12-2021

Genome Annotation Using Average Mutual Information

Garin P. Newcomb
University of Nebraska - Lincoln, garin.newcomb@huskers.unl.edu

GENOME ANNOTATION USING AVERAGE MUTUAL INFORMATION

by

Garin P. Newcomb

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Electrical Engineering

Under the Supervision of Professor Khalid Sayood

Lincoln, Nebraska

December, 2021

GENOME ANNOTATION USING AVERAGE MUTUAL INFORMATION

Garin P. Newcomb, M.S.

University of Nebraska, 2021

Adviser: Khalid Sayood

Advancements in high-throughput DNA sequencing technologies and ambitious goals for their use are resulting in the generation of a deluge of unannotated sequenced genomes. This makes computational tools that can aid in annotation increasingly valuable.

Here, we provide a detailed exploration of the utility as well as the limitations of average mutual information (AMI) in several steps of genome annotation. For a genomic sequence, AMI is a measure of the information a base contains about the base separated by a fixed lag. A profile is constructed by calculating AMI at multiple lags. In addition to traditional AMI, we employ two AMI variants: expanded AMI and expanded-adjusted AMI, both of which preserve some granular detail discarded by AMI.

First, we demonstrate AMI's capacity to assess evolutionary similarity by constructing phylogenetic trees similar to those currently accepted. The remainder of this work focuses on applications involving binary classification. We use support vector machines trained using the AMI profiles to classify sequences and evaluate predictive performance. These classification problems include predicting whether sequences come from protein-coding regions, identifying essential genes, and making functional predictions about the proteins genes produce. We conclude that AMI is particularly adept at identifying coding regions, and this behavior is consistent for species across all of life's diversity.

ACKNOWLEDGMENTS

What a long and winding road it has been! That this work exists at all is a tribute to all those who have supported and encouraged me through the years. First among them is my advisor, Dr. Khalid Sayood, who guided my research and stoked my curiosity. Outside my parents, there is no one from whom I have learned more. He gave me the freedom to develop and test my ideas, and always took care to ensure I felt I belonged at this level, in this field. His humor, wisdom, and perspicacity brought substantial richness to my undergraduate and graduate careers.

Next, I offer my appreciation to my other committee members, Dr. Michael Hoffman and Dr. Hasan Otu, for their careful reading of this thesis and insightful suggestions for improvement. Dr. Otu was an invaluable source of bioinformatics knowledge in my time at UNL, and what he taught me serves as the foundation for much of what appears here. Dr. Hoffman excels at inspiring creativity in his students' work, and I appreciate his tolerance (perhaps enjoyment) of the myriad idiosyncrasies that appeared in my assignments. I am fortunate that each member of my committee played such an important role in my academic development. In total, I took 11.5 courses taught by the three of them, and thoroughly enjoyed every one.

I am thankful to Teresa Ryans for all her help in overcoming administrative obstacles and keeping me on track to graduate this semester. I must also recognize my labmates for their contributions, especially Keith Murray and Amirsalar Mansouri. Keith was very welcoming when I first began working in Dr. Sayood's lab, and was a consistent source of ideas, commiseration, and excitement for science and engineering. And of course, my eternal gratitude goes to TAS for constant, unwavering inspiration and support.

I am very grateful to my parents, Lynn and Kathy Newcomb, for always

instilling in me that education is valuable and curiosity should be pursued. In particular, my father taught me to see the beauty in mathematics. And finally, I am so appreciative of my wife, Emma, who patiently waited for this moment and gently encouraged me to press on. It is the privilege of my life to walk beside you for all of my days.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Life is vast, diverse, innumerable, reaching even the most inhospitable edges of the earth. Recent estimates suggest as many as 10 million eukaryotic species inhabit this planet [3], of which we have only cataloged a small fraction, on the order of 15%. Nonetheless, this is utterly dwarfed by the scale of microbial diversity, whose membership may include as many as one trillion species [4]. In other words, there is a nearly limitless source of biological data all around us to collect, analyze, organize, annotate, classify, and interpret. Increasingly, data collection is outpacing the other steps in this process as technologies for doing so improve through cost reduction, automation, and computational capabilities. Nowhere is this more evident than in the field of genomics, as the number of species whose genome has been sequenced has exploded in recent years. For example, the Earth BioGenome Project, formed in 2018, seeks to sequence all 1.5 million currently known eukaryotic species in only 10 years [5].

The applications of this data are broad, and notably include technologies that improve human health. The consensus amongst reasonable people is that the pursuit of such improvements is a worthy endeavor. Topically (and obligatorily), the

swift development of vaccines against SARS-CoV-2 was enabled by the dissemination of its sequenced genome in January 2020, mere weeks after the first outbreak of COVID-19 was reported. Another application that depends on sequencing is CRISPR, the novel gene editing technology. CRISPR is already opening avenues for therapies treating genetic ailments such as sickle cell disease [6]. While it has been decades since the human genome was first sequenced, identifying genetic mutations that cause disease remains a highly active area of research.

The present glut of sequence data highlights the appeal of computational alternatives to work traditionally performed in a wet lab. The focus of this work is on a narrow class of such computational techniques, utilizing average mutual information (AMI) and its derivatives to make predictions about the biological role of genomic sequences at multiple levels. The AMI calculated over a genomic sequence of interest is known to be biologically significant, and has been used as a tool in many aspects of bioinformatics. This work investigates several new and previously demonstrated applications, with particular focus on a performance comparison between AMI variants. Additionally, we involve k-mers to supplement the information provided by AMI.

First, we explore AMI's ability to construct accurate phylogenetic trees for several species of fungi. Next, we use AMI to predict whether a sequence is drawn from a protein-coding region of a genome. Using a similar methodology, we predict whether a gene is essential for survival. Lastly, we attempt to infer various aspects of protein-coding gene annotations as described by the popular Gene Ontology (GO) framework. In addition to providing classification performance metrics, we attempt to identify AMI and k-mer features that contribute significantly to the discriminative capabilities.

Chapter 2

Biology Background

As the stated objective of this work is genome annotation, we will first introduce some background on the relevant genome biology. We will focus on DNA: its structure, its relationship to proteins, and how it is sequenced. DNA is fundamentally a vessel for information, and this is what we will subsequently exploit in making predictions about a particular DNA sequence's function or relationship to other sequences.

2.1 Biological Sequences

Heredity is a universal and readily observable feature of life as we know it. The primary mechanism behind this heredity is by now well understood: information about an organism passes to its offspring via a subset of its genome. The genome is a collection of deoxyribonucleic acid (DNA), a macromolecule that encodes a set of instructions for assembling all the component pieces needed by an organism during its life. Genes are interspersed through an organism's genome. While genes have many functions, this work is primarily interested in protein-coding genes, which are translated into amino acid sequences to produce proteins.

2.1.1 Structure

DNA is composed of two strands that intertwine to form a double helix [7]. Each strand is a chain of nucleotides, a monomer comprised of a nitrogenous nucleobase, a deoxyribose sugar, and a phosphate group. The nucleotide's phosphate group is attached to the 5' carbon of the ribose. The phosphate group of one nucleotide is bonded to the 3' carbon of the adjacent nucleotide's ribose to form the strand's backbone. Each nucleobase is hydrogen bonded to the complementary nucleobase on the opposite strand, producing a ladder-like structure. This structure is shown in Figure 2.1.

DNA's ability to store information is derived from the nucleobases. There are four possible bases, typically abbreviated to their first letter: cytosine (C), guanine (G), adenine (A), and thymine (T). Thus, each base contains two bits of information. Bases pair with only one other base: C with G, and A with T. For this reason, the sequence of bases on one strand determines the sequence of bases on the other strand. However, the strands are directional and genes are present on both. One strand's 3' end corresponds to the complementary strand's 5' end, and vice versa.

DNA is organized into chromosomes. Prokaryotes typically have a single circular chromosome, while eukaryotes have multiple linear chromosomes. The structure of a chromosome is complex: in addition to protein-coding genes, there are sequence structures that, for example, facilitate transcription, regulate gene expression, and protect the genome from degradation during replication.

Proteins are composed of one or more strands of amino acids called polypeptides. The protein's amino acid sequence is referred to as primary structure. The manner in which the protein folds into a three dimensional structure determines its

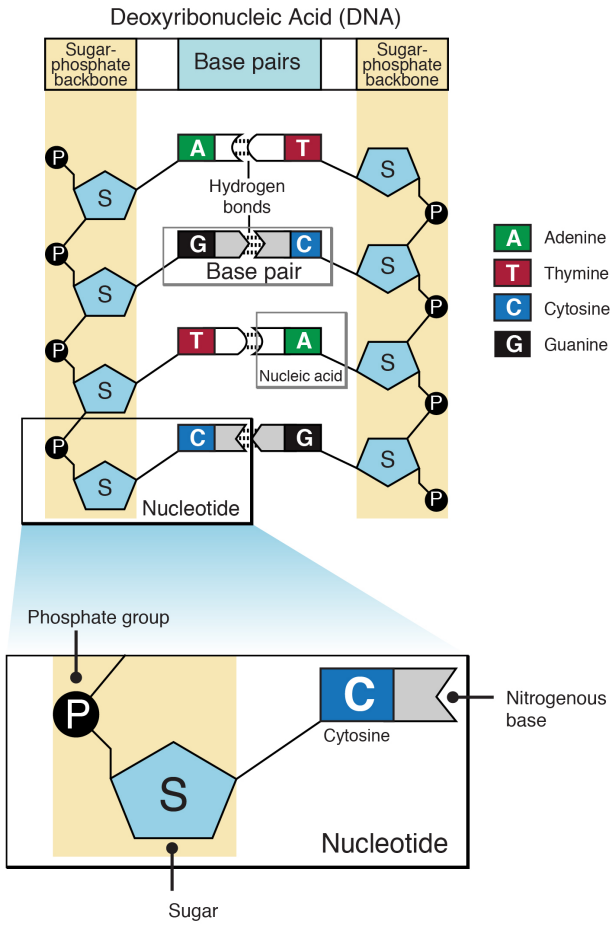


Figure 2.1: Depiction of nucleotides bonded together to form a double-stranded DNA molecule. Image freely available from the National Human Genome Research Institute.

secondary and tertiary structure. Quaternary structure describes how multiple polypeptides can organize into a single protein assembly. The primary structure dictates how the protein will fold, as bonds form between particular amino acids in particular positions in the sequence. A protein's role is determined by the following biological axiom: form fits function. That is, the shape of the protein dictates how it will function in the cell. For example, enzymes, a particular type of protein that act as catalysts in biochemical reactions, are shaped to enable binding their substrate.

2.1.2 Processes

Transcription is the process by which cellular machinery generates a complementary RNA sequence to a target gene. During transcription, an enzyme called RNA polymerase traverses the template strand in the 3' to 5' direction. As it proceeds, it produces messenger RNA (mRNA) in the 5' to 3' direction. The mRNA sequence matches the DNA's coding strand (with thymine nucleotides replaced with uracil), which is complimentary to the DNA's template strand. The transcript includes the coding segment(s), as well as 5' and 3' untranslated regions (UTR) that occur upstream and downstream of the start and stop codons, respectively. Once transcribed, the mRNA undergoes processing steps, including the addition of a 5' cap and poly-A tail, and removal of any introns.

Translation is the process by which an mRNA transcript is used to synthesize a protein. This is performed by a cellular component called a ribosome. It binds the 5' cap of a transcript and processes the transcript from 5' to 3'. When the ribosome reaches the mRNA's start codon (usually AUG) it begins translating. mRNA triplets, called codons, are mapped to amino acids by tRNA molecules. Each tRNA is attached to an amino acid and includes an anticodon that complements a particular codon. When it encounters a matching codon, it appends its amino acid to the growing chain. Translation concludes when a stop codon is reached (generally UAA, UAG, or UGA). The resulting amino acid chain then folds into a protein. Transcription and translation are depicted in Figure 2.2.

2.1.3 Genome Sequencing

Determining the nucleotide composition of a DNA sequence is not trivial. The wealth of genomic knowledge accumulated over the last half century was made

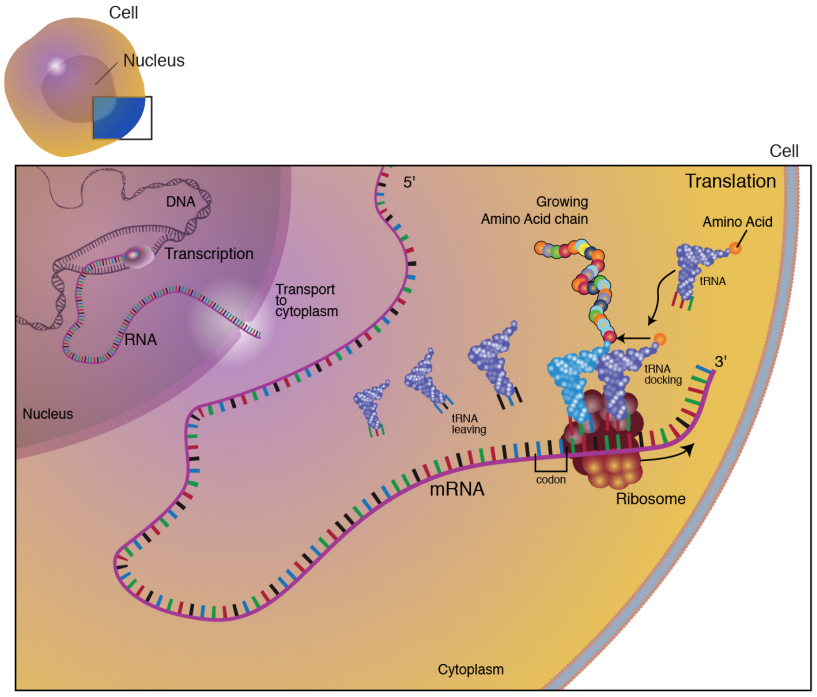


Figure 2.2: Illustration of transcription of DNA to form mRNA, and translation of that mRNA into an amino acid chain. Image freely available from the National Human Genome Research Institute.

possible by ever-advancing sequencing technologies. The human genome and its 3 billion base pairs was first sequenced in 2003 after a 13 year global project that cost $2.7 billion [8]. In contrast, so-called next-generation sequencing technologies are capable of sequencing hundreds of bases per second. Nanopore sequencing works by passing a DNA or RNA molecule through a hole in an electro-resistant membrane and measuring the current passing through each nucleotide in order to identify it [9]. This is the technology used by the portable, low cost MinION sequencer.

In contrast to next-generation sequencers, the "current" generation only sequences short segments at a time, which must then be assembled into longer sequences [10]. Whole Genome Assembly (WGA) involves the construction of longer

contigs by finding a consensus of overlapping reads. These are then placed into larger structures called scaffolds. Many sequenced genome assemblies are now available at the contig, scaffold, chromosome, or complete genome level. This work uses chromosome and genome level assemblies obtained from NCBI's assembly database in order to evaluate the effectiveness of the methods proposed. These genomes are well-studied and extensively annotated, and include model organisms such as bacteria *E. coli* and yeast *S. cerevisiae*. Our ultimate goal is to apply the proposed methods to the rapidly growing library of unannotated, newly sequenced genomes.

Chapter 3

Sequence-Derived Profiles

The composition of a given biological sequence dictates the role of the sequence in cellular processes, but primary sequences are difficult to work with. Transforming the sequence into a fixed length numerical profile introduces many options for analysis and classification. The usefulness of the profile depends on what information about the sequence it captures, so the manner in which we construct it is very important. We will next present several types of profiles that can be used independently or in concert.

3.1 Profile Construction

3.1.1 Average Mutual Information Profile

In any nucleic acid sequence, nucleotide frequency is skewed from uniform due to the biological constraints acting on it. In particular, small dependencies naturally arise between base pairs. One notable example of these dependencies occurs because of the triplet nature of translated portions of the genome. Coding regions consist of trinucleotides, where each trinucleotide corresponds to a particular amino acid.

However, there is considerable variation in trinucleotide abundance. This results in increased dependency between nucleotides that are integer multiples of 3 base pairs apart. This suggests that a measure of dependency between base pairs is a reflection of the underlying biological role of the sequence. We use average mutual information to measure such dependency.

The information learned by knowing the outcome of an event depends on the uncertainty associated with the event. This is quantified by the Shannon entropy [11], $H(X)$, which is defined as:

$$H(X) = -\sum_{x \in \mathcal{A}} p(x) \log p(x) \tag{3.1}$$

This is easily extended to multiple events. Events which are correlated have mutual information. This means that knowing the outcome of one event provides information about the other. That is, the uncertainty concerning the latter event is reduced. Average mutual information, $I(X;Y)$, measures the information contained in event X about event Y, and is defined as:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= \sum_{X \in \mathcal{A}} \sum_{Y \in \mathcal{A}} p(X,Y) \log \frac{p(X,Y)}{p(X)p(Y)} \end{aligned} \tag{3.2}$$

Initially, Shannon developed average mutual information (AMI) for studying communication [12], but it has since been applied to many fields, including bioinformatics. Specifically, it has been used to study the covariation of residues in the envelope protein of HIV [13] and other proteins [14]– [15]. It has also been used to identify coding regions of the genome [16], to aid in sequence assembly [17], and to generate a species-specific signature [18]. Whole and partial genome AMI profiles

have been used to classify fungal and mycobacterial samples [19], and study changes in HIV populations [20].

In order to generate an AMI profile based on a DNA sequence, we define X to represent the nucleotide at arbitrary location n and Y to represent the nucleotide at location $n+k$, for some lag k. The possible outcomes of both X and Y are then the four nucleotides: $\mathcal{A} = \{A, C, G, T\}$. We then estimate the marginal probability distributions $p(X)$ and $p(Y)$ by counting the occurrences of each nucleotide and dividing by the length of the sequence. Note that they are the same, since both are measured across the entire sequence. We call this estimate $\hat{p}_0(X)$. Similarly, the joint probability distribution $p(X, Y)$ is estimated by counting the occurrences of each of the 16 possible pairs of nucleotides separated by k base pairs, and dividing by the total number of pairs in the sequence. We call this estimate $\hat{p}_k(X, Y)$ for lag k. Using these probability estimates, we generate an AMI profile AMI_k for selected values of k as follows:

$$AMI_k = \sum_{X \in \mathcal{A}} \sum_{Y \in \mathcal{A}} \hat{p}_k(X, Y) \log \frac{\hat{p}_k(X, Y)}{\hat{p}_0(X)\hat{p}_0(Y)} \tag{3.3}$$

If the nucleotide occurring at position $n+k$ is independent of the nucleotide at position n, then the average mutual information between the two events is 0 (i.e. $AMI_k = 0$). Likewise, if there is a peak in the AMI profile at some lag k, this indicates inflated correlation between nucleotides k base pairs apart.

3.1.1.1 AMI for Finite Length Sequences

The AMI profile generated from an infinite-length sequence of random nucleotides will consist of all zeros. The joint and marginal nucleotide probabilities will be exact, so the independence of nucleotides at each lag will be reflected in the

profile values. For a finite-length sequence, the probabilities derived from the sequence are inexact estimates. As sequence length increases, the expected value of the error between the true marginal and joint probabilities and their estimates decreases. This results from the law of large numbers. The individual errors for each probability are random variables, and can be positive or negative. However, because the correlation between nucleotides is generally small, error in either direction will imply greater correlation than there really is. This has the effect of artificially inflating AMI profile magnitudes according to sequence length.

