Evolving Side Effect Machines to Classify DNA Sequences

MATH*4600: Advanced Research Project in Math

Tamara Charchoghlyan
Advisor: Dan Ashlock

Table of Contents

Abstract

This paper will focus on the classification of DNA sequences using Side Effect Machines (SEMs) chosen with an evolutionary algorithm. SEMs are augmented finite state machines and were created for the purpose of converting DNA (or other string data) into numerical features. In particular, the following paper will focus on a specific DNA corresponding to the Dehydration proteins (i.e. dehydrins). Dehydrins are found in plants and are known to play a large role in protecting the function of other proteins during dehydration stress. Hyperparameter optimization is carried out to determine the most optimal set of hyperparameters in order to obtain the most optimal solution (i.e. SEM).

Introduction

In the last few decades, the quantity of DNA sequences available to researchers has grown exponentially [1]. The need for accurately and automatically classifying DNA has grown with the size of the dataset. This study was inspired by the lack of accurate classification techniques for the dehydrin proteins. If it is possible to automatically classify dehydrins, further research can be carried out pertaining to their function and evolution. This will lead researchers
into understanding and controlling their expression, allowing us to protect our crops from dehydration stress and increase arable land.

SEMs transform DNA strings of variable length into fixed length numerical vectors. It is an injection of DNA into $\mathbb{R}^n$ [1]. This is very useful since a feature set is being collected that can be represented with vectors of equal length for sequences of varying lengths. This makes it significantly easier to compare the DNA sequences.

Additionally, this opens up opportunities for using many different analysis techniques as a way of evaluating the performance of the SEMs. The goal is to evolve SEMs whose maps from DNA to $\mathbb{R}^n$ allow one analysis technique to function really well. This would then suggest that it is preserving important information across the string-to-real-parameter bridge. Once a simple measure (i.e. analysis technique) has been chosen that allows for the determination of whether or not an information-preserving SEM has been found (i.e. extracts a diverse feature set), more complicated analysis can be carried out on said features.

Dehydration Proteins and Classification

Dehydration is the decrease of the amount of free water available to the cell. It is considered to be the most important environmental stress that affects plants as it causes a greater reduction in crop productivity than any other abiotic stress [2]. The dehydration proteins (i.e. dehydrins) are a multi-family of proteins that are highly expressed in plants during dehydration stress. There is an upregulation of the dehydrin proteins and mRNA during drought, cold, and salinity. Due to its high hydrophilic amino acid composition, they are intrinsically disordered proteins (IDS) [3]. This means that dehydrins lack a significant secondary or tertiary structure. In addition, the exact biochemical function of dehydrins is not fully known *in vivo*. However, there is a large body of evidence from *in vitro* experiments that suggests they are responsible for membrane protection, cryoprotection of enzymes, and protection from reactive oxygen species [3].

Due to their lack of structure and clearly defined biochemical function, dehydrins are defined by the presence of at least one copy of the K-segment sequence motif [2]. A motif is a short recurring pattern in the amino acid sequence of a protein. The following is the fifteen amino acid long K-segment: EKKGIMDKIKEKLPG. However, an inspection of a range of other reported dehydrin sequences shows that its conservation is not absolute [3]. For these reasons, it is difficult to accurately and confidently classify a protein as being a dehydrin.

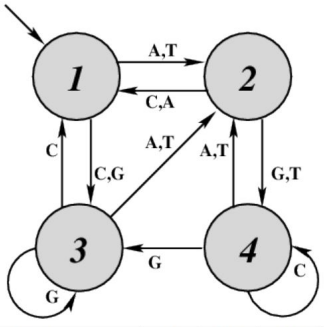
Phylogenetics can be used to observe the evolutionary relationships between genes or species. A phylogenetic tree is a graphical representation of such relationships and can be beneficial when trying to deduce where certain genes diverged. In the case of dehydrins, when inspecting the phylogenetic tree, it is evident that there are gaps. For example, the proteins are found in vascular seed plants, but not in vascular non-seed plants [2]. Due to the lack of certainty when classifying dehydrins, further evidence is required to determine whether certain species truly do not produce the dehydrin proteins or whether the proteins are being misclassified.

