New Strategies to Develop Hidden Markov Models Combining KEGG and Pfam

by
Sonica Gupta

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Scholar)

Presented June 4, 2020
Commencement June 2020

AN ABSTRACT OF THE THESIS OF

Sonica Gupta for the degree of Honors Baccalaureate of Science in Computer Science presented on June 4, 2020 Title: New Strategies to Develop Hidden Markov Models Combining KEGG and Pfam.

Abstract approved:_____

Maude David

Profile Hidden Markov Models utilize the information stored in a multiple sequence alignment such as its residues and conserved regions to assign probabilities at each column. This indicates areas in the sequences where similar functionality is observed. These probability weights are then combined to assign a sequence score, which is then able to better detect distant protein sequences. Current databases take advantage of the Profile HMMs such as Pfam and KEGG. However, where Pfam has a lot of sequences, it lacks specific annotation, making it harder to detect distant sequences that share the same functionality. Additionally, where KEGG has specific annotation, it lacks in number of sequences. Our goal is then to leverage the unannotated sequences from Pfam to generate KO-annotated level HMMs.

Key Words: Hidden Markov Model, Sequence Alignment, BLAST, BLAT

Corresponding e-mail address: guptaso@oregonstate.edu

New Strategies to Develop Hidden Markov Models Combining KEGG and Pfam

by
Sonica Gupta

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Scholar)

Presented June 4, 2020
Commencement June 2020

Honors Baccalaureate of Science in Computer Science project of Sonica Gupta presented on June 4, 2020.

APPROVED:

_____

Maude David, Mentor, representing Department of Microbiology

_____

Christine Tataru, Committee Member, representing Department of Microbiology

_____

Andrew Thurber, Committee Member, representing Department of Microbiology

_____

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, Honors College. My signature below authorizes release of my project to any reader upon request.

_____

Sonica Gupta, Author

Contents

Introduction

Cost decrease and popularization of next generation sequencing (NGS) have led to increasingly large DNA and RNA sequence datasets. These sequencing technologies can output over one billion sequencing reads within a span of a few days at a relatively low cost[1,2]. Current methods available for analyzing these sequences are computationally intensive. Furthermore, 98% of bacteria are uncultivated, meaning we do not have the information necessary to directly annotate these sequences.

In most cases, in order to assign a function to a gene, reads need to be first aligned to the reference genes or genomes [2]. To facilitate functional gene annotation, the scientific community has made several databases available online, which carry protein sequences that have been characterized and curated to various degrees. The National Center for Biotechnology Information (NCBI) allows all authors to submit and publish sequences generated from their study, including unassembled and uncharacterized metagenomes (within their nr database, which paradoxically stands for "non redundant"), resulting in extremely rich but very poorly functionally characterized sequences. On the other end of the spectrum, investigators have spent an extensive amount of time and energy experimentally characterizing enzymatic activities, and databases such as the Kyoto Encyclopedia of Genes and Genomes (KEGG[18]) present protein sequences categorized by their function, the pathway they belongs to as well as the organisms they have been identified in.

Unfortunately, the number of protein sequences present in this well-curated database is very low as less than 2% of bacteria have been cultivated [4]. Furthermore, only a small portion of the genome has been characterized through direct experiment while the rest has been characterized through a comparison of sequence similarity, which has also come to be known as

uncultured genome sequences [3,5]. Subsequently, since we lack cultivated sequence data, we are also missing key functionalities to assign to sequences and as a result, annotations often suffer.

This project proposes to address this gap by leveraging uncharacterized sequences to develop enriched Hidden Markov Models (HMMs) profiles. To do so, we will rely on protein families defined by the PFAM database[15], and develop new models that will refine the functional annotations to the KEGG Ortholog level. The first part of this thesis will consist in a detailed bibliography of three current alignment methods: BLAST, BLAT and HMMs. This bibliography will introduce the rationale that motivates developing new Hidden Markov Models. We then describe the methodology in section 2 and the results in section 3.

Background

Current Annotation Algorithms

As previously mentioned, the growth of NGS has ushered in an increasing supply of protein sequences. Several algorithms and software tools have been developed to mark specific sequence features through annotation and detect the similarities between these features through annotation. There is such a large amount of data stored within these protein sequences that it requires extensive computing power to analyze them[7]. The alignment stage alone is crucial to identify similarity between the sequences; however, due to the large number of combinations, the time complexity, O(mn), greatly increases as the number of sequences increases[7].

There are two types of sequence alignments: Pairwise Sequence Alignment, which compares a pair of sequences, and Multiple Sequence Alignment, which compares a group. Pairwise Sequence Alignment can then be further classified into a local or a global sequence alignment [7]. During a global alignment, sequences are lined up end to end in order to detect similarity. While in a local alignment, subsequence matches are searched for. One of the most frequented alignment tools, BLAST, performs local alignment and utilizes a heuristic algorithm to decrease computation time.

Local Alignment Heuristic Algorithm

The precursor to local alignment heuristic algorithms relied upon dynamic programming, which was computationally intensive especially as the sequences grew in both size and numbers. These algorithms included the Needleman-Wunsch and Smith-Waterman algorithms[6]. The algorithms emphasized similarity by minimizing the evolutionary distance through the focus of the least costly set of mutations[6]. However, given the size of databases today, these algorithms are computationally infeasible. A heuristic alignment algorithm, on the other hand, aims to reduce

the computational complexity by approximating the above algorithms, but often sacrifices sensitivity. BLAST is a widely used software alignment tool that utilizes a heuristic algorithm and BLAT has since developed upon that algorithm.

BLAST

The Basic Local Alignment Search Tool (BLAST) optimizes speed and performance by using a heuristic approach that estimates the Smith-Waterman algorithm, an accurate but slow alignment algorithm[6]. BLAST utilizes subsequence similarity to find all possible pairs of local segments whose similarities exceed a certain threshold to then create a similarity score matrix[6,9]. These similar segment pairs, who exceed the threshold, are called high-scoring segment pairs (HSP), where the highest scoring pair is the maximal-scoring segment pair (MSP)[9]. The process BLAST uses could be visualized in the following three steps[9]:

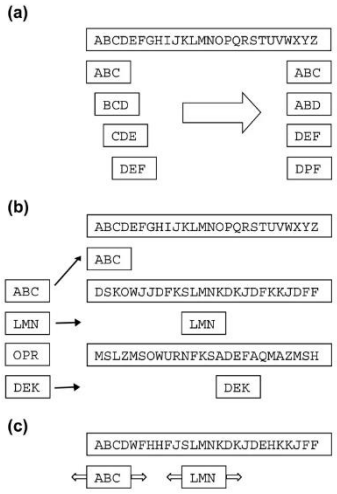


Figure 1: The three stages of a BLAST alignment

1. Generate Seeds

Part **a** of **Figure 1** shows seeds of length three constructed out of the query sequence. This list of seeds is then used to identify short matches and is a technique known as seeding. Then using a scoring matrix, high-scoring matching words are extracted from the word list that score above a

given threshold [6,9]. The use of a threshold here is important to note as at this point speed and selectivity is valued over sensitivity (i.e. the ability to capture distantly related sequences)[6]. By not considering the seeds that score below this threshold, important information that could have detected similar sequences is lost.

2. Compare Seeds to Database

Part **b** of **Figure 1** shows the seeds compared to the seeds in a sequence database. The sequences in the database already have their seeds generated and indexed so search time is minimized [9]. The goal in this step is to identify exact matches between the words in the database and the words in the query. These pairs of matches are the high scoring pairs (HSP) mentioned earlier. Since we are dealing with only high scoring words, found in step 1, the match detection ends up being limited as identification of sequences with a lower level of similarity is missed.

3. Extend Alignments

At this point the HSP has been located and the goal is now to identify the maximum- scoring segment pair (MSP). Part **c** of **Figure 1** shows the two HSP matches extended both directions. The extension occurs until the scores start to decrease [6]. The MSP is then identified as the highest scores from the entire database [9]. The limitation in BLAST is that the hits generated are too selective or too well matched and thus its sensitivity, or ability to identify distant but similar sequences, is fairly low.

BLAT

Another sequence alignment tool known as the BLAST Like Alignment Tool (BLAT) is similar to the algorithm used in BLAST; however, in an effort to increase speed, its similarity score is even lower than that of BLAST [8]. This is because the HSPs generated in Part **b** of **Figure 1** are

achieved through a more stringent threshold of a 98% match [8]. This means that fewer sequences are passed to Part **c** of **Figure 1** and thus fewer MSPs are located [7].

In both cases, BLAT and BLAST, an initial seed is built through the query sequence to generate the alignments. This is not an effective approach as it becomes too selective at the cost of sensitivity. Moreover, this seed alignment approach does not factor in the region conservation, which is key to detecting homology. This is because BLAST and BLAT indexes a large sequence with little to no curation rather than looking at the protein domains.

Hidden Markov Models

Protein domains and sequence motifs are key to identifying homologous sequences since they show the level of evolutionary conservation [3]. Domains often contain one or more motifs, which are, loosely, a set of conserved residues that show protein function [3]. Protein sequences can contain one or more domains, and comparing domains individually is critical during sequence analysis in order to identify homology between sequences[3]. Note that similarity and homology are not the same thing. Similarity will show the level of resemblance between two sequences; however, homology will show if this similarity results from common ancestors [9]. Because BLAST and BLAT do not factor in the domain structure, these algorithms are not efficient in detecting homologous sequences that do not maintain high sequence similarity.

We need to identify protein domains in order to detect protein function shared through common ancestors. This can be done through Hidden Markov Models (HMMs), which have been heavily used in a variety of applications such as speech recognition. HMMs allows us to create probabilistic models for a system with state changes [10,11]. The system in this case is the alignment and the changes correspond to the weights within the alignment that accounts for modifications such as insertion, deletions, and substitutions.

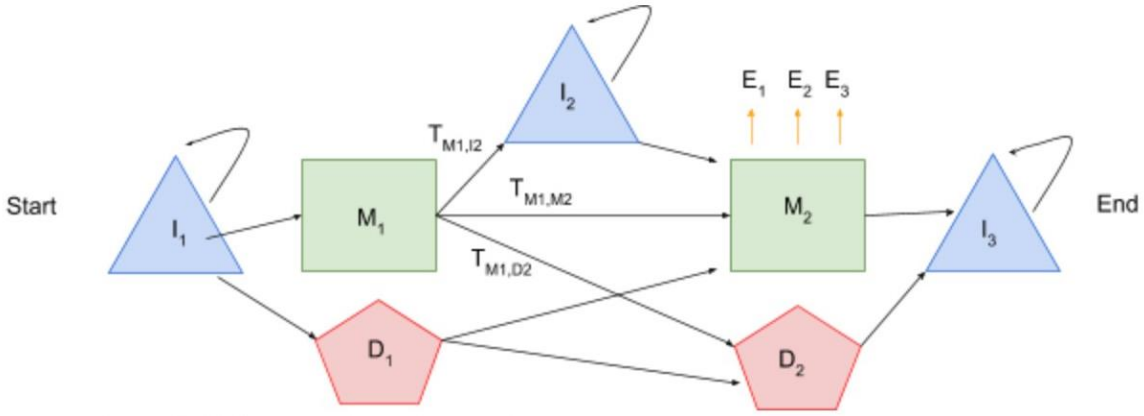


Figure 2: Diagram of the three states

Insertion (I_j), Deletion (D_j), and Match (M_j) states used to calculate the probabilities of an HMM Profile. $T_{i,j}$ represents the Transition Probability from state i to j and E_k represents the Emission Probability of each amino acid.

