

# Evolutionary Learning In Grouped Gene Networks

Josh Pattman

April 8, 2024

## 1 Introduction 1.5PM

Introduction: You need not spend more than 1.5 pages describing the original paper.

- The displayed genes of a creature, their phenotype, is not only influenced by their own genetic code, know as their genotype, but also through complex interactions between the genes of that creature.
- This means that each trait displayed by a creature is actually the result of the entire genotype, and not just a single locus in the genetic code.
- The interactions between the different genes is called the gene regulation network.
- During evolution, not only are the genes themselves evolving, but the gene regulation network is also evolving.
- The evolution of the gene regulatory network can be the cause of patterns in covarying genes emerging, where multiple genes may increase the likelihood of a particular trait being displayed.
- More importantly, previously expressed phenotypic patterns may be more likely to be expressed in the future.

## 2 Experiment Setup

- What you evolved including:
  - How you represented individuals
  - What fitness function did you define
  - What kind of GA you used

- Steady state/ generational? Tournament selection/ fitness proportionate (roulette wheel)?
- What kind of crossover (if any) you used.
- All parameters (enough detail so a reader could re-implement the GA)

## 3 Reimplemented Results

- 1 Page
- The reimplemented figures (side by side with the originals)
- Discussion of what the results show/don't show, what worked/didn't work, what you learned

## 4 Extension 1pg

The previously described model of a GRN is a significant abstraction of the true complexity of the interactions between genes. To extend this work, this paper proposes a more advanced model of a GRN which takes inspiration from protein production in genes.

In a biological GRN, the expression level of a gene is affected not directly by the expression levels of other genes, but instead by the levels of proteins produced by those genes. It also may be the case that there are significantly fewer unique proteins available for this purpose than there are genes, meaning that genes with similar functions may have to evolve to use similar proteins for regulation. These features of GRNs are not captured by using a single regulation matrix to represent the interaction between genes.

## 4.1 Goals and Hypothesis

This extension focusses on simulating these intermediate messenger proteins in a simplified manner while maintaining maximum similarity with the previously described method, enabling simple comparisons between the two.

The hypothesis of this experiment is two-fold:

1. **The gene regulation network will be able to remember previously expressed phenotypic patterns even with an overall reduction in the number of parameters when compared to a dense network.** This is because in the datasets used for these experiments, and indeed real life, there are patterns that the network should be able to encode into a single protein. Each pattern may then also affect multiple genes in a similar way. These features leave much room for the network to generalise and group certain features that behave in similar ways together.
2. **The gene regulation network may become more robust and better at generalising to new patterns, as it is forced to 'make sense' of the input genotype.** In traditional machine learning, a bottleneck is a point in a model where there are fewer features than there were inputs to the model, forcing the network to learn the patterns in its data rather than memorising it. Hypothetically, this should also apply to a GRN with fewer chemicals than genes, as the GRN will need to extract only the most important information from the inputs in order to best create an accurate output.

## 4.2 Model

To represent this new model of a GRN, the interaction matrix must be replaced by two distinct matrices, one for encoding a state of the GRN into a vector representing the protein production,  $B_e$ , and one for decoding a protein production vector into the updates to each state,  $B_d$ . Given a number of available proteins  $n_{protein}$  and the genotype size  $n_{genotype}$ ,  $B_e$  has dimensions  $(n_{protein}, n_{genotype})$ , and  $B_d$  has dimensions  $(n_{genotype}, n_{protein})$ . Given these matrices the phenotype at the next timestep,

column vector  $P(t + 1)$ , is given by the equation

$$P(t + 1) = P(t) + \tau_1 \sigma(B_d(B_e P(t))) - \tau_2 P(t) \quad (1)$$

Where  $\tau_1$  and  $\tau_2$  are the update rate and decay rate of the network respectively, and  $\sigma$  is the sigmoid function.

This method has some useful properties aside from fulfilling the aim to simulate the intermediate properties:

1. Mapping the previous state to the next state with two matrices may be a faster computation with fewer parameters compared to a single interaction matrix, as long as

$$n_{protein} < n_{genotype} \div 2 \quad (2)$$

2. As the matrices are multiplied linearly together, it is possible to find the equivalent single interaction matrix by performing the calculation

$$B = B_d B_e \quad (3)$$

Computing the single interaction matrix enables easy comparison to both the original approach, and the weights derived from Hebb's rule, which is useful when evaluating if this model maintains the tendency to move towards the Hebbian weights.

- TODO: Support the value of asking that question (using literature)

## 5 Extension Results

- 1.5 page

## 6 Conclusion

- one page
- What do you conclude from that
- significance of the extension results
- critique/evaluation

## 7 Appendix