PROTEIN STRUCTURE ANALYSIS AND PREDICTION UTILIZING THE FUZZY GREEDY K-MEANS DECISION FOREST MODEL AND HIERARHICALLY-CLUSTERED HIDDEN MARKOV MODELS METHOD

by

Cody Landon Hudson

A thesis presented to the Department of Computer Science
and the Graduate School of University of Central Arkansas in partial
fulfillment of the requirements for the degree of

Master of Science
in
Computer Science

Conway, Arkansas
December 2013

TO THE OFFICE OF GRADUATE STUDIES:

The members of the Committee approve the thesis of

Cody Hudson presented on December 9th, 2013.

Dr. Bernard Chen, Committee Chairperson

Dr. Chenyi Hu

Paul Young

Dr. Paul Young

Dr. Sinan Kockara

PERMISSION

Title	Protein Structure Analysis and Prediction Utilizing the Fuzzy Greedy K-Means Decision Forest Model and Hierarchically-Clustered Hidden Markov Models Method
Department	Computer Science
Degree	Master of Science

In presenting this thesis in partial fulfillment of the requirements for graduate degree from the University of Central Arkansas, I agree that the Library of this University shall make it freely available for inspections. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis/dissertation work, or, in the professor's absence, by the Chair of the Department or the Dean of the Graduate School. It is understood that due recognition shall be given to me and to the University of Central Arkansas in any scholarly use which may be made of any material in my thesis/dissertation.

Cody Hudson

Cody Hudson

November, 22, 2013

ABSTRACT

Structural genomics is a field of study that strives to derive and analyze the structural characteristics of proteins through means of experimentation and prediction using software and other automatic processes. Alongside implications for more effective drug design, the main motivation for structural genomics concerns the elucidation of each protein's function, given that the structure of a protein almost completely governs its function. Historically, the approach to derive the structure of a protein has been through exceedingly expensive, complex, and time consuming methods such as x-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy.

In response to the inadequacies of these methods, three families of approaches developed in a relatively new branch of computer science known as bioinformatics. The aforementioned families include threading, homology-modeling, and the de novo approach. However, even these methods fail either due to impracticalities, the inability to produce novel folds, rampant complexity, inherent limitations, etc. In their stead, this work proposes the Fuzzy Greedy K-means Decision Forest model, which utilizes sequence motifs that transcend protein family boundaries to predict local tertiary structure, such that the method is cheap, effective, and can produce semi-novel folds due to its local (rather than global) prediction mechanism. This work further extends the FGK-DF model with a new algorithm, the Hierarchically Clustered-Hidden Markov Models (HC-HMM) method to extract protein primary sequence motifs in a more accurate and adequate manner than currently exhibited by the FGK-DF model, allowing for more accurate and powerful local tertiary structure predictions. Both algorithms are critically

examined, their methodology thoroughly explained and tested against a consistent data set, the results thereof discussed at length.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

I. Introduction

Structural genomics is a field of study that strives to derive and analyze the structural characteristics of proteins through means of experimentation and prediction using software and other automatic processes [20]. Alongside implications for more effective drug design [38], the main motivation for structural genomics concerns the elucidation of each protein's function, given that the structure of a protein almost completely governs its function [15]. Currently, structural genomics is supported through a synergetic gambit of processes and applications on both the experimentation and prediction side, including (respectively) "wet lab" procedures such as x-ray crystallography [3] and nuclear magnetic resonance (NMR) spectroscopy [21], and bioinformatics algorithms [4] which include homology-modeling, threading, and de novo modeling [4]. Wet lab procedures drive the process of structural genomics such that "target" proteins are selected and their structures explicitly determined through accurate albeit extremely expensive and time consuming processes. The target proteins are selected in such a manner that allows the predictive algorithms to determine the structure of proteins that are either sequentially or structurally homologous to the target proteins, allowing for accurate structural analysis of most proteins by only explicitly determining the structure of a few.