Developing a method to detect dehydrins that does not depend on the presence of the K-segment may prove to be beneficial, increasing the accuracy of the classification, and providing a foundation for further research of their function and evolution. Understanding the mechanism by which plants can survive dehydration stress will lead to protecting crops from damage and increasing the amount of arable land.

Side Effect Machine Representation

Side Effect Machines (SEMs) are augmented finite state machines (FSM) with counters on each state. An FSM is a mathematical model of computation, consisting of states and a transition function, and can be used to efficiently identify patterns in strings. The machine starts at a defined start state and changes states based on external inputs. It can be in exactly one of a finite number of states at any given time. The data is read sequentially, one character at a time, driving the transition from state to state. For a SEM, the counters are incremented upon each visit, yielding a vector of integers in the positive orthant of $\mathbb{R}^n$ for an n-state SEM for each string that is processed.

Example 1: The following is a SEM with four states, with the results of the processed strings in the table of the figure below. C1 is the counter for state one, C2 is the counter for state two, and so forth. The arrows indicate the transitions between states, with the labels being the input values that correspond with that transition. The arrow on state 1 indicates that it is the start state. For instance, if the machine is in state one and encounters an input of “A” or “T”, it will transition to state two.



String	C1	C2	C3	C4
CATCATCAT	1	3	1	5
TACTACTAC	4	3	3	0
GGGCCCAAA	4	2	4	0
ACGTACGTA	3	4	3	0

Fig. 1. A 4-state SEM with the results of the processed strings in the table below the SEM.

Methods

Evolutionary computation is a branch of artificial intelligence and is an abstraction from evolutionary concepts in biology. Population-based trial and error algorithms are implemented to create global optimization procedures and methodologies in order to solve problems. A set of candidate solutions, which comprise the population, are randomly generated and iteratively updated through continual selection and optimization. A new generation is produced by removing less desired solutions and introducing small random changes (i.e. mutations). More specifically, algorithms use mechanisms such as reproduction, mutation, selection, and recombination in order to evolve the population. The quality of solutions is evaluated using a fitness function. The chosen fitness function of the evolutionary algorithm is an objective function that determines how close the current solution is to the optimal design solution. The population evolves in order to increase the fitness.

Evolving SEMs

The chosen approach starts by randomly generating a population of SEMs and performs a certain number of mating events, during which the population is updated by mating two SEMs and factoring in random mutations. A SEM can be represented by its transition function, described in a table in Figure 2. Each row of the table can be viewed as a chromosome (i.e. a characteristic of a SEM).

Example 2: The following is a tabular representation of a SEM. It describes the transition function for a given machine. As an example, if the SEM is in state 0 and encounters input “A”, it will stay in state 0. However, if it encounters input “T”, it will transition to state 2. Each row can viewed as a “chromosome” of the machine.

States	**A**	**T**	**G**	**C**	
0	0	2	2	3	S0
1	1	3	2	0	S1
2	0	1	3	2	S2
3	1	2	2	0	S3

Fig. 2. A tabular representation of the transition function of a SEM. The rows (S0, S1, S2, S3) represent the chromosomes of the SEM.

Given two tables representing parent SEMs, a two-point crossover (i.e. recombination) is performed to produce two offspring. In two-point crossover (Figure 3), the parent chromosomes are lined up and two crossover points are randomly chosen. The bits in between the two points are swapped, creating two new SEMs. After performing the two-point crossover, mutations are introduced where the transition functions of the offspring are randomly altered. That is, for each

machine, a random transition for a random state is chosen and set to a random state within the machine. The individual fitness values of the new offspring are evaluated and the two least fit members of the population are replaced with the new SEMs. The two-point crossover and mutation create the required diversity in the population in order to facilitate novelty, while the “artificial selection” forces the increase of quality (i.e. fitness).