For a sequence of N random independent equiprobable nucleotides, the expected value for AMI magnitude at lag of 1 (i.e. adjacent nucleotides) is as follows. The number of instances of nucleotide X and Y in the sequence are denoted by n_X and n_Y, respectively. The number of instances of the nucleotide pair XY is denoted by n_{XY}.

$$\begin{aligned} E\left[AMI_1\right] &= E\left[\sum_{X\in\mathcal{A}}\sum_{Y\in\mathcal{A}} \hat{p}_1(X,Y)\log\frac{\hat{p}_1(X,Y)}{\hat{p}_0(X)\hat{p}_0(Y)}\right] \\ &= E\left[\sum_{X\in\mathcal{A}}\sum_{Y\in\mathcal{A}} \frac{n_{XY}}{N-1}\log\frac{n_{XY}N^2}{n_X n_Y(N-1)}\right] \\ &= \sum_{X\in\mathcal{A}}\sum_{Y\in\mathcal{A}} E\left[\frac{n_{XY}}{N-1}\log\frac{n_{XY}N^2}{n_X n_Y(N-1)}\right] \end{aligned}$$

Because nucleotide probabilities are identical, all possible nucleotide pairs XY such that $X \neq Y$ produce the same expectation, and all possible nucleotide pairs XY such that $X = Y$ produce the same expectation. Henceforth, XY denotes the

case where $X \neq Y$, and XX denotes the case where $X = Y$.

$$\begin{aligned} E\left[AMI_1\right] &= 12E\left[\frac{n_{XY}}{N-1}\log\frac{n_{XY}N^2}{n_X n_Y(N-1)}\right] + 4E\left[\frac{n_{XX}}{N-1}\log\frac{n_{XX}N^2}{n_X^2(N-1)}\right] \\ &= 12\sum_{n_X=0}^{N}\sum_{n_Y=0}^{N}\sum_{n_{XY}=0}^{N-1} P(n_X, n_Y, n_{XY})\frac{n_{XY}}{N-1}\log\frac{n_{XY}N^2}{n_X n_Y(N-1)} + \\ &\quad 4\sum_{n_X=0}^{N}\sum_{n_{XX}=0}^{N-1} P(n_X, n_{XX})\frac{n_{XX}}{N-1}\log\frac{n_{XX}N^2}{n_X^2(N-1)} \end{aligned}$$

We now require the joint probabilities $P_N(n_X, n_Y, n_{XY})$ and $P_N(n_X, n_{XX})$ for sequence length N. If we append another random nucleotide to form a length $N+1$ sequence, the probability that the final nucleotide pair is XY or XX depends on the nucleotide at position N, denoted t_N. The (identical) marginal probability of a each nucleotide occurring at each position is denoted p_t. If $t_N = X$, the probability that the final nucleotide pair is XY or XX is p_t, while if $t_N \neq X$, the probability of both is 0. It is thus helpful to define the joint probabilities $P_N(n_X, n_Y, n_{XY})$ and $P_N(n_X, n_{XX})$ in terms of t_N as follows:

$$\begin{aligned} P_N(n_X, n_Y, n_{XY}) &= P_N(n_X, n_Y, n_{XY}, t_N = X) + P_N(n_X, n_Y, n_{XY}, t_N \neq X) \\ P_N(n_X, n_{XX}) &= P_N(n_X, n_{XX}, t_N = X) + P_N(n_X, n_{XX}, t_N \neq X) \end{aligned}$$

The joint probabilities for sequence length N are defined recursively as follows:

$$\begin{aligned} P_N(n_X, n_Y, n_{XY}, t_N = X) =& P_{N-1}(n_X - 1, n_Y, n_{XY}, t_{N-1} \neq X, t_N = X) + \\ & P_{N-1}(n_X - 1, n_Y, n_{XY}, t_{N-1} = X, t_N = X) \\ =& p_t P_{N-1}(n_X - 1, n_Y, n_{XY}, t_{N-1} \neq X) + \\ & p_t P_{N-1}(n_X - 1, n_Y, n_{XY}, t_{N-1} = X) \end{aligned}$$

$$
\begin{aligned}
P_N(n_X, n_Y, n_{XY}, t_N \neq X) =& P_{N-1}(n_X, n_Y, n_{XY}, t_{N-1} \neq X, t_N \notin \{X, Y\}) + \\
& P_{N-1}(n_X, n_Y, n_{XY}, t_{N-1} = X, t_N \notin \{X, Y\}) + \\
& P_{N-1}(n_X, n_Y - 1, n_{XY}, t_{N-1} \neq X, t_N \neq X) + \\
& P_{N-1}(n_X, n_Y - 1, n_{XY} - 1, t_{N-1} = X, t_N \neq X) \\
=& 2p_t P_{N-1}(n_X, n_Y, n_{XY}, t_{N-1} \neq X) + \\
& 2p_t P_{N-1}(n_X, n_Y, n_{XY}, t_{N-1} = X) + \\
& p_t P_{N-1}(n_X, n_Y - 1, n_{XY}, t_{N-1} \neq X) + \\
& p_t P_{N-1}(n_X, n_Y - 1, n_{XY} - 1, t_{N-1} = X)
\end{aligned}
$$

$$
\begin{aligned}
P_N(n_X, n_{XX}, t_N = X) =& P_{N-1}(n_X - 1, n_{XY}, t_{N-1} \neq X, t_N = X) + \\
& P_{N-1}(n_X - 1, n_{XY} - 1, t_{N-1} = X, t_N = X) \\
=& p_t P_{N-1}(n_X - 1, n_{XY}, t_{N-1} \neq X) + \\
& p_t P_{N-1}(n_X - 1, n_{XY} - 1, t_{N-1} = X)
\end{aligned}
$$

$$
\begin{aligned}
P_N(n_X, n_{XX}, t_N \neq X) =& P_{N-1}(n_X, n_{XY}, t_{N-1} \neq X, t_N \neq X) + \\
& P_{N-1}(n_X, n_{XY}, t_{N-1} = X, t_N \neq X) \\
=& 3p_t P_{N-1}(n_X, n_{XY}, t_{N-1} \neq X) + \\
& 3p_t P_{N-1}(n_X, n_{XY}, t_{N-1} = X)
\end{aligned}
$$

The joint probabilities for sequence length 2 are initialized on the set of 16 possible nucleotide pairs. Calculated expected AMI magnitudes using the

recursively defined probabilities are shown in Figure 3.1 for lengths up to 300 nucleotides. As N increases, the expectation converges to the inverse of N. This is particularly apparent if we normalize the AMI value by multiplying by the sequence length. This implies, but does not prove, the following for some constant C:

$$\lim_{N\to\infty}(N-1)E\left[AMI_1\right] = C$$
$$\lim_{N\to\infty}\sum_{X\in\mathcal{A}}\sum_{Y\in\mathcal{A}} E\left[n_{XY}\log\frac{n_{XY}N^2}{n_X n_Y(N-1)}\right] = C$$

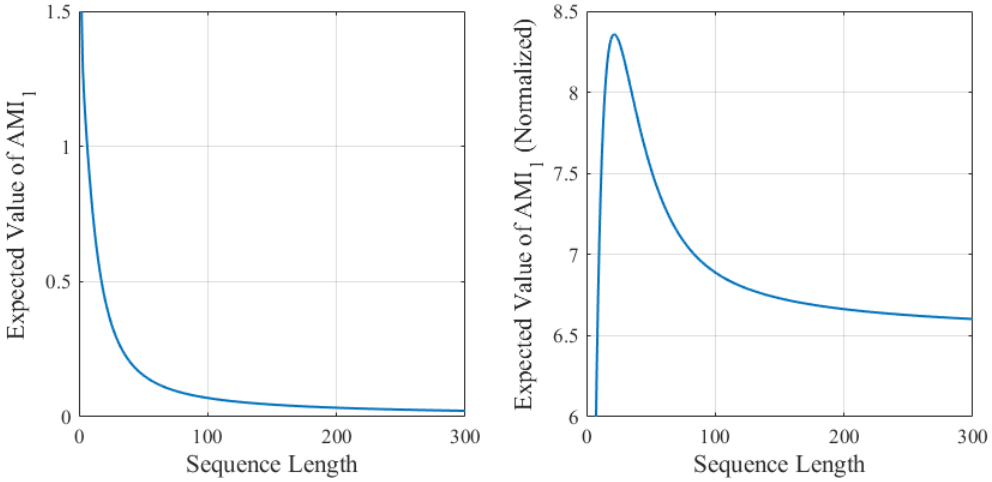


Figure 3.1: Expected AMI magnitude for lag 1 with and without normalization by sequence length

Calculating the expected value of AMI using this recursive method is untenable for sequence lengths more than a few hundred nucleotides. Instead, we estimate the expected AMI magnitudes using simulated random sequences. In this way, we can observe how the expected value of AMI profile values vary for different marginal nucleotide probabilities and lags. As seen in Figure 3.2, there is significant variation for short sequence lengths, but in all cases, AMI converges to a tight range as length increases. For our purposes, a sequence of length 200 is likely adequate for calculating reasonably accurate AMI values at a variety of lags.

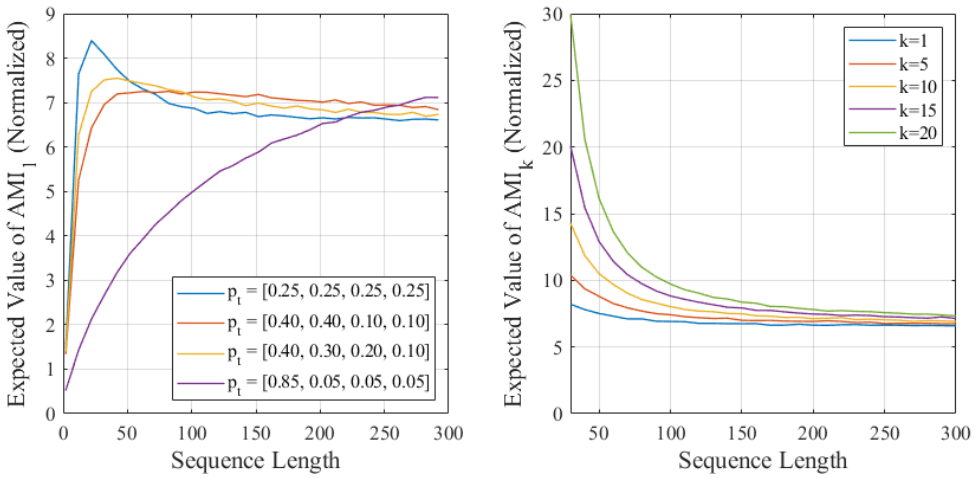


Figure 3.2: Expected normalized AMI magnitude for various marginal nucleotide probabilities (left) and lags (right)

3.1.1.2 Average Mutual Information Variants

The AMI profile provides a glimpse into how a sequence's nucleotides bias surrounding nucleotides at particular lags. A less succinct profile that provides additional information can be defined by collecting the individual terms that are summed when calculating AMI. We entitle this profile "expanded-adjusted Average Mutual Information" (eaAMI). For each value of k, the profile consists of 16 elements, one for each possible pair of nucleotides k bases apart. The frequency of each nucleotide pair is estimated, and then scaled by the dependence between the two nucleotides. That is, the profile element for lag k and nucleotide pair X, Y is defined as:

$$eaAMI_k(X,Y) = \hat{p}_k(X,Y) \log \frac{\hat{p}_k(X,Y)}{\hat{p}_0(X)\hat{p}_0(Y)} \tag{3.4}$$

Thus, the profile consists of $16k$ values, concatenated into a single vector.

A slightly simpler profile utilizes the unadjusted nucleotide pair frequencies, and is thus termed the "expanded Average Mutual Information" (eAMI). Formally, each

profile element is defined as:

$$eAMI_k(X,Y) = \hat{p}_k(X,Y) \tag{3.5}$$

As with eaAMI, the eAMI profile consists of $16k$ values. These AMI variants are easily extended to use with amino acid sequences, which may provide different (but overlapping) information about the sequence.

3.1.2 k-mer Profiles

A k-mer is a subsequence of length k that occurs in a biological sequence, such as DNA or polypeptides. k-mers have many applications, including DNA sequence assembly [21], predicting genomic regulatory elements [22], and identifying species in metagenomic samples [23]. A k-mer profile may be constructed for a given sequence by counting the frequency of occurrence of each possible k-mer. For a given value of k, there are 4^k possible k-mers in a nucleotide sequence, and 20^k possible k-mers in an amino acid sequence. The profile may include k-mer counts for multiple values of k.

Typically, k-mers are overlapping, but we will also consider special cases where k-mers do not overlap. For example, non-overlapping 3-mers have special significance to DNA sequences because they correspond to codon frequencies in protein-coding regions. For DNA and RNA sequences, 1-mers are synonymous with nucleotide counts, and for polypeptide sequences, 1-mers are synonymous with amino acid counts.

3.1.3 Dimensionality Reduction

The aforementioned profiles are selected because they present information contained in the sequences from which they are derived in an easily analyzed format. However, much of they information they provide is redundant. Additionally, many profile elements provide no discrimination between classes of interest. Lastly, high dimension data presents some practical obstacles: it is difficult to visualize, and increases the complexity of any analysis applied to it. For these reasons, techniques to reduce the data's dimension are often useful.

There exist many strategies to prune our data to a more manageable dimension. The simplest solution is to change the parameters used to develop the profiles. For k-mers, this means reducing the maximum length k considered. For AMI variants, this means reducing the maximum lag k. In both cases, increasing those parameters over a certain threshold provides diminishing returns. Nonetheless, if there is a chance that some component of the higher dimension data has value, we would like to preserve it. Generally, we can measure the individual contributions of each component by considering that component in isolation. We can then assemble a low dimension profile by ranking components and selecting only those with the highest rank.

A more sophisticated approach is Principal Component Analysis (PCA), a commonly used technique for dimensionality reduction [24]. The objective of PCA is to identify the set of vectors along which a dataset has maximum variance. This is often done using singular value decomposition, including for all uses of PCA in this work. We can then reduce the dimensionality of our profiles by projecting them onto these vectors. This preserves the maximum variation in the original data for a given number of reduced dimensions. Underlying this method is the assumption

that high variance is indicative of high information content concerning the classification of interest. This is certainly not guaranteed, and we may ultimately discard valuable lower variation data in favor of useless high variation data.

On the other hand, PCA requires no prior knowledge of the class to which training set sequences belong, which may be advantageous in some situations. Finally, PCA is particularly effective in eliminating the redundant information provided by variables that are strongly correlated with each other, as is the case with many of the profile elements.

3.2 Profile Analysis

Now that we have introduced our framework for transforming sequences into numerical profiles, we will discuss the tools we will use to analyze those profiles.

3.2.1 Clustering

One of the defining features of DNA is that species relatedness is reflected in their genomes. As species evolve through gradual, undirected genetic changes, they diverge from each other. Relationships between species are depicted in phylogenetic trees, which branch to indicate when a group's ancestors diverged. One method for developing a tree is to apply a hierarchical clustering algorithm to a distance matrix populated with the distances between all pairs of species in a set. While the mechanics of the available clustering algorithms (e.g. UPGMA, WPGMA, and neighbor-joining) vary, they each attempt to cluster elements with similar elements, and groups of elements with similar groups of elements. Numeric sequence profiles provide a natural way to calculate distance between sequences, including Euclidean distance and correlation distance. Many tools exist to perform clustering on a

distance matrix. One such popular tool is the PHYLogeny Inference Package (PHYLIP) [25].

3.2.2 Classification

Many of the biologically relevant characteristics of a sequence can be framed as a binary classification problem. Given a novel sequence, we would like to predict if it possesses that characteristic, thus eliminating the need for expensive wet lab experiments to make that determination. Myriad tools exist for making binary predictions, including support vector machines, neural networks, nearest neighbor, naive Bayes, and classification trees [26]. As a baseline, we will apply a simple nearest neighbor classifier. Given training data, we calculate the centroid of data points belonging to each class. We then predict that test sequences belong to the class whose centroid is closest.

We also make frequent use of support vector machines (SVMs), a well established supervised learning tool used for binary classification [27]. SVMs operate by constructing a hyperplane in the N-dimensional space occupied by the profile data. The objective of training is to identify the hyperplane that maximizes the margin between the two classes, and an example of data separated by such a hyperplane is shown in Figure 3.3.

The classes in the example are separable, so there is a hard margin between them that is maximized by the SVM. The objective function for doing so can be formulated for optimization using quadratic programming. Importantly, SVMs also work on inseparable data, where there is overlap between the two classes and the margin between them is "soft". This is done by penalizing the objective function for data that crosses the boundary separating the two classes.

SVMs can also be formulated to generate nonlinear classifiers by performing the

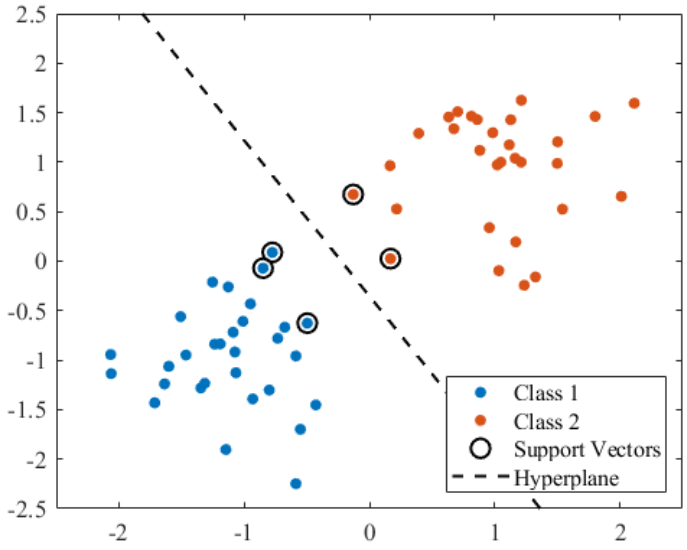


Figure 3.3: Data points from binary classes separated by a hyperplane determined by an SVM, with support vectors shown.

"kernel trick", in which a nonlinear kernel function is applied to the objective function. Generally, a linear SVM performs adequately for our purposes, as the data does not follow any well defined nonlinearities. At various times in the course of developing this work, we applied other kernels, including polynomial and radial basis function, but did not observe any noticeable improvement. There are several other configurable training parameters that can be adjusted to improve classification and convergence behavior.

Chapter 4

Phylogenetic Tree Construction

Now that we are familiar with AMI-based profiles, it is time to put them to use. Our first application is the construction of a phylogenetic tree [1]. These trees represent evolutionary distance, and how species relate to one another. Sequence profiles provide a natural means of calculating distance between sequences. If these distances correspond to evolutionary distance, then the tree we construct will closely match the accepted phylogeny.

4.1 Fungal Phylogeny Introduction

While genetically very similar, species of *Candida* and *Saccharomyces* yeasts differ widely in their net impact on humans. The *Candida* clade includes many pathogens, including *C. tropicalis*, *C. parapsilosis*, and *C. albicans* [1]. Together, *Candida* species are the most common cause of opportunistic fungal infection. In contrast, *Saccharomyces cerevisiae* (colloquially known as brewers yeast) and other

[1]Phylogeny results were published in 2017 IEEE EIT: G. Newcomb, A. L. Atkin and K. Sayood, "Use of average mutual information signatures to construct phylogenetic trees for fungi," *2017 IEEE International Conference on Electro Information Technology (EIT)*. Lincoln, NE, USA: IEEE, May 2017, pp. 398-403.

Saccharomyces species have immense economic utility. Evolutionary relationships between yeast species are accordingly of great interest. Current accepted phylogeny was assembled by performing multiple sequence alignment on 706 orthologs across 17 fungal species [1]. This phylogeny is shown in Figure 4.1.

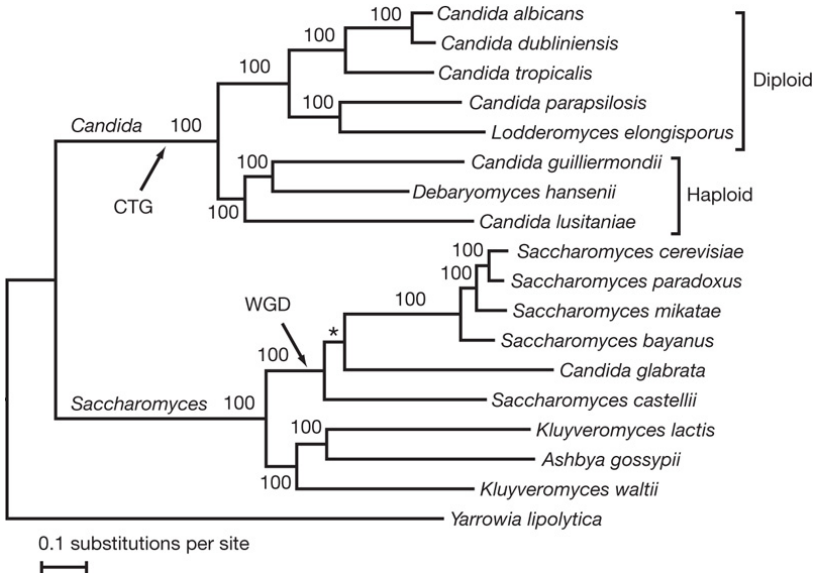


Figure 4.1: Accepted fungal phylogeny focusing on select *Candida* and *Saccharomyces* species [1]. "WGD" refers to a whole genome duplication, while "CTG" refers to the clade of fungi characterized by the translation of CTG codons as serine rather than leucine. Reprinted by permission from Macmillan Publishers Ltd: *Nature*, 472: 657-662, copyright 2009.

While this method is certainly robust, it is time consuming and requires reliable genome sequencing and annotations for all included species. Many other methods for predicting evolutionary relationships exist. The relative complexity measure (RCM) was designed as an alignment-free method for assessing sequence similarity [28]. It operates by generating a Lempel-Ziv dictionary from one sequence and evaluating how many steps are needed to generate a dictionary for the second sequence when using the first dictionary as a starting point. RCM has been shown to accurately generate a phylogenetic tree for fungi. While good at assessing sequence similarity, RCM does not provide information about what makes two

sequences similar. An alternative alignment-free method for measuring sequence similarity involves profiles defined by the average mutual information present in the sequences. These profiles have been used to accurately group subtypes of the HIV-1 [18], suggesting applications in phylogeny.

Ribosomal DNA (rDNA) is commonly used to evaluate species relatedness. The rDNA gene complex contains the 18S, 28S, and 5.8S genes, each of which are ribosomal components once transcribed. The sequences that separate these genes are called internal transcribed spacer (ITS) 1 and ITS2. The two spacers, along with the 5.8S sequence between them, are together termed the ITS region. The ITS region has been proposed as a universal barcode for fungi [29]. This is because the highly conserved rDNA genes are ideal for designing primers. However, the spacers are under less pressure to avoid mutations, and accordingly diverge more quickly. This makes the ITS region ideal for evaluating evolutionary distance between species. We obtained ITS sequences for 16 of the 18 species evaluated by [1]. ITS sequences could not be found for *C. guilliermondii* and *K. waltii*.

