The evolution of gene regulatory networks in variable environments

Csenge Petak*
Univesrity of Vermont
(Dated: October 21, 2021)

Understanding how gene regulatory networks evolve is crucial to step closer to answering fundamental questions in evolutionary biology, such as how organisms adapt to frequent environmental change, how biological innovation comes to be and how multicellularity evolves. In this study, the effect of different parameters was tested on a model of gene regulatory network evolution. Evolving networks experiences 2 alternative environments. We found that increasing the time between environmental changes increased the ability of the GRNs to learn one environment, but this came at the cost of forgetting the previous. Furthermore, as the number of genes whose concentration was considered in the fitness function increase, the average fitness decreased, but surprisingly, increasing the mutation rate of edge weights had the opposite effect. Lastly, the shape of functions describing regulatory interactions greatly influenced robustness. The experiments performed have used more biologically accurate assumptions than most other studies done on the subject previously, and tested novel parameters. Future work could further explore many of these parameters, as they can have significant effects on GRN evolution and still little is known about them.

I. Introduction

Many years have passed since the publication of the Origin of Species by Charles Darwin, who gave us the fundamental mechanism for evolution which resulted in the birth of modern biology. Thanks to the work of countless scientists since then, we know have an overall good understanding of how life evolves on Earth. However, there are still some fundamental questions that need to be answered. How do species adapt to variable environments? How do species "remember" past environmental change and adapt faster when the environment changes again? How is the effect of a mutation translated into a change in traits? How does evolutionary innovation come to be? The answer seem to be hidden in the evolution of development.

Multicellular organisms have the same sequence of nucleotides in each and every one of their cells. Yet, cells can look and behave very differently. This is due to the ability of differential gene regulation, and as a result each cell type (e.g. muscle, blood, stem, etc.) has a different specific set of genes activated or repressed. There genes are called **transcription factors**, their sole purpose is to regulate (i.e. promote or repress the expression of) other genes. Thus, we can build gene regulatory networks (GRNs) where **nodes** represent genes, either regulatory or non-regulator (so called effector genes) and **edges** represent regulatory interactions. This network is directed, weighted, has feedback loops, self-loops and its hierarchical. At the moment of fertilization, eggs only contain proteins from their mother (in sexually reproducing organisms). However, shortly after, the zygote starts expressing its own genes. This is initiated by the mother's proteins, coded by so called **maternal genes**, and they mainly activate the expression of transcription factors that themselves activate transcription factors and so on. Finally, effector genes are switched on that give different cell types their unique properties. GRNs can evolve through mutations in the genes that affect what they regulate, or by de novo gene birth, gene duplication or gene deletion. While GRNs are subject to evolution, they themselves influence the direction and speed of evolution. Thus understanding the evolution of GRNs can reveal important insights in the mechanism evolution.

A. Related work

The first attempt to understand the evolution of regulatory networks is described in [1] followed by [2] and it heavily inspired the implementation of the model this work. The former argued for the importance of gene duplication and **network clustering**, while the latter looked at the effect of stabilising selection on network **robustness**. Since then, many studies investigated the evolution of gene regulatory networks computationally, using different modelling assumptions, level of biological realism and detail, modelling techniques, and aiming to answer various questions.

Many studies have found that robustness emerges in GRNs, but they mostly investigated using static environments [3–5]. Runneburger and Rouzic 2016, for example, found that even though selection pressure on robustness is weak and indirect, mutational robustness evolved, and that interestingly complexity and size of the network had less effect than mutation rate [3].

Studies that instead investigated the effect of environmental fluctuations generally found that over time populations were able to adapt faster (as fewer mutation were needed for adaptation) to a new environment if the population has been experiencing the fluctuation in that environment in a similar fashion for evolutionary significant time [6]. Most studies investigating the effect of **environmental variability** also found the emer-

* csenge.petak@uvm.edu

gence of robustness. For example, [7] found that populations experiencing 2 alternating environments evolved GRNs that had high robustness, yet there were **critical nodes** that when removed/added had highly beneficial effects and switched the expression pattern to be optimal in the other environment, enabling the populations to adapt faster when the environment changed. Another study instead focused on the ability of GRNs that have been evolving under fluctuating natural selection to generate adaptive phenotypes in novel environment and found that variable environments did select for an increased capacity of GRNs to adapt to new environments [8].

Recent papers have made comparison between the evolution of development and learning, and found that under some circumstances **GRNs can "learn"** similar to how neural networks learn, as the structure of the GRN that evolves in changing environments can preserve information about past selection, "remembering" and "learning" that way. This could then enable faster adaptation to the reappearing alternative environment [9, 10].

In this project the aim was to create a model of GRN evolution that uses more biologically relevant assumptions than many of the previous studies and investigate the effect of variables we know little about. This would allow us to answer the following questions:

- *Is there an increase in robustness over generations?* This has been shown before to be true, so if our model shows the same we at least replicated previous results and can be more confident that we are on the right track.
- *Would we still observe robustness emerging if we introduce a cost of having more nodes in the GRN?* Previous studies attributed much of the observed robustness to increasing networks size, adding a cost to it would force evolution to select for it only if truly advantageous.
- *In a variable environment, what kind of strategies emerge, generalists or specialists?* [1] In particular, we can investigate how does the frequency of environmental change influence which of these win in the long term.
- *What kind of network structures emerge?* What are common network properties like average clustering, network density, etc.?
- *What is the relevance of the many possible parameters that influence the evolution of GRNs?* Among them, number of effector genes, steepness of the sigmoid function describing regulatory interactions, season length and edge mutation rate.

[1] Generalist: medium fitness in either environments, Specialist: high fitness in one but low fitness in the other environment.

The following assumptions have been used while designing the experimental protocol:

- The fitness is going to be a function of only the output (effector genes) of the network, not the intermediate nodes (regulatory genes). This allows for more creativity in the internal wiring and expression levels.
- Nodes will have an energy cost - this is biologically realistic and prevents our network size from exploding.
- Networks will start with a very few nodes and edges and then grow, instead of initializing a random network with many nodes and edges.