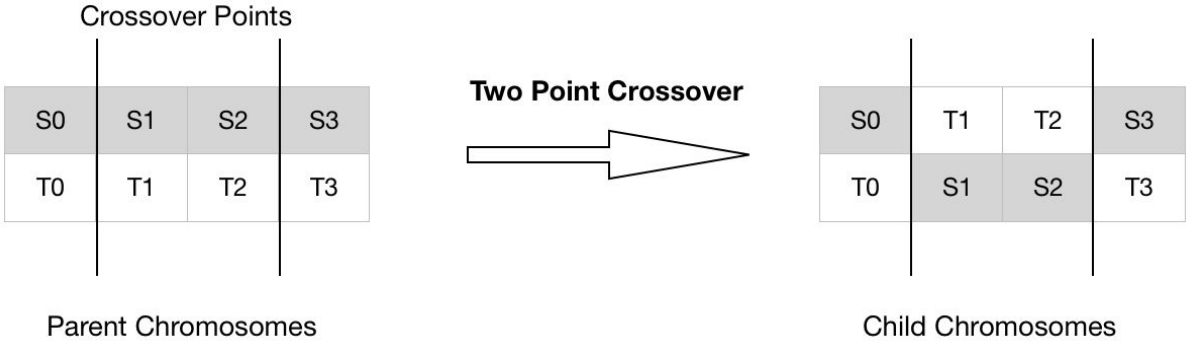


Fig. 3. Two-point crossover between 2 SEMs. S0, S1, S2, S3 represent the first parent while T0, T1, T2, T3 represent the second parent. When the bits are swapped, 2 new offspring are produced.

This process is repeated 30 times and the best SEM is saved for each run, resulting in 30 of the best SEMs and their corresponding fitness values.

Hyperparameter Optimization

In machine learning, hyperparameters are the parameters that are set before the learning process begins and do not change during training. In contrast, the value of a parameter is derived during training. Hyperparameter optimization, also known as tuning, is the problem of choosing a set of optimal hyperparameters for a learning algorithm. Hyperparameter settings are critical to many machine learning methods as the performance of the resulting model is highly dependent on them [4]. Additionally, it provides insight into the importance of different hyperparameters and their interactions [4], and the impact that they may or may not have on the performance of the model. For this specific problem, three hyperparameters were considered: population size, number of states for a SEM, and maximum number of mutations (MNM). MNM is considered during each mating event; a random number, x, is generated in the interval [1, MNM], and the offspring is mutated x times.

For the first iteration of tuning, three values were considered for each hyperparameter:

Table 1: the three different hyperparameters and the values considered during the first iteration of the tuning.

Population size	10	100	1000
MNM	3	7	9
Number of States	6	18	24

All possible combinations of the above variables were considered, yielding 27 different combinations, and saving a total of 810 of the best SEMs and their fitness values.

For the second iteration of tuning, the results from the first iteration were observed and additional values for two of the hyperparameters were tested:

Table 2: two of the hyperparameters and the additional values considered during the second iteration of the tuning.

Population size	5	500	2000
Number of States	3	10	30

Running the additional tests yielded yet another 27 different combinations and, therefore, 810 more SEMs and their fitness values were saved. In total, 1,620 distinct data points (i.e. fitness values of SEMs) were saved and analyzed for the purpose of hyperparameter optimization.

DNA Sequence Dataset

The dataset consists of known dehydrin DNA sequences as well as synthetic dehydrin DNA sequences. The synthetic sequences were derived using the codon profile of dehydrin data. A codon is a sequence of three DNA or RNA nucleotides and, in this case, corresponds to an amino acid in the protein chain. The dehydrin dataset is analyzed and patterns are extracted using a hidden Markov model. The synthetic data is generated using these patterns, resulting in sequences that closely resemble dehydrin protein sequences, but are clearly not.

Fitness Function

As previously mentioned, the fitness function of an evolutionary algorithm is used to determine how close the current solution is to the optimal solution. For this specific problem, the function must be designed to accurately reflect how well the SEM is classifying the DNA sequences. In other words, the goal is to find the best information-preserving SEMs, those that extract the features (i.e. vectors) that will allow us to determine which DNA sequences correspond to dehydrins. In a sense, it could be treated as a clustering problem. The accuracy of classification

can be evaluated using the Euclidean Distance, calculating distance between the vectors in $\mathbb{R}^n$ generated from running the machine on all inputs, both dehydrin and non-dehydrin sequences.

During the first set of experimental runs, when a vector was generated for a given DNA sequence, it was normalized based on the length of the string, yielding vectors where all members are between 0 and 1.