4.2 Phylogenetic Tree Construction

We generated AMI, eAMI, and eaAMI profiles for each of the 16 ITS sequences, for lag values $1 \leq k \leq 32$. Thus, the AMI profiles can be interpreted as vectors in 32-dimensional space, and the eAMI and eaAMI profiles as vectors in 512-dimensional space. We then populated 16 by 16 distance matrices D for each profile type by calculating the distance between profiles for each pair of sequences. The distance was calculated using two different methods, resulting in two distance matrices, and thus two phylogenetic trees for each profile type. First, we calculated a correlation difference based on the angle between each pair of profiles. That is, we

calculated d_{ij}, the distance between the profiles for sequence i (x_i) and sequence j (x_j), as follows:

$$d_{ij} = 1 - \cos\theta = 1 - \frac{\boldsymbol{x_i} \cdot \boldsymbol{x_j}}{\|\boldsymbol{x_i}\|\|\boldsymbol{x_j}\|} \tag{4.1}$$

Since all elements of the profile are non-negative, d_{ij} must satisfy $0 < d_{ij} < 1$, where 0 indicates maximum similarity (but does not require identical sequences). We also calculated Euclidean distance between profiles, as follows:

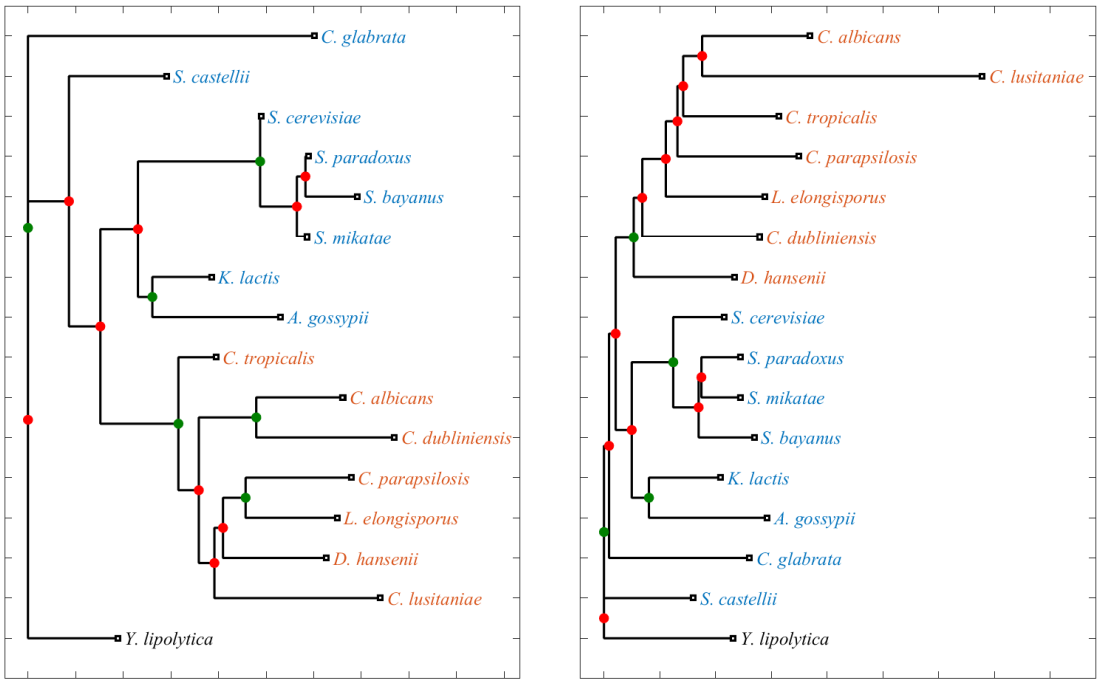
$$d_{ij} = \|\boldsymbol{x_i} - \boldsymbol{x_j}\| \tag{4.2}$$

Phylogenetic trees were generated using PHYLIP (version 3.695) [25]. The AMI-based distance matrices were used as input, and the program was run using the neighbor-joining option.

4.3 Phylogenetic Tree Results

4.3.1 Phylogenetic Tree Using Correlation Distances on AMI Profiles

The phylogenetic tree generated using a distance matrix comprised of correlation distances between AMI profiles is presented in Figure 4.2a. *Y. lipolytica* is correctly identified as having diverged from the remaining sequences. All 7 species in the *Candida* clade are correctly assigned in a monophyletic group. The tree was less accurate for the *Saccharomyces* clade: only 6 of 8 species were clustered in a monophyletic group. *C. glabrata* and *S. castellii* both should have been included in this clade but were instead shown to have diverged prior to the split between *Saccharomyces* and *Candida*.



(a) Correlation (b) Euclidean

Figure 4.2: Phylogenetic trees generated by PHYLIP. The distance matrix used to generate each tree was populated with the distances between each pair of species' AMI profiles, using the metric specified in each caption. Species in blue belong to the *Saccharomyces* clade, while species in orange belong to the *Candida* clade. Branches colored green indicate that the list of leaves descending from that branch match those in the accepted phylogeny.

4.3.2 Phylogenetic Tree Using Euclidean Distances on AMI Profiles

The phylogenetic tree generated using a distance matrix comprised of Euclidean distances between AMI profiles is presented in Figure 4.2b. *Y. lipolytica* is correctly identified as having diverged from the remaining sequences, but the tree erroneously indicates that *S. castellii* was first to diverge. All 7 species in the *Candida* clade are correctly assigned in a monophyletic group. Additionally, 7 of 8 *Saccharomyces* species were clustered in a monophyletic group. *C. glabrata* was correctly included

in the group, while it was not in the tree generated using correlation distances. However, within the *Candida* clade, the Euclidean distances did not correctly identify either of the monophyletic pairs of species (*C. albicans* paired with *C. dubliniensis*, and *L. elongisporus* paired with *C. parapsilosis*).

4.3.3 AMI Magnitudes and Sequence Length

The ITS sequences are less than 1000 nucleotides, which results in a bias in the AMI profiles. This is shown in Figure 4.3. The AMI magnitudes are inversely proportional to the length of the sequence. While inflating these magnitudes would not affect the correlation distance, which is agnostic to vector magnitude, it would affect Euclidean distance considerably. This is particularly evident in the case of *C. lusitaniae*. At 382 bp long, its ITS region is 137 bp shorter than that of any other species. This is reflected in the tree, which indicates that the branch length for *C. lusitaniae* is significantly longer than any other branch. To reduce this source of

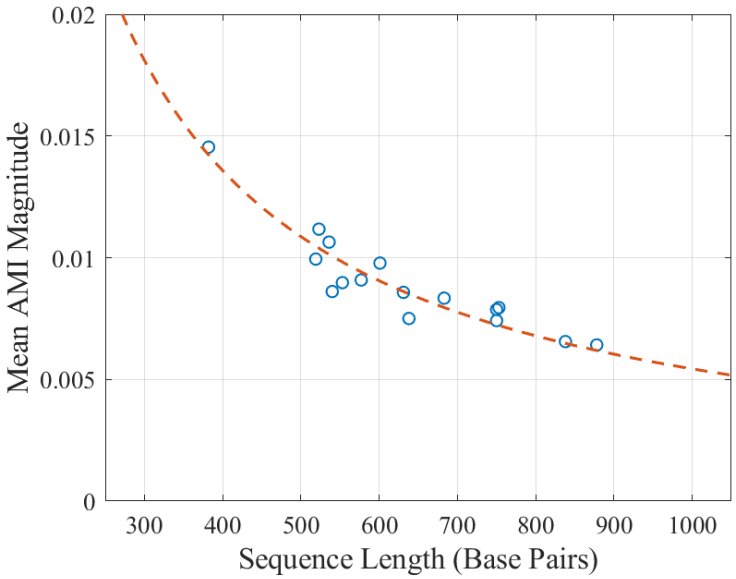


Figure 4.3: Mean AMI magnitude for each ITS sequence, versus sequence length. The data closely fit the curve $y = \frac{5.43}{x}$.

bias, we normalized the magnitudes by multiplying each element of the AMI profiles by the sequence length. A new distance matrix and new tree were generated with the normalized profiles.

The phylogenetic tree generated by this method is presented in Figure 4.4. It clearly bears a striking resemblance to the tree generated using correlation distances. Once length-induced bias was removed, Euclidean distances closely matched the corresponding correlation distances. Nonetheless, there were slight differences in the trees. Most importantly, *K. lactis* and *A. gossypii* are separated from the other members of the *Saccharomyces* clade. Thus, despite similar performance, it appears that the correlation distance method yielded a slightly more accurate tree.

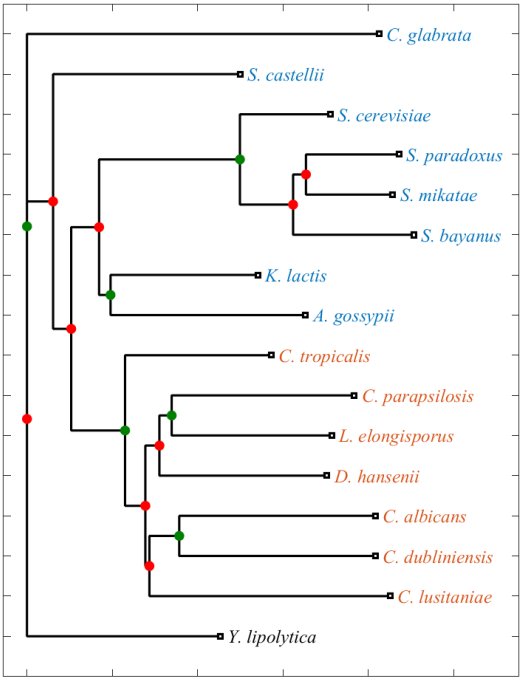


Figure 4.4: Phylogenetic tree generated by PHYLIP. The AMI profiles were normalized by multiplying each element by the sequence length. Euclidean distances were then used to populate the distance matrix.

The mean AMI profiles for *Candida* and *Saccharomyces* species after normalization are presented in Figure 4.5. While fairly similar in form, there are a few noticeable distinguishing characteristics. There are significant differences in AMI magnitude at lags 1, 2, 6, and 12 between the two clades. In fact, all *Candida* species had a local peak at lag 12, while none of the *Saccharomyces* species had such a peak. This may be the result of length 6 repeats in *Candida* ITS regions (there are no length 12 repeats in *C. albicans*). It could also result from differences in the secondary structures of the ribosomal RNA.

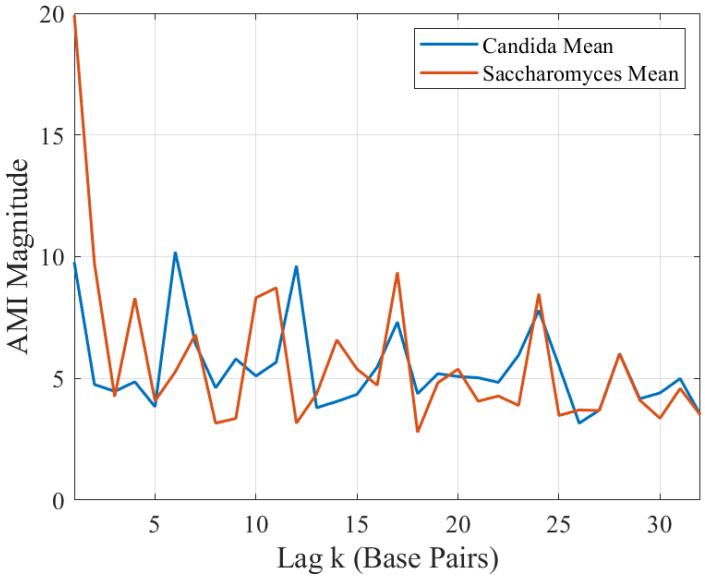
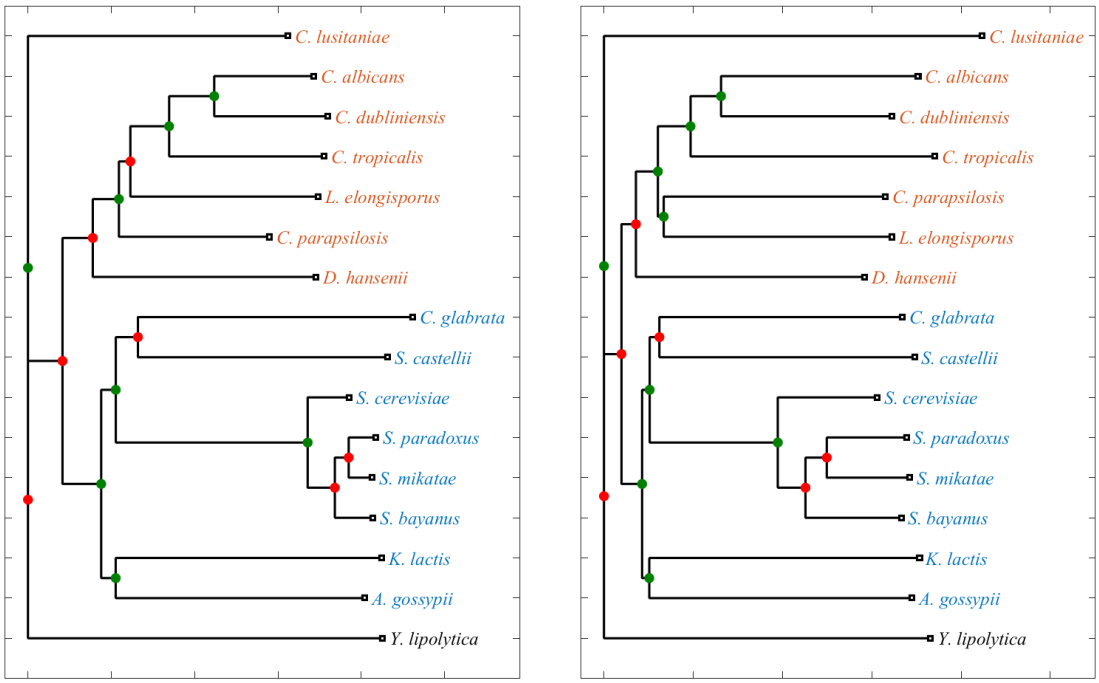


Figure 4.5: Mean normalized AMI profiles for the 7 species in the *Candida* clade and 8 species in the *Saccharomyces* clade.

4.3.4 Phylogenetic Trees Using AMI Variants

Phylogenetic trees generated using distance matrices comprised of pairwise distances between eaAMI profiles are presented in Figure 4.6. Trees using eAMI profiles are presented in Figure 4.7. For both variants, trees for Euclidean and correlation distances are included. The eaAMI trees are noticeably better, with only



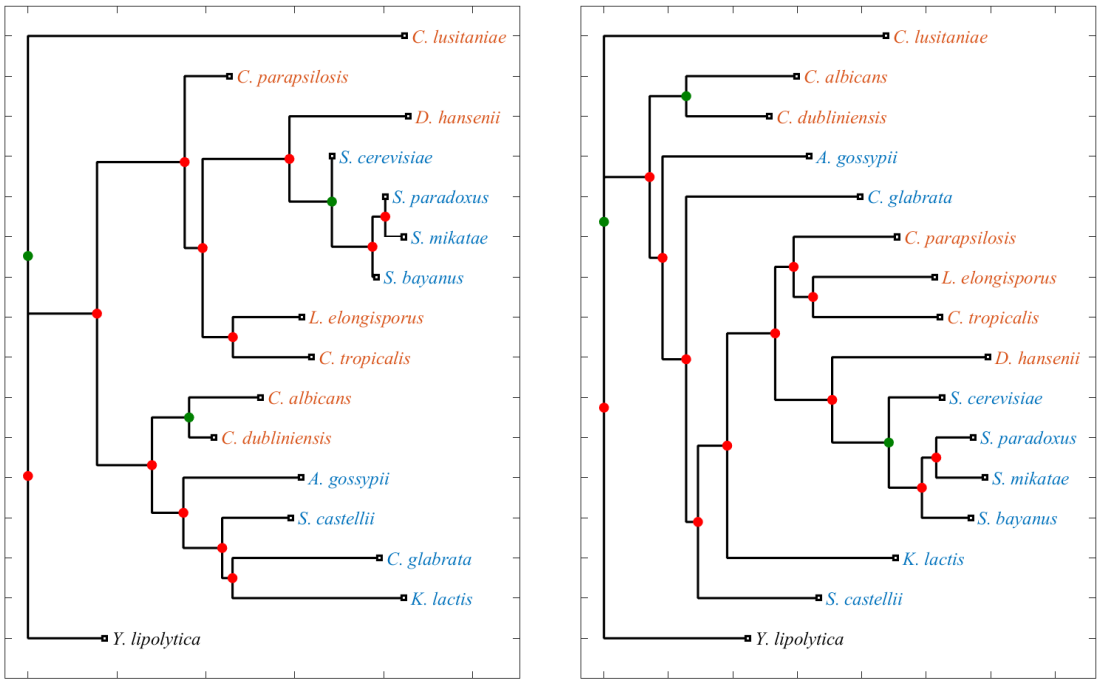
(a) Correlation (b) Euclidean

Figure 4.6: Phylogenetic trees generated by PHYLIP using distance matrices derived from eaAMI profiles.

C. lusitaniae not included in the correct *Saccharomyces* or *Candida* clade. eAMI struggled at all levels of the phylogeny, regardless of the distance matrix used. Next, we will explore differences in performance for the profile types in more detail.

4.3.5 Tree Summary and Comparison

To aid in the comparison of the trees, we focused on how well they clustered the species of the *Saccharomyces* and *Candida* clades together, and how well they matched closely related species. This gives a reasonable gauge for how closely each metric matches evolutionary distance over both the short and long term. We only considered the 16 species for which ITS sequences could be found. The accepted



(a) Correlation (b) Euclidean

Figure 4.7: Phylogenetic trees generated by PHYLIP using distance matrices derived from eAMI profiles.

phylogeny includes 8 species in the *Candida* clade, 8 species in the *Saccharomyces* clade, and 6 species in the "whole genome duplication" (WGD) clade, a subclade of *Saccharomyces*. For each distance metric, we counted the number of species in the accepted clades that were clustered in a monophyletic group in the generated tree. The accepted phylogeny also includes 4 monophyletic pairs of closely related species. Again, for each distance metric, we counted how many of these pairs also appeared in the generated tree. Finally, we count how many branches in the output trees have sets of leaf descendents identical to the corresponding branches in the accepted tree. The topology is ignored, as a branch counts as a match as long as the identity of the leaves matches. These results are presented in Table 4.1.

Table 4.1: Comparison of phylogenetic trees. Accuracy for each distance metric is presented as the number of matches with the accepted phylogeny for each category.

		AMI			eAMI		eaAMI	
Category	Accepted Phylogeny	Cor.	Euc.	Euc. (Norm.)	Cor.	Euc.	Cor.	Euc.
Candida Species	7	7	7	7	2	3	6	6
Sacchar. Species	8	6	6	4	4	4	8	8
WGD Species	6	4	4	4	4	4	6	6
Monophyl. Pairs	4	3	1	3	1	1	2	3
Branch Leaves	14	5	3	5	2	2	7	8

Of the three profiles, eaAMI produced the trees that most closely resembled the accepted phylogeny. The distance metric used had a minor impact on the resulting tree, as the tree using Euclidean distances was slightly more accurate. AMI performed better than eAMI. The dramatic difference between the performance of eaAMI and eAMI is striking given the similarity in how those profiles are constructed.

Based on these results, we conclude that correlation between AMI profiles provides a slightly better measure of evolutionary distance than Euclidean distance. For Euclidean distances, normalizing AMI magnitudes by sequence length resulted in better pairing of closely related species. However, this step also appeared to degrade the ability of the metric to correctly cluster larger clades as monophyletic groups. This latter effect suggests that divergence in ribosomal sequence lengths may reflect evolutionary distance. Indeed, the sequence lengths ranged from 382-638

bp for species in the *Candida* clade, and 577-878 for species in the *Saccharomyces* clade. As the un-normalized Euclidean distances were in part a measure of sequence length distance, it is unsurprising that this metric successfully grouped the two clades together.

4.4 AMI as a Measure of Evolutionary Distance

In general, the objective of any professed measure of evolutionary distance is a deterministic, monotonically increasing function of time. Linearity is also desirable, as such a measure would provide equivalent differentiation across all time frames. In order to gain insight into the performance of AMI-based measures of distance, we simulated the evolution of an ITS sequence in the simplest case. We started with the ITS sequence for *C. albicans* (length 536 bp), then generated two diverging sequences by probabilistically injecting point mutations in each. At each iteration, the substitution rate for each nucleotide was 0.001, and the insertion and deletion rates were both 0.00005. Thus, the substitution to indel ratio was 10, which is typical for noncoding regions in Eukaryotes [30]. The simulation was performed with only substitutions, only indels, and with both, each time with 1000 trials and 500 iterations. The average correlation and Euclidean distances across all trials at each iteration were recorded. For ease of comparison, all distances were normalized by dividing by the maximum average distance for all iterations. These results are presented in Figure 4.8.

While simple and contrived, a few conclusions can be drawn from these simulations. First, and most notably, correlation distance consistently converged to its maximum more slowly than Euclidean distance. That is, correlation distance more closely resembled a linear increase over a longer time interval. This suggests

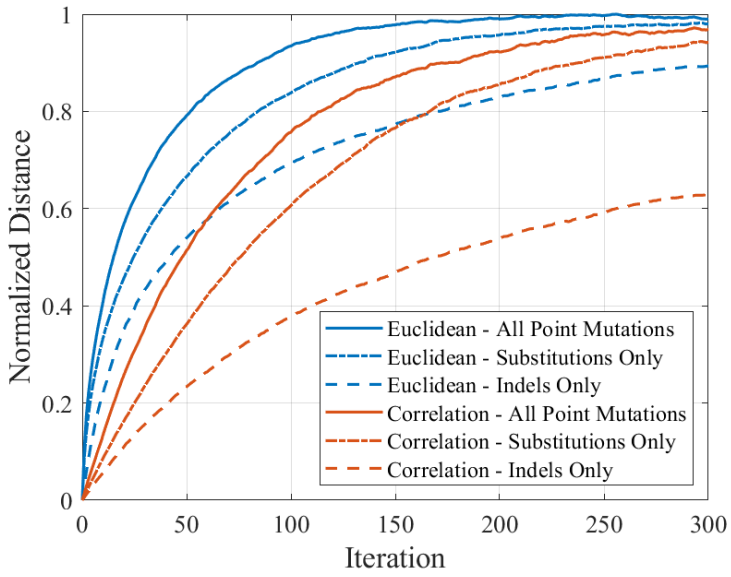


Figure 4.8: The distance between two sequences diverging via a simple point mutation evolutionary model. Distance is measured by the Euclidean and correlation distance between the sequences' AMI profiles.

that correlation distance would be more robust in correctly ranking sequences according to time since divergence, which is supported by the results of phylogenetic tree construction. Second, it suggests that Euclidean distance would perform better for more closely related species than for more distantly related species, which was true once AMI magnitudes were normalized by sequence length. Third, both distance metrics increase more quickly as indels accumulate compared to when substitutions accumulate when rates of both are the same. However, for a more realistic ratio of 10 substitutions per indel, both distance metrics increase more quickly as substitutions accumulate.

Chapter 5

Coding Region Analysis

The remaining chapters are concerned with binary genomic features that we will attempt to predict using our sequence profiles. We begin with one of the most fundamental attributes of DNA: whether or not it is protein-coding [1]. Once a genome has been sequenced, identifying the protein-coding genes strewn throughout it is one of the first and most important steps in annotation.

5.1 Coding and Noncoding Regions

Genomes are composed of coding and noncoding regions. "Coding" refers to the correspondence between a DNA sequence and an amino acid sequence that folds into a protein. The utility of coding regions has long been well understood. Proteins perform a vast array of cellular functions, and cells respond to environmental conditions by producing required proteins via transcription and translation of genes.

