Identifying the Relationship Between Evolutionary Distance and the Accuracy of *Cis*-Regulatory Module Predictions

Paulina Cueto

May 11, 2014

Abstract

Cis-Regulatory Modules (CRMs) are the portion of DNA that initiates gene expression. Gene expression is the process through which the body turns DNA into functions and cells within an organism. In this paper I build upon a program, MultiModule, created by Zhou and Wong (2007) that utilizes hidden Markov models and multiple sequence alignments to determine novel *Cis*-regulatory modules. I use the program to determine if there is a relationship between the evolutionary distance between species, and the ability to identify CRMs based on multiple alignments. The results indicated that there is a higher prediction rate between the closest species, and that the greater the variety in evolutionary distance the more precise the predictions are.

Contents

1 Introduction **3**
1.1 Biological Background 3
1.2 *Cis*-Regulatory Modules 3
1.3 Evolutionary Relationships 4
1.4 Computational Background 5

2 Identifying *Cis*-regulatory Modules **7**

3 Computational Approaches **7**
3.1 Phylogenetic Algorithms 8
3.2 Clustering Algorithms 10
3.3 Statistical Algorithms 10
3.4 Discriminative Algorithms 12
3.5 Experimental Methods 12

4 MultiModule **13**
4.1 Input 14
4.2 The Hidden Markov model 14
4.3 Coupling HMMs via Multiple Alignment 15
4.4 Evolutionary model 16
4.5 Basic Framework 17

5 Using MultiModule to Identify CRMs **18**
5.1 Methods 19
5.2 Results 20

6 Conclusion **25**

7 Acknowledgements **26**

Appendices **29**

A Running the Experiments **29**
A.1 Overview of Parameters 29
A.2 File Formats 29

1 Introduction

At the base of all organisms, there are portions of DNA that contain genes; these genes help each creature function and produce the necessary material to continue living. Genes must be regulated by another tool within the body. This tool, which will be discussed later in this paper, is a *cis*-regulatory module. At a high level, these modules are the on-and-off switches that dictate when a gene should be used.

1.1 Biological Background

Starting at the basics, we have the instructions for organsims, DNA. *DNA* (deoxyribonucleic acid) is the molecule that contains the instructions for building proteins, which control the biochemical processes in the human body. DNA is composed of four different nucleotide bases: adenine, guanine, thymine, and cytosine; the bases create a sequence that consists of A, G, T, and C, respectively. In the double stranded DNA, each letter bonds, or chemically forms attachments, with another. The result is *base pairs* of A bonding with T, and C bonding with G.

The sequence of letters that results from a consecutive read of base pairs, otherwise known as *genetic code*, is the key factor in being able to use computational power to identify problems encoded within DNA. Computational biology studies biological sequence alignments by reading in the series of letters and analyzing them as the programmer sees fit. Computationally predicting alignments in a major way computers can aid in the field of biology. In this thesis I will focus on identifying the sections of any given genetic sequence that regulate gene expression. *Gene expression* is the process by which the information in DNA is used as a blue-print for creating proteins. [**?**]. To *regulate* an expression is to control the manner in which it is expressed into proteins. Proteins are vital for most of cell functions and production [8]. Essentially, the portion of code I will be looking for controls when, and if, the instructions contained in a specific target gene become something useful for an organism.

1.2 *Cis*-Regulatory Modules

Knowing the location of these regulatory modules would aid in the research of transcriptional, and genomic, diseases. With the knowledge of where mutated genes are "turned on", then there is hope of finding a way to turn them off, if possible.I examine, and build upon, algorithms that are capable of identifying and accurately predicting the location of *cis*-regulatory modules on the genome. *Cis-Regulatory*

Modules (CRMs) refer to a section of the genetic code that initializes protein production, known as *gene transcription*. Gene transcription, is the process by which genetic information is copied from DNA to RNA (ribonucleic acid) in order to be used. RNA is important to an organism because most proteins and functions are started through RNA, rather than DNA. It is an organism's way of protecting the master copy of the body's instructions, rather than allowing the master copy of the instructions to be damaged in gene transcription.

Gene transcription begins with *transcription factors*, a special type of protein, binding to specific portions of DNA called *binding sites*. The interaction between the two create *transcription factor binding sites (TFBSs)*, which are the location where the expression of specific genes is started.

TFBSs tend to cluster and form CRMs *upstream* of a target gene, which means the CRM is somewhere before the gene it is supposed to start. The binding sites have similar characteristics called *motifs*. The motifs and the clustering are useful when trying to identify the location of the CRM because they create patterns within the genetic code that algorithms can find.

1.3 Evolutionary Relationships

The evolutionary relationships between closely related species play a large role in identifying CRMs without prior knowledge of where they reside in a genetic sequence. When two or more species are believed to have evolved from the same ancestor, it's called *common descent*. Common descent is important in this research because species with a more recent common ancestor have more similar genetic code than those who branched from the common ancestor at much earlier times in the evolutionary series. Those sequences that have similarities due to a recent common ancestor are referred to as *homologous sequences*. More specifically, there are *orthologous sequences* which are homologous sequences that have many similarities due to an evolutionary separation at the most recent ancestor. More simply, orthologous sequences come from the same evolutionary "parent" [5]. In order to be beneficial in finding the location of CRMs, the species must be similar enough to have orthologous sequences.

Through the genetic similarity we can see *evolutionary constraint*, or conservation of specific genetic sequences through the process of evolution. More importantly, evolutionary constraint implies that the conserved piece of code is important to a genetic sequence. This implication is founded with the idea that if there is code that is vital to survival, the organism will need to keep it in the DNA sequence and evolutionary conservation will keep the sequence in tact. On the other hand, *puri-*

fying selection is the deletion of characteristics that impact an organism negatively. Therefore, as we look back to evolution, the results should be that positive traits are maintained while those which impact an organism negatively are deleted.

Though similarities in genetic code are important, changes and differences within the code can be insightful as well. One important thing to note is that there are coding and non-coding portions of every genome. The use of the term *coding* in this context indicates that a portion of code is capable of gene expression. Similarly, non-coding indicates the inability of a specific sequence to be expressed. The *introns*, or coding portions of DNA, contain information that will be used during transcription. *Exons*, the non-coding portions, are the rest of the DNA that retain zero functionality within an organism. A CRM encompasses both the introns and exons. It holds both important genetic instructions, such as the binding sites, as well as background genetic sequence between the binding sites.

Though there are *mutations*, or changes in a genetic sequence that alter the composition of the DNA in a negative way, phylogenetic shadowing should be able to identify those portions of code and deal with them. *Phylogenetic shadowing*, which will be discussed in greater detail later, is the process of using evolutionary descent to identify significant similarities between species with common ancestors.

There are two major types of mutations: deletions and insertions. *Deletions* change a DNA sequence by eliminating nucleotide bases from the sequence. This can be detrimental if the deletion resides in a coding portion of DNA. *Insertions* insert new nucleotide bases into a sequence. The change is detrimental to an organism if it significantly changes the coding portion of DNA. Mutations are important to note because background, or non-coding DNA, is more likely to mutate through evolution because it's not being naturally selected for. This is important because one approach to identifying CRMs relies on the rate that species mutate over time.