Granted this, there are significant drawbacks to this current approach of wet lab driven structural genomics, the most prominent of which being that current predictive algorithms are heavily dependent on the continual explicit determination of protein structures through the resource intensive wet lab procedures. This work would propose and discuss a new predictive algorithm that analyzes protein structure not through strict homologues, but rather seeks to discover sequential patterns, or motifs, that transcend families of homologous proteins. Unknown proteins analyzed by this approach. This approach allows for the prediction of new protein structures by strictly analyzing the current record of known protein structures for shared motifs that are not aligned alongside protein families, determining the structure generated from each extracted motif, and aligning the motif (and its structure) with the sequence of the new protein. To that effect, this work presents two algorithms: the Fuzzy Greedy K-means Decisions Forest model (FGK-DF model), and the Hierarchically Clustered Hidden Markov Models (HC-HMM). The FGK-DF uses a combination of clustering to determine both the non-homologous sequential motifs and then decision tree classifiers to determine if unknown proteins share a given sequential motif and thus structural motif. The HC-HMM is an attempt at resolving several limitations presented by the FGK-DF, most notably an assumed sequential motif size. Given this, it is imperative for fully understanding the implications of the methods proposed and the problems presented by the current methodologies to have sufficient background information concerning the subject, specifically concerning protein structure anatomy, a closer examination of the aforementioned wet lab experiments and existing predictive approaches, and finally a brief overview of the FGK-DF model and HC-HMM.

1.1 Protein Structure Anatomy

In the preceding section, protein structure referred specifically to the three dimensional structure of the protein, its “shape,” so to speak. In fact, protein structure can be broken down into four primary categories, three of which this research deals with directly: primary sequence, secondary structure, and tertiary structure, all shown in Figure 1 below. The first of these categories is the primary sequence of the protein, described as an ordered sequence of chained amino acids. Amino acids are molecules hailing from a family of twenty possible amino acids that form, essentially, the alphabet of protein composition. That is to say that amino acids are the building blocks of proteins, determining, more than anything else, the characteristics that the protein assumes (most notably the "shape"). The arrangement of these amino acids can cause highly regular substructures to appear as the amino acids are chained together. These substructures form the secondary structure of the protein, the second category of protein structure, and often come in the form of three overarching motifs: alpha-helix, beta-sheet, or coils. These motifs arise from complex intermolecular reactions between components of the primary structure as they form bonds, creating folds and other such structures that appear as the secondary structure.

Once the secondary structure is stabilized, the tertiary structure, the third category of protein structure, begins to take shape in a process known as folding. A protein folds due to intermolecular forces as well as environmental conditions, such as the presence of water or temperature, and eventually reaches a stable form known as the "native state." This native state describes the correct tertiary structure of the protein, or, rather the three dimensional shape of the protein. This is the most pivotal category of structure as it is, as has been mentioned, the sole determinant of the function of the protein. It has also been the most elusive, historically, as the next section will explore.