A gene is a segment of the genome that, when expressed, produces a functional RNA molecule or protein. A protein-encoding gene consists of one or more coding

[1] Coding region results were published in *Entropy*: G. Newcomb and K. Sayood, "Use of Average Mutual Information and Derived Measures to Find Coding Regions," *Entropy*, vol. 23, no. 10, p. 1324, Oct. 2021.

regions, which directly inform the resulting amino acid sequence, as well as interspersed noncoding regions. The vast and varied roles of noncoding regions are the subject of much current research [31]. While they do not affect the content of the protein, a gene's noncoding regions play crucial roles in the gene's expression. Noncoding regions also exist between protein-coding genes. Some of these noncoding regions are transcribed into RNA molecules that facilitate and regulate gene expression. Coding regions are distributed along both strands of the genome in segments that range from a single codon to many thousands of base pairs.

The discovery of new genes (in particular, protein-encoding genes) is aided by computational predictions of coding regions. This is possible due to observable feature differences between the two sets. For example, coding regions tend to contain higher fractions of the bases guanine (G) and cytosine (C), which is referred to as GC-content [32]. Also, coding regions exhibit triplet periodicity because they are composed of triplet codons and codon abundance is not uniform. Another strategy for identifying protein-coding genes employs interpolated Markov models. The most popular example of this technique is a tool called Glimmer [33–35], which we will use as a reference against which to frame our results. Now, we will endeavor to use sequence-derived profiles to predict whether an unknown sequence is (or contains) a coding region. Additionally, we will examine aggregate profiles for sequences belonging to the two sets to identify potential novel features.

5.2 Coding Region Prediction

5.2.1 Prediction Methodology

Numerical profiles provide a mechanism to map a nucleic acid sequence into a vector space that is readily analyzed and manipulated. Each profile presented here

defines a different space, but the techniques used to operate on the space are generic. We use linear SVMs to perform binary classification. As with any classification problem, data is partitioned into training sets and test sets. In order to objectively measure the methods' performance, the data sets are intentionally contrived.

The data sets were generated from a repository of 82 species with well-annotated genomes, consisting of 70 bacteria, 1 archaea, and 11 eukaryotes. Each species is identified by its taxonomic ID. We searched NCBI's assembly database for the most recent RefSeq assembly for each taxonomic ID. We then downloaded the genomic FASTA and GTF file for that assembly. For each GTF file, we compiled all annotated CDSs. We accepted the annotation as-is and did not speculate on the existence of possible unannotated CDSs. For large eukaryotic genomes, we used only the first 3 chromosomes. Given the CDS coordinates, we extracted each from the corresponding genome, taking the reverse complement of each CDS on the negative strand. All CDSs were then concatenated into a parent coding sequence, such that there is a single coding sequence for each species. These range in length from 500 thousand base pairs to tens of million base pairs. We then extracted the noncoding sequences. Any segment of the genome that is not included in any CDS on either strand was deemed to be noncoding. This includes all noncoding genes. Noncoding regions from both strands were included, so that noncoding regions surrounding CDSs on both strands are represented. These were all extracted and concatenated into a single parent noncoding sequence for each species. These range in length from 60 thousand base pairs to hundreds of million base pairs.

The data sets were then constructed by randomly drawing 2000 nonoverlapping sequences of constant length from each of the two parent sequences. Each data set used to train and test an SVM includes sequences of constant length, but multiple data sets were constructed with a different length for each. If the genome is too

short to allow for 2000 nonoverlapping sequences at the given length, the number of sequences in the data set is reduced accordingly. This allows us to evaluate the effect of sequence length on predictive performance for a range of 25 to 10000 base pairs.

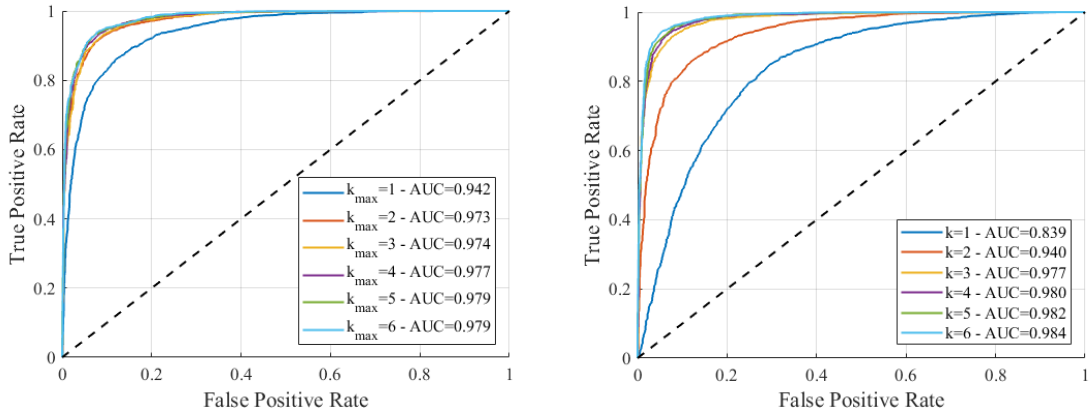
We use k-fold cross-validation, with $k = 5$. The SVM is trained on the training folds, and its performance is evaluated by using it to classify the test folds. The output of the SVM is a classification score assigned to each input profile. The magnitude of the classification score is the distance from the profile to the SVM decision boundary, and the sign specifies on which side of the boundary the profile falls. Positive scores indicate profiles on the side of the boundary corresponding to the coding region class. A higher score implies a higher probability that the test sequence was drawn from the coding region parent sequence.

We also evaluate a Euclidean distance classifier. Given a training set of sequence profiles from coding and noncoding regions, we calculate the centroid for both sets. For each sequence in the test set, we determine the Euclidean distance to both centroids, and subtract one from the other to determine classification scores. Positive values indicate that the test sequence profile is closer to the coding region centroid. Scores produced by each classifier for each type of profile are used to evaluate the classification performance. In addition to k-mer and AMI profiles, we include GC-content as a baseline.

In practice, when predicting coding regions for an unannotated genome, we would need to use a model trained on some related species. We evaluate this cross-species scenario by training a model for each species we consider, and using the model to predict coding regions in all other species.

5.2.2 Results and Discussion

Receiver Operating Characteristic (ROC) curves are generated by sweeping a prediction threshold across the entire range of scores for each prediction methodology. That is, each score that is produced by the SVM is used as a threshold to generate a point on the ROC curve. For each threshold, sequences that score higher than the threshold are declared to be coding regions, while those that score lower are declared to be noncoding regions. We then calculate the true and false positive rates, which yields a point on the ROC curve. This is repeated for all scores produced by the SVM. Example curves are shown for *S. cerevisiae* with eAMI and k-mer profiles in Figure 5.1. The area under the curve (AUC) is then used as an objective single-value metric for evaluating prediction performance. Additionally, the classifier's sensitivity and specificity are calculated using a threshold of 0. That is, coding regions assigned a positive score by the SVM are considered true positives, while noncoding regions assigned a negative score are considered true negatives.



(a) eAMI profiles (b) k-mer profiles

Figure 5.1: ROC curves for coding region prediction using SVMs on the specified type of profile

For both k-mers and AMI profiles, increasing the information available to the

classifiers yields AUC improvements. As is to be expected, the improvement tapers as k continues to increase. For eAMI, there are diminishing returns for k values greater than 2. For k-mers, AUC starts to converge once k reaches 3. For both profiles, the ROC curves are reasonably symmetric about the line y=1-x. This suggests that at the optimal decision threshold (the point on the ROC curve closest to a 0% false positive rate and 100% true positive rate), the false negative and true positive rates will be balanced.

The longer a sequence is, the more closely it will tend to resemble the aggregate profile of the set to which it belongs. Accordingly, we would expect prediction performance to increase as sequence length increases. To evaluate this effect, we constructed data sets consisting of sequences of increasing length. This is shown in Figure 5.2 for AMI profiles, and Figure 5.3 for k-mer profiles. The SVM results vary widely. For AMI, it does no better than Euclidean distance, but there is a significant performance premium for eAMI.

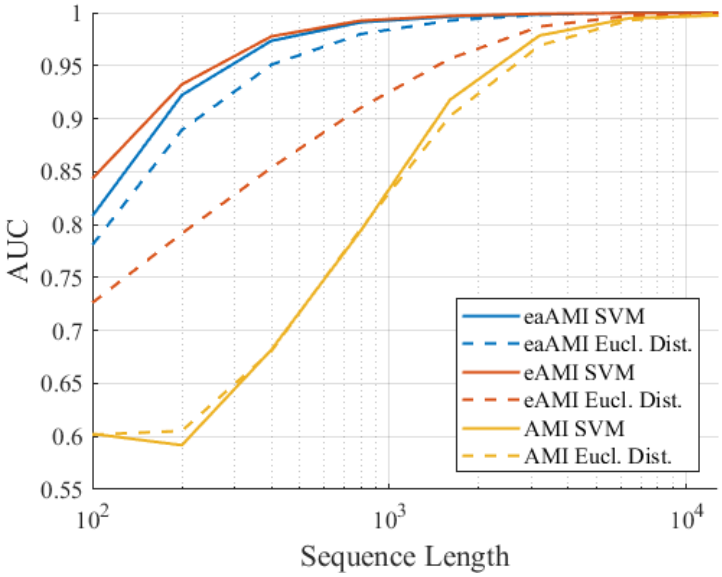


Figure 5.2: AUC for coding region prediction using SVMs and Euclidean distance on AMI profiles derived from *S. cerevisiae* sequences of increasing length

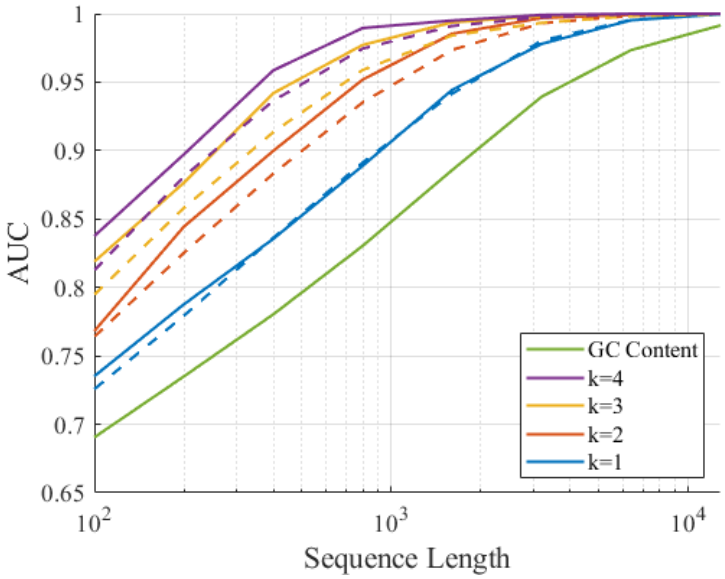


Figure 5.3: AUC for coding region prediction using SVMs (solid lines) and Euclidean distance (dashed lines) on k-mer profiles derived from *S. cerevisiae* sequences of increasing length

Additionally, the parameters used to construct the profiles impact performance. This is especially true for k-mers, as larger values of k result in more information preserved in the profile. This is shown in Figure 5.4. Notably, the performance improvement gained by the use of SVMs depends on both sequence length and the value of k. The SVM seems to become saturated with information once k reaches 6. At this point, the bulk of the information is noise, and the signal becomes lost in it. The SVM appears to perform optimally for $k = 4$, when the profile length is 256.

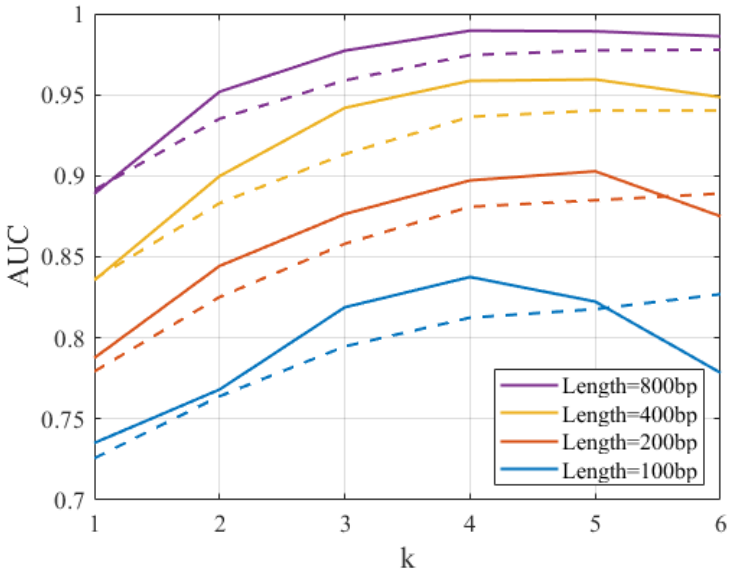


Figure 5.4: AUC for coding region prediction using SVMs (solid lines) and Euclidean distance (dashed lines) on k-mer profiles derived using increasing k values

5.3 *S. cerevisiae* Profile Analysis

For all profiles considered here, we can effectively distinguish between coding regions and noncoding regions provided we have a sequence of sufficient length. This suggests that profiles for members of both classes converge to some characteristic profile. To best represent these characteristic profiles, we calculate the centroid of profiles drawn from the longest sequences considered in this analysis (12800 base pairs). These centroids provide a visual representation of the differences between coding and noncoding regions, from which we can glean features about the class from which they were drawn.

5.3.1 AMI Profiles

Centroid AMI profiles for *S. cerevisiae* are presented in Figure 5.5. The most obvious feature is the presence of distinct peaks at multiples of three in coding

regions, resulting from the triplet periodicity conferred by codon abundance biases. The notable exception to this periodicity is the inflated values for lags 1-2, suggesting biases in the occurrences of certain nucleotide pairs and triplets. For lags greater than 4, magnitude decreases only slightly over the window considered here. The noncoding centroid also has noticeable, if less pronounced features. It too is marked by significant magnitudes for small lags, but with more gradual degredation thereafter. Curiously, there appear to be small but significant peaks at even lags from 6-14.

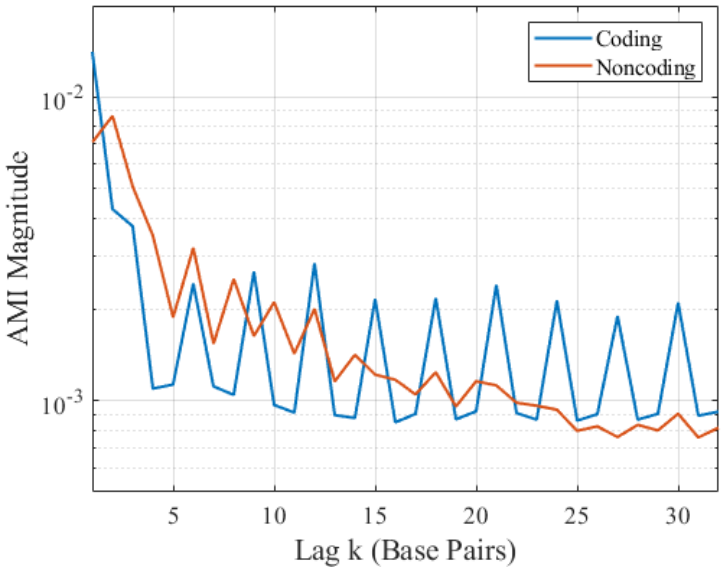


Figure 5.5: Centroid AMI profiles for coding and noncoding regions from *S. cerevisiae*

Centroid eAMI profiles are presented in Figure 5.6 for lags 1-4. Subsequent lags are omitted for brevity, but they bear resemblance to those presented. The presence of strings of thymine is most indicative of a noncoding region. This would occur in the opposite strand of a poly(A) tail downstream of a coding region. Noncoding regions are more symmetric, in the sense that complementary nucleotide pairs have similar abundance. This is due to the relatively higher likelihood of the opposing

strand of a noncoding region also being noncoding.

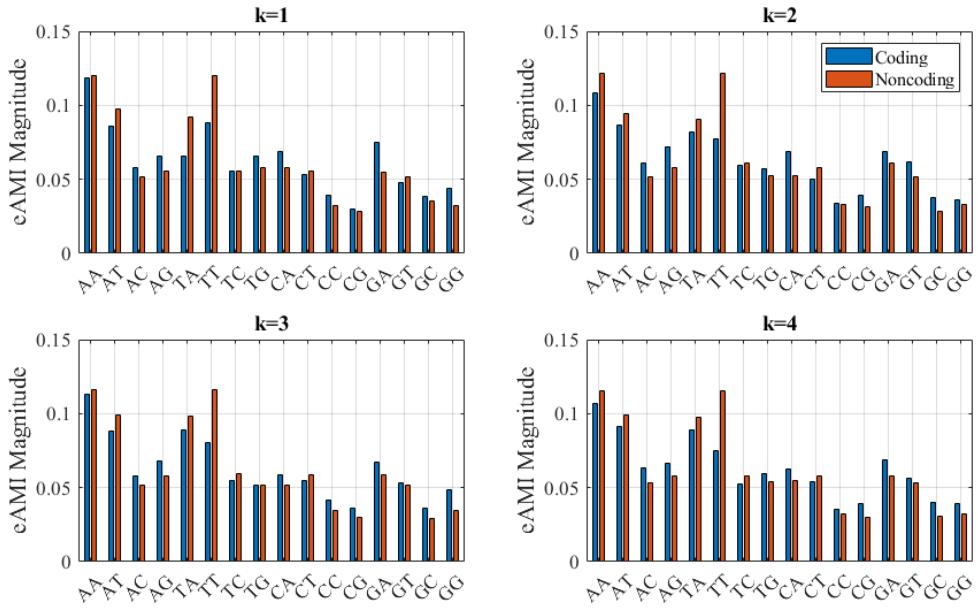


Figure 5.6: Centroid eAMI profiles for coding and noncoding regions from *S. cerevisiae*

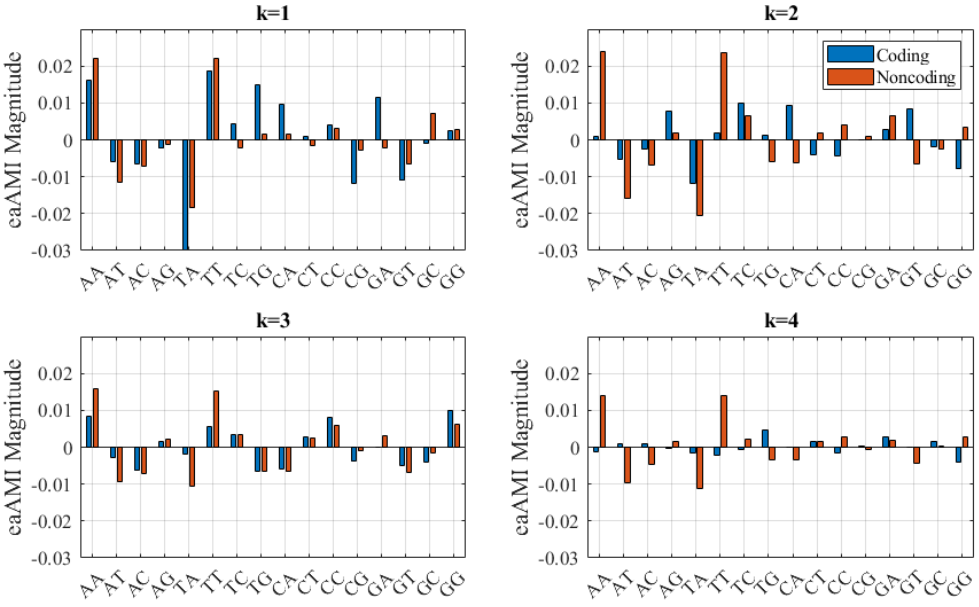


Figure 5.7: Centroid eaAMI profiles for coding and noncoding regions from *S. cerevisiae*

Centroid eaAMI profiles are presented in Figure 5.7 for lags 1-4. Again, subsequent lags are omitted for brevity.

5.3.2 k-mer Profiles

Centroid k-mer profiles for coding and noncoding regions of length 12800 are presented in Figure 5.8 for $k = 1 - 3$. Subsequent values of k are omitted for brevity. Similar themes are demonstrated in both the k-mer profiles and the eAMI magnitudes. In particular, 2-mer profiles are inherently identical to eAMI profiles for $k = 1$. k-mers consisting of consecutive thymine nucleotides are enriched in noncoding regions, as are k-mers of alternating AT. GC content is higher in coding regions, as expected, and this is captured in the 1-mer profile. It should be noted that 3-mers are not a direct measure of codon abundance. The k-mer profiles include the occurrence of each overlapping k-mer in a sequence. Codons in a coding sequence are thus included, but so are k-mers comprised of part of two adjacent codons. Several of the 3-mers that occur at higher rates in coding regions consist of adenosine and guanine, particularly AGA and GAA.