II. Methods

Code is available at:

`www.github.com/Cpetak/GRN_model`

The experiments have been set up as follows:

A. Creating individuals

A population with a fixed population size of 1000 was generated at the beginning of each experiment. Each individual started with an adjacency matrix (w) that only contained a maternal gene and a set number of effector genes (E) depending on the experiment. The adjacency matrix represents a weighted, directed GRN. Weights represent the strength and sign of the regulatory interaction and direction indicates which gene activates or represses others. Effector genes are genes whose concentration is considered the output of the GRN and whose concentration gives the individuals their expression pattern (I). The maternal gene represents the protein that comes from the mother in developing embryos, and is always expressed in our model. The initial weights in the model were 0.1 for all edges, except for the edge of the maternal gene with itself (self-loops were allowed), which was 1 to allow for continuous expression. New weight were randomly drawn from a Gaussian distribution with probability

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

with $\mu = 0$, $\sigma = 1$. Gene concentrations ($\vec{P}$) ranged between 0 and 1.

$$\vec{P}(0) := \{P_1(0), \ldots, P_N(0)\}$$

Where N is the total number of nodes in the network, including the maternal, regulatory, and effector genes.

When generating the individuals' expression pattern, the adjacency matrix is iteratively multiplied with the vector containing the gene concentrations ($\vec{P}$).

$$\begin{aligned} P_i(t+\tau) &= s\left(g_i(t)\right) \\ g_i(t) &:= \textstyle\sum_{j=1}^{N} w_{ij} P_j(t) \end{aligned} \tag{2}$$

Where τ is a time constant, w is the adjacency matrix, g_i describes the "strength" of interaction of the product of gene j with gene i and s(x) is a sigmoid function:

$$s\left(x_i\right) = \frac{1}{1+e^{-\alpha(x_i-0.5)}} \tag{3}$$

The value of α was different depending on the experiment. This is repeated until further matrix multiplication doesn't change the gene concentrations vector anymore, or matrix was iteratively multiplied $3 \times N$ times. If the loop ends before a stable concentration is reached, the gene expression pattern is determined to be invalid and a fitness of 0 is assigned. Otherwise, the individual's expression pattern is the subset of $\vec{P}$ that contains only the concentrations of the effector genes.

Let's look at an example with 4 effector genes.

Initial gene concentrations:

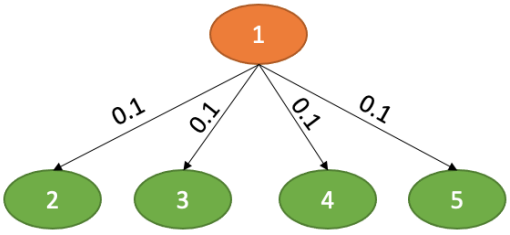
$$\vec{P}(0) := \{P_1(1), P_2(0), P_3(0), P_4(0), P_5(0)\}$$

(only maternal gene is switched on, N=total number of genes).

Initial adjacency matrix (w):

$$w = \begin{pmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Network represented by the adjacency matrix visualised:



The adjacency matrix is then iteratively multiplied with the following vector: $\vec{P} = [1, 0, 0, 0, 0]$. Once $\vec{P}$ doesn't change anymore, we take the elements that represent the concentrations of the effector genes and that is the individual's expression pattern. (In this specific example $\vec{P}$ doesn't change, but mutations can change the initial network so that it does in future generations).

B. Mutations

Four types of mutations were implemented:

1. *De novo gene mutation*

With probability 0.005, networks could gain an additional node with no edges (later, other mutations could add edge). This was implemented by adding a column and row of 0's to the adjacency matrix.

2. *Gene duplication*

With probability 0.005, networks could gain an additional node which was a copy of an already existing node, meaning that a new node was added that inherited the edges of a random node in the network. This was implemented by copying a column and a row in the adjacency matrix and inserting them at the right and bottom respectively. Effector or maternal genes couldn't be duplicated.

3. *Gene deletion*

With probability 0.01, networks could loose a random node and all its connections. This was implemented by deleting a random column and its corresponding row from the adjacency matrix.

4. *Edge mutation*

With probability depending on the specific experiment, a randomly chosen edge's weight was replaced with a value randomly drawn from the Gaussian distribution mentioned above.

C. Generating optimal expression patterns and fitness function

Optimal expression patterns (i.e. sequences of numbers that represent the expression patterns with the highest possible fitness in each environment, denoted by O) were generated for two alternative environments by two beta distributions. The probability density function for beta is:

$$o(x, a, b) = \frac{\Gamma(a+b)x^{a-1}(1-x)^{b-1}}{\Gamma(a)\Gamma(b)} \tag{4}$$

It was implemented in python using the

`scipy.stats.beta.pdf`

function. Beta takes a and b as shape parameters. For all experiments, for one of the environments the optimal expression was generated based on a beta distribution with $a = 2, b = 7$, and the other was the same distribution but mirrored on the x-axis (see Figure 1). These distributions were chosen such that there is little overlap between optimal expression patterns of the two environments and so that results will be easier to interpret (as visual inspection of the individuals' expression pattern will indicate fitness).

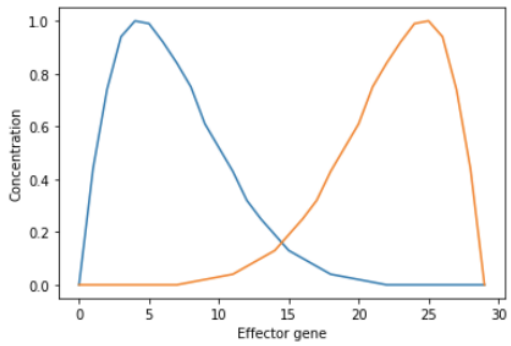


FIG. 1: Beta distributions showing optimal expression patterns, orange in one, blue in the alternative environment.

The fitness of an individual was calculated based on the distance between the individual's gene expression pattern and the optimal expression pattern for the specific environment the individual was experiencing the following way:

$$f(I_i) = \left(1 - \frac{1}{E}\sum_{i=1}^{E}|I_i - O_i|\right)^3 \quad (5)$$

Where f is the fitness function, I is the individual's expression pattern, E is the number of effector genes, and O is the optimal gene expression pattern. In some experiments, the fitness function considered the individuals' GRN size:

$$f(I_i) = \left(1 - \frac{1}{E}\sum_{i=1}^{E}|I_i - O_i|\right)^3 - c(N - (E+1)) \quad (6)$$

Where c is the cost of having many nodes (ranges between 0 and 1), N is the total number of nodes, and $E + 1$ is the number of effector genes and the maternal gene. This meant that if, for example, the node cost was set to 0.1, then the fitness of an individual would decrease by 0.1 times the number of gene in its GRN that are not maternal or effector genes.