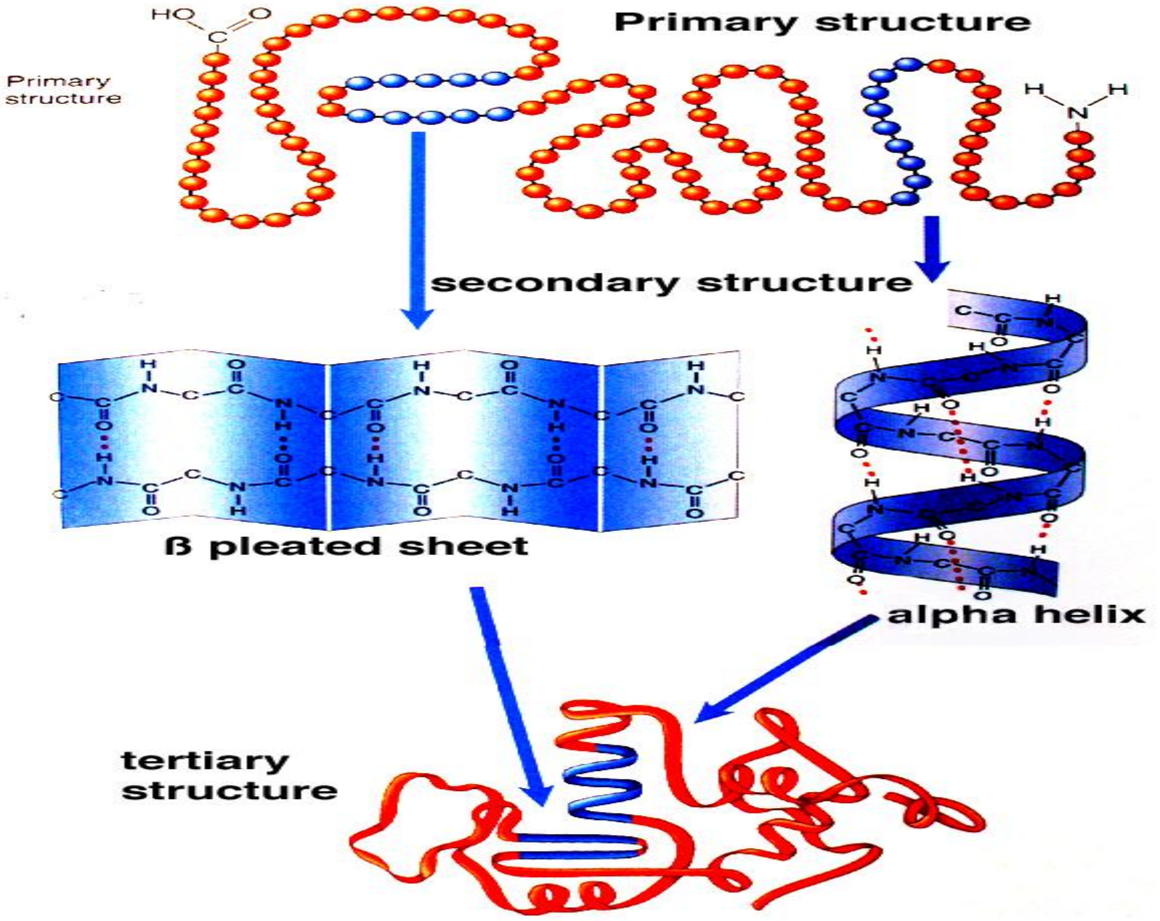


Fig. 1: Protein Structure Anatomy [28]

1.2 Wet-Lab Procedures: X-Ray Crystallography and NMR Spectroscopy

Historically, the method by which one would discern the tertiary structure of a protein is with a physical experiment using one of two major processes, x-ray crystallography and nuclear magnetic resonance spectroscopy. The former, introduced in the early 20th century, relies on firing x-rays through crystallized macromolecules, such as proteins, and recording the resulting diffraction pattern which can then be analyzed, mathematically, to produce the structure of the macromolecule. This requires that the molecule can be crystallized (which can exclude vast portions of proteins, such as those associated with membrane functions), not to mention the extensive amount of time, effort, training, and money that must be poured into the process to determine the structure of only one macromolecule. Furthermore, imperfections in the process can lead to unusable resolutions of the structures, making the already complex process frustratingly impractical in many cases [3]. Despite this, according to the Protein Data Banks, there have been over 94,715 structures (including all biological macromolecules, not just proteins) determined by this method alone [17].

The other major competing method of structure determination is the aforementioned NMR spectroscopy, which utilizes the properties of magnetic fields produced by the spin of charges in atomic structures to produce information on their chemical and physical properties. This can be extended to determine the structure of molecules [21]. Unfortunately, this method can generally only be applied to much smaller molecules and proteins, though it is often times the only method by which one can experimentally determine unstructured proteins. Perhaps more so than x-ray crystallography, this method is extremely expensive, requiring massive machinery and considerable expertise to analyze or even produce the results. Both methods (x-ray crystallography and NMR spectroscopy) are also extremely slow, relatively speaking, and can be prohibitively impractical in certain cases. This is why structural genomics, while driven by experimentation, is largely supported by predictive algorithms [BAKER-GENOME] as the next section will explore.

1.3 Bioinformatics: Predicting Protein Structures using Computer Science

In the late 21st century, after computing and the science thereof had been considerably established, a growing field concerned with extensive data analysis and *mining* had begun to crop up amongst those interested in such things. This coincided, to no one's surprise, with an overwhelmingly large need to analyze the massive amounts of data being produced by the now vastly popular tools the internet had provided. This gave birth to a wide range of fields all contained, if remotely, under a field known as data mining. Soon, the so-called "informatics" spread to other disciplines, including biology, which of course resulted in the aforementioned bioinformatics [4]. It was not long before the new study concerned itself with the mission of structural genomics, forming three distinct approaches to the problem: homology-modeling, threading, and the de novo approach [4]. Each of these approaches is explained briefly in the following paragraphs, along with a brief discussion of the inherent weaknesses of each approach in the context of protein structure prediction.