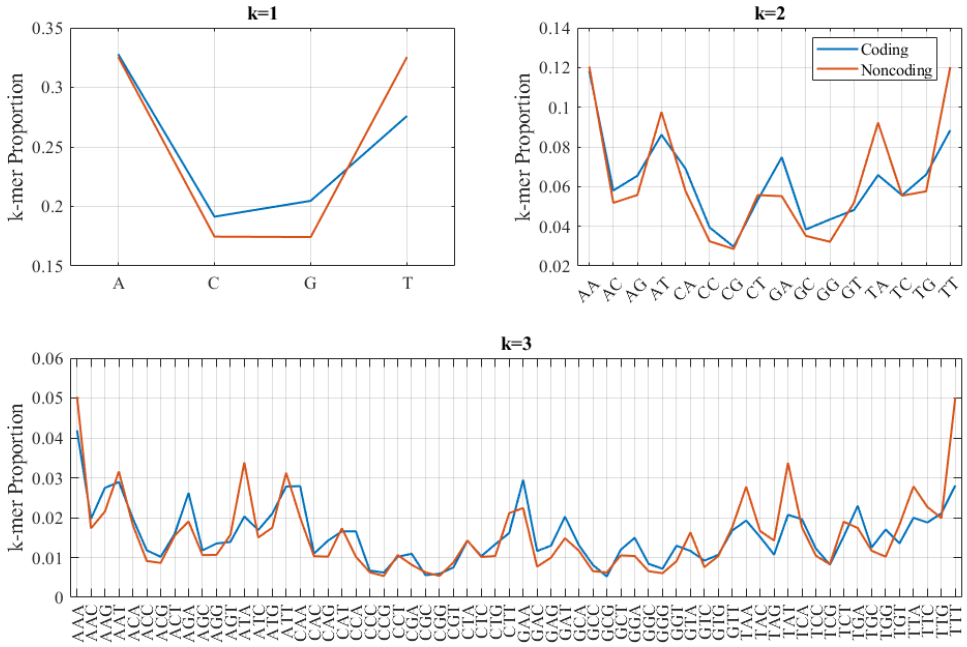


Figure 5.8: Centroid k-mer profiles for coding and noncoding regions

5.4 AMI Convergence Behavior

As noted earlier, mean AMI is approximately inversely proportional to length for randomly-generated sequences. This presents difficulties in comparing the profiles of short sequences that vary in length. One strategy for remedying the inconsistent magnitudes is to multiply AMI values by sequence length in order to normalize them. However, the utility of this normalization depends on how closely the subject sequence resembles a random sequence. A random sequence's AMI will converge to zero, while a structured sequence's AMI will converge to some value greater than zero. This is true of both coding and noncoding regions, as seen in Figure 5.9. Interestingly, the two categories have very similar AMI magnitudes, though the profiles themselves are markedly different. Sequences of roughly 1000 or more base pairs diverge from the random baseline.

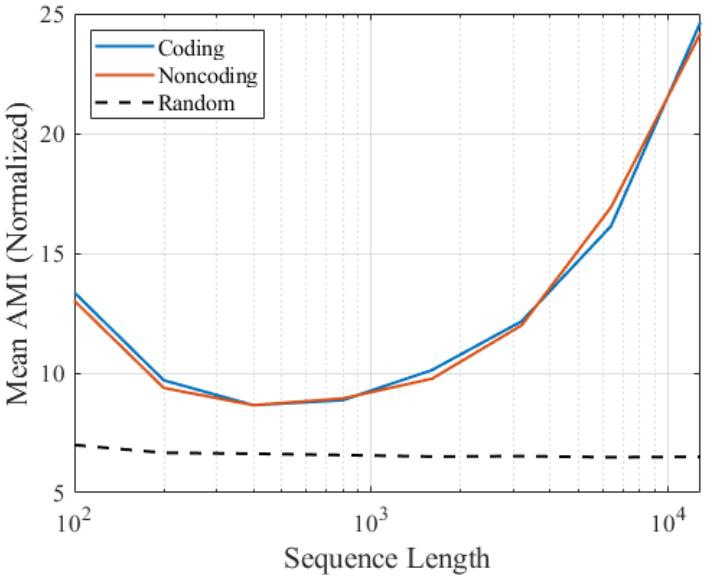


Figure 5.9: Mean AMI values (normalized by sequence length) for coding and noncoding regions from *S. cerevisiae*, with values for random sequences shown for reference

5.5 All Species Predictions

In order to evaluate the robustness of this method, we applied it to 82 genomes, consisting of 70 bacteria, 1 archaea, and 11 eukaryotes. For length 100 base pair sequences, eAMI produced the highest AUC in 78 of the 82 species considered. AUC results for all profile types and species are summarized in the histograms in Fig. 5.10. The eAMI SVM performed best for the lone archaea, *M. maripaludis* (0.971 AUC, 92.7% sensitivity, and 89.6% specificity), and worst for *A. nidulans* (0.826 AUC, 80.5% sensitivity, and 68.7% specificity).

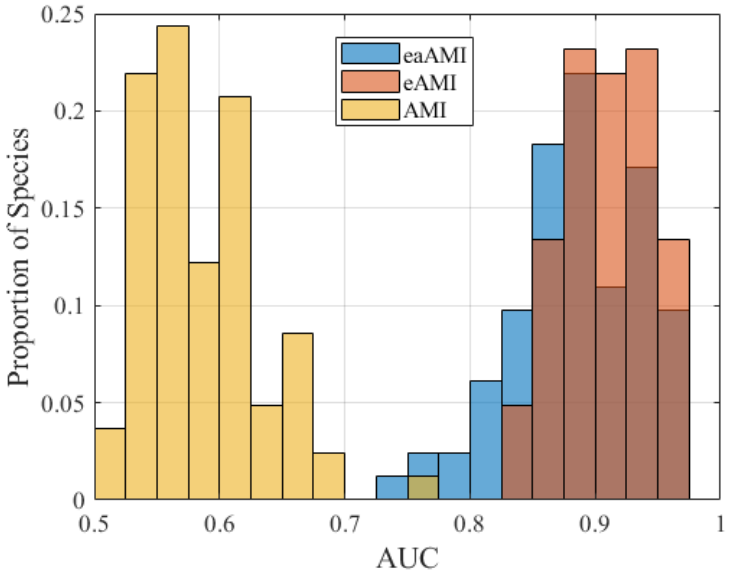


Figure 5.10: AUC distribution for all three profiles across all 82 species.

For length 1000 base pair sequences, eaAMI outperforms eAMI in 70 of the 82 species. This suggests that for shorter sequences, eAMI provides better discrimination, while for longer sequences, eaAMI is preferred. The length threshold at which eaAMI begins to outperform eAMI is about 250 base pairs, as shown in Figure 5.11. Figure 5.11 also shows the impact of using an SVM trained on a set of 1000 base pair sequences, rather than an SVM trained on a set of sequences the same length as the test sequences. There is virtually no degradation in performance for eAMI profiles, and only a significant impact on very short sequences for eaAMI profiles. This is practically beneficial, as it implies there is no need to train models for many different lengths in order to make optimal predictions on sequences that vary in length. Generally, we would only need to train two models: an eAMI model trained on 250 bp sequences, and an eaAMI model trained on 1000 bp sequences.

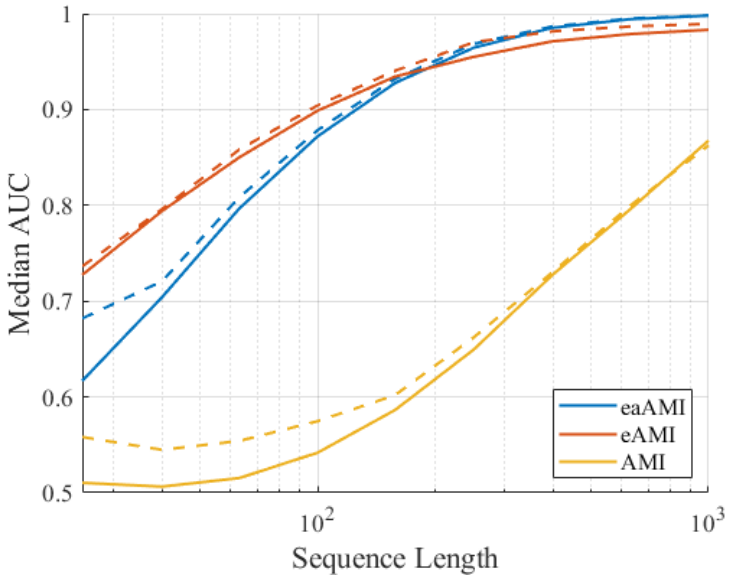
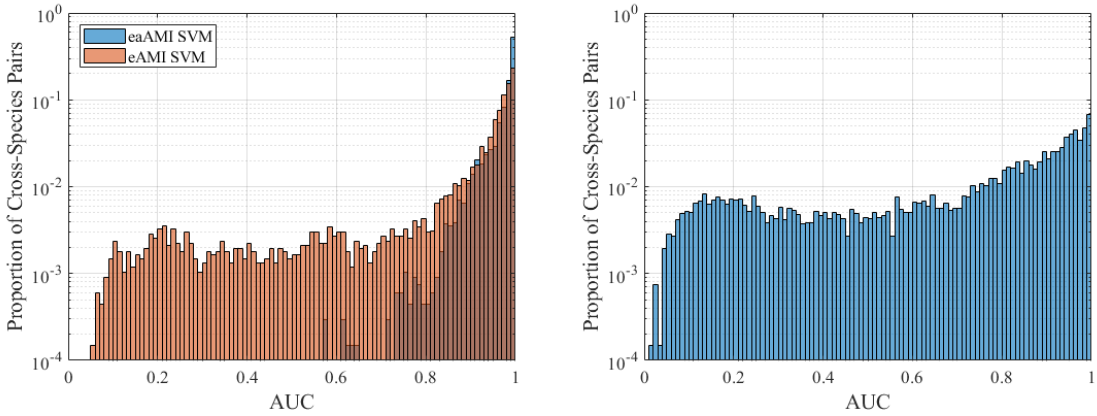


Figure 5.11: Median AUC across all 82 species for SVMs on each type of profile given sequences of increasing length. Solid lines are results when the SVM was trained on a set of 1000 bp sequences. Dashed lines are results when the SVM was trained on a set of sequences the same length as the test sequences.

5.6 Cross-Species Predictions

Many of the structural constraints imposed on coding (and noncoding) regions are common to all known life. The cellular machinery used to produce proteins is fundamentally identical irrespective of species; the chemistry dictating the protein's behavior is inescapable; the protein functions required for survival and reproduction have considerable overlap across all of life's diversity. These similarities imply the existence of universal sequence features that the profiles should be able to extract. We can evaluate this by training an SVM on one species and using that model to predict whether sequences from a different species are coding. For closely related species, there are likely to be coding region features specific to their evolutionary lineage. For disparate species, any success achieved by the model is likely to reflect universal features common to all coding regions.



(a) eAMI and eaAMI profile SVMs (b) Glimmer

Figure 5.12: AUC distributions across all pairwise cross-species predictions

AUC results for all profile types and species are summarized in the histograms in Fig. 5.12 for length 1000 base pair sequences. Analogous results obtained using Glimmer are shown for reference. Median AUC, sensitivity, and specificity for each profile are shown in Table 5.1. Sensitivity is consistently higher than specificity when applying the SVM to different species, indicating that SVM scores for both coding and noncoding sequences are shifted higher on aggregate compared to the sequences used for training. To remedy this, one may wish to increase the decision threshold. This would improve sensitivity with only limited impact on specificity. As with single-species results for this sequence length, eaAMI produces better results on average than eAMI, and is also considerably more consistent. 95.4% of cross-species predictions made using eaAMI profiles resulted in AUC greater than 0.9, compared with 76.6% of predictions using eAMI profiles.

Table 5.1: Median cross-species results for all profiles using length 100 and 1000 base pair sequences

	100 bp Sequences			1000 bp Sequences		
	AUC	Sensitivity	Specificity	AUC	Sensitivity	Specificity
eaAMI	0.788	0.606	0.837	0.991	0.904	0.985
eAMI	0.847	0.672	0.840	0.970	0.703	0.983
AMI	0.515	0.494	0.540	0.737	0.678	0.727
Glimmer	0.694	0.253	0.884	0.832	0.122	0.996

Cross-species results for several members of the *Aspergillus* genus are presented in Table 5.2. All predictions had AUC scores of at least 0.97. Results for several members of the Gammaproteobacteria class are presented in Table 5.3. All are well known human pathogens. Despite being less closely related (i.e. belonging to the same class instead of the same genus), there seem to be more identifiable similarities in their coding region sequences. Most cross-species SVM scores have near-perfect separation between coding and noncoding sequences, with no AUC scores less than 0.98.

Table 5.2: Cross-species coding region prediction results for selected *Aspergillus* species using eaAMI profiles. SVMs were trained on length 1000 sequences from the species denoted in the row headers, and used to classify sequences from the species denoted in the column headers.

	A. nidulans			*A. fumigatus*			*A. niger*			*A. oryzae*		
Train	AUC	SN	SP	AUC	SN	SP	AUC	SN	SP	AUC	SN	SP
A. nid	0.98	0.94	0.93	0.99	0.96	0.93	0.99	0.96	0.93	0.97	0.92	0.91
A. fum	0.98	0.93	0.94	0.99	0.94	0.95	0.99	0.94	0.94	0.97	0.89	0.93
A. nig	0.98	0.95	0.92	0.99	0.96	0.92	0.99	0.96	0.93	0.97	0.92	0.90
A. ory	0.98	0.97	0.90	0.99	0.97	0.89	0.99	0.97	0.91	0.98	0.94	0.91

Table 5.3: Cross-species coding region prediction results for selected species of the Gammaproteobacteria class using eaAMI profiles.

	A. baumannii			*S. pneumoniae*			*S. enterica*			*V. cholerae*		
Train	AUC	SN	SP	AUC	SN	SP	AUC	SN	SP	AUC	SN	SP
A. bau	1.00	0.97	0.99	1.00	0.88	0.99	1.00	0.96	0.99	1.00	0.96	0.99
S. pne	1.00	0.97	0.98	1.00	0.98	0.99	0.99	0.87	0.98	0.99	0.94	0.94
S. ent	0.99	0.88	0.99	0.99	0.86	0.99	1.00	0.97	1.00	1.00	0.94	1.00
V. cho	0.99	0.95	0.97	0.99	0.88	0.99	1.00	0.97	0.98	1.00	0.98	0.99

Results for a broad range of organisms are presented in Table 5.4. All three domains of life are represented, along with two Eukaryotic kingdoms. Perhaps most striking is that models trained on all four species could nearly perfectly predict coding regions in the Archaea *M. maripaludis*. The mold *A. nidulans* was most difficult to predict, with an AUC as low as 0.92. Still, considering the vast time scale over which these species have been diverging, eaAMI appears to be highly robust in differentiating coding and noncoding regions across all life.

Table 5.4: Cross-species coding region prediction results for selected highly divergent species using eaAMI profiles.

	H. sapiens			*A. nidulans*			*M. maripaludis*			*S. pneumoniae*		
Train	AUC	SN	SP	AUC	SN	SP	AUC	SN	SP	AUC	SN	SP
H. sap	0.99	0.94	0.97	0.93	0.69	0.94	0.99	0.74	1.00	0.99	0.83	1.00
A. nid	0.93	0.66	0.97	0.98	0.94	0.93	1.00	0.98	0.99	0.99	0.92	0.96
M. mar	0.89	0.59	0.98	0.93	0.66	0.94	1.00	1.00	1.00	0.99	0.84	1.00
S. pne	0.94	0.86	0.89	0.92	0.93	0.70	1.00	0.98	0.99	1.00	0.98	0.99

Table 5.5: Cross-species coding region prediction results for selected species of the Gammaproteobacteria class using Glimmer.

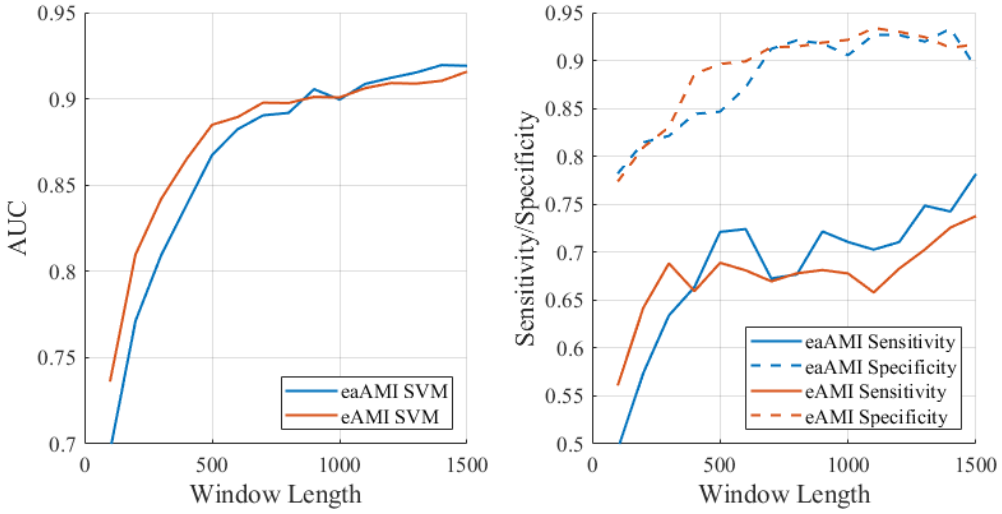
	A. baumannii			*S. pneumoniae*			*S. enterica*			*V. cholerae*		
Train	AUC	SN	SP	AUC	SN	SP	AUC	SN	SP	AUC	SN	SP
A. bau	0.92	0.94	0.44	0.79	0.69	0.83	0.30	0.12	0.65	0.64	0.62	0.57
S. pne	0.80	0.73	0.80	0.96	0.98	0.44	0.21	0.03	0.87	0.41	0.24	0.78
S. ent	0.91	0.32	0.99	0.85	0.07	1.00	0.98	0.99	0.43	0.94	0.66	0.99
V. cho	0.96	0.77	0.97	0.93	0.54	0.98	0.91	0.88	0.76	0.98	0.98	0.59

5.6.1 Genome Scanning Predictions

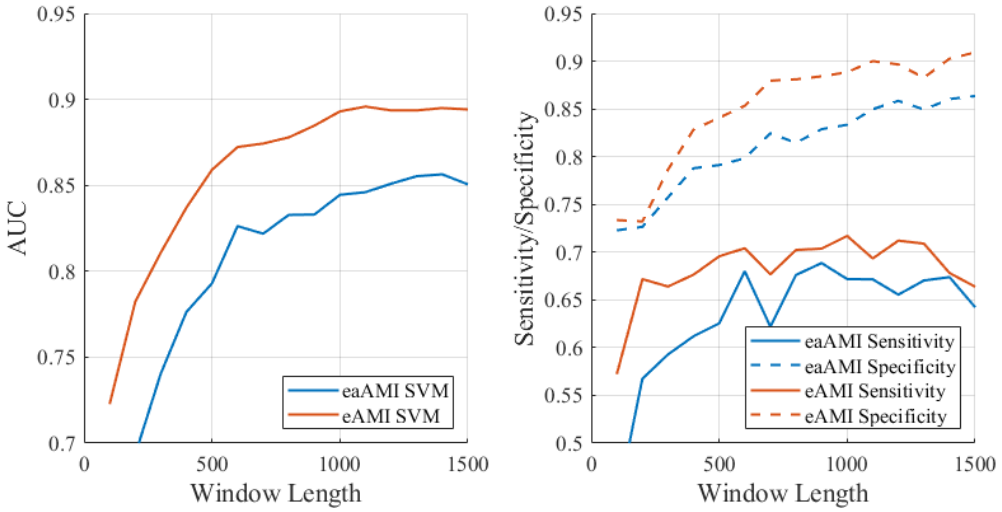
While the above results are indicative of this method's utility in differentiating coding and noncoding regions, in practice the sequences of interest will not be purely coding or noncoding. When searching for genes in an unannotated genome, we may slide a window along the genome and apply the trained model on that window. If the window includes a gene's start and/or stop codon, or if it includes both introns and exons, it will include both coding and noncoding segments. For the purpose of this binary analysis, we declare the window to be coding if it consists of at least 50% coding regions.

Results of this procedure for a pair of *Aspergillus* species and a pair of Gammaproteobacteria species appear in Figure 5.13. For 1000 base pair sequences, there is about a 0.1 AUC deterioration compared to the pure coding/noncoding region predictions. Interestingly, eAMI appears to outperform eAMI, particularly for the *Aspergillus* pair. This is likely due to eAMI's superior performance for shorter pure sequences, as this suggests eAMI requires less information in order to provide some level of differentiation. While this measure of AUC improves with an increasing window length (up to about 1000 base pairs), this is achieved by sacrificing resolution. Shorter window lengths may better facilitate identifying where a gene begins and ends, while longer window lengths are better at determining if a gene is present. The presence of introns would certainly complicate such gene-finding endeavors.

For species whose protein-coding genes have few introns, locating coding regions is much simpler. We can first identify open reading frames (ORFs) along the genome, then apply the prediction methodology to this list of ORFs. Because we observed that the models are largely agnostic to length, ORFs of virtually any



(a) Results for *V. cholerae* using models trained on *S. enterica*



(b) Results for *A. fumigatus* using models trained on *A. niger*

Figure 5.13: Prediction results for genomes partitioned into variable-length sequences. For each window length (in base pairs), an SVM is trained on the genome of a species closely related to the species of interest.

length can be tested with the same model. Since longer sequences have better predictive performance, we would have more confidence in a prediction made about a long ORF than one made about a short ORF.

Chapter 6

Essential Genes

Supposing we have now successfully identified all of the protein-coding genes on a newly sequenced genome, there are still many questions that remain. One such question is how critical is each gene to the organism's survival. Not all genes are expressed at all times, so there are many genes that a given organism could live without under normal conditions. Identifying which genes are needed and which are not has important ramifications in fighting infections diseases.

6.1 Essential Gene Background

Organisms live and reproduce in a variety of environmental conditions. These various conditions expose the genome to evolutionary pressures that produce genes functionally suitable to the environment. Accordingly, there are genes whose products increase fitness in some environments, but are not necessary (or even expressed) in others. Conversely, each genome includes a set of genes fundamental to the organism's continued existence in all environments. These are aptly termed "essential genes".

The vital nature of essential genes makes them a topic of research interest. For

example, consider the case of developing therapies to combat an infectious bacteria. If the bacteria's set of essential genes is known, such a therapy could be designed to target the product of one of those genes.

While essential genes themselves have many functions, their importance subjects them to unique evolutionary pressures. Mutations in essential genes that produce functional differences may result in a nonviable organism. This may explain why essential genes tend to be better conserved than nonessential genes, at least in bacteria [36]. Similarly, genes of younger phyletic age are less likely to be essential, as are duplicate genes [37].

6.2 Essential Gene Prediction

Many studies have endeavored to identify essential genes in various species. Given the rigor required to conduct these experiments, it would be helpful if we could predict whether a particular gene is essential. Experimental results provide a robust training set for use in developing computational predictors. Further, several groups have compiled databases of essential genes identified in these studies. The Database of Essential Genes (DEG) has been used to train classifiers on information-theoretic measures such as mutual information [38]. Our investigation focuses on the Online Gene Essentiality Database (OGEE), which includes a larger swath of experimental datasets than DEG [39]. In particular, the OGEE set contains more experimentally-confirmed nonessential genes. This is important because our prediction methodology requires a training set consisting of both classes. As with DEG, OGEE has been used to train effective computational predictors [40].