D. Calculating robustness and strategy

Each generation the robustness of 10% of the population was calculated. After the expression pattern and fitness of an individual was calculated, a random edge of its GRN was mutated and the expression pattern and fitness was calculated again. This was repeated 50 times. The expression robustness was determined by dividing the number of mutations that changed the original expression pattern by 50. The fitness robustness was determined by dividing the number of mutations that changed the original fitness by 50. A further measure was calculated to keep track of what proportion of the fitness changing mutations increased and decreased the fitness (later referred to as beneficial mutation rate and deleterious mutation rate).

Each generation the strategy of 10% of the population was also calculated.

- **Strategy "low":** in both of the environments fitness < 0.3, or in one of the environments fitness < 0.3, in the other fitness < 0.7.
- **Strategy "high":** in both environments fitness > 0.7, or in one of the environments fitness > 0.7, in the other fitness > 0.3.
- **Strategy "specialist":** in one of the environments fitness > 0.7, in the other fitness < 0.3.
- **Strategy "generalist":** in both of the environments $0.3 <$ fitness < 0.7.

E. Evolution

After the initial population and optimal expression patterns were generated, individuals were mutated, their gene expression pattern was calculated, and they were selected for reproduction with a certain probability weighted by their fitness such that individuals with higher fitness were more likely to be selected. This was done such that the population size would always stay the same. One individual could be selected multiple times. In some experiments there was elitism, where the top x individuals (depending on the variable "elite size") with the highest fitness were select in a deterministic manner, while the rest were selection fitness proportionally. Fitness was calculated based on the current environment, that determined which beta distribution to compare the individuals' gene expression patterns. The environment switch every n generations depending on the variable called season length. Individuals selected for reproduction were copied to a new population whose individuals where mutation, their gene expression pattern was calculated and they where selected for the next population in a fitness proportional manner.

Edge mutation rate	Season length	E	Alpha
0.01	100	10	5
0.05	200	20	10
0.1	300	30	20
0.8	-	-	-

TABLE I: Different parameters used for the experiments

The number of environments was always 2, population size 1000. 3 kinds of node costs were tested in preliminary experiments: 0.0001, 0.001, and 0.01. Since with higher node costs populations' GRNs weren't learning much, node cost of 0.0001 were used for further experiments. Similarly, elite sizes of 0 (everyone is selected fitness proportionally), 0.05 (top 5% of the population reproduces deterministically) and 0.1 (top 10% of the population reproduces deterministically)were tested but as individuals' fitness were much lower with higher elite sizes, for further experiments elite size of 0 was used.

See Table I for the different parameters that were used for the final, longer experiments. Every combination of these parameters was run for at least 7000 generations. All combinations of parameters were repeated 5 times, thus in total 540 experiments were run.

F. Data analysis

Temporal plots were created using Wandb. The rest of the plots and statistics were generated using mainly numpy, pandas, matplotlib, networkx and sklearn. Code is available here.

The effect of edge mutation rate, number of effector genes, season length and alpha on average fitness, average GRN size, expression robustness, and fitness robustness was explored using barplots, a pairwise correlation coefficient matrix and linear regressions. Also Wandb's parameter importance analysis was used to determine the relative importance of these variables.

Every generation, the adjacency matrix of the individual with the highest fitness was saved to a csv (replacing the previously stored network to avoid memory issues). These adjacency matricies were read into networkx objects and for each network average clustering, network density, degree centrality, total number of nodes, number of regulatory nodes (N-(E+1)), number of edges and number of communities were calculated. Again, to investigate the effect of edge mutation rate, number of effector genes, season length and alpha on these network properties, a pairwise correlation coefficient matrix was generated. Next, the CCDF (double check this) was plotted for all networks. Finally, the gene expression pattern along with the optimal gene expression pattern of each network was plotted for visual investigation.

Equation for computing the clustering coefficient for

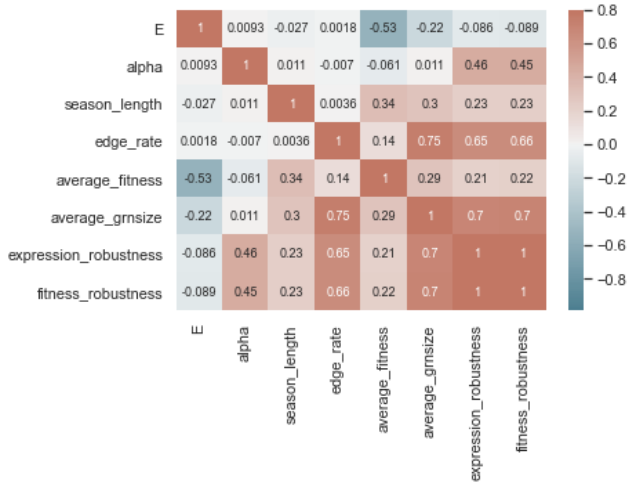


FIG. 2: Pairwise correlation coefficient matrix showing the sign and strength of interactions.

nodes:

$$c_u = \frac{1}{\deg(u)(\deg(u)-1))} \sum_{vw} (\hat{w}_{uv}\hat{w}_{uw}\hat{w}_{vw})^{1/3}$$

where deg(u) is the degree of u and edge weights $\hat{w}_{uv}$ are normalized by the maximum weight in the network $\hat{w}_{uv} = w_{uv}/maxw$.
Network density:

$$d = \frac{m}{n(n-1)}$$

where n is the number of nodes and m is the number of edges.
Degree centrality: The degree centrality for a node v is the fraction of nodes it is connected to.
For community detection the community.best_partition() function was used.

III. Results

A. Overall effects of variables

First population-wide fitness, GRN size (number of regulatory nodes, i.e. not effector or maternal genes) and robustness values were averaged over generations for every run of an experiment to look at overall trends. See Figure 2 for the pairwise correlation coefficient matrix. Also, see appendix for barplot comparing all variables and their combinations.

1. *Effect on average fitness*

The number of effector genes (E) had a strong negative effect on average fitness, regardless of the effect of

other variables. On the other hand, there was a positive relationship between season length and average fitness. While α didn't seem to have an effect, with increased edge mutation rate the average fitness increased too. Interestingly, this didn't hold when $E = 10$, here edge rate had varying effect.

2. *Effect on average GRN size*

Interestingly, similar effect was found regarding GRN size. E, again had a negative effect, while season length had a positive effect. The value of α had no effect. GRN size affected edge mutation rate the most strongly, increasing the edge mutation rate increased GRN size.

3. *Effect on average expression robustness*

Since fitness and expression robustness had almost perfect correlations (so most mutations that altered the expression pattern also affected the fitness of the individual), only one of them was further investigated. Again, interactions with similar sign were found. E had a weak negative effect on robustness (a notable exception being when the edge rate was 0.01 there was a positive effect), while season length had a weak positive effect. The strongest effect, again, edge mutation rate had, which was positive. The biggest difference from the previous two paragraphs was regrading the effect of α, which this time had a strong positive effect on robustness.