1.3.1 Homology Modeling

Homology-modeling, in a sentence, attempts to exploit the mechanism by which protein evolution operates, such that proteins that share an evolutionary ancestor are said to have similar tertiary structures. This is in part due to the fact that there are only three primary ways a protein sequence (and, subsequently, its structure) can change over time: insertions, deletions, and swaps in the amino acids in its primary sequence. The former, as one would expect, is the situation when one or more amino acids are inserted into a random location on the protein. Deletions are the opposite, such that random sequence of one or more amino acids on the primary sequence is removed. Swaps are effectively changes in place, where the deleted portion of the primary sequence is replaced with an inserted portion. The following figure demonstrates the three types of mutations:

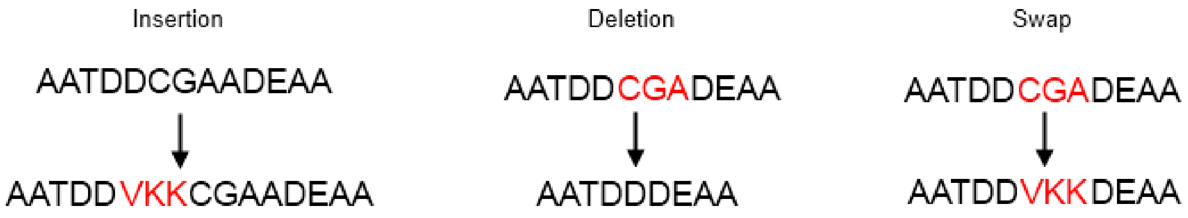


Fig. 2: Protein Sequence Mutations

These three types of mutations lead to the various branches within a given protein family, such that all the branches, presumably, lead to a "root" protein, the shared evolutionary ancestor. Each protein that is descendant of that "root" protein is just a set number of swaps, insertions, and deletions away from the root. This suggests that to relate one homologue to another (that is, two proteins that share the same "root" protein), one can simply work their way to the root and then back down again to the target homologue through a unique series of mutations. Primary sequence largely determines structure, such that if one can relate a given sequence to another sequence, they can also relate a given structure to another structure. This is compounded with the fact that function is largely conserved throughout protein evolution (as non-functioning proteins would die out), and as function and structure are also intertwined (though not to the same extent as sequence to structure), common evolutionary ancestors imply common structures through common functions [18].

Given this, homology-modeling attempts to build the roadmap, so to speak, from the potential template sequences to the root protein back to the target sequence. This is done, generally, with a process known as dynamic programing [34], specifically sequence alignment. In its most naïve form, sequence alignment attempts to find the optimal alignment between two primary sequences by inserting, deleting, or swapping characters in the template sequence, such that it optimally matches the target sequence. Once can clearly see that this approach emulates the evolutionary mechanisms of the proteins. Using substitution matrices [29] and scoring from the alignment, one can detect homologues and the strength of the evolutionary link. This produces a score known as "sequence identity," such that those homologues which are close, evolutionarily, have a higher sequence identity between each other. Homology-modeling attempts to best align those homologues with the highest sequence identity to the target sequence, such that the structures (which are assumed to be conserved) of the templates are said to be the structures of the target sequence. Those segments that can't be aligned must be filled in using "loop modeling," as those are, generally, the regions that are highly flexible in the physical protein structure (i.e. the "hinges") [35].

1.3.2 Threading

Threading, conversely, determines a template protein according to similarities in the folding of the tertiary structure between the target (the protein whose structure is being predicted) and potential template proteins from enormous databases. In more explicit terms, the approach attempts to model the target protein by aligning, or “threading,” an unknown protein’s sequence “to a known structural motif” [32]. This requires one to have a database of “spatial folding templates,” (i.e. the known structures to which one can align a primary sequence to), to perform the prediction process. In its purest and most naïve form, the unknown protein’s sequence is aligned one amino acid at a time against these templates until a best fit is found. This best fit, being one of the aforementioned templates, has the corresponding “structural motif” (i.e. the fold) that is said to be the tertiary structure of the unknown protein. In other words, the threading approach “recognizes the protein sequences likely to fold into similar structures” [32].

