Simulating the evolution of transcriptional regulatory motifs

Robert Forder (rforder1@umbc.edu)

Department of Mathematics and Statistics

University of Maryland, Baltimore County

Senior Thesis Spring 2012

Abstract

Transcription factors are proteins that bind DNA for the purpose of regulating the expression of specific genes. The process by which transcription factors identify their binding sites is poorly understood. It is possible to infer the nature of recognition strategies used by transcription factors through an examination of the information content of their associated binding motifs. Unfortunately, experimentally verified binding motifs are often incomplete and numerical methods of examination frequently suffer from under-sampling artifacts. We attempt to better understand the feasibility of specific recognition strategies by simulating the co-evolution of a transcription factor and its associated binding motif. We explore the hypothesis of positional independence and conclude that it is unlikely for an organism to rely heavily on correlations between positions in a binding motif as its primary means of binding site recognition.

1 Introduction

The regulation of gene expression is fundamental to almost all known forms of life. Gene expression is regulated through a variety of mechanisms. Transcription factors typically bind DNA at *promoter regions* upstream of specific genes in order to inhibit or facilitate gene expression. By directly influencing the behavior of RNA polymerase, transcription factors are one of the most efficient methods of gene regulation. A specific nucleotide sequence to which a transcription factor is known to bind is a *binding site* of that transcription factor. The process by which transcription factors recognize their binding sites is still not well understood. By examining patterns in their binding sites, it is possible to make inferences concerning the recognition strategies of transcription factors. This will help us to better understand the behavior of interacting networks of transcription factors.

It is of primary interest to us to quantify certain features of collections of binding sites. Specifically, we are interested in quantifying the information content of such a collection. Information arises from two distinct signals. Traditionally, conservation of bases at specific positions of binding sites has been of primary interest. Conservation can be detected with a linear recognition strategy, and makes the simplest assumptions about the capabilities of transcription factors. Alternately, we may consider information that arises as a result of correlations between positions in binding sites. A transcription factor must be capable of non-linear classification in order to utilize this signal.

1.1 Transcription factors

Transcription factors are a specific type of protein. They bind to promoter regions which are typically upstream of genes. By binding to promoter regions, transcription factors either inhibit or promote the

expression of the downstream gene by blocking or facilitating transcription by RNA polymerase. RNA polymerase is responsible for transcribing genes and constructing units called messenger RNAs (mRNAs) which are then bound by ribosomes and translated into proteins. A single transcription factor may regulate the expression of multiple genes, and very often many transcription factors form an interacting regulatory network which controls the expression of several genes in response to environmental changes. Well-known examples of such regulatory networks include the regulation of the *lac* operon [4].

The collection of all sequences to which a transcription factor is known to bind is called the *binding motif* of that transcription factor. When constructing a binding motif, we generally align all of the sequences within the motif and truncate them to a common length. Thus we may consider the frequency with which a base occurs in a *position* or *column* of a binding motif. Determining the complete binding motif of a transcription factor is not a trivial task. By examining the binding motif of a transcription factor, we can find patterns among the sequences to which a transcription factor is known to bind. By examining the nature of these patterns, we can attempt to make certain inferences concerning the mechanism by which a transcription factor identifies and binds to its sites.

1.2 Information theory

We borrow ideas from information theory to analyze binding motifs and identify these patterns. We utilize Shannon entropy to quantify the information content of a binding motif. This technique was pioneered by Schneider in [7]. We will use the notation which was developed by Schneider.

1.2.1 Entropy

The notion of entropy in the context of information theory is analogous to thermodynamic entropy. It is a quantity that represents the amount of uncertainty in a signal. We first define the notion of *genomic entropy*. Let $P_g(b)$ represent the frequency with which the base b appears throughout the genome g, where $b \in \{A, T, C, G\}$. The genomic entropy of g is then

$$H_g = -\sum_b P_g(b) \log_2 P_g(b). \tag{1.1}$$

We may apply the same notion of entropy to any nucleotide sequence. Next, consider the entropy of a specific position within a binding motif. Let $P_x(b)$ represent the frequency with which the base $b \in \{A, T, C, G\}$ appears in position x of the binding motif of some transcription factor. The entropy of position x in the motif is then