1.4 Computational Background

I describe a few implementations commonly used in computational biology. The tools are are *position weight matrices*, *hidden Markov models*, and *sequence alignments*. Postition weight matrices (PWMs) summarize the frequency distribution of nucleotides for each motif, because motifs are very similar, but not identical on each genetic sequence. Motifs themselves are summarizations of TFBSs, and the PWMs create a more accurate visualization of the motifs. In other words, a PWM takes a specific window of a given sequence and determines how often a nucleotide base is shown at each location in the window. In Figure 1, the matrix on the right is an example of the frequencies of the genetic sequences on the left. PWMs are useful

because the ability to depict the most likely sequence for a given motif will prove helpful when identifying novel motifs given orthologous sequences.

$$\begin{array}{cccccccccc} A & G & T & C & A & A & G & T & C & A \\ A & C & G & C & A & G & G & T & C & A \\ A & G & T & T & A & A & T & T & G & A \\ C & T & A & C & G & A & C & T & G & A \\ A & T & T & A & A & C & G & T & C & A \end{array} \qquad \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{bmatrix} .8 & 0 & .2 & .2 & .8 & .6 & 0 & 0 & 0 & 1 \\ .2 & .2 & 0 & .6 & 0 & .2 & .2 & 0 & .6 & 0 \\ 0 & .4 & .2 & 0 & .2 & .2 & .6 & 0 & .4 & 0 \\ 0 & .4 & .6 & .2 & 0 & 0 & .2 & 1 & 0 & 0 \end{bmatrix}$$

Figure 1: (Left) An example set of sequences and (Right) the resulting matrix of the frequencies. The information in this diagram is not real DNA sequence, but rather an example used to explain PWMs.

Sequence alignment is simply the mapping of one DNA sequence onto another to identify the regions that are similar. The two different types of alignment are local and global. *Local alignment* allows gaps to appear between matches, or will return matches that only contain a sub-section of the total sequence. *Global alignment* searches for exact matches with fewer deletions, or mutations allowed between two sequences. Sometimes it's important to keep gaps out of matches - in heuristic algorithms it's easier to keep gaps for the overall understanding of an alignment, especially considering the possibility of evolutionary changes over time.

Markov models are a stochastic model describing a sequence of events.A *stochastic model* is a collection of random variables used to represent the progression of a sequence over time. A markov model is represented with states and transition probabilities. A *state* is simple the current location of a given random sequence, and the *transition probability* is the probability that the state will change, and what it will change to, depending on the model. In a general Markov model, the probability of each state is dependent on the previous event. However, for hidden Markov models (HMMs), the state is not directly visible, but output, dependent on the state, is visible. In biological application, HMMs are used to predict the composition of gene sequences. The HMM then contains states that represent the overall section of the sequence is currently in, and *emission probabilities* that indicate what nucleotide base is most likely at that location.

2 Identifying *Cis*-regulatory Modules

In this thesis I will focus on identifying novel CRMs without the use of a motif library. The difficulty in identifying CRMs is the lack of knowledge about where in the sequence they reside and what nucleotide composition is. Though it is known that the modules reside upstream of the gene they will regulate, that location may be within 10 base pairs or 10,000 base pairs of the target gene. In addition, when looking through a sequence, there isn't a set length that CRMs can be; they can vary from 5 to 200 base pairs.

Transcription factors aid in the identification of the modules because they form clusters as they interact with binding sites upstream of the target gene. However, in order to identify the transcription factors and the respective binding sites, the location of the binding site on the sequence must be known. Only a small number of the motifs of binding sites have been experimentally proven. Therefore, there is not a complete collection of known sites, which can lead to motifs being overlooked during an alignment process.

Identifying CRMs through biological experiments is the most accurate way of determining the location of a CRM. However, the process is a long and tedious approach that doesn't give results in a reasonable amount of time. Computational biology offers a solution with the ability to analyze sequences of DNA in shorter amounts of time. There are alogorithms capable of predicting CRMs by manipulating and aligning gene sequences to identify the most useful information within each. Some algorithms compare evolutionary relationships of multiple species in order to locate where potential CRM locations are. Other approaches to solving the problem will be discussed in the following section. In this thesis I look into the relationship between evolutionary distance between species and the accuracy of CRM predcitions.

3 Computational Approaches

Predicting CRM's isn't a straightforward alignment problem, but there are computational approaches that use the biological patterns within the composition of a CRM to help identify the location of CRMs on a genome. The four general categories of algorithms that predict CRMs are: statistical, clustering, phylogenetic, and discriminative.

Binding site locations on a sequence create a pattern that can be identified by multiple sequence alignment. The basis of statistical and discriminative algorithms is the inherent pattern of motifs that make up CRMs because those approaches use those patterns found within many sequences to create mathematical models to

emulate the patterns. Those models are used to predict the most probable CRM composition at given a new input sequence. The underlying idea of phylogenetic and clustering algorithms is using evolutionary relationships to isolate similarities between homologous sequences. Closing the amount of space in which an algorithm must search for CRMs on a sequence increases the chances of finding a CRM. Most of the current algorithms, however, use a motif library to find the CRMs on a sequence. Algorithms are more confident in predictions when a motif library is used, because it's easier to make a prediction when the motifs are known. However, not all binding sites or CRMs have been identified through experimental means, which implies that the programs that rely on the databanks of motif information would not be able to identify CRMs composed of yet-to-be identified motifs. In addition, using only known information doesn't allow for identifying new, unique CRM predictions.

In order to more accurately compare the different approaches, and determine what is the most effective, there are measurements of accuracy *sensitivity*(SN) and *specificity*(SP) which are most often used to compare results. The equations below show how to calculate the sensitivity and specificity of an algorithm, given the expected and actual results. Sensitivity is proportional to the true positive rate (TP) over the sum of the true positive rate and the false negative rate (FN). It indicates how many true CRMs are found from all the annotated CRMs by the program being tested. Specificity depends on the true negative (TN) rate over the sum of the true negative and false positive rates, indicating how many true exons (the portion of DNA that is used to express genes) regions are found from the dataset [14]. Both specificity and sensitivity will be used as an indication of how well a program fared through experiments [10].

$$SN = TP/(TP + FN)$$

$$SP = TN/(TN + FP)$$

3.1 Phylogenetic Algorithms

When considering the biological aspects of predicting CRMs, the high level biology can be abstracted, for computational purposes, to a string that contains the characters A, C, T, and G. Using a string alignment is advantageous because it's a more studied problem in the the computer science field. Instead of trying to indentify the chemical interactions between binding sites and transcription factors, an algorithm will take in multiple strings and find the most similar alignment, or match, between all the strings.

Phylogenetic shadowing is a method that takes advantage of string comparison to make genomic comparisons. It is the process of identifying evolutionary constraint between multiple species and highlighting the regions of code that have been constrained throughout evolutionary descent. The results help identify the most important aspects of a genetic codes. A section of code is "important" if it has survived natural selection, and stayed the same through evolution. Therefore, being able to identify the constrained portions of genetic code is extremely beneficial in identifying CRMs, because they are necessary for gene functionality in organisms [5].

One potential issue that has been recognized with this approach is that comparisons only work well where there is strong purifying selection on the CRMs. Therefore, genetic sequences have a chance of being similar simply because there hasn't been enough time since the last evolutionary split. The possibility must be taken into account when determining similarities between DNA, otherwise the results would be too ambiguous. One side effect of comparative approaches is that they must have more than one species' sequence as input, they are by definition not able to predict CRM locations given only one sequence. A productive aspect of this approach is that it is not dependent on a motif library, because it relies on the evidence of purifying selection to recognize new motifs.