Both threading and homology-modeling are part of a larger subset of prediction approaches known as template-based modeling, in which a target protein is modeled against templates selected based either on homologies or folding. Despite this, both approaches have significantly different drawbacks to their methodology. Homology modeling has a clear limitation: prediction is limited to only those proteins with existing and identified homologues. This is best exhibited by the clear correlation between low sequence identities and lower prediction accuracy, especially when sequence identity falls below 20% [7]. Furthermore, those regions that must be modeled using loop modeling can be incredibly inaccurate [35]. Threading, while it can have the loop modeling problem (as it runs into the same issue of modeling the “hinges”) is not limited to protein family boundaries, but the presence of conserved folds and structural motifs, which are conserved across protein families much more so than evolutionary homologies. In fact, it was found, according to the 2010 CATH release notes, that there are 1,282 folds conserved across protein family boundaries, as opposed to the highly specialized 2,549 "homologous superfamilies." This causes homology-modeling to have a much larger search space and inherent data complexity than threading. Despite this, both approaches fail to generate novel structures through prediction. In other words, since both approaches rely on structural templates based on known structures, any predictions made will be based off of those known structures, and thus no unique structure predictions can be made. This is an incredibly important drawback as it forces continued reliance on the aforementioned wet lab procedures to produce new protein structures through explicit experimentation to allow for further predictions.

1.3.3 De Novo Modeling

The other method, de novo modeling [8], takes a radically different approach to predicting protein structures, in that the mechanisms by which protein folding occurs is simulated and modeled rather than the structures themselves being formed based on matching templates. The simulation environment can be given an input of only the primary sequence of a protein and, without referring to a database of known protein structures, it can produce a simulation of the folded protein and thus its end tertiary structure. The simulation environment itself can be generated through sampling a "conformation space," (i.e. the possible and expected structure of the proteins given constant conditions), from which possible structures are generated, scored, and refined. This, in turn, is supported by a plethora of mathematical models and equations that model physical laws, free energy minimization, water and amino acid hydrophobicity, and so on. The sheer complexity and range of approaches contained under the de novo approach prohibits further discussion in this work on the matter, but it should be noted that the same complexity prohibits the de novo approach from being applicable to all but the smallest proteins. Furthermore, template-based modeling, such as threading and homology-modeling, have consistently outperformed de novo approaches in the past [31], and thus, despite the fact that it is the only branch of structure prediction that can produce novel folds, the de novo approach will not be considered further in this work.

1.4 The Fuzzy Greedy K-means Decision Forest Model and Hierarchically Clustered Hidden Markov Model

In short, the base FGK-DF algorithm is a hybrid template-based approach that, instead of folds or homologies, uses subtle, conserved primary sequence motifs (that is, sequential motifs) that, much like folds, transcend protein family boundaries. It does this on a local tertiary structure level rather than the conventional global (or total) tertiary structure level, which not only affords the algorithm a much higher resolution at the prediction level, but also allows the algorithm, amongst its other mechanisms, to provide semi-novel folds, a great improvement over the shortcoming described for the conventional template-based approaches. In more explicit terms, the FGK-DF model trains itself on a large, non-homologous training dataset (over a half million protein segments), granulizing and clustering the information in the training set based on the shared presence of these so-called sequential motifs extracted using a sliding window technique with a fixed size. From there, decision trees are trained on each cluster, utilizing both primary and secondary information such that each tree can discern if a target protein segment contains the sequential motif within its primary sequence. Once the entire forest of decision trees are trained, they can be searched according to primary sequence "distance" to find the best fit (in the a similar manner as that described by homology-modeling), and then using the decision tree to decide if the target protein contains the motif that characterizes the cluster the tree is trained on. If it does, the tertiary structure is predicted to be the average (and thus novel) structure of either the cluster or a given branch on the tree, depending on the particular setup of the model. This is repeated for each target protein, such that the end product is a cheap, accurate, and quickly determined tertiary structure for a vast number of protein segments (and thus proteins themselves) that does not rely on continual support from wet lab procedures to produce new, explicit structures. Rather, the FGK-DF model allows for finer grain analysis and data extraction on the already considerable wealth of data existing in the Protein Databanks and other relevant databases.