$$H_x = -\sum_b P_x(b) \log_2 P_x(b). \tag{1.2}$$

Say, for example, we wish to consider the entropy of two positions simultaneously while taking into account possible correlations between those two positions. Instead of considering a signal over the alphabet of nucleotides, we now consider a signal over the alphabet of dinucleotides, or nucleotide pairs. Let $b = (b_1, b_2)$ be an ordered pair of bases and let $P_{xy}(b)$ be the frequency with which b_1 appears in position x of a binding site while b_2 is in position y of a binding site (that is, b_1 and b_2 are in positions x and y within the same site). Then the *joint entropy* of positions x and y within the binding motif is

$$H_{xy} = -\sum_b P_{xy}(b) \log_2 P_{xy}(b). \tag{1.3}$$

1.2.2 Information content

Entropy is in many ways the opposite of what we are primarily interested in. Presumably, if a transcription factor is able to distinguish a binding site from its genomic background, then there must be some form

of information present within that site which the transcription factor is able to recognize. That is, some pattern must be present in a binding site for a transcription factor to distinguish that site from the genomic background. We wish to quantify that information content. We define information content of a signal as the decrease in entropy upon receiving the signal. Within the context of transcription factor binding sites, the information content of a position in a binding motif is the difference between the genomic entropy and the entropy of that position in the binding motif. Suppose we randomly sample a single nucleotide in the genome of *Escherichia coli*. Let b denote this mystery nucleotide. We may predict the value of b by examining the frequency with which each nucleotide appears in the E. coli genome. If we then learn that b was sampled from position x of some binding site of a transcription factor whose complete binding motif is known, then we can consider the frequency with bases appear in position x of that binding motif to aid us in predicting b. If the entropy of position x in the binding motif is less than the genomic entropy, then we have gained information. In fact, we have gained precisely

$$R_x = H_g - H_x \tag{1.4}$$

bits of information. A defining property of Shannon entropy is that it is additive for independent random variables. So, one way of defining the information content of an entire binding motif is

$$R_{sequence} = \sum_x R_x \tag{1.5}$$

where x varies over the positions in the binding motif. Of course, here we are only considering the information content found in each position of the motif and summing it. Hence, we are assuming that binding of a transcription factor to its binding sites operates in a linear, positionally independent manner, and we are explicitly ignoring any correlations which may occur between positions within the motif.

We may also define the minimum information content that would be required of binding motif in order for a transcription factor to distinguish the sites that constitute it from the genomic background by considering

$$R_{frequency} = -\log_2 \frac{\gamma}{G} \tag{1.6}$$

where γ denotes the number of sites in the binding motif and G denotes the length of the genome within which the motif is found.

Say, position x and position y are always identical within the motif, but in general bases appear with equal frequency in position x and y. Then if we considered only (1.5) we would conclude that the information content is quite low. In fact, the identity or, more generally, any linkage between columns is a clear pattern which we have overlooked. We refer to this as *mutual information*. We can quantify the mutual information between any two positions in a binding motif as

$$I(x; y) = H_x + H_y - H_{xy} \tag{1.7}$$

or, the decrease in entropy which we experience when we learn that we are examining positions x and y are within the same binding site. Schneider assumed that mutual information played no significant role in transcription factor binding, which is to say that correlations between positions in binding motifs have no biological relevance. We refer to this as the *hypothesis of positional independence*. Subsequently, (1.5) has most often been used as the definition of information content within a binding motif and (1.7) has been regarded as irrelevant.

2 Problem

We wish to explore the hypothesis of positional independence. The assumption of positional independence is fundamental to many techniques widely used in bioinformatics today. Hence, further justification, or

refutation, of this assumption would aid in providing a more sound theoretical basis on which to develop future methods.

There are several fundamental impediments to testing the hypothesis of positional independence. Perhaps the largest impediment is the lack of information concerning protein function. In general, the mechanisms by which proteins bind are not well understood. The scale at which this binding occurs only serves to exacerbate this problem.