Overall, the effect of E was negative, but season length and edge mutation rate had a positive effect, and α only had an effect in robustness.

When correlating population-wide fitness, GRN size and robustness values, fitness had no correlation with GRN size (R-squared = 0.08), or robustness (R-squared = 0.04), however, there was a strong positive relationship between GRN size and robustness (R-squared = 0.49). See Figure 3

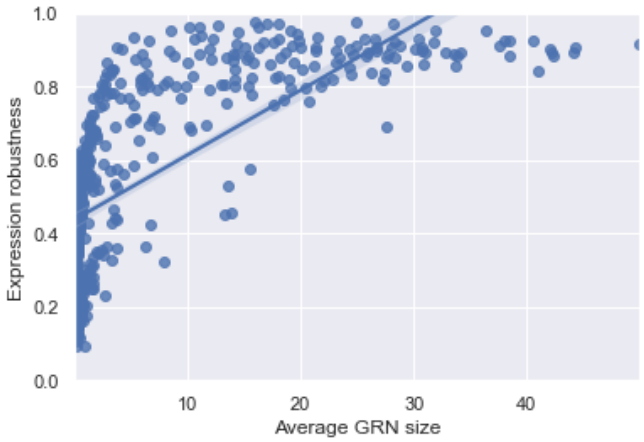


FIG. 3: There was a positive relationship between GRN size and expression robustness, R-squared=0.49

B. Temporal effects of variables

Next, population-wide fitness, GRN size and robustness values were plotted over generation. In addition, the beneficial and deleterious mutation rate, as well as the frequency of strategies in the population were plotted.

As seen in Figure 4, the variation in fitness was increasing with decreasing E. With $E = 10$, the fitness ranged between 0.18 and 0.82, while for $E = 20$ it ranged between 0.25 and 0.5, and for $E = 30$ between 0.3 and 0.41. Interestingly, when the season length was increased from 100 to 500, the lower limit was the same for all E, but the higher limit still decreased with increasing E. When the edge mutation rate was increased, these differences became smaller.

In Figure 5, the frequency of beneficial and deleterious mutations is shown over time. The patterns were similar as before, with increasing E, the range decreased, and increasing the season length and edge mutation rate had similar effects as well.

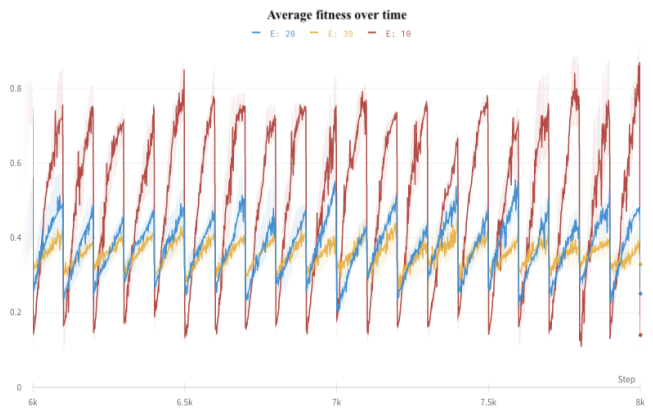
Lastly, in Figure 6, the frequency of generalist and specialist strategies is shown over time. Specialist and generalist frequencies don't add up to 100% as there can also be "high fitness strategies" and "low fitness strategies" that had high fitness in both or neither environments. However, the frequency of "high fitness strategies" was always 0.

When season length was set to 100, at the beginning of a season the frequency of generalists increased until about halfway through the season when specialists took over and their frequency increased up until the next change in the environment. When the season length was increased, at the beginning of a season while the frequency of generalists still increased, the specialist strategy took over relatively quickly and their frequency stay around 100% until the end of the season when the environment changed again. Finally, when E was increased, specialists never took over.

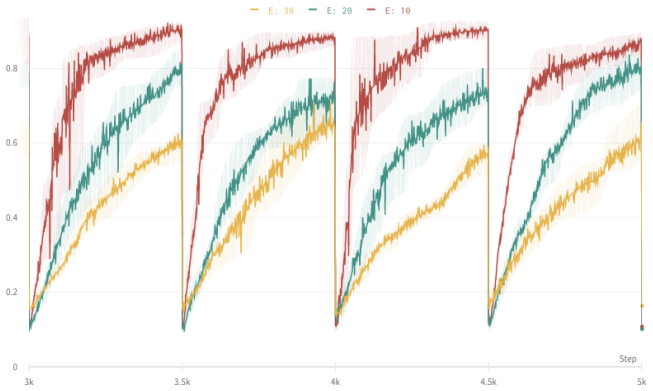
C. Effects of variables on final best GRN

The adjacency matrix of the fittest individual in the last generation was saved and analysed. Since degree centrality highly correlated with network density, the former was taken out of further analysis to avoid redundant variables. For the rest of the variables, a pairwise correlation coefficient matrix was generated, See Figure 7.

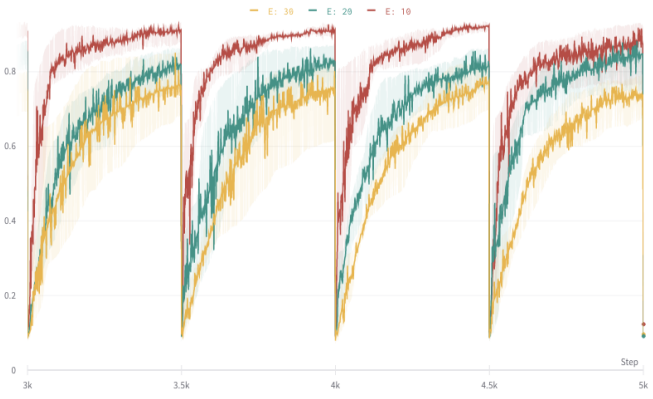
Edge mutation rate and season length had positive effect on all network properties, although the letter had weaker relationships. Alpha seem to have no effect on any of the network properties. The most interesting variable was E, which negatively affected all except number of