Calculation

Hidden Markov Models allow us to detect distant homologs by utilizing a probability distribution to classify sequences based on residue alignments. A probability is assigned to each column of the alignment based on the frequency of a particular amino acid at that position. This then gets factored in when assigning an overall score to the sequence. An application of Hidden Markov Models known as Profile HMM is used to represent multiple sequence alignments with its corresponding probability scores [12]. Another way to describe this is to look if a particular amino acid is present in all sequences at the same region, the area is considered a conserved region. During a multiple sequence alignment, these conserved regions of the sequences become more apparent as a higher number of sequences are aligned.

Profile HMM is especially efficient in detecting distant homologues because it accounts for insertions and deletions of each column in a multiple sequence alignment [11]. The probability

assigned depends on three states: the match state, the insert state, and the delete state. An instance of this could be visualized in **Figure 2**. The insert state (I_j) allows for the insertion of a new, random amino acid that is present in the query sequence [11]. Note that this state can transition back to itself, indicating multiple insertions. The delete state (D_j) allows for the conserved column to be deleted[11]. In absence of a delete state and an insert state is the match state (M_j). This occurs when a column of a query and target align, where the amino acid between the two do not necessarily need to match.

Each of these states has a certain likelihood, known as the transition probability ($T_{i,j}$), of transitioning from state i to state j, as indicated by the arrows in **Figure 2**. The transition probabilities are only dependent on what happens in the previous state [13]. For instance, the match state, M_1 in **Figure 2**, has three possible states to transition to: insertion, deletion, or the following match state, M_2. Then the transition probabilities would be captured in a matrix[13]:

$$P = [T_{M1,I2}, T_{M1,M2}, T_{M1,D2}], \text{ where } T_{M1,I2} + T_{M1,M2} + T_{M1,D2} = 1.$$

The emission probability occurs at every match and insertion stage to represent the likelihood of each of the twenty amino acids appearing. All these states and probabilities are used to build a Profile HMM to represent the specific characteristics of the sequences.

Logo Representation

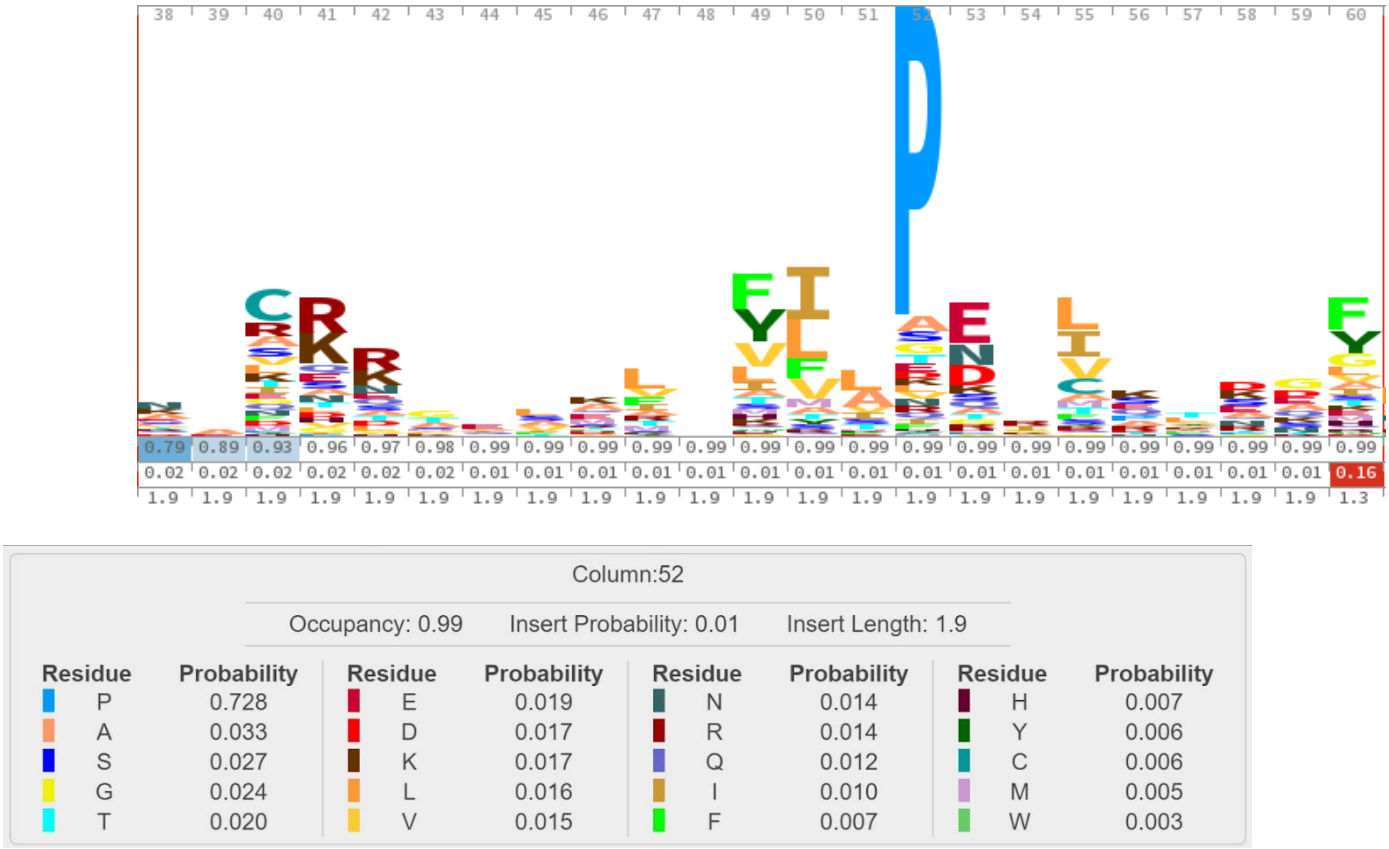


Figure 3: HMMLogo with Column Probability

Top half is of an HMM Logo created by Skylign[14], a web server that builds interactive logos. Bottom is of the amino acids pictured in column 52 of the top half.

A graphical representation, known as an HMM logo, of a sample Profile HMM is seen in **Figure 3**. Each column contains twenty letters that correlates to the twenty amino acids built off of a basic set of proteins. The original input was a multiple sequence alignment constructed out of five sequences taken from the zf-CCCH_8 protein family. Here, the alignment information was used to reveal a stack of letters at each position, where the stack's height corresponds to that position's conservation[14]. In the case of **Figure 3**, column 52 would represent the largest measure of invariance, as the stack height is the largest. The height of the individual letters then corresponds to the number of occurrences of the letter at that position[14]. The largest letter, P,

signifies that that amino acid occurs the most at that position. A closer look of the probabilities in column 52 is shown in the bottom half of **Figure 3**. In the HMM logo, the residue corresponding to P is the largest letter pictured in column 52. Similarly, the probabilities show that P has the largest probability at 0.728. This means that another sequence with a P in the same position would have a higher likelihood to be similar to this protein family because it closely resembles this conserved area. Since Profile HMMs leverage the information stored in a multiple sequence alignment to assign probabilities at each column, it can better detect distant protein sequences.

Current Hidden Markov Model Databases

Hidden Markov Models have already been widely applied by the scientific community for functional annotation, as they focus on the protein domains in order to determine the level of conservation for specific residues, and therefore allow the identification of distant domains. The two main HMM databases, Pfam Database[15] and TIGRfam Database[16], are detailed below.

PFAM

The Pfam database is a large database of 13,672 protein families composed of alignments constructed at a very high functional level through Hidden Markov Models[15]. Moreover, Pfam is a protein domain database, meaning that the Pfam entries are not protein sequences but rather alignments of the most conserved domains of related proteins [3]. Since Pfam uses protein domains, which are a set of conserved residues that show protein function, sequence functionality is preserved within the database [3]. This means that alignments are then based on these protein domains, or functionalities, rather than three letter seeds that are used with BLAST and BLAT.

The first step taken in the construction of a full alignment within a Pfam database is the creation of a *seed alignment*. This seed alignment is built off of a *representative* set of sequences and then is manually verified [17]. Then an HMM Profile is created from the seed alignment. Lastly, the full alignment is created by aligning members of the Swissprot database to this HMM Profile [17]. To eliminate false positives, two gathering thresholds are put in place, a sequence threshold and a domain threshold, so that only sequences reaching a high level of similarity are detected [15]. The goal of these gathering thresholds is to minimize false positive matches[15]. It is based on the idea that long profiles tend to be generic and thus capture sequences that overlap between families [15]. The gathering threshold is then meant to exclude these sequences in the overlap. This is comparable to what is happening in BLAST and BLAT as we are now limiting the sequences that we are detecting by being too selective. However, in this case we are limiting sequences that share functionality, whereas with BLAST and BLAT the results only contained sequences similar to each other based on three letter seeds unrelated to the protein domains.

TIGRFam

TIGRfam is similar to the Pfam database except that it emphasizes protein function whereas Pfam stresses domain architecture[16]. TIGRfam still goes through the process of using seed alignments to produce HMM Profiles, but it also takes it one step further to generate an equilog model. Each protein within TIGRfam is considered in terms of its function to see how they differ from the protein family's function[16]. A protein family is deemed an equilog when all members of a protein family share the same functionality[16]. Pfam, on the other hand, was only looking for similarities in sequences' domains. In this way, TIGRfam would identify fewer protein similarities than Pfam would[16].

Project Rational

While both HMM databases developed above allow the detection of distant orthologs, motifs detected via their pipeline only allow small domains to be identified. As a consequence, TIGRfam and Pfam allows more sequences to be annotated than heuristic approaches using whole length sequence alignment. But annotations of those short domains are difficult to integrate with the metabolic pathways' ontology available in the literature, such as MetaCyc or the KEGG database[18.]

Furthermore, in order to find conserved domain within the protein considered, both HMM databases use a seed alignment, where they remove any sequences that are not properly aligned with their bulk of sequences. This first step results in critical loss of information as sequences are thrown away when a perfect match is not found.