Analytic methods of examining current experimental data also fall short. Mutual information, among other methods of detecting correlation, is often unreliable due to under-sampling. This is particularly unfortunate because many transcription factors are only known to bind to relatively few sites (with some notable exceptions like the cAMP receptor protein). As a consequence, even though we can detect sizable amounts of information content in almost all known binding motifs, we cannot know whether or not these correspond to true genetic signal or are artifacts of under-sampling.

3 Methods

In [1, 2] we investigate the evolution of information content within transcription factor binding motifs to assess the viability of positional independence. In our original work, we do not address the added complexity which arises from the necessity of the coevolution of the transcription factor and its binding motif. Rather, we evolve the binding motif and assume that the associated transcription factor is capable of recognizing all information content within the binding motif at all times. We continue our work here using a more sophisticated evolutionary model in which our artificial transcription factor must evolve in conjuntion with its binding motif to recognize patterns therein.

To circumvent problems associated with traditional approaches to testing positional independence, we have developed a software platform called ESTReMo. ESTReMo is an evolutionary simulation of transcriptional regulatory motifs. Though ESTReMo is capable of conducting a wide variety of simulations pertaining to the evolution of regulatory networks, we have used ESTReMo to simulate the co-evolution of a single artificial transcription factor and its binding sites. Using this approach, we can impose specific constraints on the behavior of our artificial transcription factor and examine changes in the corresponding binding motif. This helps us in two ways. First, we are able to observe if there is an evolutionary preference for encoding information in the form of $R_{sequence}$ or mutual information by examining which is actively used and at what levels. Secondly, we are able to prevent the artificial transcription factor from utilizing correlations between positions in its recognition strategy. This allows us to determine the level of mutual information that may appear artifactually in an under-sampled binding motif. As a consequence, we can compare known artifactual levels of mutual information to levels observed in evolved organisms. If we see an increase in mutual information in organisms which are capable of utilizing it, then it would suggest that mutual information is being actively used in those organisms.

3.1 The genetic algorithm

ESTReMo uses a genetic algorithm to simulate the evolution of artificial organisms. From a mathematical perspective, a genetic algorithm is an optimization technique which was inspired by Darwinian evolution [3]. In this light, the function by which we quantify the fitness of each organism is an objective function which is to be optimized.

There are three primary components of a genetic algorithm. Perhaps most importantly, a fitness function is used to quantify the fitness of each organism within the population. A selection strategy is then used to choose which of these organisms will produce offspring, and point mutations and cross-over events are applied to the offspring of the chosen parents. This cycle is repeated in successive iterations known as

generations. Ideally, each generation of the genetic algorithm produces successively fitter organisms until some stopping criterion is reached.

Though ESTReMo is capable of using a wide variety of different genetic algorithms, we will focus on a k-tournament selection strategy with full parental replacement. Parents are chosen through a series of tournament rounds. Each tournament round pits k (a positive integer) organisms against one another. The fittest of the k is chosen to produce offspring. A single organism may win multiple tournament rounds, and that organism will produce a single offspring for each tournament round which was won. Tournament rounds are conducted until enough offspring are produced to completely replace the parent generation. The population size remains constant throughout the course of a simulation. Larger values of k imply stronger selective pressures and may result in more rapid convergence of the genetic algorithm at the expense of decreased genetic variety within the population. Lower k values allow for broader exploration of the solution space, but may increase the number of generations required for convergence. Finally, we halt the genetic algorithm when an organism achieves a fitness of zero (which is optimal).

3.2 The artificial organism

The artificial organism used by ESTReMo has many interdependent components. To understand the purpose of each component, it is necessary to understand the criteria by which an organism is judged. Essentially, the goal of an organism is to properly regulate the expression of its genes. Its effectiveness at this task is quantified by a fitness function. The expression level of a particular gene within an organism is determined by the affinity that its *recognizer* (our artificial transcription factor) has for the *promoter region* which is upstream of that gene. A recognizer that has a stronger binding affinity to a site in a particular promoter region will result in the associated gene having a higher expression level. A recognizer that binds weakly to a promoter region will result in lower expression levels for the associated gene. The recognizer is encoded within a *coding region* in the genome of the organism. Point mutations are applied to coding and non-coding regions, allowing the recognizer and its binding sites to co-evolve. The recognizer is modeled as an artificial neural network (ANN). ANNs are only capable of evaluating vectors, so there is also a need for a *translator* which will map sequences in promoter regions to real-valued vectors to be evaluated by the recognizer.