When there is evidence of evolutionary constraint, there is much more confidence in the results containing a CRM. Frazer et al. [5] implements a multi-step process to accuractly define homologous sequences and take advantage of the information an evolutionary relationship offers. The high-level description of the process is identifying species with a common ancestor, annotating the sequences being compared for known patterns, and aligning them again for most accurate results. Through this process, an algorithm can refine it's own understanding of what the CRM will look like and check against the input sequences to make sure it's accurate.

An advantage of phylogenetic shadowing is that the information gathered from an alignment can then be used with other methods of computation. One example is the program written by Zhou and Wong (2007) - MultiModule, which uses phylogenetic shadowing to identify motifs, then uses hidden markov models and create a more fundamental understanding of the composition of a CRM. Combining the comparitive technique with other methods is beneficial because knowing what has been conserved over time alone doesn't confidently indicate the portions of code that are definitely CRMs. The practicality of this approach has allowed it to be a part of many different algorithms, such as CisPlusFinder [10] and MorphMS [13], which will be discussed in detail later. However, using only genomic comparisons doesn't allow for the precision to identify exact locations of CRMs. The lack of precision is due to the similarity that genetic code can have with species similar to it. It is much more effective to

apply the phylogenetic approach with another algorithm to refine the search, such as MultiModule which will be discussed later.

3.2 Clustering Algorithms

Motifs appear multiple times throughout a sequence, but tend to be concentrated in areas that initiate gene expression. The clustering approach takes advantage of motif patterns and identifies clusters of high density binding sites on a sequence. Identifying clusters of motifs can used with single genomic approaches, or through multiple sequences. Motifs are identified using a library of known motifs, if enough motifs are identified in one area then it's flagged as a potential CRM location. Therefore, looking for significant clusters of transcription factor binding sites means a greater dependence on the motif libraries mentioned before. The dependency is due to the need to verify that the predictions are correct. In single sequence identification, where there is only one input sequence, there is not another means of determining whether a result is or is not a motif.

High density indicates that binding sites are in a close proximity to each other and within a specified window. The window used to look for these clusters is generally specified as a parameter at the beginning of a run. As mentioned, a CRMs length can't be known before it is identified. Without knowing the length of the CRM, there is potential of excluding CRMs by choosing a length that is too small. However, because algorithms can run with varying lengths of windows, the variety of CRMs predicted can adjust as well.

CisPlusFinder is an effective clustering algorithm that uses clustering to identify perfect local ungapped sequences (PLUSs) to identify CRMs. *PLUSs* are sequences that are perfectly conserved in multiple genomes and contain multiple motif similarities. The algorithm identifies such sequences and labels them as clusters, which in turn indicates CRMs. This approach gives it a 99 percent sensitivity if the motifs are known, but only 56 percent if the motifs must be determined at the same time [10]. One problem already identified with this approach is the dependence on previous knowledge of TFBS in order to effectively use an algorithm with this approach.

3.3 Statistical Algorithms

Probability theory, or the branch of mathematics that deals with the analysis of random phenomena, is the bases of *statistical algorithms* [4]. Genetic sequences are a series four nucleotide bases A, C, T and G, and statistical programs use theory to predict which nucleotides will occur at any given position within a sequence. Of

the multiple methods of creating a theoretical model, the most prevalent is hidden markov models (HMMs) which will be discussed more later in the paper. Markov models emulate an unknown sequence and determine the probabilities that a given position in a sequence is part of a CRM.

Statistical approaches use of theoretical models help give an example CRM given specific known binding sites. They model a CRM sequence as being generated by a combination of a set of binding sites [14]. Therefore, when given input sequences the algorithm will be able to compare the actual with the theoretical and determine where the CRM on the actual sequence should reside. Then, generally, the next step will be to compare the results with a motif or CRM library in order to eliminate incorrect results, and validate the correct predictions. Though checking with a known library can increase the accuracy of an algorithm, it isn't effective if the goal is to learn where new CRMs are located. The goal is to remove the necessity of the biological experiment aspect of identifying CRMs and still have a high sensitivity rate.

One good example of a statistical algorithm is MorphMS, [13] which combines sequence alignments with hidden markov models to identify the most likely location of CRMs.

The input is two orthologous sequences that are aligned by probabilistically summing over all possible alignments by their matches to the potential binding sites given by the user. An additional input parameter are the transcription factor binding site motifs of interest [14]. The program MorphMS has a sensitivity rate between 60-90 percent depending on the parameters defined within a given run [13]. This is an extremely good rate, but the problem lie with the use of a motif library. The attachment to a library defeats the purpose of being able to identify CRMs in a purely computational way.

Another example that doesn't require the use of predetermined motif libraries is MultiModule. It has motif and CRM prediction in one step. More discussion will be given to the different models, as well as more specifics on MultiModule later. Generally, the algorithm has three defining steps: updating motifs, updating alignment, and sampling the product motifs through a dynamic program [15].

There are many different sub-approaches to the overall statistical approach, however the main point to gather from these techniques is the ability to take in orthologous sequences and create models that will help predict CRMs based off the probability of sequence location.

3.4 Discriminative Algorithms

Discriminative algorithms are based on creating two separate models for the CRM and non-CRM sequences in order to precisely identify where CRMs are located. Understanding the two models aids in determining if a sequence is a CRM, and to eliminate other portions of code that are definitely not CRMs. Instead of only outlining the structure of a CRM, discriminative takes into account sequences that will definitely not be a CRM.

For example, HexDiff, an algorithm than uses hexamers (a molecule consisting of 6 subunits) to identify CRMs, seeks to solve the discriminative motifs problem. The solution presented is to find PWMs (position weight matrices) that are present in one sequence and absent in another, indicating a positive and negative set respectively. PWMs illustrate the frequency of nucleotide bases within a transcription factor binding site, which is helpful in determining the probability that a nucleotide will be at that position. HexDiff produces relatively good results with a sensitivity rate of 69.23 % [2].

The downside to this algorithm, that may effect the sensitivity rate in a negative way, is that PWMs have maximum magnitude that is searchable. Essentially cutting out some of the possible CRMs by not taking into account larger portions of code. This characteristic is similar to the window used in the clustering algorithm. Though it's helpful, it is still hindered by the inability to know the length of any given CRM.

Determining motifs through the use of PWMs is an important aspect of the MultiModule program previously alluded to. Using PWMs decreases the dependency on a motif library as the programs identifies new motifs and binding sites as it runs. Discriminative modeling creates a cleaner difference between background nucleotides and nucleotides that are part of the regulatory module. The key with this modeling is being able to accurately discriminate between background sequence and functional sequence [12].

3.5 Experimental Methods

The most accurate method of identifying CRMs is not a computationally motivated approach. Though there are many computational approaches to solving the problem of identifying CRMs, there is an experimental method that is highly accurate, and allows biologists to identify TFBS and read short sequences in parallel. The non-computational approach that is important to mention uses epigenetic marks, where *epigenetics* is the study of heritable changes in gene activity which are not caused by changes in the DNA sequence. The method looks at any changes that are visible, but not DNA related [1]. This is useful in biological experimentation because epigenetic

changes can affect the activation of certain genes, which is helpful in determining if a portion of sequence is a CRM.

A more specific example is *chromatin immunoprecipitation* followed by high-throughput sequencing [8]. Broken down, what this means is individually identifying transcription factors and isolating the DNA portions bound to it. Then having multiple parallel short-sequence reads that then align the reads into a sequence.

The use of more biochemical signals is most beneficial in post-processing CRM predictions and in confirming novel CRMs, otherwise this route is still the biological experimental route and would not be saving time. This paper does not look into this method as it is based on the more biological aspect of CRM predictions. However, it may be useful closer to the end of the research in order to have more accurate validation rates.