This project proposes to address some of these limitations by developing new HMMs using the KEGG database[18] framework, which would allow us to refine the annotation of HMMs at the level of KEGG orthologs, and make distant ortholog detection compatible with multiple visualization and analytics pathways analytics softwares[19]. The KEGG database characterizes its sequences with a KO identifier used to group by functionality and aimed to detect orthologs between different organisms.

As mentioned previously, it is important to note that only about a third to a half of the characterized genomes have KO annotations[19]. To overcome this challenge, we propose here to propagate KO annotations to unannotated sequences using a ClustalO alignment. Then, we will construct an HMM Profile per KO using more complete sequence alignments generated from the annotation propagation. The resulting HMM Profiles will then take into account a more diverse set of sequences.

We conducted an evaluation of our newly generated HMMs using a ten-fold cross-validation approach to see how well we were able to predict the KO annotations of our unannotated sequences. Through this validation, we set aside a unique 10% set at each fold. This 10% set contains both annotated and unannotated sequences. We compare our profiles against the annotated sequences of this set to see if our profiles are linking to the correct KO annotations.

Methods

The code used to generate this pipeline is available at: https://github.com/MaudeDavidLab/hmm-project.

Overview Diagram

The overall methodology can be visualized in **Figure 19**.

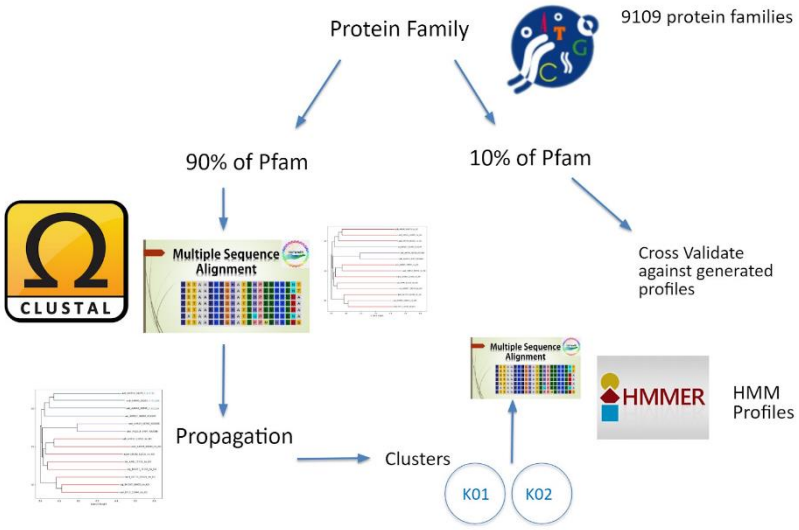


Figure 4: Overview of Methods

The first part of our pipeline involved fetching protein sequences from the GenomeNet DB, described in detail in the section **Protein Retrieval**. The we separate our sequences so that we are currently working in 90% of our sequences. Next, we propagate our known KOs to unannotated sequences as explained in **KO Annotation Propagation**. Following that, we are able to build our HMM Profiles based on the annotated sequences as detailed in **Cross Validation**. Lastly, we use the other 10% of our sequences to verify our new HMM Profiles as reported in **Cross Validation**.

Protein Retrieval

9,109 protein families were retrieved from GenomeNet Database. From there, the sequences within the protein families were filtered out so that they only contained the prokaryotes. Within the GenomeNet Database, about one-third to a half of the genes contain a KO annotation[19].

For each protein family, we performed a ten-fold cross validation in order to evaluate our model's performance. To do this, a unique ten percent of sequences were randomly pulled out from each protein family at each fold. With the remaining 90% at each fold, we built a phylogenetic tree based off of the initial guide tree generated by ClustalO [20]. This tree, shown in **Figure 4**, contains both annotated and unannotated KOs. All the unannotated KOs were colored in red, while the annotated KOs were uniquely colored depending on the KO.

KO Annotation Propagation

For each protein family, the trees generated from the alignment were used to propagate KO annotation to the unannotated sequences using the ETE python toolkit[21]. The propagation was performed by traversing down the tree and looking at each annotated leaf and its unannotated children. If the unannotated child's branch is shorter than the annotated leaf's branch, the KO annotation is propagated to the child. Frequently, at its initial stage, there are many more unannotated genes than annotated, as seen on the left of **Figure 4**. There are two main cases to consider:

Propagation Up/Down the Tree

The purple sequence in **Figure 4** represents K12284 and three red, unannotated branches directly below it. Additionally, these three branches are shorter than the annotated branch. Therefore, since these three branches match our condition for propagation, these three branches will inherit the KO annotation: K12284

Propagation in between Two Branches

In another case, there could be unannotated branches that are placed in between two annotated branches. For instance, in the case of the green branches in **Figure 4**, that represent K02656, there is a red branch in between these. The longest branch would be looked at first that corresponds to aaa_Acav_1461_K02656. Then the first unannotated child sequence will be considered: aae_aq_854_no_KO. Since, this unannotated branch is shorter than the annotated branch, the KO annotation propagates. The next child branch that is considered is aacn_AANUM_0066_K02656. However, since this branch is already annotated the propagation from aaa_Acav_1461_K02656 stops and the next annotated branch with child nodes is considered.

The right of **Figure 4** shows 2 instances of propagation taking place based on the two cases described above. In this way, we are left with fewer unannotated sequences than when we started, as seen on the right.

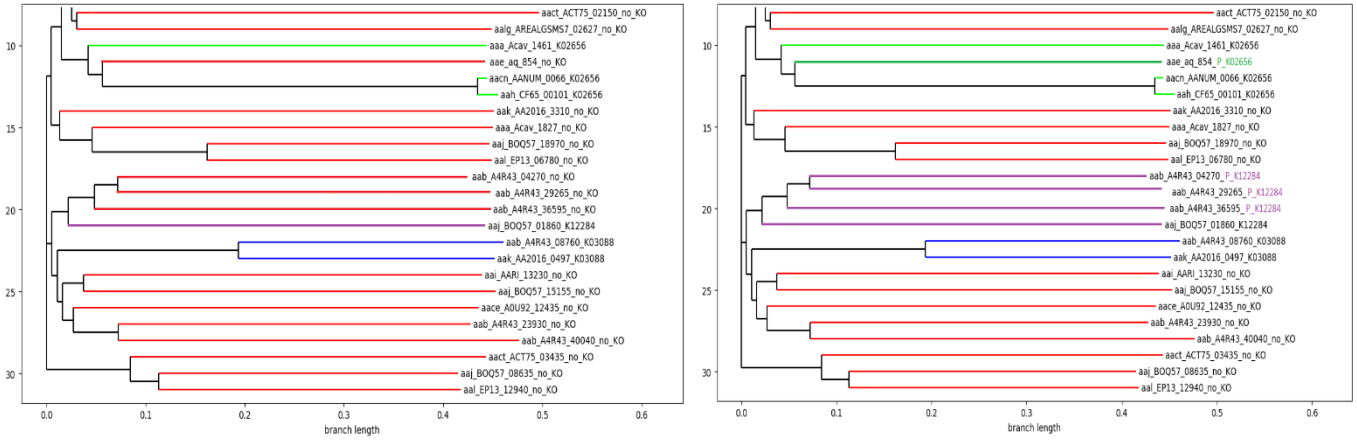


Figure 5: Before and After Annotation Propagation

Subset of 24 sequences from the TPR_21 protein family using the initial guide tree built by ClustalO. The left panel represents the tree before propagation of the unannotated KO sequences and the right after propagation.

Sequence Clustering for HMM Building

At this point, after propagating the known KO's to the unannotated sequences, the next step involves building an HMM profile to represent each of these KOs. As shown in **Figure 5**, the tree is broken up into separate clusters based on the KO annotation. Since the tree **Figure 5** only contains three KOs, K02656, K12284 and K03088, only two HMM Profiles would be built. The sequences in the tree without a KO annotation are ignored as their representation is unknown. The size of the clusters range in the amount of sequences they hold. In order to build a representative HMM profile, a threshold was placed so that a profile was only built if the KO cluster contained at least five sequences. The results of this are further discussed in the discussion section.

After the formation of clusters of KOs, each cluster was aligned using ClustalW2 version 2.1[22]. The HMM Profiles were then constructed using the hmmbuild software available within HMMER package version 3.1.b1.

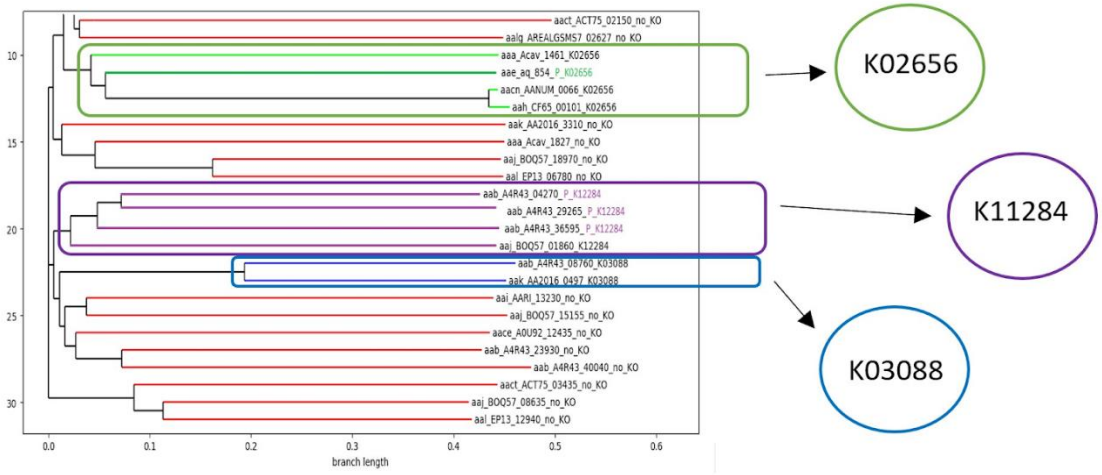


Figure 6: HMM Clustering

The annotated KOs are broken up into their own clusters. For instance, all sequences annotated with K02656 get grouped together.

Cross Validation

10% of our protein family's sequences were set aside to conduct the ten-fold validation. The unannotated sequences from the 10% sequence set were then filtered out so that only the sequences with the known KOs are included. Using the tool hmmsearch through the HMMER package version 3.1.b1, the profiles we created were searched against this filtered 10% sequence set. With the hits generated, the results consist of our prediction, based on the HMM Profiles, the ground truth, generated through our known 10% set, and the e-value that gives us the level of which this similarity occurs.

Results

Propagation Increases the Number of Annotated Sequences by 14%

As described above, we propagated KO annotation of sequences from the GenomeNet Database, by using a guide tree (see K03088, K02656, and K12284 in **Figure 4** as an example).