3.2.1 Recognizer

The recognizer is a transcription factor model. Here we use an artificial neural network (ANN) as a recognizer. An ANN is a biologically inspired computational model which is capable of identifying complex patterns in data sets. For our purposes, an ANN is a function from $\mathbb{R}^n$ to $[0, 1]$. We map sequences of DNA to vectors in $\mathbb{R}^n$ and score them using this recognizer. The score which the recognizer assigns to a sequence is analogous to the binding affinity that the recognizer has for that sequence. Typically, a method known as *back-propagation* is used to optimize the weights which govern the behavior of an ANN. Back-propagation is not relevant to our application of ANNs. Within the context of ESTReMo, the weights of each ANN are optimized by the genetic algorithm.

There are two primary flavors of ANNs: the single-layer perceptron (SLP) and the multi-layer perceptron (MLP). The SLP is the simpler of the two. The SLP possesses an input layer and an output node. The input layer consists of nodes which accept input values. For the output node, the value of each input node is multiplied by a weight, summed, and passed through an *activation function*. Let $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ be a vector that we wish to evaluate using some SLP. Let $w_1, \ldots, w_n \in \mathbb{R}$ be the weights which define this SLP. Then the score which is assigned to x by this SLP is

$$\phi\left(\sum_{i=1}^{n} w_i x_i\right) \tag{3.1}$$

where ϕ is an activation function (typically a sigmoid function). Specifically, we are using

$$\phi(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

as the activation function for both SLPs and MLPs. The SLP model, as will be seen shortly, is to be used as a control. Most importantly, SLPs are only capable of identifying linear features in data sets, since it is not possible for an SLP to use correlations between input nodes to classify elements.

The MLP is capable of approximating any real-valued function, and, as such, can be used to identify complex patterns in data sets. The MLP functions similarly to an SLP. An MLP has an input and output layer, as well as at least one hidden layer which rests between the two. For each hidden node, the the value of each input node is multiplied by a weight, summed, and passed through the activation function, which determines the output value of the hidden node. For each hidden node in the layer which follows, the value of each hidden node in the previous layer is multiplied by a weight, summed, and passed through the activation function. The output of the activation function determines the output value of the hidden node. This process is repeated until the output layer is reached, where the value of the output node is determined similarly.

We will restrict our discussion to MLPs employing a single hidden layer. Again, let $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ be a vector that we wish to evaluate using some MLP. Let $w_{i,j} \in \mathbb{R}$ be the weight associated with the i-th input node (of which there are n) with respect to the j-th hidden node (of which there are m). Then the score assigned to x by this MLP is

$$\phi\left(\sum_{j=1}^{m} \phi\left(\sum_{i=1}^{n} w_{i,j} x_i\right)\right). \tag{3.3}$$

The MLP, in contrast to the SLP, is capable of exploiting correlations between input nodes. As a model for transcription factor binding, this means that the MLP can utilize mutual information when developing a recognition strategy.

3.2.2 Recognizer coding regions

The recognizer of an organism is encoded within a recognizer coding region in the genome of the organism. This recognizer coding region is an array of floating point numbers. Each of these numbers represents a weight in the recognizer. Subsequently, the size of a coding region is determined by the particular type of recognizer that is in use and the number of nodes that it possesses, because the length of the coding region must accommodate the number of weights which the recognizer requires. Mutations are applied to coding regions by adding a value sampled from a small uniform distribution to a value at some index in the array. While it would be possible to store coding regions as sequences of base-pairs, it would still be necessary to convert those base-pairs to rational numbers using some system. Such a system would be necessarily arbitrary, and the influence of mutations applied to base-pairs in coding regions would be more difficult to understand.