Granted that, the FGK-DF model has its own inherent weaknesses, the most prominent of which is an assumed motif size. As the FGK-DF model relies on a slide windowing technique with a set window size to extract segments from proteins in order to determine potential sequential motifs, there is an implicit assumption on the maximum size a motif can be. This can cause motifs that are much larger than the assumed size to be needlessly segmented, and protein motifs that are smaller than the assumed size to be hidden by the "noise" of non-conserved local amino acids. The HC-HMM algorithm was developed to resolve this issue and be reactive to the size of potential sequential motifs, rather than force users of the algorithm to make assumptions on the expected or average size of all sequential motifs in a given dataset. To do this, each protein sequence used to build the database of known, transcending sequential motifs is converted into a Hidden Markov Model [24]. HHMs are based on a system of states and transitional probabilities that exist between those states. This allows one to model a protein sequence as a series of states reflecting both the composition and aforementioned evolutionary behaviors of proteins (insertion, deletion, etc.). Using simple distance calculations on each ‘node’ of the generated HMMs, it becomes a simple process of aligning and clustering each HMM with another HMM exhibiting minimum distance. This process of aligning and clustering is repeated until there is only one HMM cluster left, from which sequential motifs can be extracted based on a minimum number of overlapping, aligned HMM nodes. Since the motifs are based on alignment overlap in a given cluster, no size is assumed on the motif, presumably extracting more accurate and complete sequential motifs than the FKG-DF model.

Granted all of this, the model of the base FGK-DF model and HC-HMM will be extensively explained and defended. The development history, methodology, and short exploration of the results of the FGK-DF model will be explored. Discussion will then turn to the extensions and utility provided by the HC-HMM, how exactly the data is modeled, further discussion and explanation of the proposed algorithm, and analysis of the algorithm’s effectiveness at extracting motifs. Of course, before any further discussion can begin, it is pivotal to understand, exactly, what constitutes the basis of this entire work, the FGK-DF model. What are the underlying algorithms? And, of course, what is the data being used? Each question, in turn, is answered in the subsequent chapters.

II. Fuzzy Greedy K-Means (FGK) Algorithm

At the core of the FGK-DF model is the FGK (Fuzzy Greedy K-Means) algorithm, which clusters protein information based on sequential motifs found in the primary sequence, supplemented by secondary structure information. As the following sections will explore, the FGK algorithm accomplishes this by first roughly granulizing the data using Fuzzy C-Means clustering [19], and then finely clustering the data using Greedy K-Means [43]. The following sections will provide an extensive and detailed walkthrough of not only just what the various steps required in the FGK algorithm, but also the data and parameters that are required at each step, as well as the final output and usage of the algorithm.