Prior to propagation, one protein family, TPR_2, contained 49,586 sequences with only 12,877 of these sequences containing a KO annotation. This means that 36,709 did not contain a KO annotation.

Using a ten-fold cross validation, we pulled a unique 10% out, so that we are only working with 90% of the sequences at each fold. This means that at each fold we are working with a total of about 44,620 sequences. On average, at each fold, 11,589 sequences were annotated with a KO. The results at each fold is shown in **Table 1**.

After propagating the known KO annotations to the unannotated sequences, about 6,157 sequences were annotated. This means that the number of annotated sequences increased by an average of 53%. Out of the total subset, the number of annotated sequences increased by an average of 13.7%.

Total	Annotated	Annotated After Propagation	% increase in annotated sequences	% increase in total annotated sequences
44,628	11,542	5,705	49	12.7
44,628	11,596	5,799	50	13
44,628	11,549	6,083	53	13.6
44,628	11,618	6,290	54	14.1
44,628	11,574	5,979	52	13.4
44,628	11,569	6,529	56	14.6
44,628	11,616	6,339	55	14.2
44,628	11,608	6,636	57	14.7
44,628	11,606	6,012	52	13.5
44,622	11,615	6,200	53	13.9

Table 1

Each KO Generates Multiple HMMs

HMM Profiles were built for each KO at each fold and within every protein family (one KO being sometimes present in multiple protein families). **Figure 6** shows the number of models created per KO for a random set of 100 protein families. In total there were 2,104 unique KOs across the 100 protein families. From these KOs, 22,260 models were created.

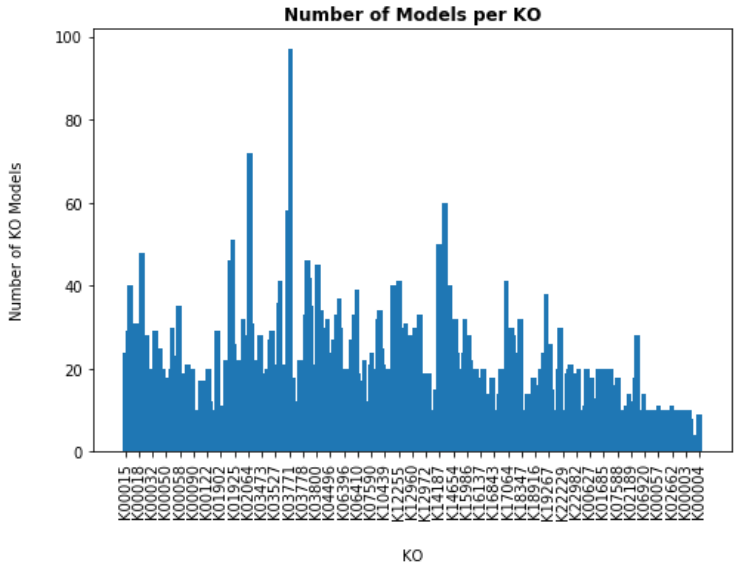


Figure 7:Models per KO

Overall Model Performance

The goal of the HMM Models is to correctly label the sequences without a KO annotation. To do so, we looked at 100 randomly sampled protein families and the HMM Profiles built from their KOs. These HMM Profiles contained the sequences that we annotated using our propagation method. Using a 10-fold cross validation approach, we have also set aside 10% of our sequences at each fold to align against these HMM Profiles. We subsequently match these sequences with known KOs to the corresponding HMM Profile to validate the models. After searching our

known KO sequences from the 10% set against our HMM Profiles, we are able to characterize our results in four ways based on the level of similarity defined by the e-value:

1. True Positive: Given an e-value below our threshold, our prediction was correct
2. True Negative: Given an e-value above our threshold, our prediction was correctly wrong
3. False Positive: Given an e-value below our threshold, our prediction was wrong
4. False Negative: Given an e-value above our threshold, our prediction was missed

We focused on reporting: precision and sensitivity, whose box plots can be seen in **Figure 7**. The F1 Score is also included to show our classifier's performance. Conceptually, our metrics tells us:

1. Precision - When something was predicted positive, how often was it actually positive?
2. Sensitivity - From our actual positive data, how often did we predict correctly?

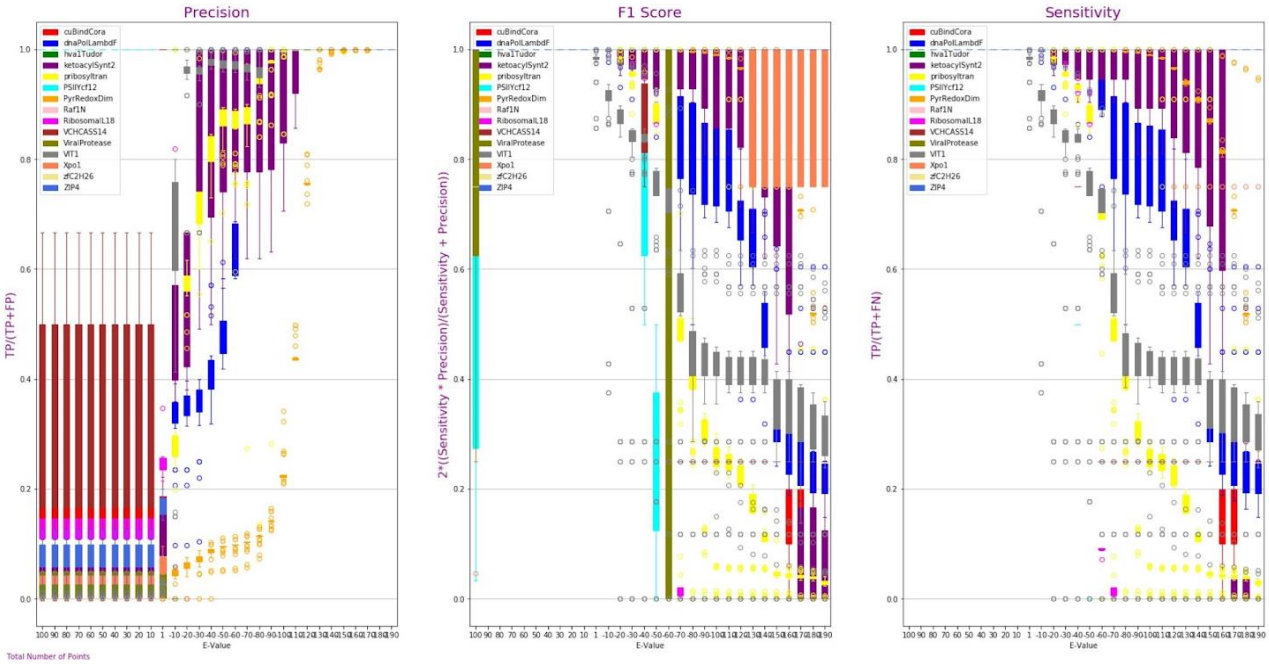


Figure 8: Overall performance of HMMs on 11 Randomly Sampled protein families

Figure 7 shows the performance of 226 HMM models that were built for the KOs of 11 random protein families across an e-value threshold with a range of 1e100 to 1e-190. With a more stringent threshold (a lower e-value), there is an apparent tradeoff between precision and sensitivity. So, with a lower e-value we are able to achieve higher precision but then our sensitivity suffers.

Filtering out KO Models with Low Counts of Sequences

The propagation step may result in producing clusters of sequences for each KOs with drastically variable size. **Figure 8** shows the distribution of the number of sequences (x-axis) used for each KO model (y-axis). Out of 22,260 HMM Profiles, 3,528 HMM Profiles were built from only two sequences. In general, **Figure 8** shows that most KO models contain fewer sequences while fewer KO models contain a lot of sequences.

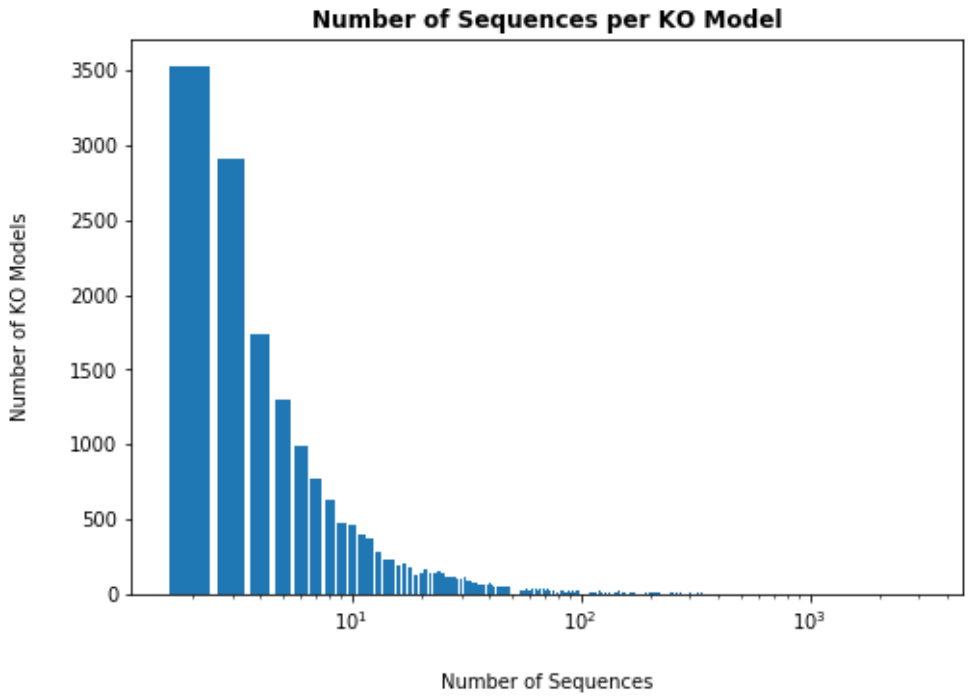


Figure 9: Number of Sequences per KO Model

Total Sequence Number Impacts the Models' Overall Performance

Since the number of sequences per KO model can change drastically (see **Figure 8**), we wanted to estimate how the total number of sequences fed into the pipeline would affect the overall performance. We used the protein family TPR_21 that contained 21,544 sequences to test this. A random 2000 sequences were set aside at the start and 492 of these sequences had KO annotations. Using the remaining 19,544 sequences, we started out with a random set of 2,000 sequences. An HMM Profile was built off of the propagation conducted from this set of 2,000 sequences. We repeated this process of building a tree, propagating, and generating HMM Profile nine more times, each time adding 2,000 more sequences to the mix.

We used our constant ten percent set that we set aside at the beginning of the process to estimate our performance by searching these sequences against our 10 sets of HMM Profiles that we continuously added 2,000 sequences to. In this way we are able to gage how our performance changes with the inclusion of more sequences. **Figure 9** shows how the performance changes based on precision, sensitivity, and accuracy.