3.2.3 Promoter regions

Promoter regions are represented as character arrays containing nucleotide bases in addition to potential non-sequence data. All binding sites reside within promoter regions. The length of a binding site is determined by a window size, which is provided as a parameter to the simulation. The window size must be no larger than the length of the promoter region in which it resides, allowing for multiple (possibly overlapping) binding sites within a promoter region. Point mutations are applied to promoter regions by spontaneously changing a character from one nucleotide base to another at an arbitrary point in the array.

3.2.4 Translator

A promoter region may include other components in addition to sequence data. Furthermore, an ANN is not capable of evaluating nucleotide sequences directly. There must be an intermediary which translates sequence and non-sequence data into an array of values that can be evaluated by the recognizer. The translator fulfills this role. Each base in a nucleotide sequence is mapped to four values in the output sequences. The mapping is as follows: `A` $\rightarrow (0,0,0,1)$, `C` $\rightarrow (0,0,1,0)$, `G` $\rightarrow (0,1,0,0)$, and a `T` $\rightarrow (1,0,0,0)$. The vectors to which each base in a sequence are mapped are then concatenated. For example, the sequence `AATG` would map to $(0,0,0,1,0,0,0,1,1,0,0,0,0,1,0,0)$. Thus, a recognizer has four input nodes for each base to be evaluated. At this stage of development, no non-sequence data is considered, though in the future we may use curvature to complement the current sequence-based encoding.

3.2.5 Genomic background

Finally, the genome of each organism is not contiguous. Rather, the genome is divided into 3 segments: recognizer coding regions, promoter regions, and the genomic background. This occurs for several reasons. Perhaps most obviously, recognizer coding regions must be encoded separately because they are encoded using an array of floating point numbers rather than a character array. Additionally, we chose not to apply mutations to the genomic background. An initial concern was the possibility that any organism that wishes to evolve a binding motif which can be identified by some transcription factor model within an unconstrained genome, may find it easier to prune the background rather than develop a coherent recognition strategy. That is, it may be easier for such a system to classify binding sites correctly by degenerating the background rather than developing a realistic recognition strategy (e.g. everything but the promoter regions is mutated to an A, so the recognizer rejects anything containing an A). Of course, a real genome is subject to many constraints, so artificial constraints need to be imposed on the genomic background. Rather than creating arbitrary constraints that may or may not be realistic, no mutations are applied to the background whatsoever. This forces the recognizer to develop a more robust recognition strategy. To prevent memorization of the background by the recognizer (a possible outcome when forgoing mutation of the background), the genomic background is randomly sampled from a larger genomic sequence at each generation, presenting evolving organisms with a continuously changing, yet inherently consistent background.

3.3 Fitness evaluation

We quantify the fitness of each organism by comparing the expression level of each of the organism's genes against a set of predetermined targets. First we determine the *occupancy* of each promoter region, which is the likelihood that a unitary recognizer is bound there at any point in time. We then determine the *activation* of the promoter region, a function of the occupancy and the recognizer concentration which represents the extent to which the associated gene is up-regulated. The fitness of the organism is a function of the activation of the promoter region and a target value, which is provided as a parameter.

Occupancy represents the likelihood that a unitary recognizer is bound to a specific promoter region at some point in time. It is defined as

$$occ_{i,j} = \frac{out_{i,j}}{\sum_b out_b + \sum_k \sum_l out_{k,l} + G\tau} \tag{3.4}$$

where $out_{i,j}$ is the output of the recognizer in the i-th position of the j-th promoter region, out_b is the output of the recognizer at the b-th position in the genomic background, G is the length of the genomic background, and τ is a universal transit constant. Roughly speaking, we can consider the denominator here to be the unit time for the recognizer. The numerator can be considered the amount of time spent bound

to a specific position in a specific promoter. Hence, the ratio of the two is the probability that a single recognizer is bound at that position in that promoter at any given time. The activation of a promoter region is defined as

$$act_j = \rho \max_i(occ_{i,j}) \tag{3.5}$$

where ρ represents the number of recognizer units which are present in the organism. This is also provided as a parameter. In other words, activation is the maximum occupancy over the entire promoter region multiplied by the number of recognizer units in the organism. Finally, the fitness of the organism is defined as

$$f = \sum_j \phi_j \left(1 + cos\frac{\pi act_j}{t_j}\right) \tag{3.6}$$

where $\phi_j \in [0, 1]$ is a weight which defines the importance of the j-th gene to the survival of the organism (in the simulations described here, we have set $\phi_j = 1$ for all genes) and t_j denotes the target activation level of the j-th gene.