(a) Season length: 100, alpha: 5, edge mutation rate: 0.01

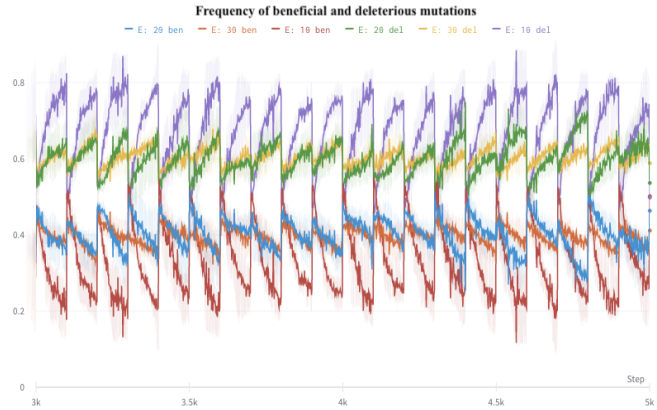


(b) Season length: 500, alpha: 5, edge mutation rate: 0.01

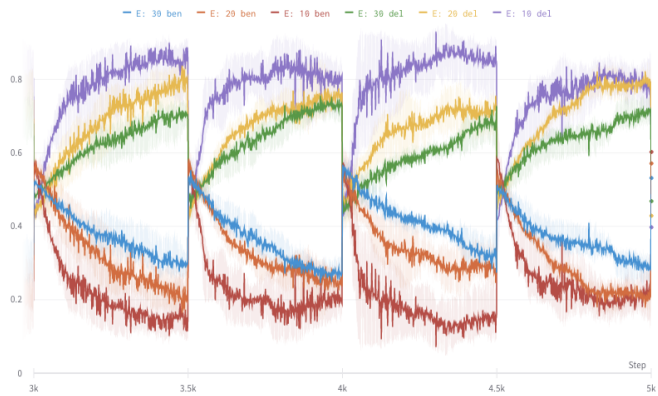


(c) Season length: 500, alpha: 5, edge mutation rate: 0.1

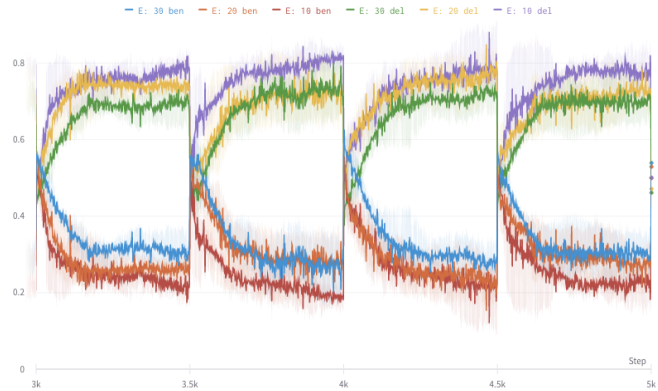
FIG. 4: Average fitness over generations with different E, season length and edge mutation rate. No smoothing was applied. Means with minimum and maximum are shown.



(a) Season length: 100, alpha: 5, edge mutation rate: 0.01



(b) Season length: 500, alpha: 5, edge mutation rate: 0.01

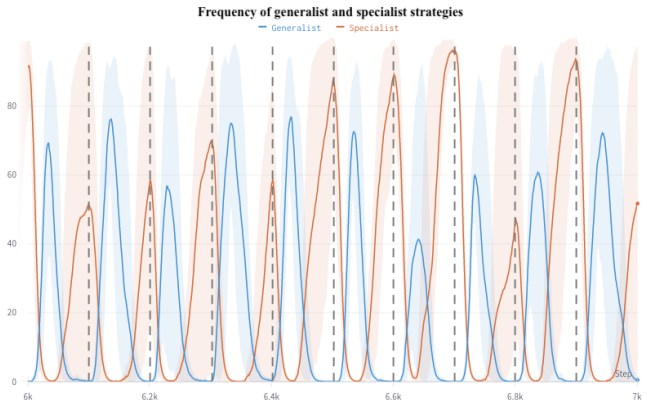


(c) Season length: 100, alpha: 5, edge mutation rate: 0.1

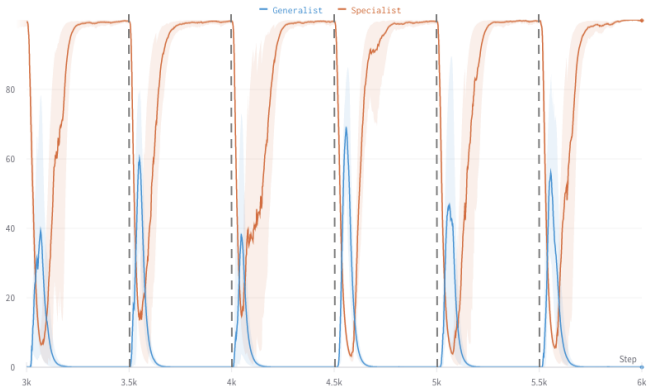
FIG. 5: Frequency of beneficial and deleterious mutations over generations with different E, season length, and edge mutation rate. No smoothing was applied. Means with minimum and maximum are shown.

nodes, but this is misleading since increasing the number of effector genes by default increases the total number of nodes and so we have to look at the number of regulatory genes to get a clearer picture of what was selected for.

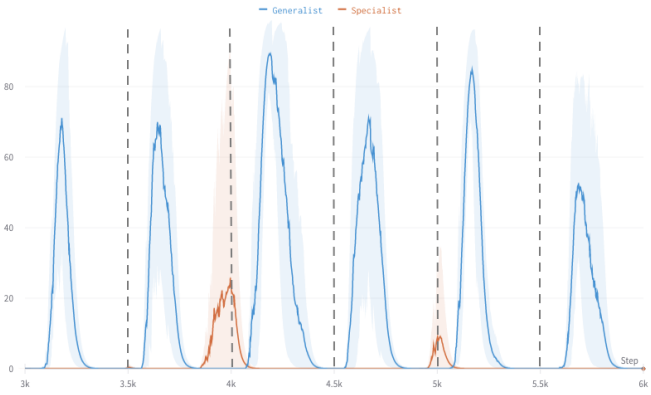
Next, the complementary cumulative distribution function (CCDF) of node degree distribution of each of the best network was plotted. The shape of the CCDF didn't depend on any of the variables we tested (E, season length, edge mutation rate, alpha). Instead, it de-



(a) Season length: 100, alpha: 5, edge mutation rate: 0.01, E:10



(b) Season length: 500, alpha: 5, edge mutation rate: 0.01, E: 10



(c) Season length: 500, alpha: 5, edge mutation rate: 0.01, E: 30

FIG. 6: Frequency of generalist and specialist strategies over generations with different E, and season length. Smoothing of 0.7 was applied. Means with minimum and maximum are shown. Dashed vertical lines show change in environment.