Example 3: If the vector generated from running a DNA sequence of length 9 through a four-state SEM is $[1, 3, 1, 5]$, the normalized vector would be obtained by dividing all members by 9; $[\frac{1}{9}, \frac{1}{3}, \frac{1}{9}, \frac{5}{9}]$.

When calculating the fitness for a given SEM, loop over all pairs of vectors; if they are in different categories (i.e. one corresponds to a dehydrin and the other corresponds to a non-dehydrin), calculate the Euclidean Distance between the two vectors.

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Formula 1: Euclidean Distance (between vectors p and q) as the chosen simple analysis technique for determining whether an information-preserving SEM has been found.

Sum all such distances and divide by the total number of dehydrin-non-dehydrin pairs. In essence, take the average distance between pairs in different categories. If this average value is low, then the SEM was not able to evolve in such a way to tell dehydrins apart from non-dehydrins. The category of a DNA sequence can then be distinguish based on its proximity to these two groups of vectors in space.

However, when observing the resulting fitness values for the best SEMs, the same fitness value was generated over and over again, regardless of the values of the hyperparameters. It seemed as though the SEMs were evolving to reach a local maximum. While this meant that they were evolving to classify the data accurately, it also meant that the features being extracted were not as diverse as desired. When the fitness function was more closely considered, it was evident that due to the normalization of the vectors, there was a built-in hard maximum of $\sqrt{2}$. Since the vectors were normalized such that all members were numbers between 0 and 1, they could essentially be viewed as probability distributions. The population was essentially evolving to maximize the distance between two sets of probability distributions, for which the largest value is $\sqrt{2} \approx 1.41421$. The fitness value that was being observed for most of the SEMs was approximately 1.38513. Figures 4-6 show the resulting box plots for the first iteration of tuning values. It can be observed that the plots do not have a large box, if they have one at all, visualizing the very low diversity in features.

Fig. 4. Extracting the marginal distribution of fitness based on population size. Each plot has 270 data points.

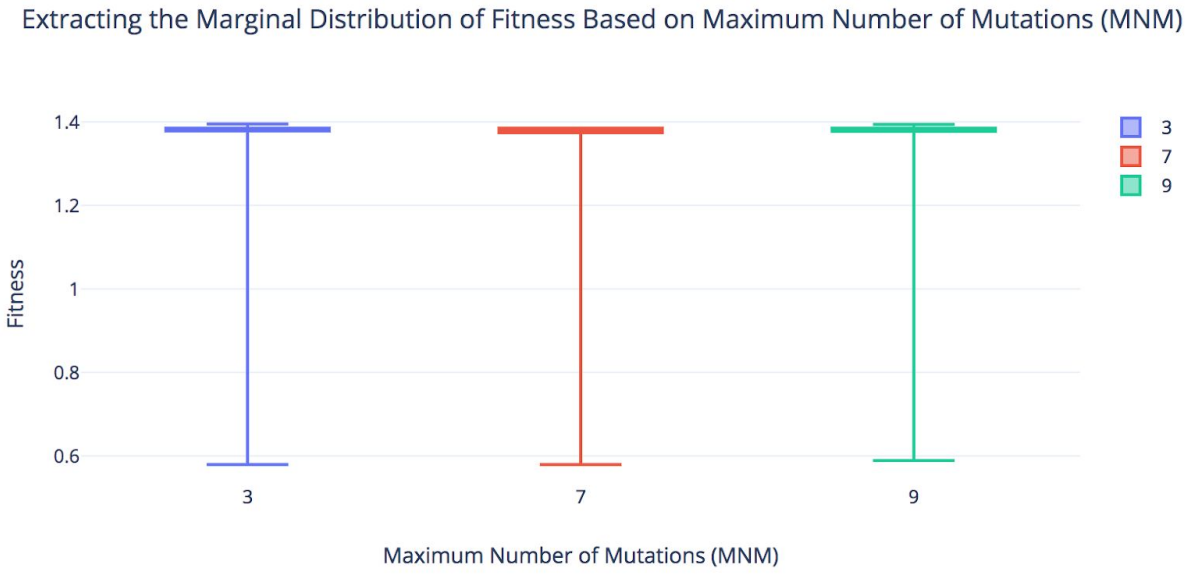


Fig. 5. Extracting the marginal distribution of fitness based on MNM. Each plot has 270 data points.