3.4 Experiment

Simulations were conducting using SLPs and MLPs with 2, 4, and 8 hidden nodes. The coding region mutation rate was 0.05 mutations per index per generation. The non-coding region mutation rate was 0.005 per base per generation. A k value of 2 was used. Population size was 1024. The genome of each organism consisted of 10,000 randomly generated base pairs sampled from a uniform distribution. Each organism contained 16 promoter regions, each consisting of 8 base pairs. The recognition window was also 8 base pairs long, so only a single site was able to evolve within the promoter region of each gene. Each experiment was run in quintuplicate. We recorded the information content of the first organism to achieve a fitness of zero (that is, the first organism to meet or exceed all of its target expression levels).

4 Results

The results of each simulation are recorded in Table 4.1. The population of organisms using a single-layer perceptron behaved roughly as anticipated. Using a genome size of 10,000 base pairs and a binding motif of 16 sites, we have $R_{frequency} = \log_2 \frac{10{,}000}{16} \approx 9.29$ bits. That is, we expect $R_{sequence}$ to be roughly equal to 9.29 bits in an organism which can correctly classify all of its binding sites. The mean $R_{sequence}$ using an SLP was 10.138 bits with a standard deviation of 0.554, a minimum of 9.518 bits and a maximum of 10.754 bits. Mutual information (which again must be purely artifactual as a SLP cannot utilize such information in its recognition strategy) was present with a mean of 1.241 bits and a standard deviation of 0.289.

In contrast to the SLP, the MLP is capable of utilizing correlation signal in its recognition strategy, and thus it is possible for correlation signal to be actively conserved. A clear trend of increasing mutual information content can be seen as the number of hidden nodes in the multi-layer perceptron increases.

N	$R_{sequence}$	*Mutual Information*
0	10.138 ($\pm$ 0.554)	1.241 ($\pm$ 0.289)
2	10.067 ($\pm$ 0.491)	1.483 ($\pm$ 0.452)
4	8.912 ($\pm$ 0.633)	2.224 ($\pm$ 1.332)
8	8.476 ($\pm$ 0.229)	2.670 ($\pm$ 0.145)

Table 4.1: $R_{sequence}$ and mutual information content in bits (standard deviation in parenthesis) of first organism to achieve ideal fitness using N hidden nodes. $N = 0$ denotes the use of a single-layer perceptron.

The population of organisms using a MLP with two hidden nodes did not diverge dramatically from the behavior of the population using the SLP. Mean $R_{sequence}$ was 10.067 bits with a standard deviation of 0.491 bits. Mean mutual information was 1.483 bits with a standard deviation of 0.452 bits. Notice that the mean $R_{sequence}$ still exceeds $R_{frequency}$ and mean mutual information is still within a standard deviation of levels seen using a SLP.

We see our first evidence that organisms may be utilizing the correlation signal when using a multilayer perceptron with four hidden nodes. Mean $R_{sequence}$ drops considerably to 8.912 bits with a standard deviation of 0.633 bits. Mutual information also increases noticeably to a mean of 2.224 bits with a standard deviation of 1.332 bits. Notice that $R_{sequence}$ now fails to meet $R_{frequency}$, despite the fact that all binding sites are correctly classified, suggesting that $R_{sequence}$ may be partially replaced by correlation signal. Standard deviations for mutual information and $R_{sequence}$ are their highest under four hidden nodes.