2,000 was the fewest amount of sequences, seen in **Figure 9** in light blue. In terms of sensitivity this set of sequences performed lower than the other. In terms of accuracy, a more stringent e-value threshold is necessary for it to perform on par with the other sets of sequences. Additionally, about 20,000 sequences was the max number. In terms of precision, it performed less well than the other sets; however, in terms of sensitivity, it performed better.

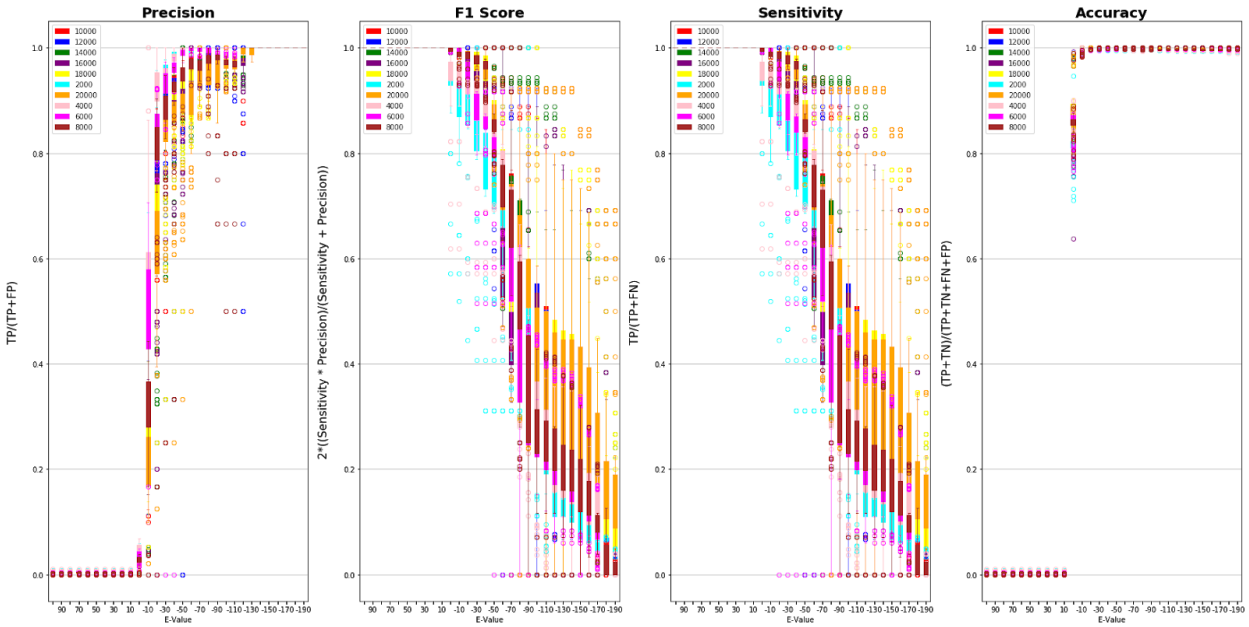


Figure 10: Overall Impact of Number of Sequences on Performance

Performance Over Selected E-Values

Based on **Figure 9**, the performance varies based on the e-value. In order to figure out an ideal e-value threshold, we looked at three specific e-value points: 1, 1e-40, and 1e-170. We considered the same set of sequences and performance as in the previous section and the results could be viewed in **Figure 10**. As previously mentioned, there is a tradeoff between precision and sensitivity. Looking at **Figure 10**, this becomes more apparent as the e-value threshold becomes more stringent.

Similarly, with a bit more stringent hold starting 1e-40, the sequence set with the least amount of sequences, light blue in **Figure 10**, performs less well in terms of sensitivity in comparison to other sets of sequences.

At the most stringent threshold, 1e-170, the sequence sent with the greatest number of sequences, orange in **Figure 10**, is one better performing sets according to all three metrics.

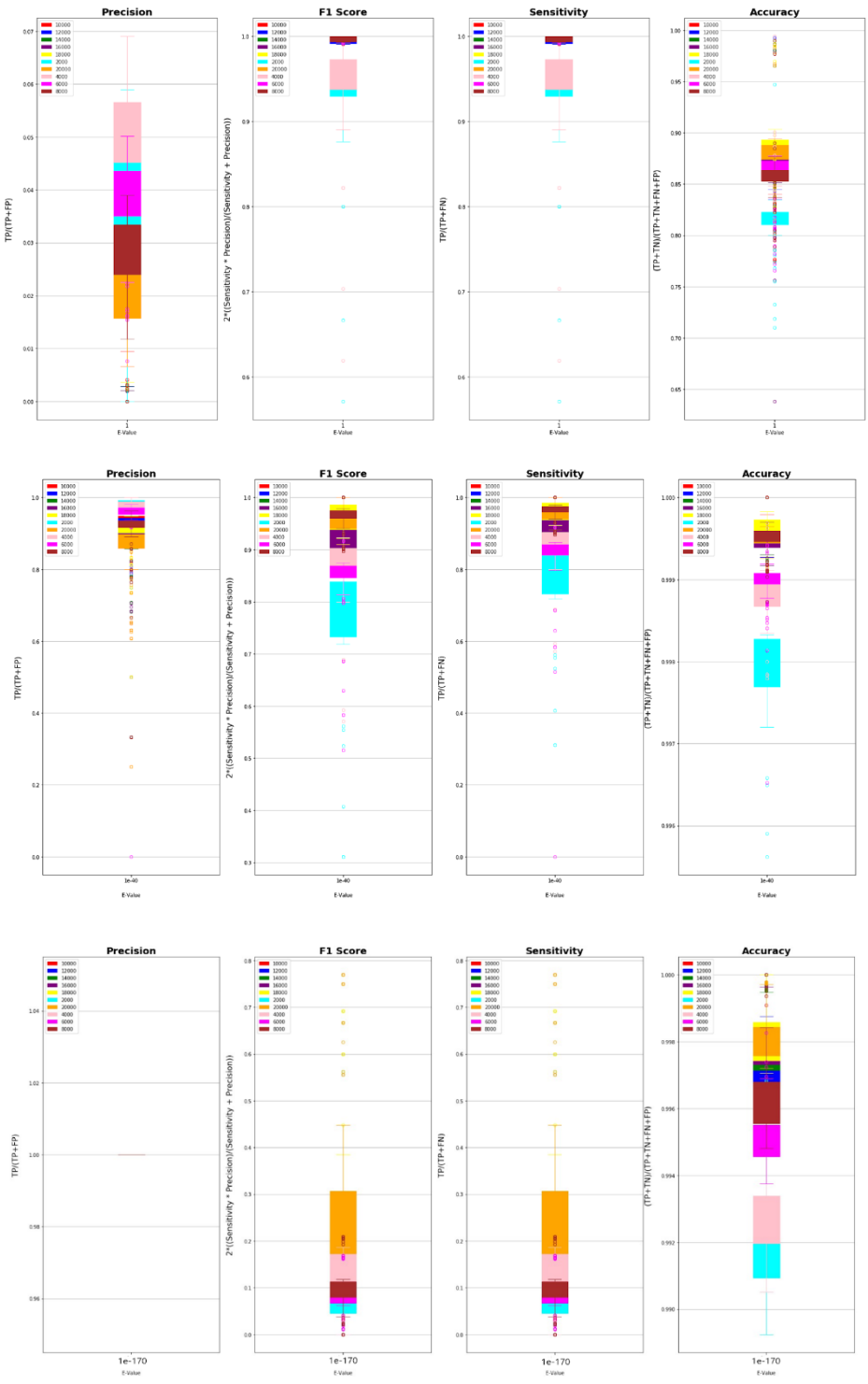


Figure 11:Overall performance at 3 E-values

a shows the metrics at an e-value of 1, **b** at e-value of 1e-40, and **c** at 1e-170

Number of Sequences per Cluster does not Significantly Impact Model's Precision

Since there is a tradeoff between sensitivity and precision, we wanted to see if we did lean in favor of developing precise models, if that would be influenced by the number of sequences in the model. **Figure 11** shows the precision vs the number of sequences at three e-values: 1, 1e-40, and 1e-170 along with the maximum precision achieved at any e-value.

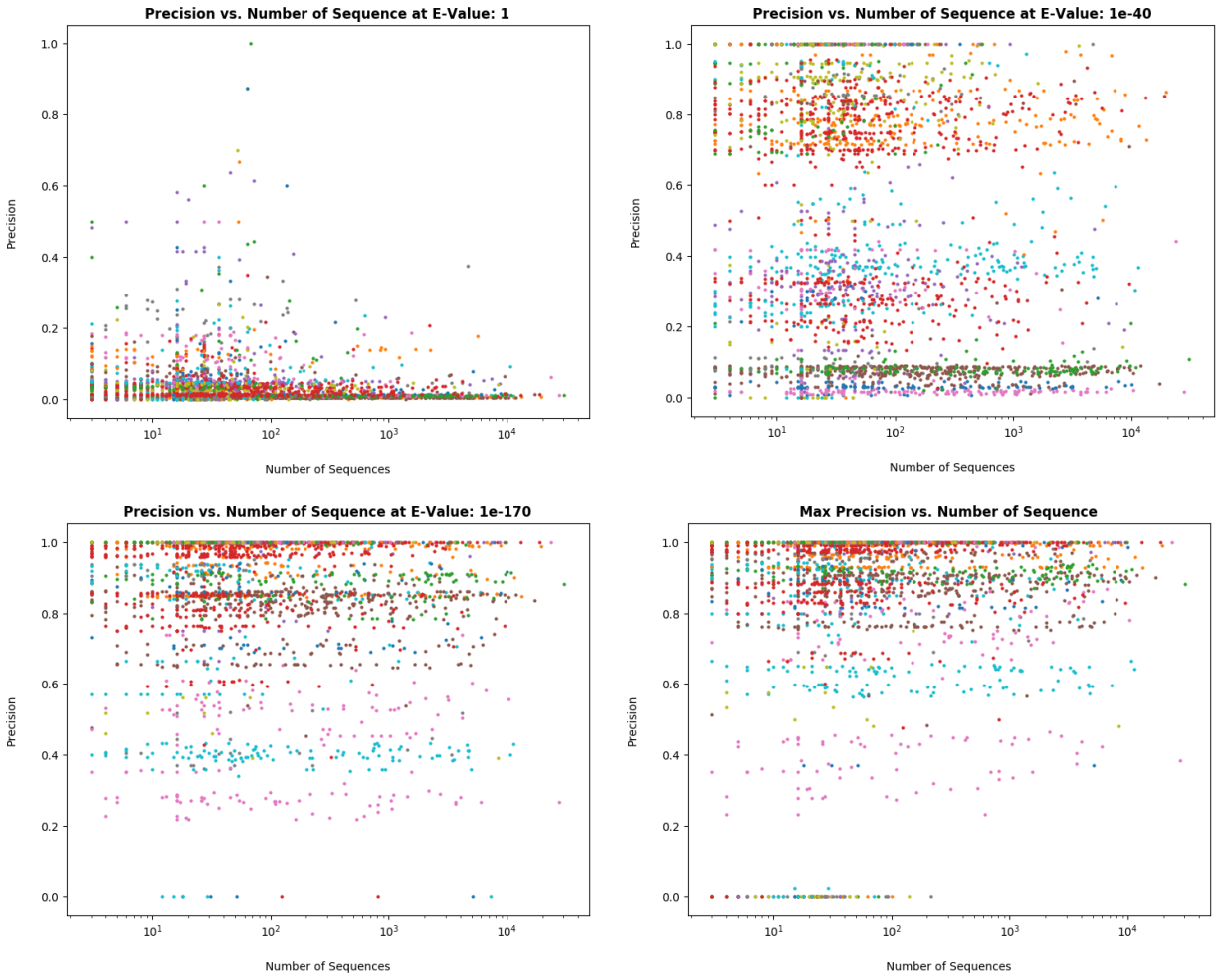


Figure 12: Precision vs Number of Sequence per KO

The general trend shows that the higher the e-value, the greater the precision. However, as the number of sequences in a cluster increases, there does not appear to be a trend between the number of sequences and precision.