pended entirely on the number of regulatory nodes in the GRN, see Figure 8. There was no indication of power-law

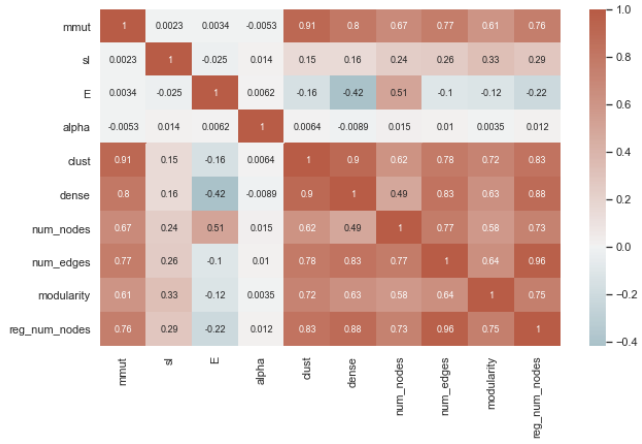
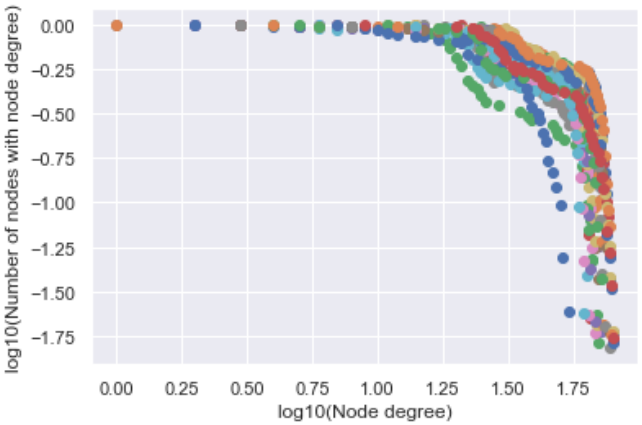
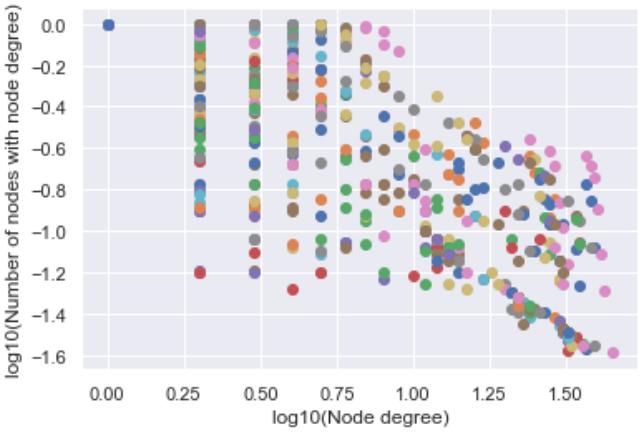


FIG. 7: Pairwise correlation coefficient matrix showing the sign and strength of interactions.

distribution.



(a) CCDF of networks with more than 30 regulatory nodes



(b) CCDF of networks with less than 10 regulatory nodes

FIG. 8: Complementary cumulative distribution function (CCDF) of all highest fitness networks separated based on number of regulatory nodes.

Finally, the expression patterns generated by these GRNs were visually inspected. There were expression patterns that fit the optimal expression pattern well for one of the environments but not the other (i.e. specialist)(Figure 9.a), and there were some that fit the optimal expression patterns for both environments (Figure 9.b). There were also some that seemed to have a random expression pattern and didn't "learn" any of the optimal patterns (Figure 9.c).

IV. Discussion

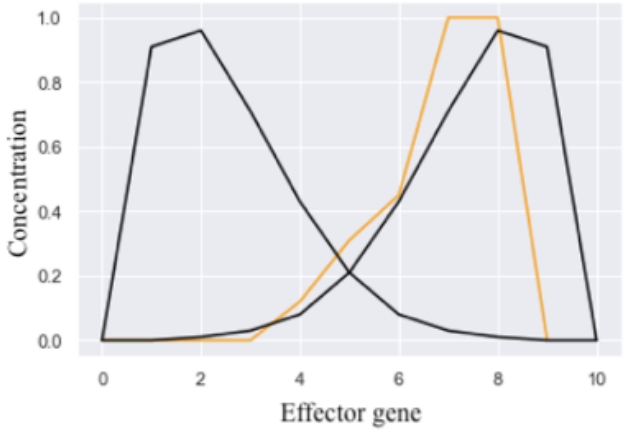
A. Value of α

The steepness of the sigmoid function (s) increases with increasing alpha (α). Experiments were run with 3 levels of steepness. Of all the tests that were run, alpha only had a significant effect on the robustness of the network. The steeper the sigmoid function, the more robust the networks have become. This result is in line with our expectations and observations and theory from previous studies. If the function is steep, most output gene concentrations will either be 1 or 0, and a great percentage of mutations in the network won't have any effect as even if the input to the sigmoid function changes (output of the g function in equation 2) the output of the sigmoid will likely be the same. On the other hand, when the function is more linear, mutations that lead to a different output of g in equation 2, will also likely effect the output of the sigmoid function. However, I suspect that while these networks are robust on average, they are also fragile, as rare specific mutations could dramatically change the output of a steep sigmoid function, from 0 to 1 or 1 to 0.

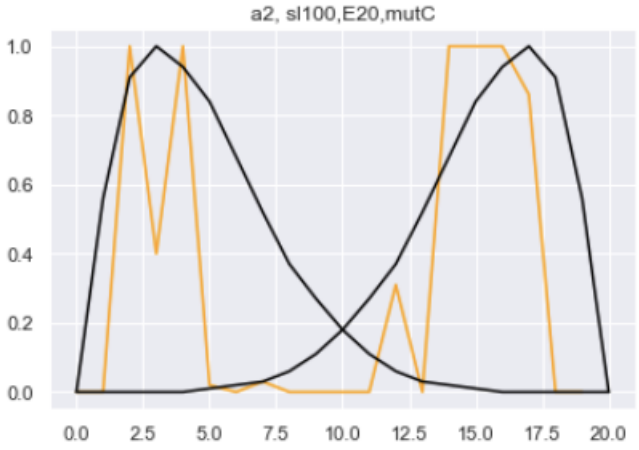
It has been experimentally shown that the activation of genes earlier in development usually happens in a switch-like manner, meaning that when a certain concentration of an activating transcription factor is reached, the target gene is switched on. On the other hand, downstream genes are usually more sensitive to the concentration of the activating transcription factor as the regulatory interaction is more linear. It has been argued that this has evolved to buffer variation earlier in development but fine tune the expression of effector genes [11]. The effect of steepness of these regulatory interaction describing functions on GRN evolution has not been explicitly tested to the best of my knowledge, but it was predicted that the more steeper the sigmoid function, the more robust the network becomes against mutations [12], so the results here are in agreement with the results of previous studies.