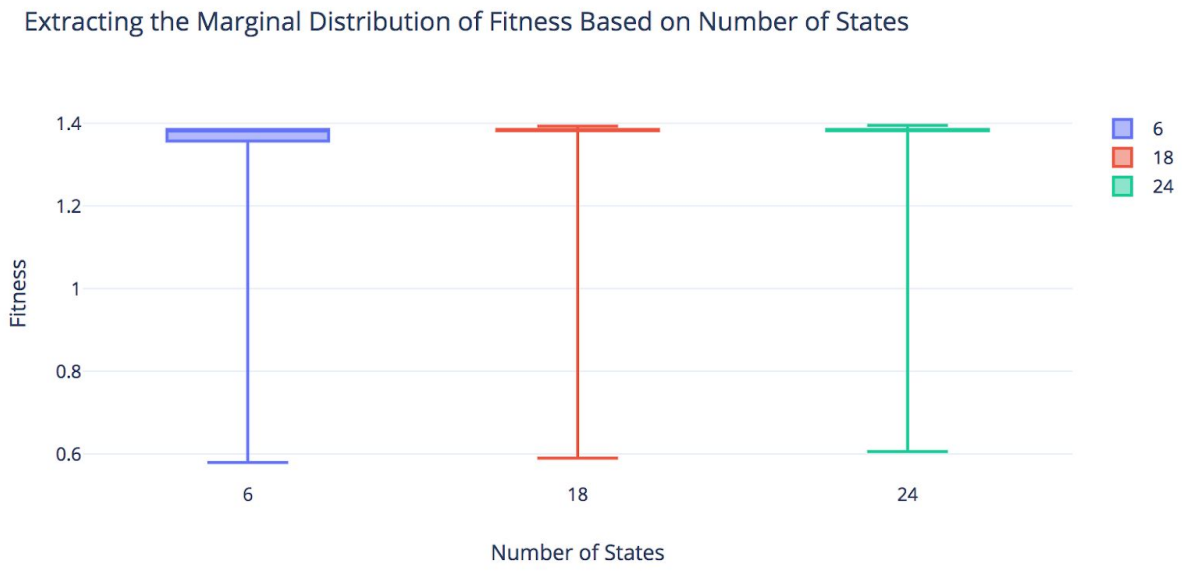


Fig. 6. Extracting the marginal distribution of fitness based on number of states. Each plot has 270 data points.

The only way that a maximum distance can be obtained between two sets of probability distributions is if there are two distinct events, one of which has all the probability in the first distribution and one of which has all the probability in the second distribution. In the terms of the problem at hand, this means that the machines that were evolving were forcing all dehydrins to one state and all non-dehydrins to another state, creating sink states. A sink state, also known as a trapping is state, is a state that only has a transition back to itself. Once a machine is in a sink state, it will never leave to another state. Note that a few state transitions are needed before a sink state can be reached. This accounts for the difference between $\sqrt{2}$ and the observed maximum fitness value of approximately 1.38513.

Example 4: if p_1 and p_2 are different probability distributions, then the distance is at most $\sqrt{2}$. This can shown with the following:

$$p_1(x_i) = 1 \qquad p_1(x_j) = 0 \qquad where\ j \neq i$$
$$p_2(x_k) = 1 \qquad p_2(x_j) = 0 \qquad where\ j \neq k$$
$$where\ i \neq k$$

In other terms, the distance between two such vectors v and u is being calculated:

$$v = [1, 0, ..., 0]$$
$$u = [0, 1, ..., 0]$$

In essence, different versions of the same perfect recognizer were being generated through the genetic algorithm, extracting the same feature over and over again. However, the goal is to evolve SEMs that recognize different things, i.e. extract diverse features.

A new fitness function was chosen, which was similar to the old one but allowed the algorithm to evolve SEMs such that they extracted more diverse features. The new function did not normalize the vectors based on the length of the DNA sequence and, therefore, yielded integer vectors. The Euclidean Distance was then calculated between all pairs where the categories were different (i.e. a vector corresponding to a dehydrin sequence and a vector corresponding to a non-dehydrin sequence), and all such values were summed. This total value was then divided by the sum of the distances between vectors in the same category, plus one (to prevent against a divide-by-zero error). Essentially, the new function is attempting to maximize the distance between categories and minimize the distance within categories. It is desired that the transitions associated with one class and the transitions associated with another class to be disjoint, and that those sets be relatively small. Figures 7-9 show the resulting box plots for the first and second iteration of tuning values when SEMs were evolved using the new fitness function.