Additionally, looking at the Spearman Rank Correlation Coefficient, based on the average values, there does not appear to be a relationship between the two, as seen in the following table:

E-value	Average Spearman Correlation Coefficient	Average p-value
1	0.35	0.37
1e-40	0.05	0.603
1e-170	-0.006	0.533

Table 2

Shows the Spearman Rank Correlation Coefficient for each of the 3 e-values comparing the number of sequences vs. precision

Refining KO Models with Larger Sequence Cluster

Figure 8 shows it is seen that the majority of HMM Profiles correspond to a lower number of sequences. However, to have a profile built on very few sequences means that these profiles are not well characterized as there are not enough sequences present to fully develop the probability scores assigned to the alignments. So, we explored the effects of assigning a threshold that HMM Profiles may only be built if there are at least five sequences present in the KO cluster. The following goes through the effects of this threshold and how that would change our results.

This does not affect how many sequences are propagated, only how many HMM Models are built. As such that would affect the overall performance of our models.

Number of Models Developed Per KO

For the same set of 100 random protein sequences as used in **Figure 6**, the number of HMM models per KO were compared, given that now the HMM Profiles can only be generated if they

contain at least five sequences. **Figure 12** shows the results. In this case, there are only 14,085 HMM Profiles that represent 1,699 KOs. This means that with the introduction of the threshold, 8,175 fewer HMM Profiles were generated and 405 fewer KOs were present.

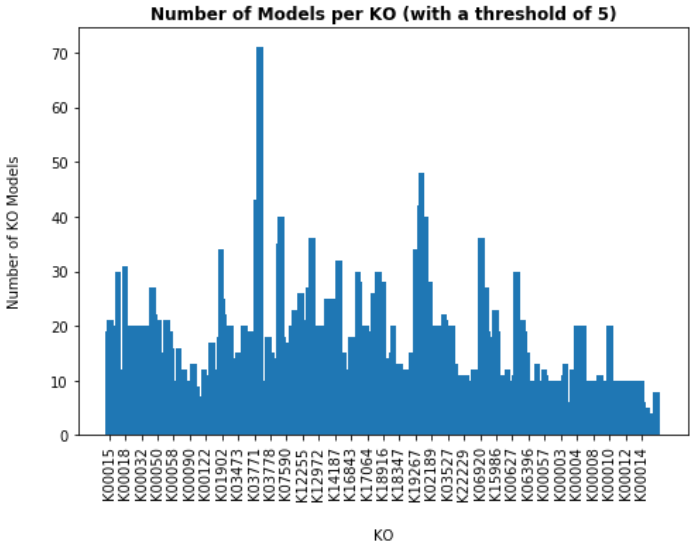


Figure 13: Models per KO with a minimum of 5 sequences

Overall Model Performance of Larger Cluster Show More Variation between KOs

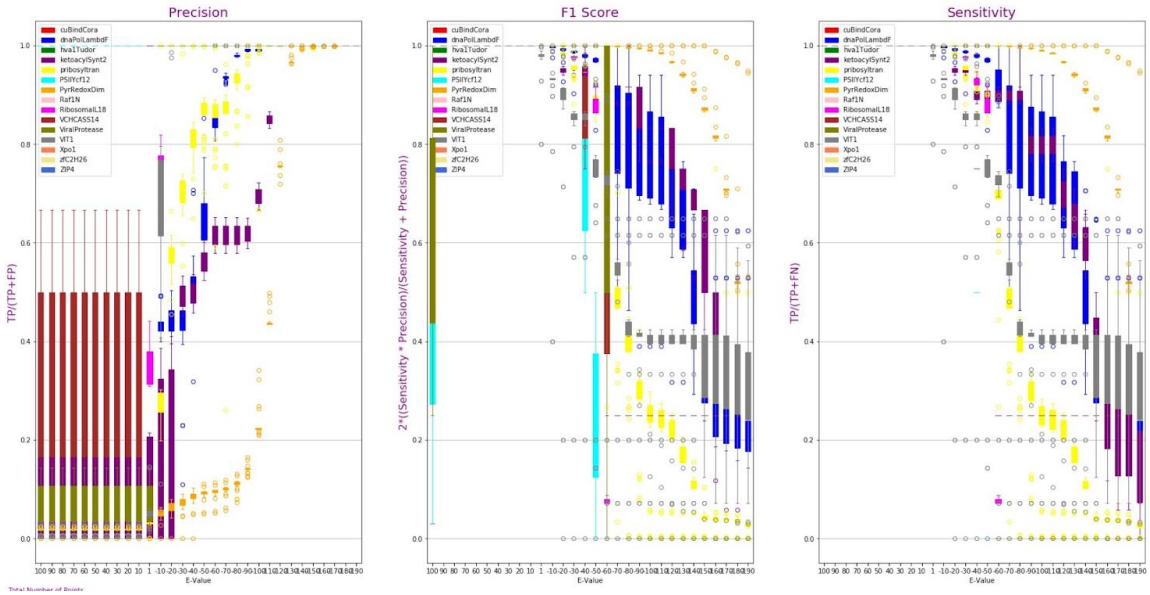


Figure 14: Overall performance for 11 random protein sequence, only considering at least five sequence for profile generation

Figure 13 shows the performance of the HMM models that were built for the KOs of 11 protein families. These are the same protein families that were used in **Figure 7**. The same observation is made that with a tradeoff between precision and sensitivity. However, comparing precision in **Figure 13** and **Figure 7**, it can be seen that certain protein families such as letoacylSynt2 perform more uniformly as the range is smaller.

Impact of Sequence Number on Model Performance

Overall Impact

Figure 15 shows the impact the number of sequences now have with the performance metrics. In comparison to **Figure 9**, there are not any apparent differences. The sets of the least amount of

sequences (2,000) and the sets of the greatest amount of sequences (20,000) seem to follow the same patterns as shown in **Figure 9**.

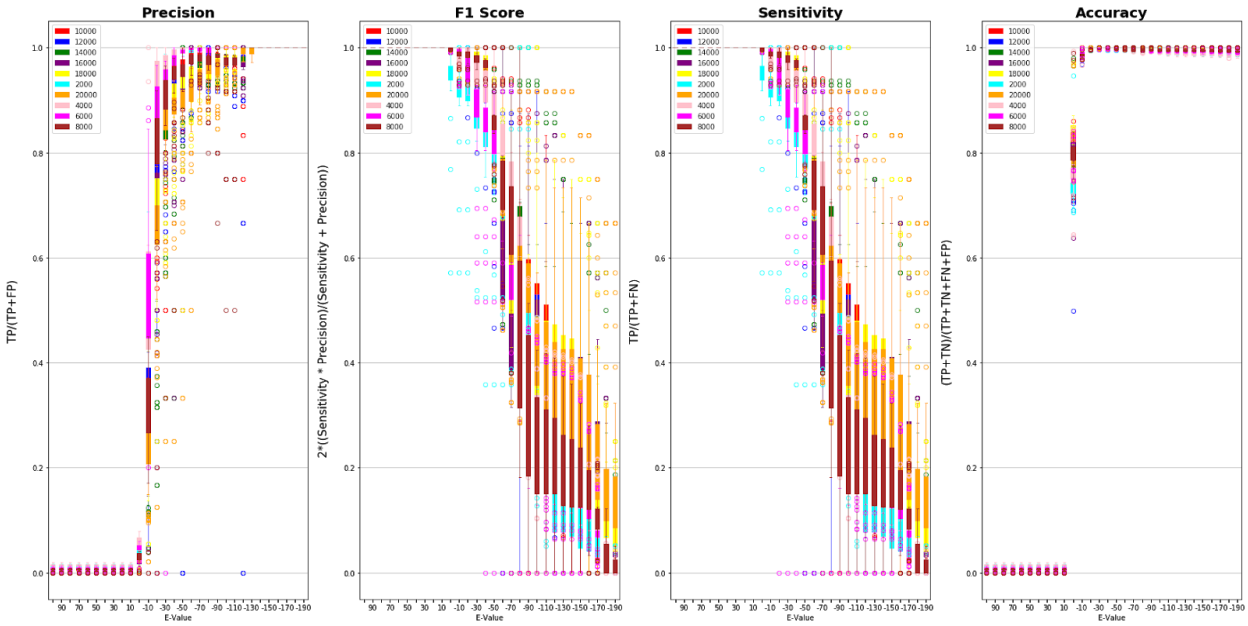


Figure 15: Overall Impact of Number of Sequence on Performance with KOs that contain at least 5 sequences

Precision over Selected E-Values

Based on **Figure 10**, we saw that the performance varies based on the number of sequences, as the sequence set with the least amount of sequences compared less well than the others in terms of sensitivity. We also say that with a stringent threshold, at 1e-170, the sequence set with the greatest number of sequences performed the best.

Figure 16 shows how this performance would change if we limited the HMM Profiles generated if at least 5 sequences make up the KO. At a threshold of 1e-40, while the smallest set of sequences don't perform the best in terms of sensitivity and accuracy, they perform better than they did in **Figure 10**.With the most stringent threshold at the e-value of 1e-170, there is not as much difference between **Figure 16** and **Figure 10**.