4 MultiModule

I will be using the aforemenotioned program, MultiModule, as the basis of my thesis. The program combines the statsitical and phylogenetic approaches mentioned in the previous section. The layout of the algorithm is as follows: multiple species alignment produces results that are checked against a theoretical model, then updated to produce more accurate predictions. It is a self-correcting method that helps to more accurately identify where it has ambiguities or errors, and decreases the magnitude of error before the final results.

Three main aspects of MultiModule are its use of: hidden markov models (HMM), position-weight matrices, and multiple species alignment. It uses HMMs to capture the module structure in each species and coupling the HMMs through multiple species alignments [15], where *coupling* implies the use of multiple species in order to identify the most similar motifs accross multiple species' sequences, whereas position-specific weight matrices take transcription factor site motifs into account during the alignment process.

The portion of the algorithm that initally aligns sequences seeks to enhance the performance of new motif discovery because it is able to use information from the spatial correlation of TFBSs of the input sequences. Using multiple genomes provides the algorithm with more details about the evolutionary constraint between the species. Using these methods in combination help increase the accuracy of MultiModule's predictions of novel CRMs.

4.1 Input

The input data of MultiModule consists of n (co-regulated) sequences from N species, giving a total of $n \times N$ sequences. For the purposes of our initial experiments, the number of genes will be $1(n = 1)$, giving $1 \times N$ sequences.

The algorithm is based on the assumption that CRMs are composed of multiple binding sites and the corresponding transcription factors. It only considers closely related species where orthologous sequences have the same patter of motifs and factors. Therefore, the user must choose a number of motifs K, for the algorithm to identify in the initial stages. The goal is to identify the motifs, and therefore create PWMs based on those found within the sequences.

4.2 The Hidden Markov model

In this section, I will detail the markov model that is created through MultiModule. The model is designed to be able to predict a CRM composition based on the assumption that any given sequence is composed of sequence containing modules which is split intermittenly by background sequence. Again, within a module the multiple TFBSs are separated by background nucleotides as well. The overall representation of the HMM is two states: module (M) or background (B), which both have emission probabilities of whether the current nucleotide is a background nucleotide, or part of a motif. Therefore, the HMM in MultiModule is outputting both the path between the different states, as well as the sequence of emissions that predicts the most likely sequence of a CRM.

Given states in a markov model, there must be the probability that a nucleotide is going to transition from one state to another. Those probabilities are given in a transition matrix (T), which contains the probabilities of going from a background state to a module state is r, and module to a background state is t, assuming the expected length of a module is $1/t$. In addition, the probability of staying within a module is $1 - t$, while staying within the background state is $1 - r$.

$$T = \begin{bmatrix} T(B,B) & T(B,M) \\ T(M,B) & T(M,M) \end{bmatrix} = \begin{bmatrix} 1-r & r \\ t & 1-t \end{bmatrix}$$

The module can be further broken down to $K + 1$ states corresponding to background (M_0) and K motif sites ($M_1 - M_K$), that is the set of substates within a module $M : M = M_0, M_1, ...M_K$. Each state, assuming that the width of a motif is w_k, is the piece of sequence of length w_k containing that motif.

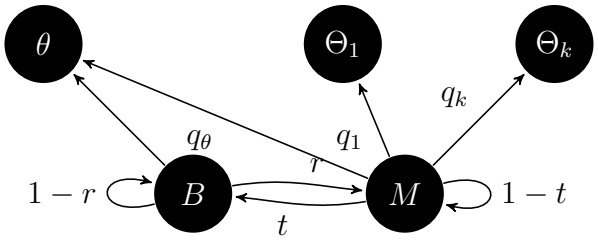


Figure 2: The HMM that represents the transitions from module (M) and background (B). The transitions shown in the figure are correlated to the transition matrix T mentioned before.

The module-background composition of a CRM gives a basis for a Markov model with two states: a module state (M), or a background state (B). In a module state, the HMM will either emit a nucleotide from the background θ_0 or a binding site of one of the K motifs, or PWMs $(\Theta_1, \Theta_2 ... \Theta_K)$ The probability that a module state emits θ, a background nucleotide, is q_θ, or $K(k = 1, 2...K)$ a binding site of one of the K motifs is q_k. The transition are shown by the Markov Model in Figure 2.

4.3 Coupling HMMs via Multiple Alignment

The HMMs different orthologs are coupled through multiple alignment, so that the hidden states that are aligned are combined into one common state. This means that if there are similarities between the orthologous sequences they are depicted all as one node.

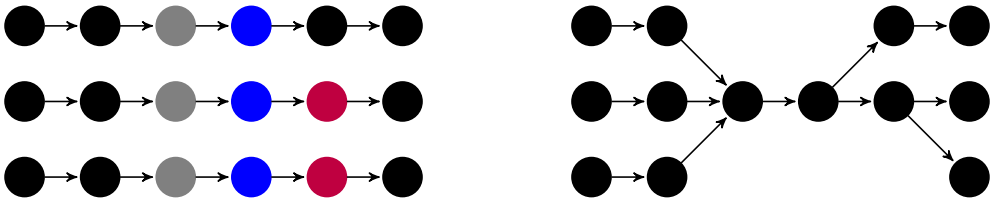


Figure 3: The figures above show a coupling a Hidden Markov model with multiple alignment. (Left) The alignment and (Right) the resulting graphical representation of similar motifs .

The graphical representation of coupled-hidden Markov models (c-HMMs) as

shown in figure 3 illustrate each hidden state as a node in a graph with arrows indicating the dependence between them. The nodes with the same color indicate that the nucleotides emitted from that state are aligned, therefore, they are collapsed into one node in the representation on the right of figure 3.

For nodes with only one parent, the transition probabilities are defined by the same transition matrix T. Otherwise, if there are multiple states leading into a collapsed node, then we say node Y with m parents, each in state $Y_i (i = 1, 2, ..., m)$, then the probability is

$$P(Y|Y_1, ..., Y_m) = \frac{1}{m} \sum_{i=1}^{m} P(Y|Y_i) = \frac{C_B}{m} T(B, Y) + \frac{C_M}{m} T(M, Y)$$

where C_B and C_M are the numbers of the parents in states B and M , respectively. The transition probability to a node with multiple parents is defined as the weighted average from the parent nodes in both states. The c-HMM first emits the ancestral nucleotides, and then different models are used for the evolution from the ancestral to descendant nucleotides depending on which state they are in.

4.4 Evolutionary model

A neutral substitution matrix, which describes the process in which a sequence of characters changes to another set of traits, is used for the evolution of aligned background nucleotides, with a transition rate of α and a transversion rate of β [15]. *Transversion* is substitution of nucleotides through mutations.

$$\phi = \begin{bmatrix} 1-\mu_b & \beta & \alpha & \beta \\ \beta & 1-\mu_b & \beta & \alpha \\ \alpha & \beta & 1-\mu_b & \beta \\ \beta & \alpha & \beta & 1-\mu_b \end{bmatrix}$$

The rows and columns of the matrix are ordered as A, C, G, T and $\mu_b = \alpha + \frac{2}{3}$ is defined for the background mutation rate. Each position in the matrix is assumed to evolve independently of the others. Ancestral nucleotides are denoted by Z, and are assumed to follow a distribution of the probability (weight) vector on A, C, G, T. The probability that a corresponding X, nucleotide from a descendent species, inherits Z is μ_f, while the probability that it is independently generated from the same weight vector is $1 - \mu_f.\mu_f$ is the mutation rate of the TFBSs within a motif,

which is identical for all positions in the matrix. If X inherits Z, then $X = Z$, otherwise$X$ is independent of Z.