Fig. 7. Extracting the marginal distribution of fitness based on population size, using the new fitness function. Each plot has 270 data points. Refer to table 3 in the appendix for median, maximum, minimum, quartile 1, quartile 3, and IQR values for each box plot.

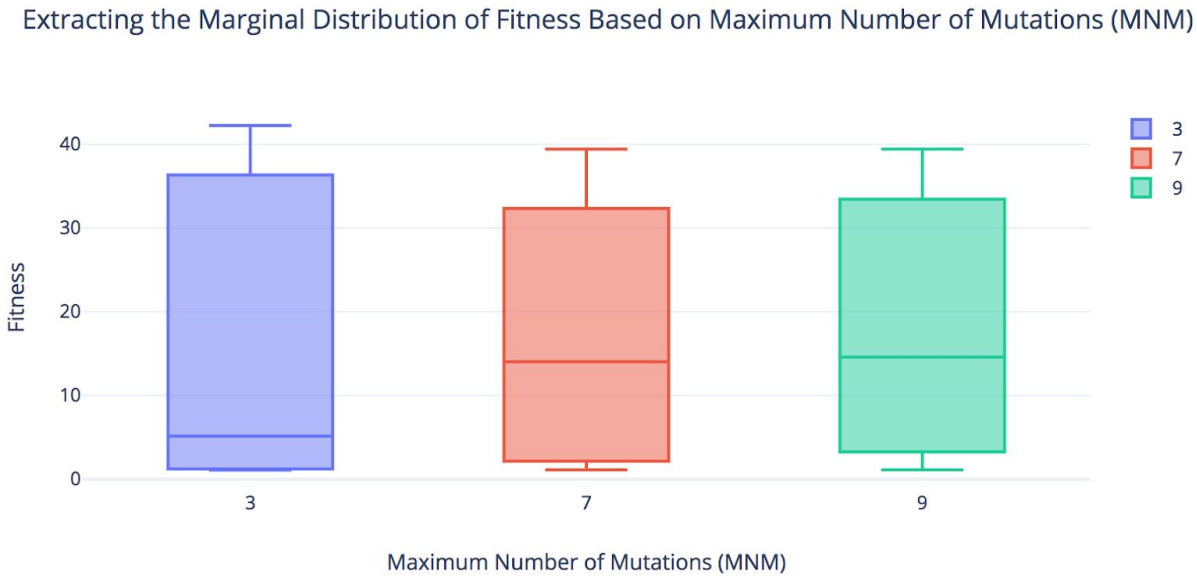


Fig. 8. Extracting the marginal distribution of fitness based on MNM, using the new fitness function. Each plot has 540 data points. Refer to table 4 in the appendix for median, maximum, minimum, quartile 1, quartile 3, and IQR values for each box plot.

Fig. 9. Extracting the marginal distribution of fitness based on number of states, using the new fitness function. Each plot has 270 data points. Refer to table 5 in the appendix for median, maximum, minimum, quartile 1, quartile 3, and IQR values for each box plot.

Results and Discussion

The box plots were generated using marginalization. Since there are many fitness values which have different values for each critical parameter (i.e. hyperparameter), they were grouped by fixed values of one critical parameter, in effect averaging over the other critical parameters. The distribution of one critical parameter across all the other critical parameters is called extracting the marginal distribution.

It was clear from Figure 7 that a population size of 100 was the most conducive to evolving SEMs such that they had a high fitness value and yielded a diverse feature set. This is shown by the median fitness value of approximately 35.4819 (Table 3), and the long tail on the box plot. This figure also shows that having a lower population size yields a lower fitness value. Similarly, Figure 8 shows that a MNM of 7 is the most optimal choice from all of the tested values. Note that a MNM of 9 is comparable in both median fitness and the diversity of features that are generated.