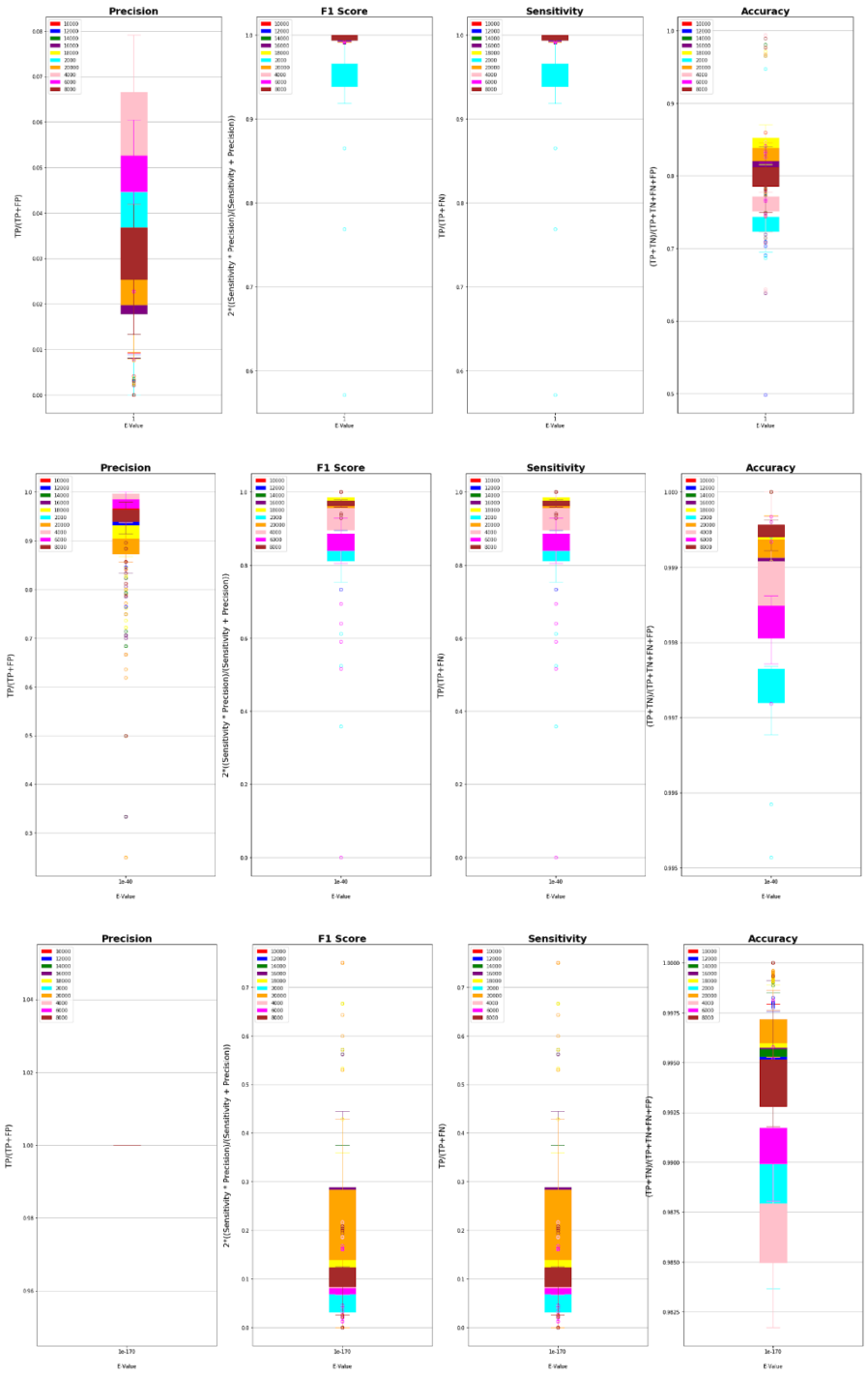


Figure 16: Overall performance at 3 E-values

a shows the metrics at an e-value of 1, **b** at e-value of 1e-40, and **c** at 1e-170.

Guide Tree Reliability

Due to the number of sequences in the protein families and the number of protein families themselves, the initial guide trees outputted by ClustalO were used for the propagation. A comparison between this initial guide tree and a tree built from a full alignment is done below.

Using a protein family CPL, which contained 71 sequences, an initial guide tree and a full alignment was done through ClustalO. This initial guide tree can be viewed in **Appendix 1**. Using the full alignment, a tree was built and replicated 100 times along with its bootstrap value. One instance of this tree with its bootstrapped values can be viewed in **Appendix 2**.Then using CompareToBootstrap.pl (a Fast Tree Comparison Tool), the initial guide tree was compared against the bootstrapped tree. This generated a new tree with the initial guide branch lengths and new bootstrap values, which can be viewed in **Appendix 3**. These bootstrap values at each node shows the fraction of times that leaves within the nodes is maintained within the 100 replicated trees. This process was repeated to compare the initial guide tree to a bootstrapped replicated 500 times. **Figure 18** shows the distributions of the new bootstrapped values after comparing the trees.

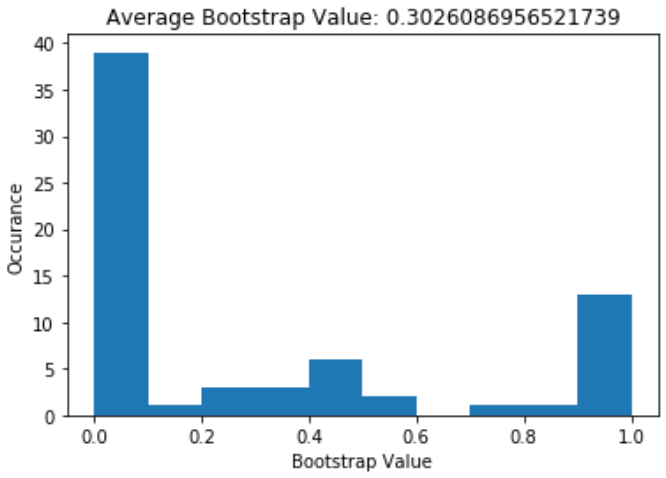


Figure 17: Compared tree bootstrap value distributions with 500 tree replications

Discussion

Propagation Control

When propagating our sequences from a known KO to an unknown KO, it is important to note that there is no limitation to how much propagating can take place. If it turns out that there is a known KO with 100 unannotated children, these children will be propagated from that one known KO. Based on **Table 1**, the average number of sequences increased by about 53%. We want to avoid the case where we are incorrectly annotating KOs. However, we also do not want to lose sensitivity by imposing conditions on our propagation. Somewhere in there we need to find the balance between control of our propagation and freely annotating.

Threshold to Use

E-Value

For much of the results section, we considered three e-values: 1, 1e-40, and 1e-170. We did this to see if there is a threshold we should enforce. There is a really high threshold, 1, which is loose as you would expect more sequence hits to generate here, as it also did. At a loose threshold of 1, there were a lot of hits, as expected. Similarly, at a stringent threshold of 1e-170, there were less hits, again, as expected.

Through this, we saw that there is a tradeoff between precision and sensitivity based on how strict the e-value got. So, if we were to use a lower e-value for our cutoff, the precision will perform really well but we will sacrifice sensitivity.

Total Number of Sequences

A considerable amount of the results went into the effects of imposing a limitation on the number of sequences to actually build an HMM Profile from. In our case, we just tested using a threshold of 5 sequences, which overall seemed to improve the performance. Further exploration will be needed to refine this threshold. By placing this threshold, we are being more selective on the HMM Profiles we create, but would potentially waste some of the newly annotated sequences. For example, **Figure 12**, from a random 100 protein families, we constructed 8,175 fewer HMM Profiles and identified 405 fewer KOs. In this case, we could also be losing valuable information and annotation.

Validation

While the 10-cross validation allowed us to develop these models, we need to identify an independent set of sequences in order to make sure our HMM Profiles are effectively predicting KO annotation and not overfitting our dataset. We plan to pull in all sequences from the KEGG database, which already contain a KO annotation, and remove all sequences that were used during the training set (hoping that KEGG latest update was not yet taken in account by PFAM).

Conclusion

Over 98% of bacteria are uncultivated, meaning there is a lot of information that we do know. Our propagation method aims to define this gap by characterizing sequences based on what they are similar to. The aligned trees and Profile HMMs are based on the formed protein families and the annotated KEGG database. We now are left to develop the thresholds and constraints we use in order to develop quality protein annotations and in turn assign some characterization in order to improve our understanding of genetic-based biological processes.

Works Cited

1. **Wadapurkar, R.M., & Vyas, R.** (2018). Computational analysis of next generation sequencing data and its applications in clinical oncology. *Informatics in Medicine Unlocked, 11*, 75-82.

2. **Shen, L., Shao, N., Liu, X. *et al.*** ngs.plot: Quick mining and visualization of next-generation sequencing data by integrating genomic databases. *BMC Genomics* **15,** 284 (2014). https://doi.org/10.1186/1471-2164-15-284

3. **Koonin EV, Galperin MY.** Sequence - Evolution - Function: Computational Approaches in Comparative Genomics. Boston: Kluwer Academic; 2003. Chapter 3, Information Sources for Genomics. Available from: https://www.ncbi.nlm.nih.gov/books/NBK20256/

4. **Wade W**. (2002). Unculturable bacteria--the uncharacterized organisms that cause oral infections. *Journal of the Royal Society of Medicine*, *95*(2), 81–83. https://doi.org/10.1258/jrsm.95.2.81

5. **Garza, D. R., & Dutilh, B. E**. (2015). From cultured to uncultured genome sequences: metagenomics and modeling microbial ecosystems. *Cellular and molecular life sciences : CMLS*, *72*(22), 4287–4308. https://doi.org/10.1007/s00018-015-2004-1

6. **Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ.** Basic local alignment search tool. *J Mol Biol*. 1990;215(3):403-410. doi:10.1016/S0022-2836(05)80360-2
7. **Haque, W., Aravind, A.A., & Reddy, B.** (2009). Pairwise sequence alignment algorithms: a survey.

8. **Kent W. J.** (2002). BLAT--the BLAST-like alignment tool. *Genome research*, *12*(4), 656–664. https://doi.org/10.1101/gr.229202

9. **Pertsemlidis, A., Fondon, J.W.** Having a BLAST with bioinformatics (and avoiding BLASTphemy). *Genome Biol* 2, reviews2002.1 (2001). https://doi.org/10.1186/gb-2001-2-10-reviews2002

10. **Eddy, S.** What is a hidden Markov model?. *Nat Biotechnol* 22, 1315–1316 (2004). https://doi.org/10.1038/nbt1004-1315

11. **L. R. Rabiner,** "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, Feb. 1989, doi: 10.1109/5.18626.

12. **Eddy, Sean R.** 1998. "Profile Hidden Markov Models." *BIOINFORMATICS REVIEW* 14 (9): 9.

13. **Amit, Tomer.** (2019). "Introduction to Hidden Markov Models." *Towards Data Science*.