4.5 Basic Framework

Putting the different pieces of the program toether, the full model involves: a transition matrix T, the emission probabilities $q_0, q_1, ..., q_K$, the motif widths $w, w_1, ..., w_K$, the PWMs $0, 1, ..., K$, the evolutionary parameters α, μ_f, and β, the background models for ancestral nucleotides and the current species. The parameters of the program are the number of motifs assumed within the CRM, (K),the expected module length L, with the transition probability fixed$t = \frac{1}{L}$ in T, the number of species being compared, the number of sequences, and a few others that will be described in the appendix.

The length and composition of a CRM is unknown, therefore MultiModule uses *Independent Poisson priors*, which are probability distributions that express the probability of a given number of events throughout a fixed interval of space to help identify the width of a CRM because it can be a variety of widths. *Flat dirichlet distributions* are used as priors for all other parameters, because given an uncertain parameter, distributions express uncertainty about that parameter before any data is taken into account.

MultiModule is a *Gibbs Sampler* that samples from the joint posterior distribution of all unknown parameters and missing data. A gibbs sampler is a Markov chain Monte Carlo (MCMC) algorithm for obtaining a sequence of observations which are approximated from the joint probability distribution of two or more random variables, when direct sampling is difficult [7]. In order to accomplish this, however, A Metropolis-Hasting step is added to the method to take uncertainty in multiple alignment, the step dynamically updates the alignment according to the current sampled parameters[15].

Initially, MultiModule take groups of orthologous sequences and builds an initial alignment of each orthologous group by a standard HMM-based multiple alignment algorithm. Each iteration of the program thereafter is composed of three main steps:

1. Given the alignments and all other missing data (A, Y, Z and V) Where A is the multiple alignments of the input sequences; Y represents hidden states' in the model, which indicate whether observed nucleotides are in a background state or module state; Z denotes the ancestral sequence; V denotes the evolutionary bonds of aligned TFBSs, update the motif widths and other parameters by their conditional posterior distributions.

2. Given the current parameters with probability u, update the alignments of each ortholog group.

3. Given the current alignment and parameters, use the Metropolis-Hastings algorithm as a dynamic programming approach to sample module and motif locations, ancestral sequences and evolutionary bonds.

More simply, once the alignments have been calculated, and the motifs have been identified, MultilModule will then go over the data and refine the parameters to fit the information found from the alignments. Once it has accomplished this section, it then uses dynamic programming to decrease the time it takes to sample where the module and motif locations would be identified. It goes through this process as many times as the user would like it to.

Motif and module predictions are based off of marginal posterior distributions constructed by the samples generated by MultiModule after around 50 percent of the iterations. The length of each motif is estimated by its rounded posterior mean. The following posterior probabilities are recorded: 1) P_k, the probability that the positions is within a motif site; 2) P_m, the probability that the position is within a module; 3) P_a, the probability that the position is aligned.

5 Using MultiModule to Identify CRMs

Sequence alignment is a large part of determining where motifs reside on orthologous sequences. As mentioned before, it can either be local alignment or global alignment. In either case, the relationship between the sequences being compared may have a significant effect on how accurate the motif predictions are. In addition, when using more than 2 alignments, the order in which sequences are compared may also play a role in predictions. Motif identification is one of the fundamental steps in MultiModule. If MultiModule is based on a local heuristic alignment tool, then much can be gained to switching to a more powerful exact algorithm to generate PWMs so that motif prediction can improve.

My focus is on the relationship between evolutionary relationships of the species and the number of CRMs accurately predicted. Phylogeny plays a large part because the evolutionary distance effects how similar two sequences are. If they are too similar, there may not have been enough time for the background to mutate and the different species to a dinstinct pattern of nucleotide bases. This is significant because if the program cannot determine the variation between sequences, then it will not recognize any motifs, or the motifs recognized may to too ambiguous. Increasing the

accuracy in determining similarities between multiple sequences would increase the overall success of the program as it would affect both the construction of PWMs as well as the HMM that is used to predict the initial alignment.

I'll be looking into whether breadth or depth matters more on the phylogenetic tree. Especially because the program's input is specifically groups of orthologous sequences. Also, I will be analyzing the results in order to discern a pattern throughout the predictions MultiModule outputs. In order to determine the best orthologous sequences to match when looking for CRMs, I was able to use the source code of MultiModule [15]. I have set up a series of MultiModule runs to gather the output information and analyze patterns between slightly different results. The outputs contain the motif predictions of many different runs. The goal is to look at all of the data and determine the most accurate CRM prediction and use the parameters given during those runs in order to most accurately set up MultiModule runs for later experiments and research. Using a variety of evolutionary relationships, we are trying to determine the best phylogenetic relationship in order to most accurately predict the region of the module.

5.1 Methods

The main goal of the experiment is to find the relationship between the accuracy of CRM prediction and phylogenetic relationship between input species. MultiModule requires the user to know how many motifs are in the CRM being located. However, in order to bypass this requirement and still gather the most accurate results, the value K (number of assumed motifs) varied from $1 \leq K \leq 20$ with the intention of catching all the possible predictions. By taking into account the predictions assuming each value of K, I can identify the most accurate location. In each of the different experiments the value of K changing each run, starting at 1 and increasing to 20. To distinguish the most beneficial evolutionary relationship, the experiment has a variety of input sequences with varying degrees of evolutionary relationships. In order to recognize the accuracy in relation to the evolutionary distance, I compared the orthologous sequences of the most similar species and then continued adding species in the comparison in order of decreasing similarity. To generate genomic variety, I chose to predict the CRM of two genes contained in two different species.

The first gene is the yellow (y) gene in Drosophila. I chose the yellow gene in Drosophila because each species contains the gene and the location of the gene is known through Flybase.org. Knowing where the gene is on each sequence narrows the search space I would need to scan over. For testing purposes, it's much easier to test the accuracy of the program with a well known gene. In addition, it is much

easier to identify sequences within the Drosophila genus rather than a mammalian genus because the genetic make-up is much shorter, and would therefore take less time. Another factor of choosing Drosophila was the great diversity of the insect. The generational rate of Drosophila is so much shorter compared to mammals that the similarity between two Drosophila species is farther away than the similarity between a human and a mouse. Therefore, we are able to see how the program will work with a variety of differences. The similarity between Drosophila melanogaster and 10 other species is shown below. For instance, D. *melanogaster* and D. *erecta* are most similar, while D. *melanogaster* and D. *persimilis* are least similar [3]. Theoretically, through the experiments we will be able to determine at what level of phylogenetics similarity will we be able to find accurate motif predictions.

The other gene is the eat-4 gene in Caenorhabditis. I chose the eat-4 gene because the species is well studied by Professsor Phil Meneely who has provided sequences from C. *elegans*, C. *remanei*, C. *briggsae*, C. *brenneri*, and C. *japonica*. The evolutionary relationship between the species is as listed with C. *elegans* and C. *remanei* most similar. As with the Drosophila genome, the Caenorhabditis genome is much simpler than a mammalian genus, and is more well-studied so the information of known CRMs is available to check the accuracy of the predictions.

The predictions given by MultiModule are nicely organized and formatted so that a user could potentially look at only the motifs identified or location of the predicted CRMs. I took all of the files containing CRM predictions and gathered the predictions located on the D. melanogaster and the C. elegans input sequences. In order to keep track of the different predictions, I initialized an array to zeros to the length of the sequence search space, then increased the values at each index of the predictions. For example, given a location prediction of $1 - 4$, the array would then contain $[0, 1, 1, 1, 1...]$. In doing this, a histogram of the location frequencies was created. The histograms are represented in the figures 5 and 4. For a more detailed description of how the information was gathered and processed, please see Appendix A.