The behavior of organisms with an MLP recognizer using eight hidden nodes continues in a similar trend. Mutual information appears slightly higher than levels found in organisms with MLPs containing four hidden nodes. Mean $R_{sequence}$ is 8.476 bits with a standard deviation of 0.229 bits and mean mutual information is 2.670 bits with a standard deviation of 0.145 bits. The standard deviation is lower using eight hidden nodes than when using four hidden nodes. Mean $R_{sequence}$ is now considerably below $R_{frequency}$.

5 Conclusions and future work

The amount of mutual information in the binding motif of the first organism to achieve a fitness of zero within a population of organisms using a MLP recognizer appears to increase steadily with the number of hidden nodes used by the MLP. Furthermore, the mutual information content of the binding motifs of such organisms is notably higher than the artifactual levels found in those same organisms utilizing SLP recognizers. This suggests that organisms utilizing the MLP recognizer model make use of mutual information in their recognition strategies, resulting in the conservation of mutual information.

5.1 Improvements on results

Whether or not the mutual information found in said binding motifs remains stable beyond the stopping point of the aforementioned simulations is currently unknown. It is of great interest to us to determine whether or not the mutual information which is present at the conclusion of the experiments we have conducted is stable, and what the steady state for these experiments may be (assuming they will reach a steady state). We stop when an organism achieves zero fitness for two reasons. First, it allows us to compare our results to those of Schneider [6] who utilized the same stopping criterion. Secondly, we wanted these simulations to run for relatively few generations because the run-time was already quite long (the run-time of a single simulation using an MLP with 8 hidden nodes was slightly over 110 minutes, each of these were run in quintuplicate). To run these simulations until they achieve a steady state will very likely take a prohibitive amount of time. To allow us to conduct this research, we intend to first reduce run-time by parallelizing ESTReMo.

5.2 Parallelization

At present, there exist a variety of conditions under which the wall-clock run-time of an ESTReMo simulation is prohibitively long. Run-time appears to increase almost linearly with respect to genome size. If any mutation occurs in the recognizer coding region, then the behavior of the recognizer changes and every position in the genome must be re-evaluated for that organism. This occurs very frequently and for large genomes becomes computationally expensive. There are several options for parallelization that would ease this burden. Arguably the most powerful option would be to execute the simulation on a

computational cluster and divide the population over several nodes. It would also be possible to offload the most computationally intensive tasks onto a GPU. Research has been conducted on evaluating ANNs using GPU frameworks such as NVIDIA's CUDA [5].

6 Acknowledgments

Thanks are due to Dr. Matthias Gobbert of the Department of Mathematics and Statistics at UMBC and Dr. Ivan Erill of the Department of Biological Sciences at UMBC, who have served as research mentors on this project, to Joe Cornish with whom this work was partially collaborative as part of Interdisciplinary Training for Undergraduates in Biological and Mathematical Sciences (UBM@UMBC www.umbc.edu/ubm), and to Patrick O'Neill who has been a font of valuable ideas and constructive feedback.

The hardware used in the computational studies is part of the UMBC High Performance Computing Facility (HPCF). The facility is supported by the U.S. National Science Foundation through the MRI program (grant no. CNS–0821258) and the SCREMS program (grant no. DMS–0821311), with additional substantial support from the University of Maryland, Baltimore County (UMBC). See www.umbc.edu/hpcf for more information on HPCF and the projects using its resources.

References

[1] Joseph Cornish, Robert Forder, Ivan Erill, and Matthias K. Gobbert. Simulation of the evolution of information content in transcription factor binding sites using a parallelized genetic algorithm. Technical Report HPCF–2011–9, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2011.

[2] Robert Forder. A parallel simulation of the evolution of transcription factor binding sites. Technical Report HPCF–2011–1, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2011.

[3] J. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[4] F. Jacob and J. Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 3:318–356, 1961.

[5] C. Patulea, R. Peace, and J. Green. CUDA-accelerated genetic feedforward-ANN training for data mining. *Journal of Physics: Conference Series*, 256, 2010.

[6] T. D. Schneider. Evolution of biological information. *Nucleic Acids Research*, 28:2794–2799, 2000.

[7] T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht. Information content of binding sites on nucleotide sequences. *Journal of Molecular Biology*, 188:415–431, 1986.