14. **Wheeler, Travis J., Jody Clements, and Robert D. Finn.** 2014. "Skylign: A Tool for Creating Informative, Interactive Logos Representing Sequence Alignments and Profile Hidden Markov Models." *BMC Bioinformatics* 15 (January): 7

15. **Punta, M., Coggill, P. C., Eberhardt, R. Y., Mistry, J., Tate, J., Boursnell, C., Pang, N., Forslund, K., Ceric, G., Clements, J., Heger, A., Holm, L., Sonnhammer, E. L., Eddy, S. R., Bateman, A., & Finn, R. D.** (2012). The Pfam protein families database. *Nucleic acids research*, *40*(Database issue), D290–D301. https://doi.org/10.1093/nar/gkr1065

16. **Haft, D. H., Selengut, J. D., & White, O.** (2003). The TIGRFAMs database of protein families. *Nucleic acids research*, *31*(1), 371–373. https://doi.org/10.1093/nar/gkg128

17. **Erik L. L. Sonnhammer, Sean R. Eddy, Ewan Birney, Alex Bateman, Richard Durbin**, Pfam: Multiple sequence alignments and HMM-profiles of protein domains, *Nucleic Acids Research*, Volume 26, Issue 1, 1 January 1998, Pages 320–322, https://doi.org/10.1093/nar/26.1.320

18. **Minoru Kanehisa, Miho Furumichi, Mao Tanabe, Yoko Sato, Kanae Morishima**, KEGG: new perspectives on genomes, pathways, diseases and drugs, *Nucleic Acids Research*, Volume 45, Issue D1, January 2017, Pages D353–D361, https://doi.org/10.1093/nar/gkw1092

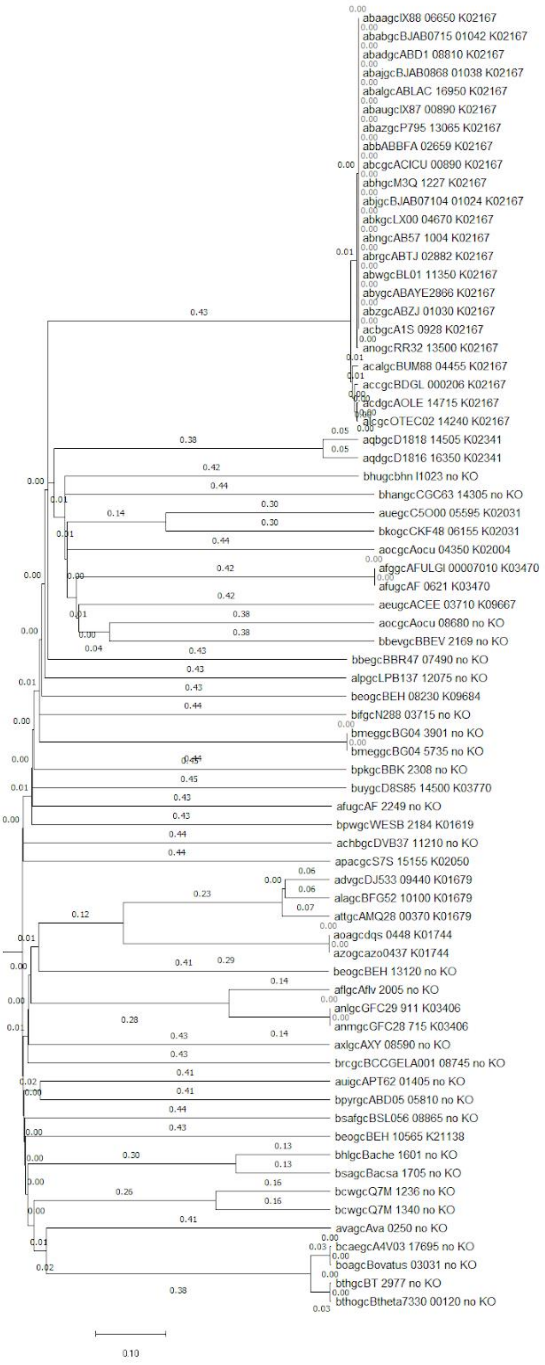
19. **Xizeng Mao, Tao Cai, John G. Olyarchuk, Liping Wei,** Automated genome annotation and pathway identification using the KEGG Orthology (KO) as a controlled vocabulary, *Bioinformatics*, Volume 21, Issue 19, , Pages 3787–3793, https://doi.org/10.1093/bioinformatics/bti430

20. **Sievers F, Wilm A, Dineen DG, Gibson TJ, Karplus K, Li W, Lopez R, McWilliam H, Remmert M, Söding J, Thompson JD, Higgins DG** (2011). **Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega**. Molecular Systems Biology 7:539 doi:10.1038/msb.2011.75

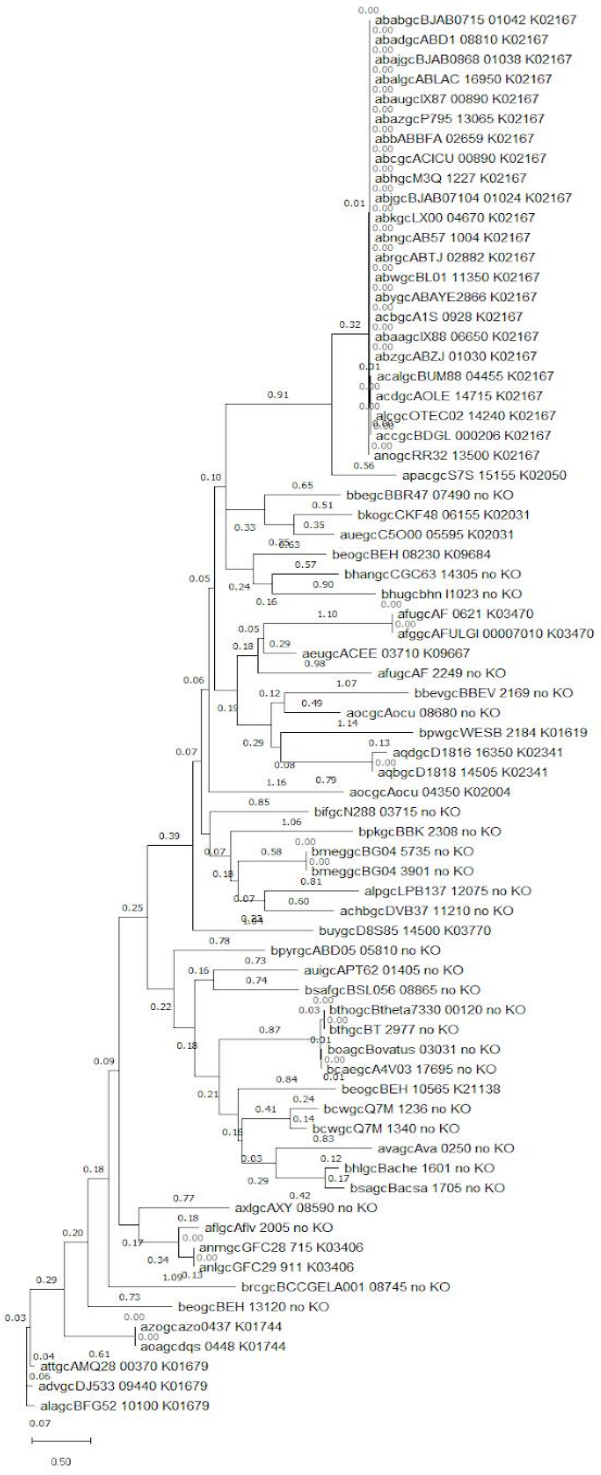
21. **Jaime Huerta-Cepas, François Serra, Peer Bork**, ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data, *Molecular Biology and Evolution*, Volume 33, Issue 6, June 2016, Pages 1635–1638, https://doi.org/10.1093/molbev/msw046

22. **Thompson, J. D., Higgins, D. G., & Gibson, T. J**. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, *22*(22), 4673–4680. https://doi.org/10.1093/nar/22.22.4673

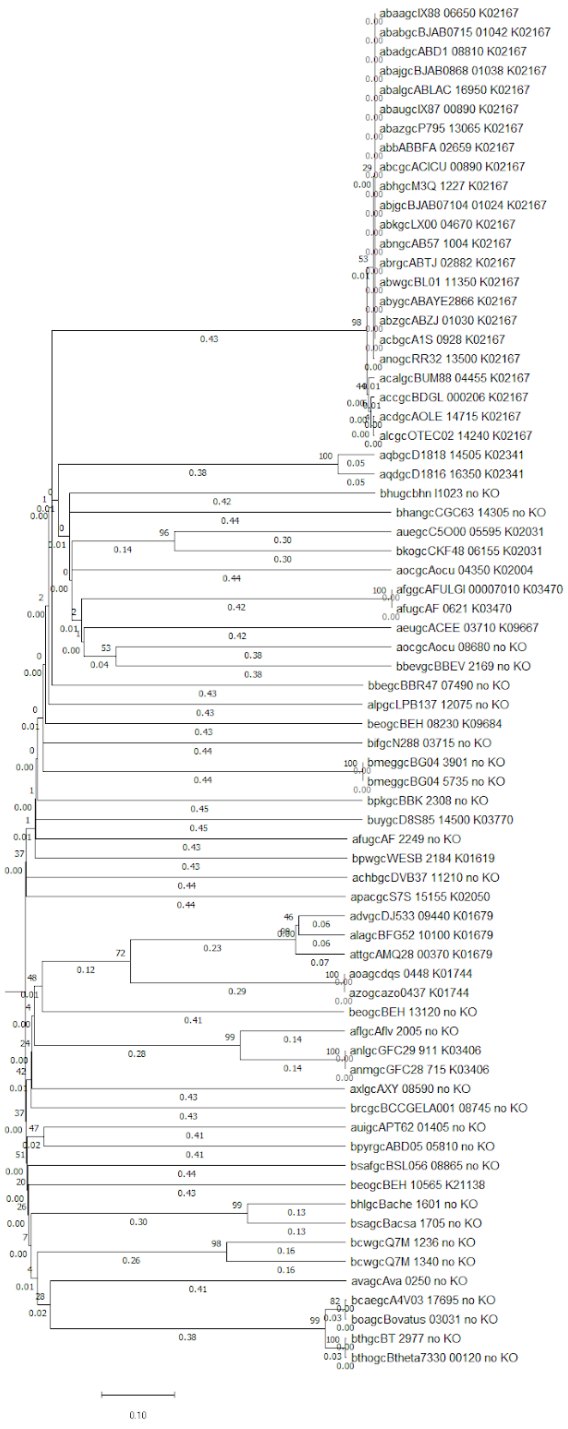
Appendix



Appendix 1: Initial Guide Tree built by ClustalO



Appendix 2: One instance of the bootstrapped tree



Appendix 3: Compared Tree