5.2 Results

Based on the results for Table 2 we can see that the higher the K value, the fewer motifs found between the species. In addition, the evolutionary relationship between the species is shown in table 1 and indicates that the closer the species are, the harder it is to accurately identify the CRM. I hypothesize this because the results of the experiment indicate that those in the "middle" range of distance display more motifs.

Species	Orthologous Regions (kbp)	Evolutionary Relationship (% similarity to D. melanogaster)
D. erecta	197,900 - 199,900	86.4%
D. virilis	3,895,500 - 3,897,500	82.7%
D. yakuba	182,500 - 184,500	81.3%
D. grimshawi	2,749,00 - 2,751,00	81.3%
D. sechellia	114,000 - 116,000	81.2%
D. mojavensis	2,470,000 - 2,472,000	80.8%
D. simulans	8,000 - 10,000	80%
D. willistoni	5,315,500 - 5,317,500	78.8%
D. psuedoobscura	4,237,500 - 4,239,500	78.2%
D. persimilis	1,177,000 - 1,179,00	72.6%

Table 1: The table indicates the orthologous regions of Drosophila sequences that were used in the experiment, and the evolutionary relationship with relation to D. *melanogaster*

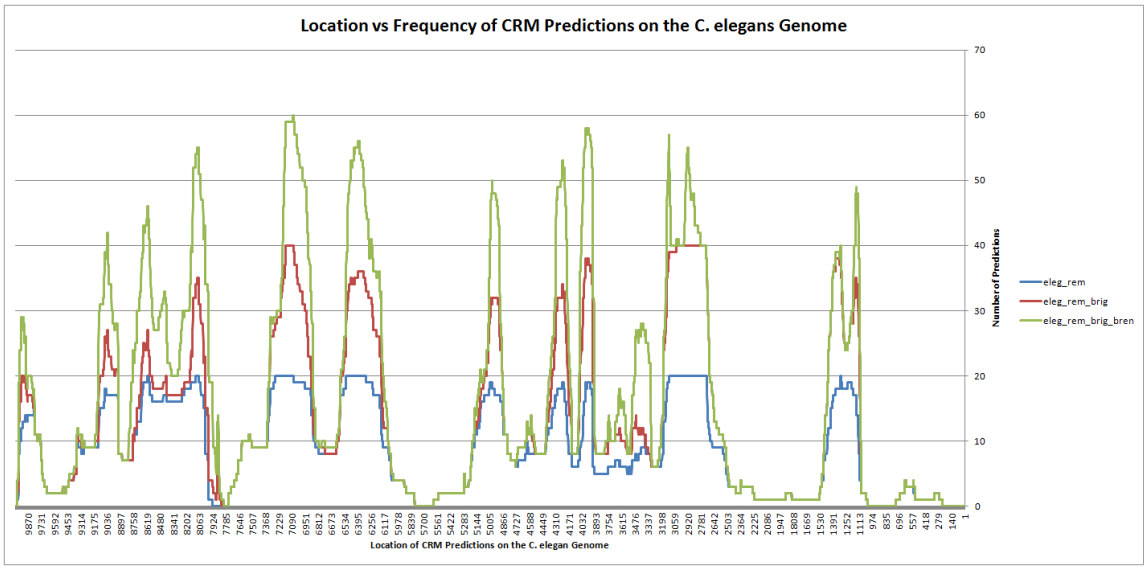
Species	Orthologous Regions (kbp)
C. elegans	9,126,916 - 9,136,916
C. remanei	2,043,034 - 2,049,386
C. briggsae	10,632,507 - 10,646,082
C. brenneri	66,909 - 73,867
C. japonica	9,811 - 13,301

Table 2: The above table show the orthologous regions that were compared in the experiments. In addition, the species are in order of evolutionary similarity. C. *elegans* and C. *remanei* are the most similar with C. *japonica* being the least similar.

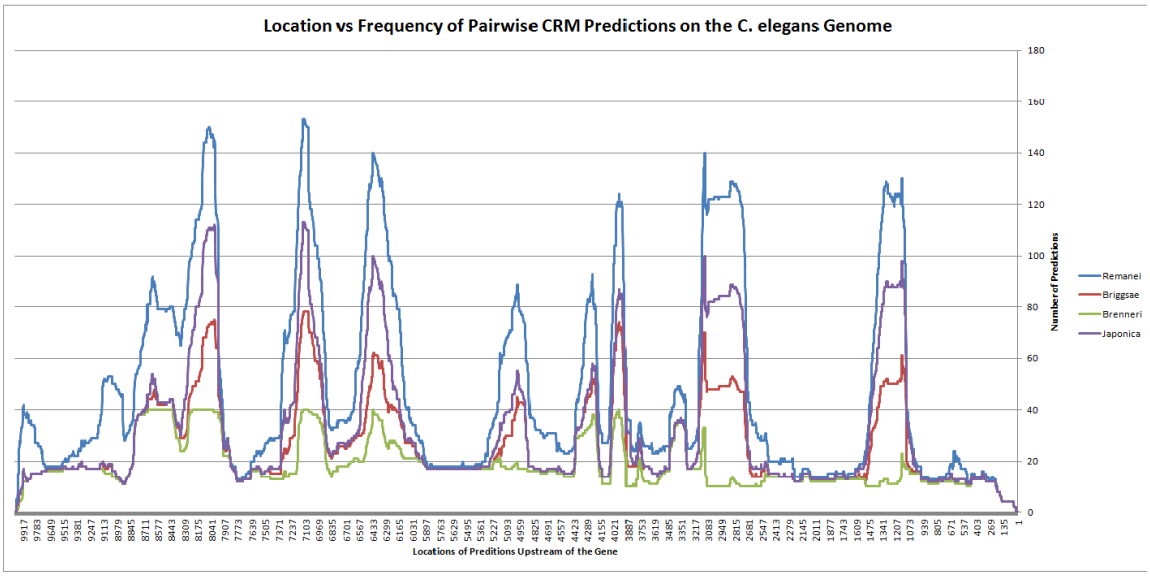
When comparing the D. *melanogaster* sequence, the files contained melanogaster and one other evolutionary related species' sequence, creating an overall format of two species with one sequence each. The same format was followed for the C. *elegans* comparisons.

The results of the pairwise comparisons indicate that the closer two species are related the higher the prediction count over the entire sequence. However, even though there are a greater quantity of predictions from MultiModule [15], it doesn't improve the accuracy of the predictions. This can be seen by investigating results of the evolutionary comparisons. When adding species one at a time into the input sequences, the predictions are decreased where there are no CRMs and increased in locations that have a higher probability of being a CRM.

Based on the graphs above, it can be concluded that there is a higher level of accuracy with the CRM predictions when MultiModule is given multiple orthologous sequences to align. It can be seen as the comparison between multiple species yield much sharper peaks, and therefore more precise predictions of CRM locations.