B. Number of effector genes

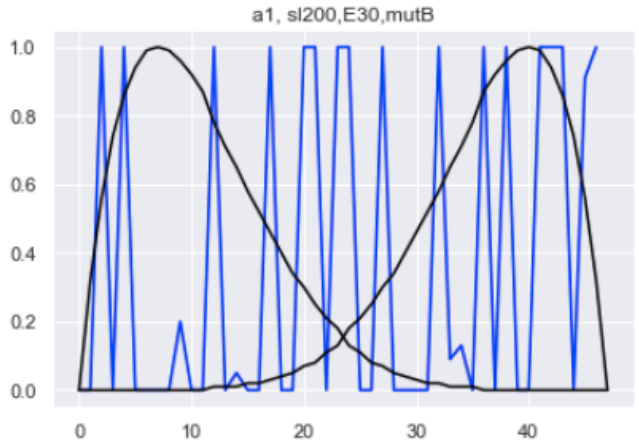
Increasing E had a negative effect on GRN size, robustness and most strongly on average fitness. Reduced aver-



(a) Season length: 100, alpha: 20, edge mutation rate: 0.01, E:10



(b) Season length: 100, alpha: 20, edge mutation rate: 0.01, E: 20



(c) Season length: 200, alpha: 10, edge mutation rate: 0.8, E: 30

FIG. 9: Expression patterns of individuals with the highest fitnesses.

age fitness could be because it is much harder to optimize for more dimensions. This could also explain the differences in fitness ranges in the temporal figures. $E = 10$ GRNs could "learn" the most, and get their expression patterns closest to the optimal, thus their average fitness quickly increased, but then dropped when the environment changed. Similarly, since they were "learning", the fraction of mutations beneficial for them decrease the most for E=10 over time (and thus the fraction of mutations deleterious for them increased). Another increasing observation was that increased E favored the emergence of the generalist over the specialist strategies. This, again, could be because they were "learning" slower than their lower E counterparts and thus information from the previous environment was still retained. This idea was introduced by Rago et al. 2019, that argued that a reduced learning rate could actually be advantageous to retain information and thus have a long-term advantage when the environment switches back [10]. Interestingly, E also negatively affected average clustering, density, number of regulatory gene, and number of edges of the final best GRN. In a future study, I would like to further investigate why there was a negative effect on GRN size (i.e. number of regulatory nodes) and therefore robustness (as they are strongly positively correlated), as having many effector genes whose concentration is what matters for natural selection is biologically realistic and hasn't been studied in depth as far as I know.

C. Season length

Increasing the season length (i.e. number of generations between environment changes) had a positive effect on GRN size, robustness and most strongly on average fitness. The reason for the increased average fitness is clearly visible on Figure 4. When the season length is short, the environment changes much more frequently. When the environment changes, individuals' that were selected for before now had expression patterns very different from the optimal expression pattern, hence very low average fitness was observed after the switch. Season length also changed the range of fitness the populations reached over the generations, since in longer seasons populations could evolve longer and get to a much higher fitness. This of course, in many cases, also meant that the fitness dropped lower when the environment finally changed, as the populations "learned" the new environment more and thus also forgot the previous environment more as well. Based on a similar argument to the one described for E, when seasons were longer, deleterious mutation rates increased much more while beneficial mutation rate decrease much more. Unsurprisingly, increasing the season length also increased the frequency of specialists. Generalists go extinct if the environment doesn't switch back quickly to the environment they are still "remembering" despite their long-term advantage (given that the environment switches back at one point). Season length had only weak positive relationships with the investigated network properties of final best GRNs. Again, I think it would be really interesting to further study the interaction between season length and GRN size and robustness.

D. Edge mutation rate

Surprisingly, edge mutation rate had a weak positive effect on average fitness (also visible on temporal figures) and a strong positive effect on GRN size and robustness. Also, edge mutation rate very strongly positively affected average network clustering, network density, total number of nodes, number of regulatory genes and number of edges. I hypothesize that this is because, since individuals with high edge mutation rates weren't "learning" much through adjusting their weights (especially the ones with edge mutation rate = 0.8), they instead "learned" through adjusting their network structure through additions of nodes [2] and edges. New nodes are initiated with 0 weight edges (meaning no interaction), but due to the high edge mutation rate a lot of these edge were likely mutated to non-zero weights thus the number of edges in the networks increased with increased edge mutation rate. These reasons also explain why there was a positive relationship between edge mutation rate and robustness (remember, this was calculated based on the effect of edge mutations). Edge mutation rate didn't have an effect on strategy.

V. Future work

In follow up studies, it will be essential to run the experiment for many more generations and with more individuals in the populations. It would be interesting to play more with elite size, have the sigmoid function for each node mutable independently, have the mutation rate itself have a mutation rate and run everything with more than two environments that alternate. To test for modularity the number of communities were calculated, however, this metric is not really that informative without knowing what these communities are exactly and what their modularity score is. I would also like to run everything again with random instead of fitness proportional reproduction to see exactly what network properties are due to how I grow the network and what is the result of selection. Finally, I would be curious to see what happens if the envionmental change is unpredictable (i.e. it doesn't happen every x generation but with a certain probability). For example, Tsuda et al. 2010, found

[2] de novo and node duplication rates were the same always, but when edge mutation rate was 0.8, individuals with more nodes were selected

that gene duplications were much more commonly fixed when environments were unpredictably fluctuating [13] and Steiner 2012 found that GRNs were became much less evolvable with increasing randomness of environmental fluctuations [14].

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

[2] A. Wagner, Does evolutionary plasticity evolve?, Evolution **50**, 1008 (1996).

[3] E. Rünneburger and A. Le Rouzic, Why and how genetic canalization evolves in gene regulatory networks, BMC evolutionary biology **16**, 1 (2016).

[4] S. Ciliberti, O. C. Martin, and A. Wagner, Innovation and robustness in complex regulatory gene networks, Proceedings of the National Academy of Sciences **104**, 13591 (2007).

[5] T. Rohlf and C. R. Winkler, Emergent network structure, evolvable robustness, and nonlinear effects of point mutations in an artificial genome model, Advances in Complex Systems **12**, 293 (2009).