As it pertains to the number of states, it is not as clear which value yields the most optimal solutions. However, it is not necessary to choose just one. By using different values for this hyperparameter, many diverse SEMs will be generated, the feature sets of which can then be analyzed further using more complicated analysis techniques. Figure 9 shows that neither a low number of states nor a very high number of states is conducive to evolving optimal SEMs. Having 5 states produces SEMs with a high fitness values but low diversity of features. Having 10 states produces SEMs with relatively high fitness values but and an average diversity of features. Having 18 states produces SEMs with relatively lower fitness values but high diversity of features.

Based on the results shown in Figures 7-9 and the discussion above, a final set of hyperparameter values were chosen and 900 production runs were carried out. MNM was chosen to be 7 and population size was chosen to be 100. Figure 10 shows the marginal distribution of fitness based on the number of states. Fixing the other two hyperparameter values, the following can be observed regarding the choice of the number of states. Choosing the number of states to be 18 yields SEMs with relatively high fitness values and a highly diverse feature set. Having 6 states yields SEMs with very high fitness values. However, there is essentially no diversity in the feature set. Lastly, evolving SEMs with 10 states yields high fitness values but a relatively uniform feature set. Finally, Figure 11 shows the aggregated fitness values for a total of 900 production runs.

Fig. 10. Extracting the marginal distribution of fitness based on number of states, using the new fitness function. Population size and MNM are fixed; 100 and 7, respectively. Each plot has 300 data points. Refer to table 6 in the appendix for median, maximum, minimum, quartile 1, quartile 3, and IQR values for each box plot.

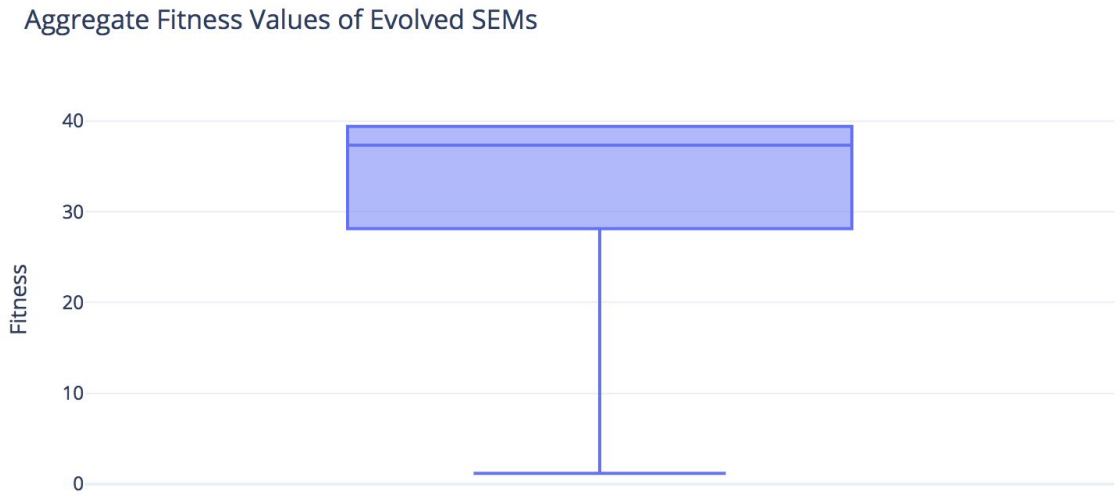


Fig. 11. Aggregate fitness values of evolved SEMs over 900 production runs. Population size and MNM are fixed; 100 and 7, respectively. The number of states varies; 6, 10, or 18. Refer to table 7 in the appendix for median, maximum, minimum, quartile 1, quartile 3, and IQR values for each box plot.

Conclusion and Next Steps

It has been shown that it is possible to evolve SEMs to extract features from a dataset of DNA sequences of two different categories with a relatively high fitness. A set of hyperparameter values were found to be conducive to evolving SEMs such that they had high fitness values as well as a diverse feature set. Fixed values for population size and MNM were found (100 and 7, respectively), while the number of states varied (6, 10, or 18). This is important since the next step is to generate an abundance of SEMs using these results and test on other data.

As previously mentioned, once a simple technique has been used for determining which SEM preserves features best, a more complicated analysis tool must be used in order to select the features that describe the sequences uniquely. In other words, if two features have different opinions about different sequences, they contain different information. Such unique features can be found using a feature selection algorithm. An example of this is to use the Pearson Correlation to find a group of features that contain different information from one another (i.e. have low correlation).