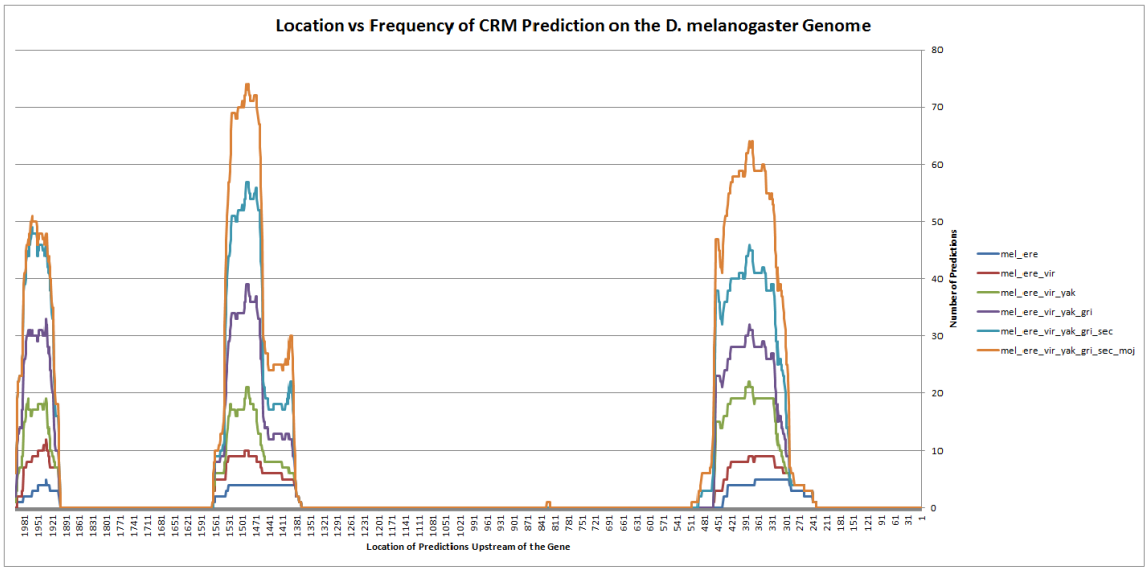


(a) The evolutionary comparison between C. *elegans* and other Caenorhabditis species. Each line indicates that another species that is farther away from C. elegans has been added to the comparison.The abbreviations shown on the graph above are as follows: rem - remanei; eleg - elegans; brig - briggsae; bren - brenneri.

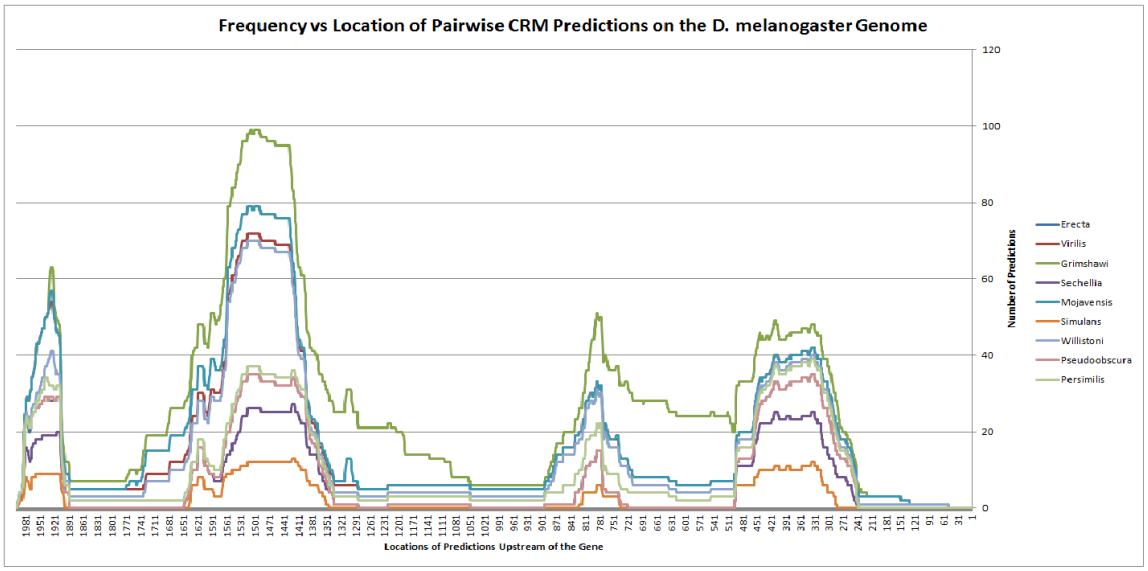


(b) The pairwise comparisons between C. *elegans* and other Caenorhabditis species. Each line indicates a unique pairwise compairison. In the legend, the species are shown in a descending order that is correlated to how closely related the species are to C. *elegans*.

Figure 4: The two figures above illustrate CRM predictions comparisons between C. elegans and related species 4a and 4b.



(a) The evolutionary comparison between D. *melanogaster*. Each line indicates another species, that is farther away from D. melanogaster, added to the comparison. The abbreviations shown on the graph above are as follows:mel - melanogaster; ere - erecta; vir - virilis; yak - yakuba; gri - grimshawi; sec - sechellia; moj - mojavensis.



(b) The pairwise comparisons between D. *melanogaster* and other Drosophila species are shown. Each line indicates a unique pairwise compairison. In the legend, the species are shown in a descending order that is correlated to how closely related the species are to D. *melanogaster*

Figure 5: The figures above show the CRM predictions given the comparison between species that have a common ancestor

The pairwise comparisons are in the same general areas as the evolutionary comparisons, but allow more ambiguity. This is in line with the phylogenetic theory that the constrained regions found between multiple species with a common ancestor are more likely to be a CRM. However, when determining the success of the results, it must be taken into consideration the method in which the results were extracted. MultiModule is a program that recognizes the functional sections of a sequences. This method of identification can result in the isolation of functional regions on the genome, but not necessarily predictions of CRMs. There is confusion when any functional portion of the sequence is mistaken as a CRM. Therefore, the excessive predictions made by MultiModule could indicate the identification of conserved regions that are not CRMs, or they could potentially be CRMs that have not been identified experimentally.

Drosophila		
Predicted Region	Known CRM	Known CRM Region
$[-296, -460]$	No	
$[-1,388, -1,541]$	Yes	$[-1,265, -1,890]$
$[-1,914, -1996]$	No	

Figure 6: The table shows that the known CRM was predicted by MultiModule.

The regions referenced in tables 6 and 7 are gathered from the graphs in figures 5 and 4. A predited region is defined by a location on the sequence that was predicted more than 20 times. The data given by the tables indicate that MultiModule has a very high false positive rate, or possibly more significantly that the program is capable of identifying novel CRMs. Based on the information provided with the Drosophila genus, MultiModule accurately predicted the only known CRM for D. *melanogaster*. There were two additional false positive predictions on the same sequence. This information is known through the public database redfly.ccr.buffalo.edu. When considering the Caenorhabditis genus, MultiModule predicted five of the seven known CRMs, giving a true positive rate of 71.42%. Conversely, MultiModule predicted seven out of twelve incorrectly, yeilding a false positive rate of 58.33%. The non-matched predictions could be due to the bidirectionality of DNA, and MultiModule could have identified CRMs for another gene. Or, MultiModule could be predicting CRMs that have not been experimentally verified. It could also be an incorrect prediction altogether, but MultiModule's predictions narrow down the possible locations of a CRM dramatically. If a new CRM was found it could then be experimentally confirmed where the CRM actually resides.