[6] T. D. Cuypers, J. P. Rutten, and P. Hogeweg, Evolution of evolvability and phenotypic plasticity in virtual cells, BMC evolutionary biology **17**, 60 (2017).

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

[8] J. Draghi and G. P. Wagner, The evolutionary dynamics of evolvability in a gene network model, Journal of evolutionary biology **22**, 599 (2009).

[9] R. A. Watson and E. Szathmáry, How can evolution learn?, Trends in ecology & evolution **31**, 147 (2016).

[10] A. Rago, K. Kouvaris, T. Uller, and R. Watson, How adaptive plasticity evolves when selected against, PLoS computational biology **15**, e1006260 (2019).

[11] D. A. Garfield, D. E. Runcie, C. C. Babbitt, R. Haygood, W. J. Nielsen, and G. A. Wray, The impact of gene expression variation on the robustness and evolvability of a developmental gene regulatory network, PLoS Biol **11**, e1001696 (2013).

[12] D. Siegal-Gaskins, M. K. Mejia-Guerra, G. D. Smith, and E. Grotewold, Emergence of switch-like behavior in a large family of simple biochemical networks, PLoS Comput Biol **7**, e1002039 (2011).

[13] M. E. Tsuda and M. Kawata, Evolution of gene regulatory networks by fluctuating selection and intrinsic constraints, PLoS Comput Biol **6**, e1000873 (2010).

[14] C. F. Steiner, Environmental noise, genetic diversity and the evolution of evolvability and robustness in model gene networks, PloS one **7**, e52204 (2012).

[15] K. Voordeckers, K. Pougach, and K. J. Verstrepen, How do regulatory networks evolve and expand throughout evolution?, Current opinion in biotechnology **34**, 180 (2015).

[16] R. Jovelin and P. C. Phillips, Evolutionary rates and centrality in the yeast gene regulatory network, Genome biology **10**, R35 (2009).

[17] I. S. Peter, E. Faure, and E. H. Davidson, Predictive computation of genomic logic processing functions in embryonic development, Proceedings of the National Academy of Sciences **109**, 16434 (2012).

A. Appendixes

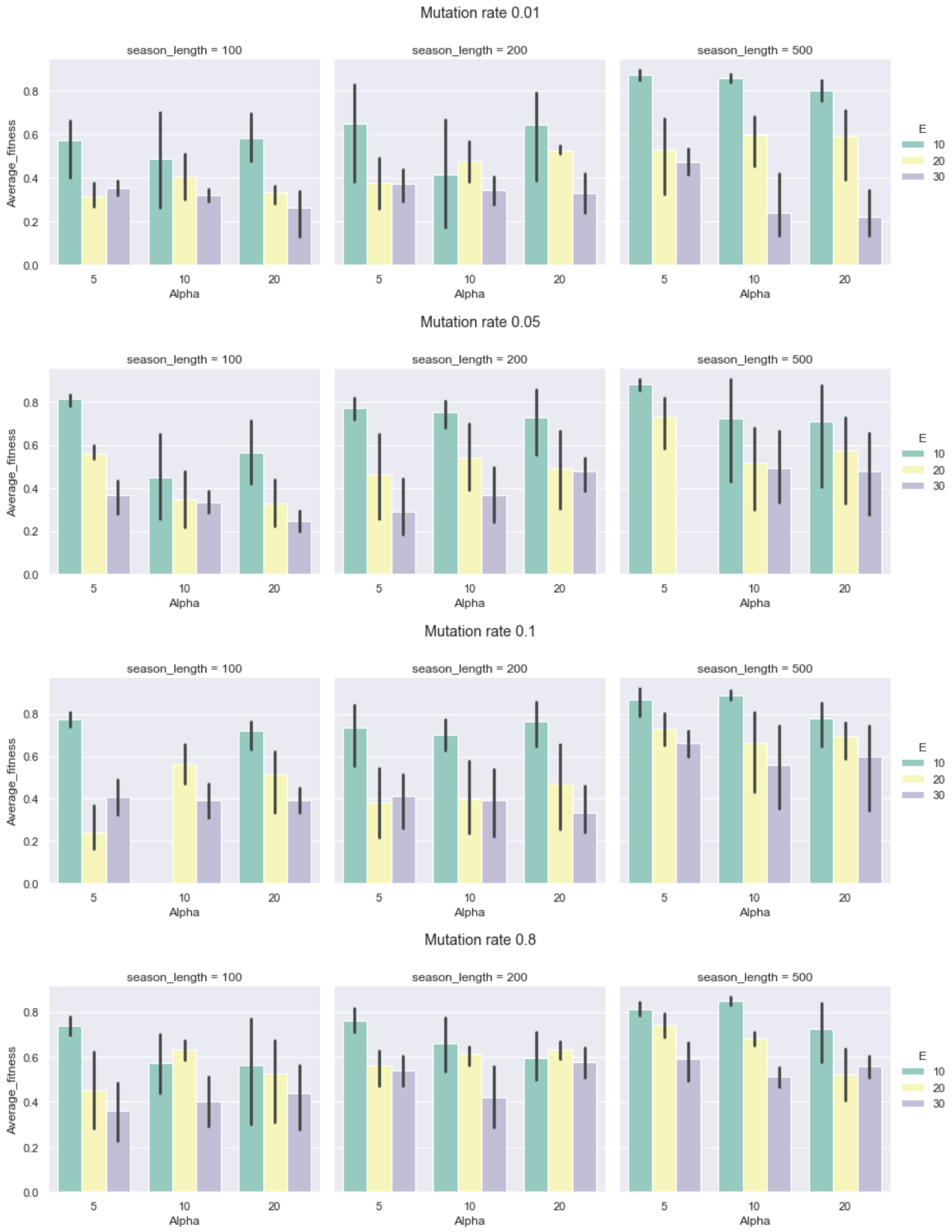

Mutation rate 0.01
season_length = 100
season_length = 200
season_length = 500
Average_fitness
0.0
0.2
0.4
0.6
0.8
5
10
20
Alpha
E
10
20
30
Mutation rate 0.05
season_length = 100
season_length = 200
season_length = 500
Average_fitness
Alpha
E
10
20
30
Mutation rate 0.1
season_length = 100
season_length = 200
season_length = 500
Average_fitness
Alpha
E
10
20
30
Mutation rate 0.8
season_length = 100
season_length = 200
season_length = 500
Average_fitness
Alpha
E
10
20
30

FIG. 10