenotypic plasticity facilitates mutational variance, genetic variance, and evolvability along the major axis of environmental variation. *Evolution: International Journal of Organic Evolution* 66, 9 (2012), 2891–2902.

[8] Michal Joachimczak, Borys Wróbel, et al. 2010. Evolving gene regulatory networks for real time control of foraging behaviours.. In *ALIFE.* 348–358.

[9] Loizos Kounios, Jeff Clune, Kostas Kouvaris, Günter P Wagner, Mihaela Pavlicev, Daniel M Weinreich, and Richard A Watson. 2016. Resolving the paradox of evolvability with learning theory: How evolution learns to improve evolvability on rugged fitness landscapes. *arXiv preprint arXiv:1612.05955* (2016).

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

[11] Daniel K Okamoto, Stephen C Schroeter, and Daniel C Reed. 2020. Effects of ocean climate on spatiotemporal variation in sea urchin settlement and recruitment. *Limnology and Oceanography* 65, 9 (2020), 2076–2091.

[12] Helen Olofsson, Jörgen Ripa, and Niclas Jonzén. 2009. Bet-hedging as an evolutionary game: the trade-off between egg size and number. *Proceedings of the Royal Society B: Biological Sciences* 276, 1669 (2009), 2963–2969.

[13] Joshua L Payne and Andreas Wagner. 2019. The causes of evolvability and their evolution. *Nature Reviews Genetics* 20, 1 (2019), 24–38.

[14] Alfredo Rago, Kostas Kouvaris, Tobias Uller, and Richard Watson. 2019. How adaptive plasticity evolves when selected against. *PLoS computational biology* 15, 3 (2019), e1006260.

[15] Alexander Spirov and David Holloway. 2013. Using evolutionary computations to understand the design and evolution of gene and cell regulatory networks. *Methods* 62, 1 (2013), 39–55.

[16] Kirsten H Ten Tusscher and Paulien Hogeweg. 2011. Evolution of networks for body plan patterning; interplay of modularity, robustness and evolvability. *PLoS computational biology* 7, 10 (2011), e1002208.

[17] Andreas Wagner. 1994. Evolution of gene networks by gene duplications: a mathematical model and its implications on genome organization. *Proceedings of the National Academy of Sciences* 91, 10 (1994), 4387–4391.

[18] Andreas Wagner. 1996. Does evolutionary plasticity evolve? *Evolution* 50, 3 (1996), 1008–1023.

[19] Richard A Watson, Günter P Wagner, Mihaela Pavlicev, Daniel M Weinreich, and Rob Mills. 2014. The evolution of phenotypic correlations and "developmental memory". *Evolution* 68, 4 (2014), 1124–1138.