Caenorhabditis		
Predicted Region	Known CRM	Known CRM Region
$[-1,105,-1,440]$	Yes	$[0,-1,618]$
$[-2,687,-3,177]$	Yes	$[-2,155,-3,354]$
$[-3,310,-3,476]$	Yes	$[-3,354,-4,450]$
$[-3,906,-4,061]$	Yes	$[-3,354,-4,450]$
$[-4,159,-4,369]$	Yes	$[-3,354,-4,450]$
$[-4,873,-5,069]$	No	
$[-6,134,-6,576]$	No	
$[-6,867,-7,327]$	No	
$[-7,966,-8,290]$	No	
$[-8,342,-8,765]$	No	
$[-8,921,-9,132]$	No	
$[-9,887,-9,961]$	No	

Figure 7: The table shows the regions predicted by MultiModule and whether the locations were also experimentally known CRMs. There are two known CRMs that were not predicted by the program: $[0,-951]$ and $[-5,300,-5,000]$.

For previous comparison of efficiency I used sensitivity and specificity. I will be using sensitivity in order to compare MultiModule's predictions with the previously mentioned approaches. I will not be using specificity because it requires that I know the true negative rate of the predictions, which I was unable to obtain. The sensitivity rate for the evolutionary comparison of the D. *melanogaster* is 100%. Though this number looks enticing, it must be considered that there were two false positives that would be taken into the calculation given the true negative rate. For the C. *elegans* the sensitivity is 71.4%, which is a good number considering there is no use of motif library, and that there is potentially new CRMs being identified within the predictions.

6 Conclusion

Reproducing MultiModule in order to determine the location of CRMs on a given input sequence was a successful endeavor in terms of gathering information. However, a flaw of the program would be the inability to discover CRMs that may have diverged through the evolutionary process. Even still, MultiModule has the poten-

tial to identify CRMs that have not been identified through experimental methods. There is potential future work in determining whether the predicted CRMs are actually CRMs or if they're an unrelated functional region of the sequence. In addition, the order in which the input sequences are aligned needs to be varied for the purpose of determining if the difference in order would be beneficial to the program. Another important edition would be to increase the range in determining which value of K had the most accurate results. In knowing the best assumption of K to give MultiModule, it would decrease the amount of time to gather results from the program. An additional compelling aspect of MultiModule is the prediction of motifs. Research into the location and composition of the binding site in order to identify the transcription factors could prove benefical.

7 Acknowledgements

I would like to thank Professor Sorelle Friedler for advising me through the process of understanding MultiModule, and guiding my work in the right direction throughout this research. I would like to acknowledge the authors of the MultiModule program Zhou and Wong (2007) who graciously gave me the source code of MultiModule in order to produce my experiments. I would also like to acknowledge Professor Philip Meneely who guided me through the biological aspects of this thesis and provided me with the sequences and information of the Caenorhabditis genus.

References

[1] Adrian Bird. Perceptions of epigenetics. *Nature*, (447), May 2007.

[2] Bob Y. Chan and Dennis Kibler. Using hexamers to predict cis-regulatory motifs in drosophila. *BMC Bioinformatics*, 6, 2005.

[3] Droshophila 12 Genomes Consortium. Evolution of genes and genomes on the drosophila phylogeny. *Nature*, 450:203–218, November 2007.

[4] Brittanica Academic Edition. Probability theory, May 2014.

[5] Kelly A. Frazer, Inna Dubchak Laura Elnitski Deanna M. Church, and Ross C. Hardison. Cross-species sequence comparisons: A review of methods and available resources. *Genome Research*, 22, 2003.

[6] Martin C. Frith, John L. Spouge, Ulla Hansen, and Zhiping Weng. Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences. *Nucleic Acids Research*, 30:3214–3224, 2002.

[7] George Casella; Edward I. George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, August 1992.

[8] Ross C. Hardison and James Taylor. Genomic approaches towards finding cis-regulatory modules in animals. *Nature Reviews*, 13:469–483, 2012.

[9] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[10] Nora Pierstorff, Casey M. Bergman, and Thomas Wiehe. Identifying cis-regulatory modules by combining comparative and compositional analysis of dna. *Bioinformatics*, 22:2858–2864, 2006.

[11] Mark Robinson, Yi Sun, Rene Te Boekhorst, Paul Kaye, Rob Adams, and Neil Davey. Improving computational predictions of cis-regulatory binding sites. Technical report, 2006.

[12] Saurabh Sinha. On counting position weight matrix matches in a sequence, with application to discriminative motif finding. *Bioinformatics*, 22:e454–e463, 2006.

[13] Saurabh Sinha and Xin He. Morph: Probabilistic alignment combined with hidden markov models os cis-regulatory modules. *PLoS Comput Biol*, 3, 2007.

[14] Jing Su, Sarah A. Teichmann, and Thomas A Down. Assessing computational methods of cis-regulatory module prediction. *PLoS Computational Biology*, 6, 2010.

[15] Qing Zhou and Wing Hung Wong. Coupling hidden markov models for the discovery of cis-regulatory modules in multiple species. *The Annals of Applied Statistics*, 1, 2007.

Appendices

A Running the Experiments

A.1 Overview of Parameters

Any of the code used for this thesis can be found at Sorelle Freidler's website : http://www.haverford.edu/computerscience/faculty/sorelle/teaching.html. The source code for MultiModule was given to me by the authors of the paper Coupling Hidden Markov Models for the Discovery of Cis-Regulatory Modules in Multiple Species, Qing Zhou and Wing Hung Wong (2007). The MultiModule source code will be available at the website mentioned above.
The input parameters for any given command line prompt are as follows:

- i: input file name
- o: output file name (default: output.txt)
- n: maximum number of iterations (default: 1,000)
- p: cutoff of posterior probabilities (default: 0.5)
- N: number of species
- K: number of assumed motifs
- L: Expected module length(default: 200)
- w: minimal motif width (default: 8)
- W: maximal motif width (default: 15)
- u: probability of updating alignments at each iteration (default: 0.2)
- c: collapsed version of the program

A.2 File Formats

The input file for the program must be .fasta fomat. For my experiments I did not vary many of the parameters. I kept them at the default settings and used the majority of the input parameters as controls and varied only K during the pair-wise comparisons, and vaired only K and N during the evolutionary comparisons. Therefore, the results were able to indicate the effect of K and N on the program.

The format of the output files was important when creating the scripts that would gather the neccessary information. The output file, with the particular suffix "modules.txt", outputs the module predicitons in a five column format. The first two columns are indices of the species and the sequence within the species. The third and fourth columns hold the start and end positions on the sequence relative to the

end of the sequence. The last column displays the composition of the module. The files that contain the motif predictions have the suffix "motif_.txt", where the blank is one of the motifs that the program has tried to identify.

In order to run MultiModule in a more automated fashion, I created three python scripts that allowed me to automate the command line prompts of the program and automate the collection of the prediction locations. A detail of these scripts is listed below:

1. MultiLineToSingle.py In order to correctly input sequences into MultiModule, the input file must be in Fasta format. In addition, the sequences be contained in one line each. MultiToSingle.py takes any fasta file that has newline charaters and creates a new one that contains a header with the information of the gene and sequence and a single line of the entire sequence below it. It's beneficial to have this section automated as the number of species and genes to input increase in size and quantity. MultiLineToSingle.py should be placed wherever the input files of MultiModule will be.

2. Automated_runs.py: The script available is equipped to take one input file form the command line and run it with a varying K value. This is beneficial when trying to run multiple files at one time. This file should be located in the same directory as the makefile of the MultiModule program.

3. histogram_maker.py: This script takes the output module predictions from any given MultiModule prediction and collects the predictions on the initial sequence. The initial sequence indicates the first species' sequence of the input Fasta file. I placed histogram_maker.py within the output file directory of MultiModule to ease the flow of information when parsing the data.