6.2.1 Prediction Methodology

OGEE provides a list of genes along with their essentiality status. In order to use a binary classifier, we included only those genes classified as essential (E) or nonessential (NE). For each taxonomic ID represented, we retrieved the GTF and CDS files from the current RefSeq assembly. We then searched the GTF file with the OGEE gene entries and populated sets of essential and nonessential gene sequences with hits. We consider all taxonomic IDs with at least five annotated essential and nonessential genes. This leaves 25 bacteria, 1 archaea, and 9 eukaryotes. As with the other binary prediction methodologies documented here, we train an SVM on profiles derived from the sequences. We use k-fold cross-validation, with $k = 5$. In order to ensure a balanced dataset, we exclude all excess members of whichever class is larger.

6.2.2 Results

A list of species along with their respective AUCs is provided for prokaryotes in Table 6.1 and for eukaryotes in Table 6.2. The median AUC was 0.719 for eukaryotes and 0.741 for prokaryotes, indicating marginally better performance for species belonging to the latter kingdom. In both kingdoms, the range over which AUC varies is quite wide.

Even among strains of the same species, performance is not entirely consistent. For example, *Salmonella enterica subsp. enterica serovar Typhi str. Ty2* (Taxonomic ID 209261) has an AUC of 0.817, while *Salmonella enterica subsp. enterica serovar Typhimurium str. SL1344* (Taxonomic ID 216597) has an AUC of 0.700. However, this performance difference may be exaggerated by the underlying data. The former strain has 2309 documented essential genes, while the latter strain

Table 6.1: AUC for essential gene prediction for prokaryotes included in the OGEE database

Taxonomic ID	Species Name	Essential	Nonessential	AUC
1140	Synechococcus elongatus	691	1699	0.742
62977	Acinetobacter baylyi	251	2493	0.733
83332	Mycobacterium tuberculosis	693	3132	0.64
83333	Escherichia coli	296	3994	0.816
93061	Staphylococcus aureus	347	2420	0.774
176299	Agrobacterium fabrum	5	90	1
192222	Campylobacter jejuni	267	1224	0.62
205921	Streptococcus agalactiae	290	1614	0.818
208963	Pseudomonas aeruginosa	444	5348	0.741
208964	Pseudomonas aeruginosa	336	5179	0.779
209261	Salmonella enterica	2309	2065	0.817
216591	Burkholderia cenocepacia	162	3218	0.666
216597	Salmonella enterica	896	2782	0.7
220341	Salmonella enterica	421	3686	0.74
224308	Bacillus subtilis	228	3945	0.81
243273	Mycoplasma genitalium	381	94	0.645
267377	Methanococcus maripaludis	415	1066	0.736
272635	Mycoplasma pulmonis	407	272	0.704
293653	Streptococcus pyogenes	85	131	0.715
373153	Streptococcus pneumoniae	110	166	0.648
388919	Streptococcus sanguinis	217	1979	0.803
471876	Streptococcus pyogenes	240	1132	0.67
528354	Neisseria gonorrhoeae	695	1154	0.691
565050	Caulobacter vibrioides	543	3105	0.758
633149	Brevundimonas subvibrioides	414	2760	0.743
941322	Escherichia coli	294	4088	0.781

has 896. This suggests that different methods may have been used to determine essential genes in each strain, and those differing methods have produced gene sets with somewhat dissimilar feature profiles. In effect, the annotations provide a lower

bounds on the performance of a computational predictor. Annotations that are inaccurate or incomplete will necessarily be detrimental to such efforts.

Table 6.2: AUC for essential gene prediction for eukaryotes included in the OGEE database

Taxonomic ID	Species Name	Essential	Nonessential	AUC
3702	Arabidopsis thaliana	484	115	0.559
5823	Plasmodium berghei	1191	905	0.675
6239	Caenorhabditis elegans	503	7255	0.903
7227	Drosophila melanogaster	399	7028	0.719
36329	Plasmodium falciparum	3284	1989	0.653
185431	Trypanosoma brucei	82	132	0.747
330879	Aspergillus fumigatus	100	156	0.777
425011	Aspergillus niger	7	75	0.776
580240	Saccharomyces cerevisiae	919	3616	0.651

6.2.3 Profile Analysis

This method produced the most accurate predictions for *C. elegans* and *S agalactiae*. This excludes *A. fabrum*, which has only 5 annotated essential genes. The majority of the elements included in the profiles used to train the SVMs contribute little. We can evaluate the merits of each by sweeping a prediction threshold across the range of magnitudes and calculating the AUC. The distribution of AUC values for all features considered is shown in Figure 6.1. The bulk of these values are near 0.5, indicating that the feature magnitude is similar in essential and nonessential genes. Such features are not good candidates for distinguishing between the two sets.

Despite the evolutionary distance between them, there is still considerable overlap in high-AUC features between the two species considered here. *C. elegans* has 104 features with an AUC above 0.65, while *S. agalactiae* has 23. Of those, 17

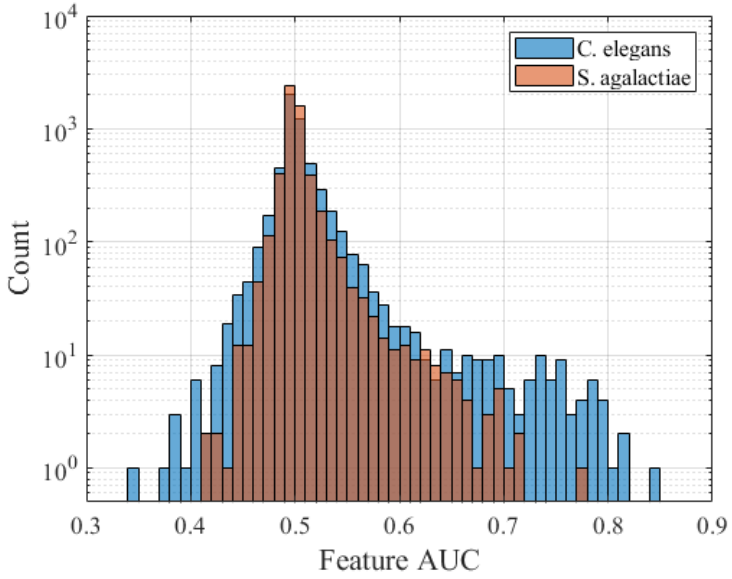


Figure 6.1: Distribution of AUCs for each individual feature included in the profiles considered

are in common. The average magnitudes of those selected features in essential and nonessential genes are shown in Figure 6.2. It is clear that many of those features will be highly correlated with each other. For example, since AT is a substring of ATA, the abundance of each will be linked. Likewise, AT is a significant nucleotide pair in the eAMI profiles for both $k = 1$ and $k = 2$. This partially explains why the SVM performs only slightly better than the most discriminating individual feature.

6.2.3.1 Effect of Dimensionality Reduction

In order for the SVM to discriminate between classes, there must be separation between the training data for members belonging to those classes. This is true of any classifier utilizing a supervised machine learning model. This separation can be present in any dimension(s). As long as one such dimension exists, the other dimensions in the data will have negligible effects on the classifier's accuracy even if they have no predictive value. For this reason, it is tempting to be liberal in the

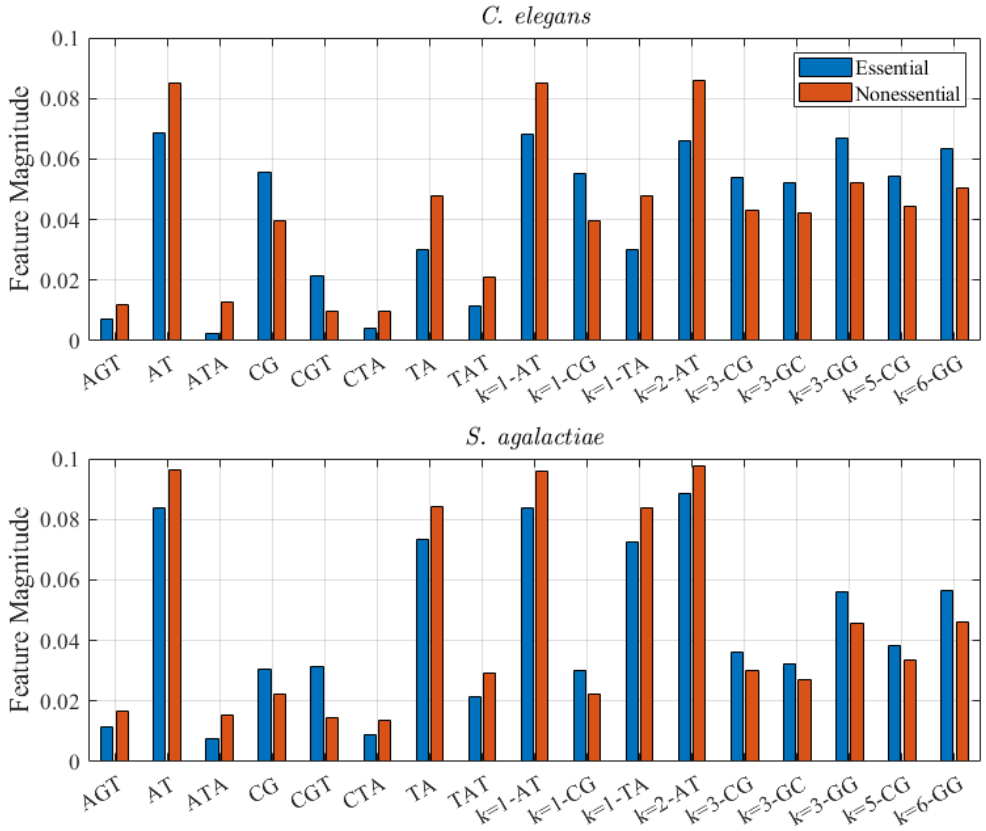


Figure 6.2: Comparison of selected features from profiles for essential and nonessential genes in a prokaryote and a eukaryote. Labels of the form "$k = n - N_1N_2$" indicate an element of the eAMI profile with lag n and nucleotide pair N_1N_2. The remaining labels represent k-mers.

quantity of information included in the training data. However, there is a practical reason to be judicious. Given higher dimension data, the SVM will require more time to converge. Thus, we would prefer to avoid saturating the model with information for no other reason than blind hope. As noted in Figure 6.1, the vast majority of the elements in the profile have virtually identical compositions in both classes. This is not a guarantee that the element will not assist the classifier, but it is strongly indicative.

As mentioned earlier, we can measure the individual contributions of each

component by calculating the AUC derived by considering that component in isolation. We can then assemble a low dimension profile by ranking components according to their AUC and selecting only those with the highest rank. As long as we include a sufficient subset of the original profile, we can achieve the same predictive performance while significantly reducing the time required. In order to determine how many components we need to keep, we repeat the essential gene prediction methodology while increasing the cardinality of the profiles. At each step, we calculate the average relative AUC across all species. "Relative AUC" refers to the ratio of AUC for low dimension data to the AUC for full dimension data. The results of this are presented in Figure 6.3 (plot labeled "High-AUC Features"). They suggest that our profile needs to include at least 20 of the features with the highest AUC.

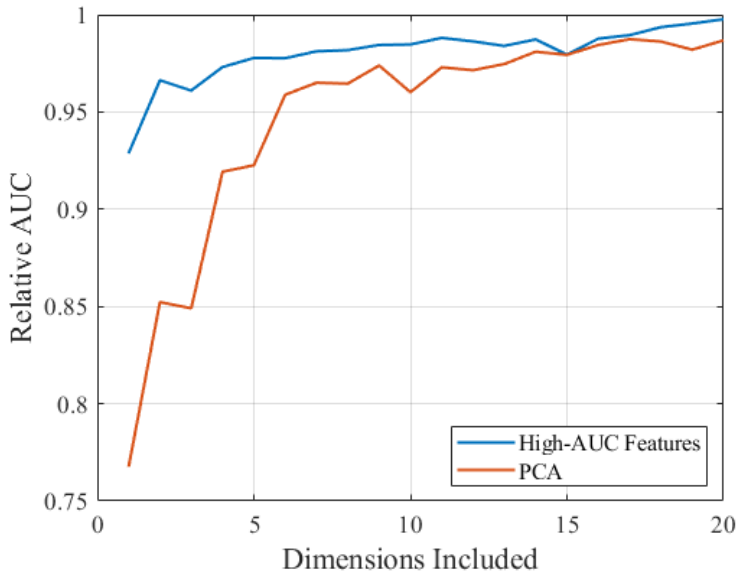


Figure 6.3: Effect of dimensionality reduction on predictive performance, as measured by AUC (relative to AUC without dimensionality reduction applied)

We also examine the impact of applying PCA to the data. Average relative AUC given dimensionality-reduced data via PCA is shown in Figure 6.3. While AUC

degrades significantly if only a few principle components are included, it increases quickly as more are added. AUC continues to increase slowly for more than 6 components.

6.2.4 Leave-One-Out Predictions

Training and testing a different SVM for each species provides a proof of concept and yields insights into what distinguishes essential and nonessential genes for that species. However, a more practically useful application would be to train the SVM on some composite set of annotated genomes and to use this model to predict essential genes for a novel genome. We can test this case by assigning one genome as "novel" and training the model on the remaining annotated genomes. This approach is called leave-one-out cross-validation. Repeating this for each genome gives us a general idea of the accuracy of such real-world predictions. These results are provided in Table 6.3. For reference, analagous results reported in [2] are also provided.

For single genome predictions, the results using the combined AMI and k-mer profiles are very similar or even slightly better. The median AUC for our method was 0.741, compared to 0.735 for Liu et al. In contrast, our method performed markedly worse in making leave-one-out predictions (0.685 median AUC, compared to 0.771). This is somewhat surprising given that there was substantial similarity in the high-AUC elements of the profiles for even distantly related species. It is possible that better results could be achieved if the test were repeated using profiles consisting of only the high-AUC elements. Alternatively, cross-species training could be used rather than leave-one-out, which would still work for predicting essential genes for a newly sequenced genome. A single species used for training may provide a better template for the SVM than the potentially noisier leave-one-out framework.

Table 6.3: Comparison of results with those obtained by Liu et al. [2] for both single genome predictions (denoted "AUC-S") and leave-one-out predictions (denoted "AUC-LOO")

Taxonomic ID	Species Name	AUC-S	AUC-LOO	AUC-S	AUC-LOO
		Liu et al.		Our Method	
62977	A. baylyi	0.775	0.753	0.733	0.687
83332	M. tuberculosis	0.674	0.699	0.640	0.547
83333	E. coli	0.735	0.833	0.816	0.762
93061	S. aureus	0.789	0.802	0.774	0.685
192222	C. jejuni	0.557	0.552	0.620	0.502
208963	P. aeruginosa	0.658	0.648	0.741	0.658
208964	P. aeruginosa	0.670	0.657	0.779	0.692
209261	S. enterica	0.721	0.845	0.817	0.585
216597	S. enterica	0.738	0.788	0.700	0.482
224308	B. subtilis	0.803	0.771	0.810	0.792
272635	M. pulmonis	0.768	0.642	0.704	0.645
293653	S. pyogenes	0.724	0.832	0.715	0.692
388919	S. sanguinis	0.751	0.813	0.803	0.740

Chapter 7

Gene Function and Location

Perhaps the most interesting (and most difficult) question about a protein-coding gene concerns the function of that protein. While this is not necessarily a question with a binary answer, we can formulate it as a binary classification problem by asking whether a gene belongs to a particular functional category or not. This allows us to continue with our prediction methodology from the previous chapters unabated. To supplement this, we also apply enrichment analysis to lists of genes sorted by profile distance to demonstrate the statistical link between function and profiles.

7.1 Gene Ontology Background

A protein's function is determined by its underlying amino acid sequence and the environment in which it exists. While environment varies widely and is difficult to characterize computationally, its sequence is reasonably fixed, and increasingly accessible. Thus, the protein's primary amino acid sequence (and the DNA sequence of its corresponding gene) can be used to predict functional information about the protein. The accuracy of such predictions can be measured by comparing them to

the protein's experimentally-determined gene ontology (GO) annotations. The GO Consortium has developed a systematic, hierarchical vocabulary of terms for describing the characteristics of proteins under three domains: molecular function, biological process, and cellular component [41, 42]. The annotation for each protein consists of a list of GO terms, and, implicitly, all of those terms' ancestors. Thus, functional predictions are made by selecting a set of GO terms that are believed to apply to a given protein.

7.2 GO Term Enrichment Analysis

Prior to attempting GO term predictions, we will attempt to demonstrate that these profiles contain information about the GO term annotations generally. Evaluation of the various profiles is done using GO term enrichment analysis. This is performed by the **ermineJ** gene set analysis tool, which identifies enriched terms in ranked lists of genes [43]. A simple method of generating such lists is to sort all genes' profiles according to their distance to some target gene of interest. For each distance metric, the same subset of 1000 randomly selected target ORFs are analyzed. All other ORFs are ranked with respect to each of these target ORFs, and the 1000 ranked lists are used as input to ermineJ. We use ermineJ's ROC method of determining GO term significance. GO terms that cluster towards the beginning of the ranked lists will produce an ROC curve with higher AUC, and this indicates that the term is significant. If the profiles have no correlation to the GO terms, these lists will be random and no enriched GO terms will be identified.

Instead, we find that the lists provided to ermineJ each produce many purportedly enriched GO terms. This is shown in Figure 7.1. This is evidence that the GO terms are related to the profiles in some way. The nature of that

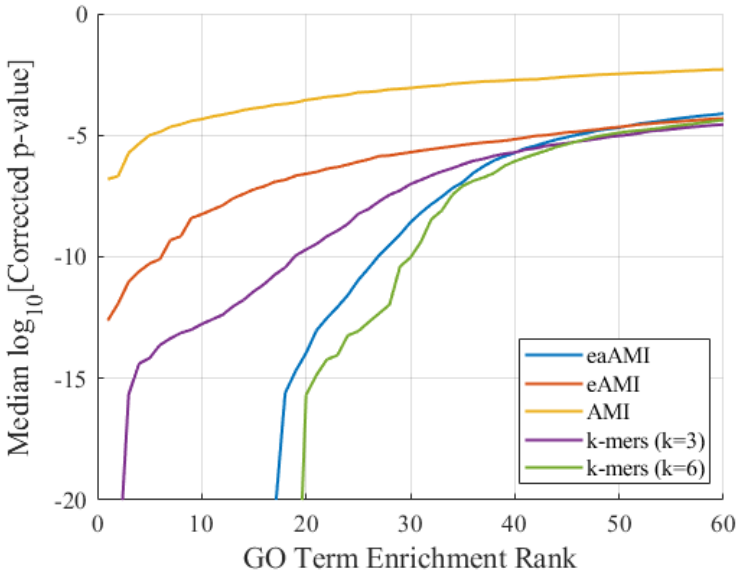


Figure 7.1: Median GO term enrichment (as measured by $\log_{10}$[Corrected p-value]) for the 60 most enriched terms output for each of the 1000 target genes.

relationship is murky, however. It is tempting to suppose that a statistically enriched term is likely to belong to the target gene's GO annotation. This turns out to not be the case. In fact, there is significant overlap in the sets of enriched terms identified for each target. These common terms may still have some relation to the type of profile used to produce them, but it is clear they do not provide information about the gene in which we are interested.

Thus, we discard enriched terms if ermineJ indicates they are enriched in at least 10% of the target genes analyzed. We are left with much more realistic p-values, as shown in Figure 7.2. The average number of enriched terms identified using each profile is presented in Figure 7.3. Interestingly, 6-mers produced terms with the lowest p values, but the fewest enriched GO terms after pruning common terms. This illustrates the misleading nature of the unpruned term lists. The pruned lists likely give a more accurate picture of the various profiles' capacity to predict GO terms.

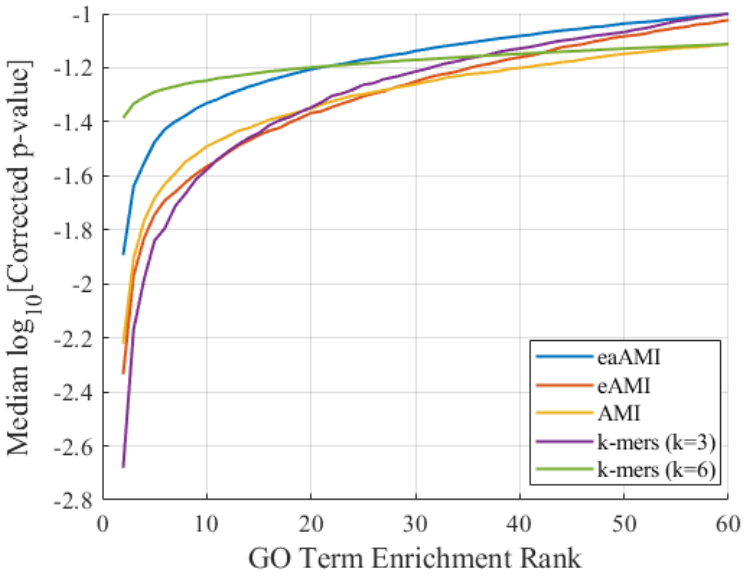


Figure 7.2: Median GO term enrichment (as measured by $\log_{10}$[Corrected p-value]) for the 60 most enriched terms output for each of the 1000 target genes. Commonly enriched terms are removed.