Example 5: Choose the best feature out of all the ones that have been generated. Sort the rest of the features by quality and by pearson correlation with the best feature, favouring those close to 0 (i.e. low correlation). In the next iteration, choose the next best feature and sort the rest. However, this time, sort the features by their Pearson Correlation with respect to the two previously chosen features. If the Pearson Correlation is treated as distance, a tree building algorithm can be used to generate a tree, on which leaves close to one another will be features where their “distance” from one another is near 0 (i.e. low Pearson Correlation, indicating diverse features).

This method, when used to classify the dehydrin proteins, can prove to be very useful as it will automate the classification and provide a higher level of confidence when looking for the proteins in plants where they were not previously believed to have existed. Being able to classify these proteins paves the way to further research regarding its evolution and biochemical function, as well as into understanding and controlling their expression [2]. This will allow us to better protect our crops from dehydration stress and increase arable land.

Appendix

Tables 3-7 in the Appendix show median, maximum, and minimum values for each box plot, as well as values for quartile one, quartile three, and the interquartile range (IQR).

Table 3: Table representing the values from the box plots in figure 7.

	Population Size: 5	Population Size: 10	Population Size: 100	Population Size: 500	Population Size: 1000	Population Size: 2000
Median	1.6002	2.6557	35.4819	12.7237	21.1231	7.6419
Maximum	39.4285	39.4285	42.2489	39.4285	39.4285	39.0313
Minimum	1.0517	1.1024	1.1286	1.1268	1.1380	1.0846
Quartile 1	1.1597	1.1714	1.1811	5.1135	14.3368	5.1135
Quartile 3	13.2731	37.3542	39.2565	36.0961	32.6515	19.1010
IQR	12.1134	36.1828	38.0754	30.9826	18.3147	13.9875

Table 4: Table representing the values from the box plots in figure 8.

	MNM: 3	MNM: 7	MNM: 9
Median	5.1135	13.9944	14.5660
Maximum	42.2489	39.4285	39.4285
Minimum	1.0517	1.0858	1.0712
Quartile 1	1.1910	2.1282	3.2449
Quartile 3	36.3269	32.3220	33.4311
IQR	35.1359	30.1938	30.1862

Table 5: Table representing the values from the box plots in figure 9.

	Number of States: 3	Number of States: 6	Number of States: 10	Number of States: 18	Number of States: 24	Number of States: 30
Median	5.1135	37.3542	26.6477	13.4657	10.2302	3.1540
Maximum			39.4285	39.4285	42.2489	39.3996
Minimum	1.1670	1.1672	1.1266	1.1024	1.1100	1.0516
Quartile 1		26.8880	18.3938	1.1717	1.1736	1.1528
Quartile 3		39.4285	39.0107	35.1892	21.4136	15.2899
IQR		12.5405	20.6169	34.0175	20.24	14.1371

Table 6: Table representing the values from the box plots in figure 10.

	Number of States: 6	Number of States: 10	Number of States: 18
Median	39.4285	37.3542	30.5912
Maximum			39.4285
Minimum	1.1714	1.1629	1.1520
Quartile 1		30.8036	1.1834
Quartile 3		39.4285	36.4075
IQR		8.6249	35.2241

Table 7: Table representing the values from the box plots in figure 11.

	Aggregated Values from Production Run
Median	37.3542
Maximum	
Minimum	1.1520
Quartile 1	28.1528
Quartile 3	39.4285
IQR	11.2757

References

[1] A. Mceachern and D. Ashlock, “Shape control of side effect machines for DNA classification,” 2014 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, 2014.

[2] A.C. Riley, D. Ashlock, and S.P. Graether, “Evolution of the modular, disordered stress proteins known as dehydrins”, PLOS One, 2019.

[3] S.P. Graether and K.F. Boddington, “Disorder and function: a review of the dehydrin protein family”, Front. Plant Sci. 5:576, 2014.

[4] F. Hutter, H. Hoos, and K. Leyton-Brown. "An efficient approach for assessing hyperparameter importance." International Conference on Machine Learning. 2014.