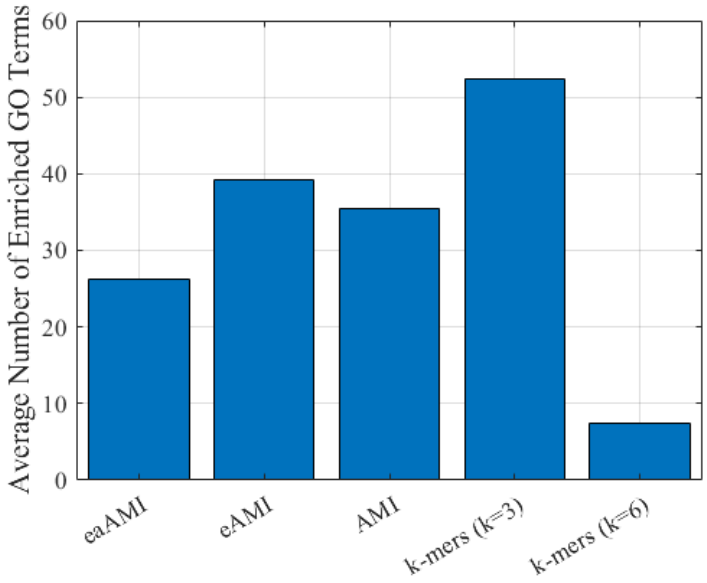


Figure 7.3: Average number of enriched terms ($p<0.05$) identified using each profile across all target genes

The terms that remain following pruning are both statistically significant and

specific, insofar as each gene is assigned a predominately unique set of terms. These terms are likely to be in some way related to the target gene, but this is difficult to verify, as few of them match any of the gene's experimentally-assigned terms. We can reasonably conclude that such enrichment analysis is not effective in making binary GO term predictions, though it may have utility in some contexts. We will next explore more effective prediction methodologies.

7.3 GO Term Prediction

Ultimately, we would like to select an unannotated gene of interest and determine to which GO terms it belongs. For several years, this has been the objective of the Critical Assessment of Functional Annotation (CAFA) project [44]. Researchers submit computational methods for predicting protein function, and CAFA evaluates them on a test set of GO terms. There have now been 3 published iterations of this assessment, with another currently in progress. CAFA provides a useful framework for evaluating predictions on both term-centric and gene-centric bases, and we will use that framework to evaluate an AMI-based classifier.

7.3.1 Prediction Methodology

We predict GO terms using SVMs trained on the gene's profile vectors. An SVM is trained and tested for each GO term annotated to at least 2 *S. cerevisiae* genes. Genes belonging to the term comprise one class, and all other genes comprise a second class, resulting in a simple binary classification problem. K-Fold cross-validation is used to evaluate the models generated from the training sets, with $K = 5$. Once each of the 5 models is trained, they are used to score the genes in the corresponding test set. Raw SVM scores are converted to an estimate of the

probability that each gene belongs to the term given its profile, resulting in scores between 0 and 1. This mapping is done using a sigmoid function modeled on the training data. These scores are used to evaluate the performance of the predictor using two methods: the term-centric average AUC, and the gene-centric F_{max} measure.

7.3.2 Performance Metrics

Receiver operator characteristic (ROC) curves are generated by incrementing a threshold over the range of scores and "predicting" that all genes whose SVM scores are higher than the threshold belong to the term in question. Based on these predictions, we calculate the true positive rate (correctly predicting a gene belonging to the term) and false positive rate (incorrectly predicting a gene belonging to the term) at each threshold value. True positive rate (TPR) and false positive rate (FPR) for a given threshold τ are defined for each term f as:

$$TPR(\tau) = \frac{\sum_i \mathbb{1}(f \in P_i(\tau) \wedge f \in T_i)}{\sum_i \mathbb{1}(f \in T_i)}$$

$$FPR(\tau) = \frac{\sum_i \mathbb{1}(f \in P_i(\tau) \wedge f \notin T_i)}{\sum_i \mathbb{1}(f \notin T_i)}$$

where $P_i(t)$ is the set of terms that have a score greater than or equal to t for gene i, and T_i is the set of terms annotated to the gene. $\mathbb{1}()$ is the standard indicator function. The area under the curve (AUC) is used to evaluate the effectiveness of the predictor for individual GO terms. It is calculated using the trapezoid rule. AUC is a good performance metric in part because of its intuitive interpretation. Given one randomly-selected gene belonging to a term and one randomly-selected gene not belonging to the term, AUC is the probability that the former will have a

higher score than the latter. Thus, an AUC of 0.5 indicates that the predictor has no value in predicting which genes belong to terms. An AUC of 1.0 indicates that all genes belonging to the term are assigned scores higher than any of the genes that do not belong to the term (i.e. the two sets are perfectly separated).

In order to assess the overall effectiveness of the predictor, the AUCs for all GO terms are averaged. Additionally, GO terms are grouped according to their domain, and the average AUC for each domain is calculated. This is used to gauge how predictor performance varies depending on what broad category of descriptor is subject to prediction.

Precision-recall curves are also generated by incrementing a threshold over the range of SVM scores. For each threshold, each gene is individually predicted to belong to all GO terms for which its SVM score was above the threshold. The precision for a particular gene i and threshold τ is defined as the number of correct predictions over the total number of predictions:

$$pr_i(\tau) = \frac{\sum_f \mathbb{1}(f \in P_i(\tau) \wedge f \in T_i)}{\sum_f \mathbb{1}(f \in P_i(\tau))}$$

Recall is defined as the number of correct predictions over the total number of terms annotated to the gene:

$$rc_i(\tau) = \frac{\sum_f \mathbb{1}(f \in P_i(\tau) \wedge f \in T_i)}{\sum_i \mathbb{1}(f \in T_f)}$$

Both precision and recall are averaged over all genes to obtain curves that vary with threshold. Then, $F_{measure}$ is determined by calculating the harmonic mean of precision and recall at each threshold as such:

$$F_{measure}(\tau) = \frac{2 \cdot pr(\tau) \cdot rc(\tau)}{pr(\tau) + rc(\tau)}$$

F_{max} is, appropriately, the maximum value of $F_{measure}$ over all thresholds:

$$F_{max} = \max_{\tau} F_{measure}(\tau)$$

7.3.3 Profile Generation

First, profile vectors are generated for each gene by calculating the frequency of occurrence of each of the 64 possible codons. That is, each gene is assigned a vector in $\mathbb{R}^{64}$ space, and the SVM is trained on this space.

Second, profile vectors are generated for each gene by calculating the eAMI. Lags k of 1-6 (inclusive) are used. Since eAMI produces 16 values for each selected lag, the resulting profile vectors are in $\mathbb{R}^{96}$ space.

7.3.4 Baseline Method

In following the performance evaluation used by CAFA, we use the "naive" baseline method to compare our results against. The naive method is so-called because it does not use any gene/protein information to form its predictions. It simply calculates the frequency of each GO term and applies this value as the prediction score to all genes for that term. Because all genes receive the same score, this method inherently produces AUC values of 0.5. That is, the method is not useful in determining which genes belong to a particular term, but only which terms apply to a particular gene. As such, it does produce F_{max} values greater than 0. This method demonstrates the importance of centering ones prediction scores around the term frequency. If this is not done, F_{max} will be "bad" regardless of how well the method predicts individual terms (as measured by AUC), because the predicted term frequencies will deviate from the actual frequencies, resulting in either inflated false positives or negatives.

Table 7.1: Comparison of different methods by the F_{max} they produce for each GO domain across all *S. cerevisiae* genes

Method	**Biological Process**	**Cellular Component**	**Molecular Function**	**All**
Naive	0.337	0.582	0.218	0.399
eAMI	0.370	0.600	0.275	0.432
***k*-mer**	0.379	0.606	0.287	0.441
eAMI and *k*-mer	0.382	0.608	0.292	0.443

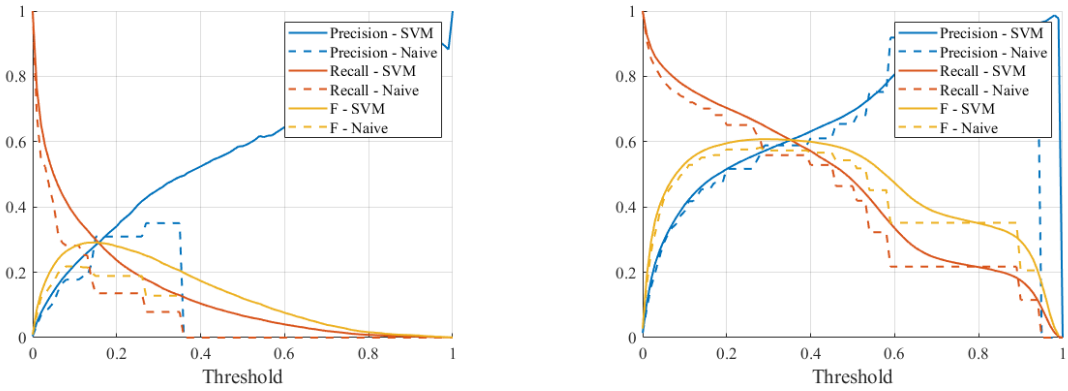
7.3.5 Results

Both profiles produced results that indicate some degree of utility in discriminating genes according to function. The results for F_{max} are summarized in Table 7.1. Precision, recall, and F curves are presented for eAMI and k-mer profiles in Figure 7.4. eAMI profiles yielded slightly better F_{max} values than the naive method in all domains. Codon frequencies performed better still, albeit not to a significant degree. Overall, they provided about 10% improvement over the naive method. The predictions yielded significantly better F_{max} values for the cellular component domain than the other two GO domains.

The results for AUC are summarized in Table 7.2. Terms are separated into a "low abundance" set, consisting of those terms annotated to fewer than 10 genes,

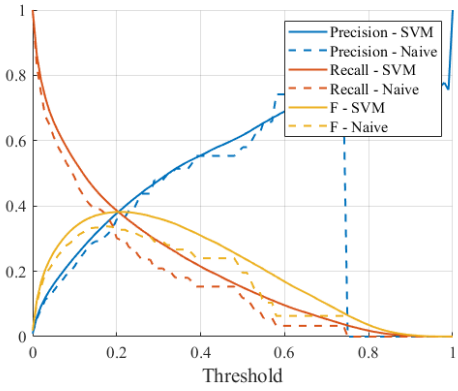
Table 7.2: Comparison of different methods by the average AUC they produce across all terms annotated to fewer than 10 *S. cerevisiae* genes.

Method	**Biological Process**	**Cellular Component**	**Molecular Function**	**All**
Naive	0.5	0.5	0.5	0.5
eAMI	0.736	0.737	0.777	0.747
***k*-mer**	0.750	0.764	0.782	0.760
eAMI and *k*-mer	0.756	0.765	0.796	0.768

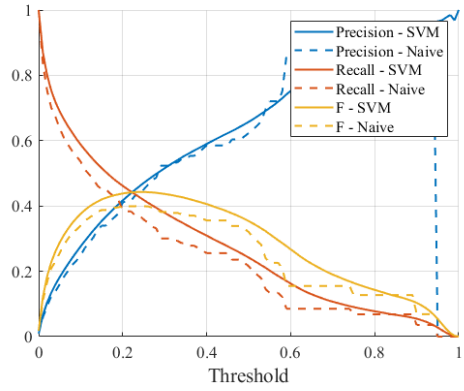


(a) Molecular Function

(b) Cellular Component



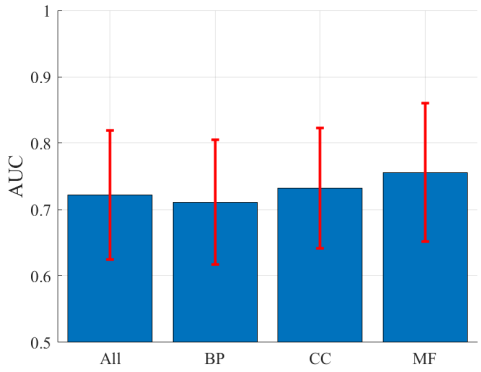
(c) Biological Process



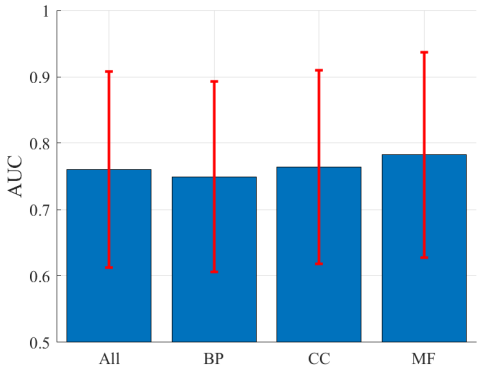
(d) All Domains

Figure 7.4: Precision, Recall, and F measure curves generated by predicting terms in all three domains using eAMI ($k = 1 - 6$) and k-mers ($k = 1 - 3$) for gene profiles. The same curves are also presented for the "naive" baseline method.

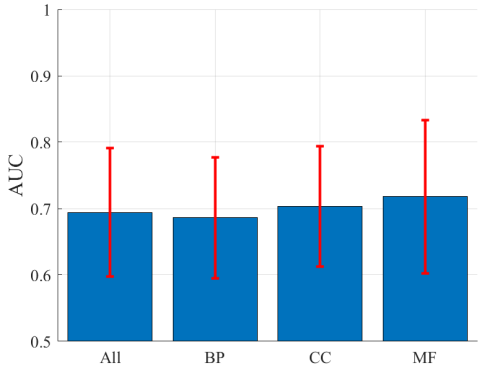
and a "high abundance" set, consisting of those terms annotated to more than 10 genes. The average AUCs and the standard error for all the AUCs obtained for each gene set are presented in Figure 7.5. Again, the codon frequencies did noticeably better than the eAMI profiles. Additionally, for both types of profiles, predictions were more accurate (though also more variable) for low abundance terms. Interestingly, performance was best for the molecular function domain, which is the inverse of the analogous F_{max} results. This likely results from how the distribution of term frequencies varies between the three domains.



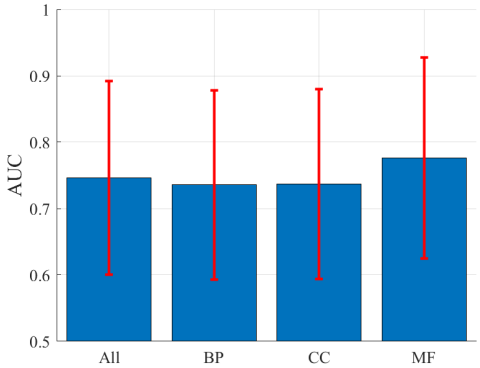
(a) "High abundance" GO terms using codon frequency for gene profiles



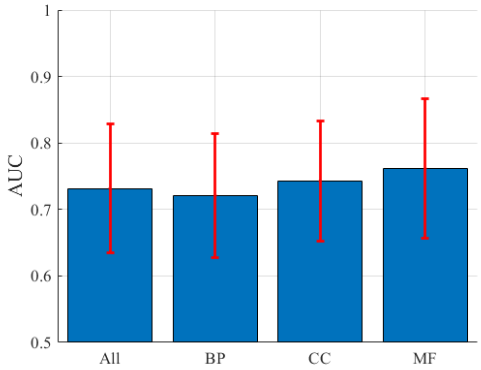
(b) "Low abundance" GO terms using codon frequency for gene profiles



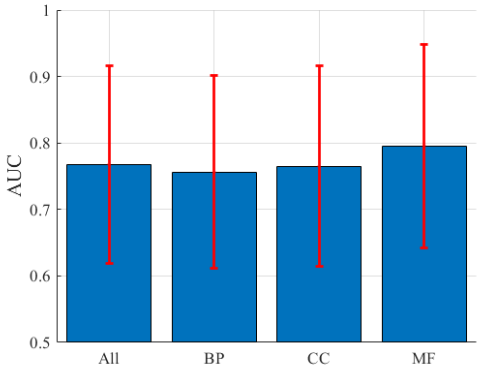
(c) "High abundance" GO terms using eAMI ($k = 1 - 6$)



(d) "Low abundance" GO terms using eAMI ($k = 1 - 6$)



(e) "High abundance" GO terms using eAMI ($k = 1 - 6$) and k-mers ($k = 1 - 3$)



(f) "Low abundance" GO terms using eAMI ($k = 1 - 6$) and k-mers ($k = 1 - 3$)

Figure 7.5: Average AUC in each of the three domains (BP - Biological Process, CC - Cellular Component, MF - Molecular Function), as well as the set of all GO terms. "Low abundance" terms are those annotated to fewer than 10 *S. cerevisiae* genes.

As with any naturally-derived data set, not all elements of the profile vectors provide the same information about the genes' function (or, for that matter, the same *quantity* of information). In order to gauge the relative importance of each codon, we used the individual codon frequency as the prediction score for each gene and calculated the resulting AUC. For individual terms, this indicates which codons are inflated or deflated relative to those genes not belonging to the term. When averaged over all terms, the AUCs for each codon provide a rough measure for how important the occurrence of that codon is in determining what a gene does and where it is located.

To normalize results across all terms, the relative AUC is used. This is the AUC obtained using a single codon divided by the AUC obtained for that term using an SVM trained using all codons. To filter out noise, only terms with an SVM-derived AUC over 0.8 were included. The average relative AUCs are presented in Figure 7.6 for each GO domain. Relative AUC ranges from about 0.7 to 0.8. While all three domains follow the same general trend, there are a few notable exceptions. In particular, ATA has a higher contribution for molecular function terms than the

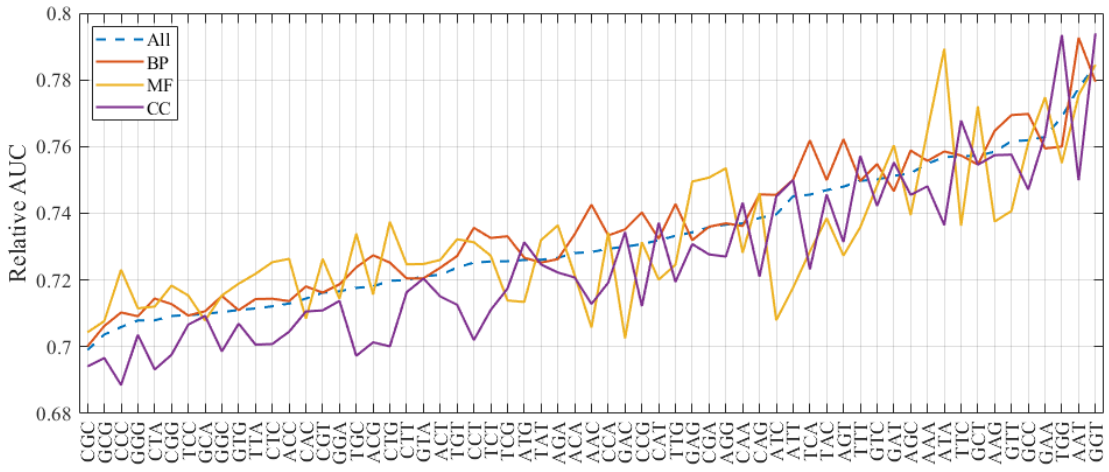


Figure 7.6: Average AUC each codon yields across all GO terms relative to the AUCs obtained using a linear SVM trained using codon frequency profiles.

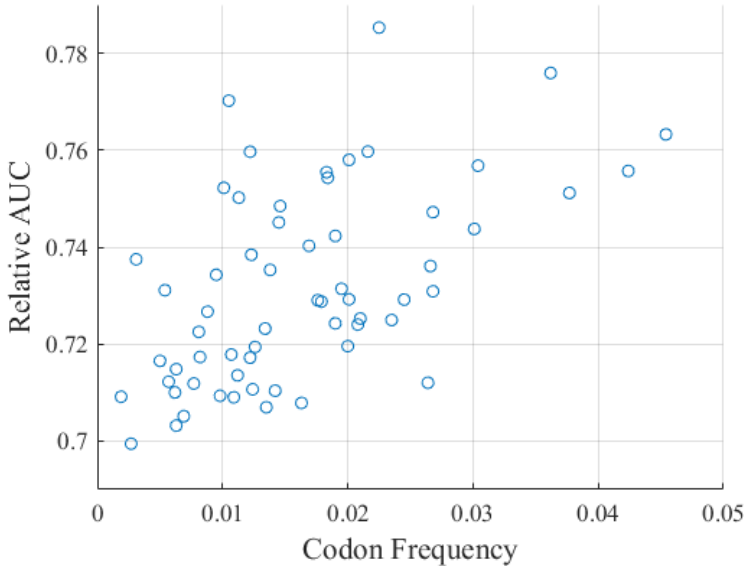


Figure 7.7: The relationship between the frequency of occurrence for a codon across all *S. cerevisiae* genes and the influence of the codon in determining the function of the gene, as measured by the average AUC the codon yields across all GO terms relative to the AUCs obtained using a linear SVM trained using codon frequency profiles.

other domains, while ATC and ATT have lower contributions. All three of these code for the amino acid isoleucine.

Interestingly, this measure of functional importance is correlated with the frequency of occurrence for each codon in all *S. cerevisiae* genes. This is shown in Figure 7.7. While the positive correlation is relatively weak, there appears to be a reliable lower bound on relative AUC that is dictated by codon frequency. An evolutionary interpretation of this is that codons do not achieve high abundance unless they play particularly important roles in determining what genes do.

7.3.6 Performance Discussion

It is clear that for at least some GO terms, belonging to a particular term exerts some observable influence on a gene's profile. This influence is sufficient to produce

strong statistical enrichment in gene lists ranked by profile distance to some target gene. For most terms, it is also sufficient to allow a trained classifier to make predictions that are more accurate than a naive classifier. Realistically, the utility of this is limited. If given a particular GO term, this method would generally not be able to identify unannotated genes that belong to it. Likewise, if given a particular gene, this method would generally not be able to identify the terms annotated to it. At best, it could provide a list of genes of interest for investigation. The likelihood that members of such a list are "hits" would vary widely depending on the term.

There are several reasons why predicting GO terms is difficult in general, as well as for this method specifically. First, most genes are assigned to multiple terms for each namespace, while some have no annotations at all. This lack of exclusivity results in significant overlap in the sets used to train and test different SVMs. It also means that there is not a particularly meaningful negative set. Consider a term that has a child term. If training a classifier for the parent term, all those genes belonging to that term will be in the positive set, while all others will be in the negative set. If training a classifier for the child term, the positive set will be a subset of the parent term's positive set. The other genes from the parent term's positive set will move to the negative set for the child term. One can contrive cases in which classifiers for both parent and child terms can effectively identify genes in the positive set. In practice, however, success for one of the two classifiers is likely to come at the expense of the other. Of course, we can mitigate this by limiting the set of GO terms we consider. By only considering level 1 terms, we eliminate the problems introduced by the hierarchy. This does nothing to make this classification methodology more useful, however. Further, prediction metrics for level 1 terms do not outperform those for all terms, suggesting that many of the level 1 terms are too general to imbue their member genes with some unifying sequence theme. This is

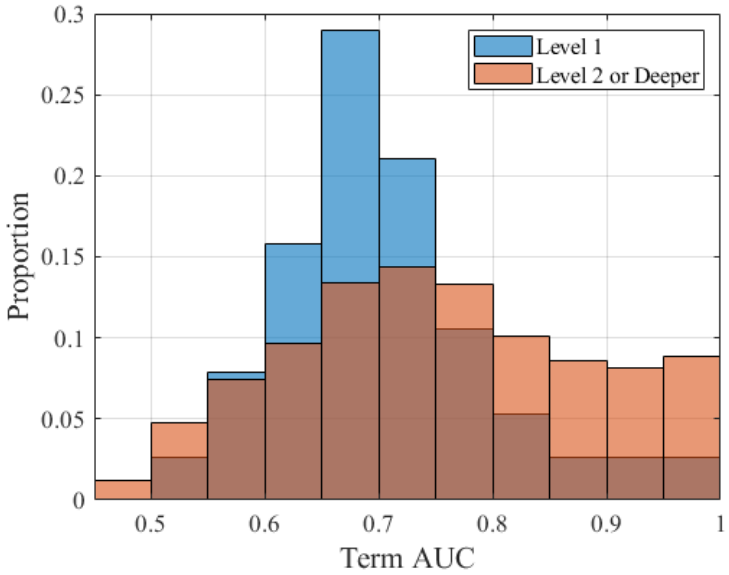


Figure 7.8: Distribution of GO term AUC according to the term's level in the hierarchy, for profiles consisting of k-mers and eAMI vectors

shown in Figure 7.8. Level 1 terms are considerably less likely to be well predicted than those terms deeper in the hierarchy. However, this trend plateaus at level 2, after which there is not a significant change in the shape of the AUC distribution.

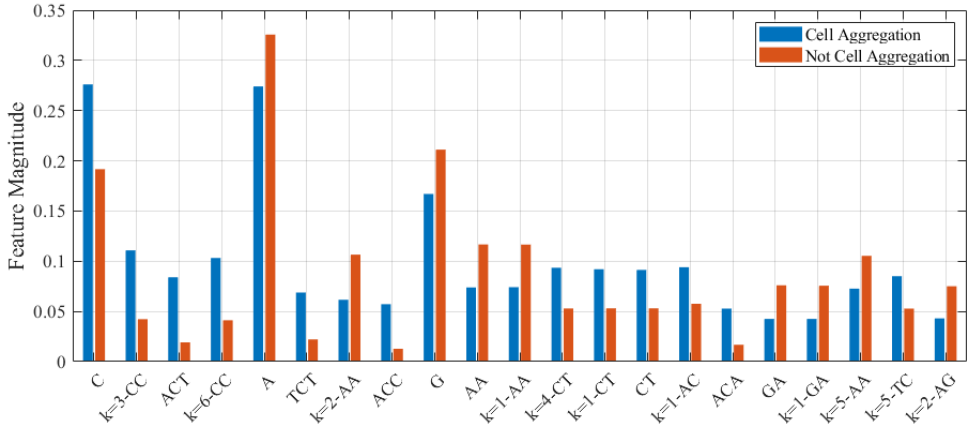
Second, while term frequency across genes varies widely, most terms are not annotated to a significant number of genes. This is especially true when considering the full set of GO terms. Even when limiting to level 1 terms, term frequency still varies from 10^{-3} to 1. The SVMs output probability estimates for each gene/term pair. The estimates are centered around the term frequency. This is intuitive, because if the average estimate was higher or lower than the term frequency, the classifier would predict too many or too few genes, respectively. The problem is that this results in estimates that are unimpressive. If a term occurs in 5% of genes (highly abundant, even for level 1 terms), then probability estimates will likely be on the order of 0-10%. This means that even those genes with the highest scores still have only a 10% chance of being a hit. Low abundance terms are generally

deeper in the hierarchy, and thus more specific. Specific terms are more likely to have some unifying characteristic that can be exploited by a classifier. Yet, low abundance terms also present the classifier with less information with which to exploit those characteristics.

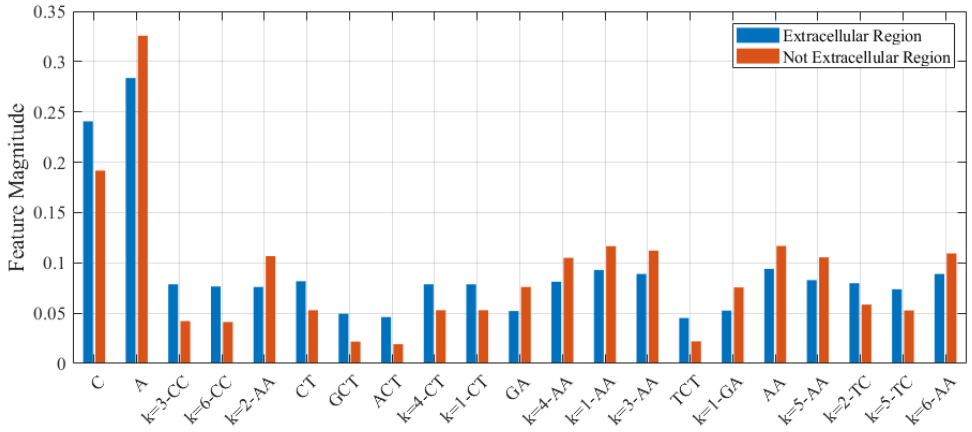
Lastly, and most importantly, there is simply a limited amount of information captured in the profiles. The information that is preserved has practical relevance to several aspects of genome biology, including gene function and location. It is sufficient to statistically discriminate between populations of genes belonging to different functional classes, but not sufficient to identify genes belonging to those classes. This is not surprising. *S. cerevisiae* genes range from hundreds to a few thousand nucleotides. This is on the low end of the required length for AMI-based profiles to capture trends in the sequence. Further, many of the GO terms are inherently difficult to characterize using a generalized sequence-based profile. For example, enzymes have an active site where the substrate binds [7]. The amino acid sequence that composes the active site will have an inflated influence on the enzyme's molecular function, which will be reflected in the GO terms.

7.3.6.1 Analysis of Well-Predicted Terms

While they are the exception, there are a few GO terms that we can accurately predict using sequence profiles. If we consider only level one terms, two terms exceed an AUC of 0.9: "Cell Aggregation" ($AUC = 0.994$) and "Extracellular Region" ($AUC = 0.939$). "Cell Aggregation" is a member of the Biological Process namespace and there are 9 genes with annotations for the term. "Extracellular Region" is a member of the Cellular Location namespace and there are 27 genes with annotations for the term. Centroid profiles for the terms and their respective negative sets are shown in Figure 7.9.



(a) Cell Aggregation



(b) Extracellular Region

Figure 7.9: Centroid profile for genes annotated to the specified GO term, along with centroid profile for genes without such an annotation. Only the features with maximum difference between the two profiles are shown, with difference descending from left to right.

Abundance of the codons ACT, ACC, ACA, and TCT is elevated in genes annotated with "Cell Aggregation". ACT, ACC, and ACA correspond to the amino acid Threonine, and TCT codes for Serine. Both amino acids appear to be required for the GO term. "Extracellular Region" does not appear to have such a strong relationship to one or more amino acids. Instead, the nucleotide pair CT occurs frequently, generally occupying the last two slots in a codon.

Chapter 8

Conclusion

We have demonstrated the utility of average mutual information (AMI) and its derivatives in providing insight on the behavior of genomic sequences. Without overstating their significance, in each of the proposed applications, all evaluated profiles produced results better than random chance. This is a low bar, but suggests that each tested feature is somehow embedded in the genome sequence, and each profile preserves some of this information. A consistent theme in the results was that eAMI and eaAMI outperformed AMI. From this we infer that the averaging stage of computing AMI is indeed discarding useful information.

AMI profiles provide useful insight into the divergence of evolving species. Because they do not require sequence alignment, distances based on AMI profiles offer a major advantage over most alternative methods of measuring distance. Phylogenetic trees generated by measuring pairwise distances between species' AMI profiles resemble the accepted phylogeny. Distance between the AMI profiles of different species reflects, to an extent, the time since the species diverged. In general, correlation distance between AMI profiles appears to be a more robust measure of evolutionary distance.

Ostensibly, the most impressive results presented in this work were those for coding region prediction. This in itself is not surprising, as it was undoubtedly the simplest classification problem posed. Still, the demonstrated robustness of our methodology to accurately predict coding regions for species across the entire spectrum of life is very promising. Given the rapid pace of novel genome sequencing, projected demand for computational gene-finding tools is high. A marker that is indicative of coding regions and can be applied universally to any species is an important step towards that goal.

8.1 Future Work

8.1.1 Phylogenetic Tree Construction

Future work will focus on identifying what classes of sequences work best for constructing phylogenies using AMI profiles. Surprisingly, when we attempted to use whole genomes, the trees were not as accurate as when using only the ITS sequences, so we do need to be selective. We will further investigate how evolution (when occurring under real biological constraints) of different sequences affects the AMI profiles. We are also interested in pursuing why eAMI performed so poorly when it did well in other applications. Finally, we would like to more thoroughly characterize how features of the AMI profile arise from the underlying biology of the sequence.

8.1.2 Classifier Optimization

The classifications described in this work were performed almost exclusively using linear Support Vector Machines. This was deliberate, as SVMs offer many appealing attributes, including speed, convergence behavior, and performance in the

presence of noise. Our limited digressions into alternative classifiers (or SVMs using different parameters, including kernels) were largely fruitless. Still, for each classification problem we consider, there is unquestionably some methodology that would provide better discrimination than the SVM.

8.1.3 Genome Annotation

Ideally, the classification techniques presented in this could be included in an annotation pipeline for newly sequenced genomes. While we focused on proof of concept for coding region prediction, we believe that this could be adapted to effectively identify protein-coding genes, certainly in prokaryotes and perhaps in eukaryotes as well. In the case of prokaryotes, it is likely sufficient to identify all open reading frames in a genome and evaluating each to determine the likelihood it is a gene. For eukaryotic genes, we would need to devise a strategy for determining boundaries between introns and exons. Once a potential gene is identified, we could further make predictions about its function, location, and essentiality, according to the methods described in this work. Another potential application is in assessing the translational efficiency of a gene. This refers to how readily the organism expresses the gene, and is determined by the gene sequence, as well as features upstream and downstream of the gene. While these predictions may not have a high degree of accuracy, they could be used to provide direction for future research.

Bibliography

[1] G. Butler *et al.*, "Evolution of pathogenicity and sexual reproduction in eight Candida genomes," *Nature*, vol. 459, no. 7247, pp. 657–662, Jun. 2009. [Online]. Available: http://www.nature.com/articles/nature08064

[2] X. Liu, B.-J. Wang, L. Xu, H.-L. Tang, and G.-Q. Xu, "Selection of key sequence-based features for prediction of essential genes in 31 diverse bacterial species," *PLOS ONE*, vol. 12, no. 3, p. e0174638, Mar. 2017. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0174638

[3] C. Mora, D. P. Tittensor, S. Adl, A. G. B. Simpson, and B. Worm, "How Many Species Are There on Earth and in the Ocean?" *PLoS Biology*, vol. 9, no. 8, p. e1001127, Aug. 2011. [Online]. Available: https://dx.plos.org/10.1371/journal.pbio.1001127

[4] K. J. Locey and J. T. Lennon, "Scaling laws predict global microbial diversity," *Proceedings of the National Academy of Sciences*, vol. 113, no. 21, pp. 5970–5975, May 2016. [Online]. Available: http://www.pnas.org/lookup/doi/10.1073/pnas.1521291113

[5] H. A. Lewin *et al.*, "Earth BioGenome Project: Sequencing life for the future of life," *Proceedings of the National Academy of Sciences*, vol. 115, no. 17, pp.

4325–4333, Apr. 2018. [Online]. Available:
http://www.pnas.org/lookup/doi/10.1073/pnas.1720115115

[6] H. Frangoul *et al.*, "CRISPR-Cas9 Gene Editing for Sickle Cell Disease and β-Thalassemia," *New England Journal of Medicine*, vol. 384, no. 3, pp. 252–260, Jan. 2021. [Online]. Available: https://doi.org/10.1056/NEJMoa2031054

[7] N. A. Campbell and J. B. Reece, *Biology*. San Francisco, CA [etc.: Pearson/Benjamin Cummings, 2005, oCLC: 1149588155.

[8] "Human Genome Project FAQ." [Online]. Available:
https://www.genome.gov/human-genome-project/Completion-FAQ

[9] "MinION." [Online]. Available: http://nanoporetech.com/products/minion

[10] "Assembly basics." [Online]. Available:
https://www.ncbi.nlm.nih.gov/assembly/basics

[11] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948. [Online]. Available:
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6773024

[12] K. Sayood, *Introduction to data compression*, 4th ed. Waltham, MA: Morgan Kaufmann, 2012.

[13] B. T. Korber, R. M. Farber, D. H. Wolpert, and A. S. Lapedes, "Covariation of mutations in the V3 loop of human immunodeficiency virus type 1 envelope protein: an information theoretic analysis." *Proceedings of the National Academy of Sciences*, vol. 90, no. 15, pp. 7176–7180, Aug. 1993. [Online]. Available: http://www.pnas.org/cgi/doi/10.1073/pnas.90.15.7176

[14] R. Román-Roldán, P. Bernaola-Galván, and J. Oliver, "Application of information theory to DNA sequence analysis: A review," *Pattern Recognition*, vol. 29, no. 7, pp. 1187–1194, Jul. 1996. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/003132039500145X

[15] B. G. Giraud, A. Lapedes, and L. C. Liu, "Analysis of correlations between sites in models of protein sequences," *Physical Review E*, vol. 58, no. 5, pp. 6312–6322, Nov. 1998. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.58.6312

[16] I. Grosse, H. Herzel, S. V. Buldyrev, and H. E. Stanley, "Species independence of mutual information in coding and noncoding DNA," *Physical Review E*, vol. 61, no. 5, pp. 5624–5629, May 2000. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.61.5624

[17] H. H. Otu and K. Sayood, "A divide-and-conquer approach to fragment assembly," *Bioinformatics*, vol. 19, no. 1, pp. 22–29, Jan. 2003. [Online]. Available: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/19.1.22

[18] M. Bauer, S. M. Schuster, and K. Sayood, "The Average Mutual Information Profile as a Genomic Signature," *BMC Bioinformatics*, vol. 9, no. 1, p. 48, 2008. [Online]. Available: http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-48

[19] K. Sayood, D. Bastola, P. Iwen, and S. H. Hinrichs, "Use of Average Mutual Information based Species Signature for Fungal and Mycobacterial Differentiation," in *International Conference on Bioinformatics & Computational Biology, BIOCOMP 2007, Volume I, June 25-28, 2007, Las*

Vegas Nevada, USA, H. R. Arabnia, M. Q. Yang, and J. Y. Yang, Eds. CSREA Press, 2007, pp. 289–295.

[20] K. Sayood, F. Hoffman, and C. Wood, "Use of average mutual information for studying changes in HIV populations," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Sep. 2009, pp. 3861–3864, iSSN: 1558-4615.

[21] P. E. C. Compeau, P. A. Pevzner, and G. Tesler, "How to apply de Bruijn graphs to genome assembly," *Nature Biotechnology*, vol. 29, no. 11, pp. 987–991, Nov. 2011. [Online]. Available: http://www.nature.com/articles/nbt.2023

[22] M. Ghandi, D. Lee, M. Mohammad-Noori, and M. A. Beer, "Enhanced Regulatory Sequence Prediction Using Gapped k-mer Features," *PLoS Computational Biology*, vol. 10, no. 7, p. e1003711, Jul. 2014. [Online]. Available: https://dx.plos.org/10.1371/journal.pcbi.1003711

[23] S. C. Perry and R. G. Beiko, "Distinguishing Microbial Genome Fragments Based on Their Composition: Evolutionary and Comparative Genomic Perspectives," *Genome Biology and Evolution*, vol. 2, pp. 117–131, Jan. 2010. [Online]. Available: https://academic.oup.com/gbe/article/doi/10.1093/gbe/evq004/568285

[24] I. T. Jolliffe, *Principal component analysis*, 2nd ed., ser. Springer series in statistics. New York: Springer, 2002.

[25] B. R. Baum, "*PHYLIP: Phylogeny Inference Package. Version 3.2* . Joel Felsenstein," *The Quarterly Review of Biology*, vol. 64, no. 4, pp. 539–541, Dec. 1989. [Online]. Available: https://www.journals.uchicago.edu/doi/10.1086/416571

[26] P. Larrañaga *et al.*, "Machine learning in bioinformatics," *Briefings in Bioinformatics*, vol. 7, no. 1, pp. 86–112, Mar. 2006. [Online]. Available: https://academic.oup.com/bib/article/7/1/86/264025

[27] C. Campbell and Y. Ying, "Learning with Support Vector Machines," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 5, no. 1, pp. 1–95, Feb. 2011. [Online]. Available: http://www.morganclaypool.com/doi/abs/10.2200/S00324ED1V01Y201102AIM010

[28] D. R. Bastola, H. H. Otu, S. E. Doukas, K. Sayood, S. H. Hinrichs, and P. C. Iwen, "Utilization of the relative complexity measure to construct a phylogenetic tree for fungi," *Mycological Research*, vol. 108, no. 2, pp. 117–125, Feb. 2004. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S095375620862050X

[29] C. L. Schoch *et al.*, "Nuclear ribosomal internal transcribed spacer (ITS) region as a universal DNA barcode marker for Fungi," *Proceedings of the National Academy of Sciences*, vol. 109, no. 16, pp. 6241–6246, Apr. 2012. [Online]. Available: http://www.pnas.org/cgi/doi/10.1073/pnas.1117018109

[30] J.-Q. Chen, Y. Wu, H. Yang, J. Bergelson, M. Kreitman, and D. Tian, "Variation in the Ratio of Nucleotide Substitution and Indel Rates across Genomes in Mammals and Bacteria," *Molecular Biology and Evolution*, vol. 26, no. 7, pp. 1523–1531, Jul. 2009. [Online]. Available: https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/msp063

[31] A. Montalbano, M. C. Canver, and N. E. Sanjana, "High-Throughput Approaches to Pinpoint Function within the Noncoding Genome," *Molecular*

Cell, vol. 68, no. 1, pp. 44–59, Oct. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1097276517306688

[32] A. E. Vinogradov, "DNA helix: the importance of being GC-rich," *Nucleic Acids Research*, vol. 31, no. 7, pp. 1838–1844, Apr. 2003. [Online]. Available: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkg296

[33] S. L. Salzberg, A. L. Delcher, S. Kasif, and O. White, "Microbial gene identification using interpolated Markov models," *Nucleic Acids Research*, vol. 26, no. 2, pp. 544–548, Jan. 1998. [Online]. Available: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/26.2.544

[34] A. Delcher, "Improved microbial gene identification with GLIMMER," *Nucleic Acids Research*, vol. 27, no. 23, pp. 4636–4641, Dec. 1999. [Online]. Available: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/27.23.4636

[35] A. L. Delcher, K. A. Bratke, E. C. Powers, and S. L. Salzberg, "Identifying bacterial genes and endosymbiont DNA with Glimmer," *Bioinformatics*, vol. 23, no. 6, pp. 673–679, Mar. 2007. [Online]. Available: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btm009

[36] I. K. Jordan, I. B. Rogozin, Y. I. Wolf, and E. V. Koonin, "Essential Genes Are More Evolutionarily Conserved Than Are Nonessential Genes in Bacteria," *Genome Research*, vol. 12, no. 6, pp. 962–968, Jun. 2002. [Online]. Available: http://genome.cshlp.org/lookup/doi/10.1101/gr.87702

[37] W.-H. Chen, K. Trachana, M. J. Lercher, and P. Bork, "Younger Genes Are Less Likely to Be Essential than Older Genes, and Duplicates Are Less Likely to Be Essential than Singletons of the Same Age," *Molecular Biology and*

Evolution, vol. 29, no. 7, pp. 1703–1706, Jul. 2012. [Online]. Available: https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/mss014

[38] D. Nigatu, P. Sobetzko, M. Yousef, and W. Henkel, "Sequence-based information-theoretic features for gene essentiality prediction," *BMC Bioinformatics*, vol. 18, no. 1, p. 473, Nov. 2017. [Online]. Available: https://doi.org/10.1186/s12859-017-1884-5

[39] W.-H. Chen, P. Minguez, M. J. Lercher, and P. Bork, "OGEE: an online gene essentiality database," *Nucleic Acids Research*, vol. 40, no. D1, pp. D901–D906, Jan. 2012. [Online]. Available: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkr986

[40] S. Nandi, P. Ganguli, and R. R. Sarkar, "Essential gene prediction using limited gene essentiality information–An integrative semi-supervised machine learning strategy," *PLOS ONE*, vol. 15, no. 11, p. e0242943, Nov. 2020. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0242943

[41] M. Ashburner *et al.*, "Gene Ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, May 2000. [Online]. Available: http://www.nature.com/articles/ng0500_25

[42] The Gene Ontology Consortium *et al.*, "The Gene Ontology resource: enriching a GOld mine," *Nucleic Acids Research*, vol. 49, no. D1, pp. D325–D334, Jan. 2021. [Online]. Available: https://academic.oup.com/nar/article/49/D1/D325/6027811

[43] H. K. Lee, W. Braynen, K. Keshav, and P. Pavlidis, "ErmineJ: Tool for functional analysis of gene expression data sets," *BMC Bioinformatics*, vol. 6,

no. 1, p. 269, Nov. 2005. [Online]. Available:
https://doi.org/10.1186/1471-2105-6-269

[44] N. Zhou *et al.*, "The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens," *Genome Biology*, vol. 20, no. 1, p. 244, Dec. 2019. [Online]. Available:
https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1835-8