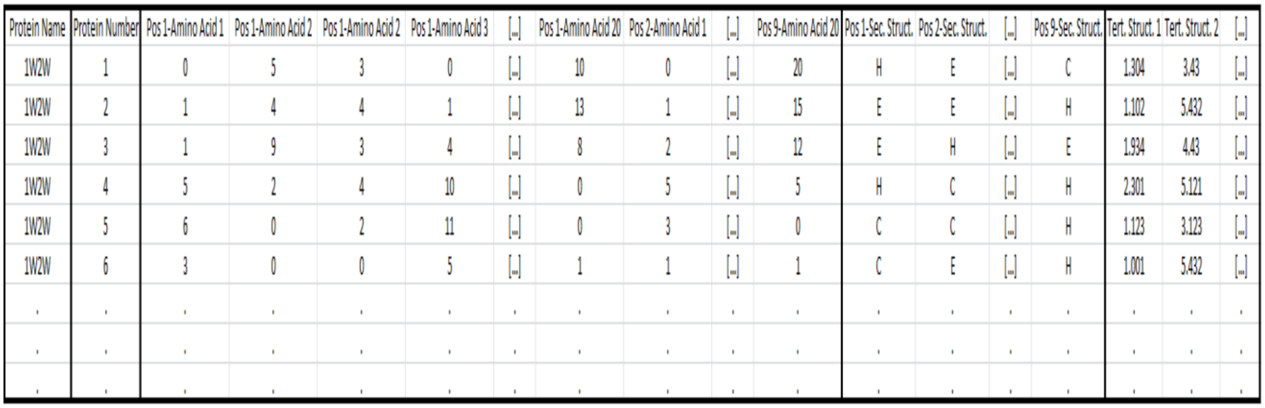
2.1 FGK Algorithm Data Set

The information required by the FGK algorithm includes the primary sequence (represented as a frequency profile), secondary structure, and tertiary structure for each protein segment described in the data. Thus, this can leave one with the simple question: where does this protein information come from? The answer, almost in every current model and algorithm, can be linked to a massive database known simply as the Protein Data Bank (PDB) [17]. The database began as a “grassroots effort in 1971,” to replace the inefficient process of exchanging structure data at the time (each atom of the entire protein was represented by a single punch card). The database did more than streamline the process of structure information exchange: by producing a centralized store, the PDB started an era of structure research based on the free and open exchange of protein information to anyone with an internet connection. Although the PDB started out modestly (in 1976, less than thirty protein structures had been archived in the database), by 2006 the number of structures archived was over 40,000, such that many of the new structures were much more complex and detailed than the older structures.

Unfortunately, as the PDB has an extensive history of radical changes in the structure of its datasets, and is primarily for tertiary structure information, other databases and programs are often needed to either make sense of or add to the data to that found in the PDB [33]. The FGK model makes use of three such intermediate databases, those being the Protein Sequence Culling Server (PISCES), Homology derived Secondary Structure of Proteins (HSSP) and Definition of Secondary Structure of Proteins (DSSP). PISCES is a “public server for culling sets of protein sequences from the PDB by sequence identity and structural quality criteria” [16]. In other words, PISCES allows one to generate a subset of the sequences and structure information found in the PDB based on certain constraints on the data. The most important criteria, in the context of the FGK-DF model, is the sequence identity. As will be explained in later sections, the FGK algorithm requires a dataset composed of proteins with a very low sequence identity (i.e. a set of protein primary sequences that are measurably dissimilar from one another). The PISCES server makes the process of “culling” proteins that fit that criteria very straight forward. The FGK algorithm uses PISCES to retrieve primary and tertiary structure information for building the dataset. To fill the secondary structure void, the DSSP is used. Put simply, the DSSP is a database describing the secondary structure of each protein in the PDB [41], making it simple and straight forward to request and append the required information to the primary sequence and tertiary structure information generated by PISCES. However, the data is still not in the form that the FGK algorithm requires it, as the primary sequence for each protein segment is still in its native form, not in the required frequency profile form [27]. To produce a frequency profile, one first must produce a multiple sequence alignment. Once this task is complete, to generate a frequency profile, one simply notes the "frequency of occurrence of each of the amino acids" at each position in the primary sequence. Assuming one is looking at only one position, this results in a table with twenty columns (one for each amino acid) with a value that ranges from zero to one-hundred describing the percentage that particular amino acid represents in the entire sequence. However, using only one position is not entirely useful, so the concept of "window size" was introduced. In that, one examines not just one position in the alignment, but multiple *contiguous* positions up to the window size. This would result in a table with 20*n columns, where n represents the window size. However, this is not, in itself, adequate as amino acid sequences are far larger than the typical window size (which has been, in this research, nine). Thus, one can introduce a "sliding window" technique [12]. In this process, each frequency profile of size n (the window size) of a protein sequence is captured by the sliding window, which is also of size n. When it starts at position 0 in the multiple sequence alignment, it captures positions 0 to n and adds it to the frequency profile table as a row. The window then "slides" over to position 1 and captures positions 1 to n + 1 and adds it as a second row to the profile table. This repeats until there are no more positions in the protein. The end result is a frequency profile table with 20*n columns and p-n rows, where p is the size of the protein's primary sequence.

To generate this frequency profile, HSSP is put to use, a database that uses multiple sequence alignment to produce the frequency profile for proteins found in the PDB [11]. Using PISCES, DSSP, and HSSP to generate tertiary structures, secondary structures, and frequency profiles, respectively, one can generate the data set required for the FGK algorithm.

Granted this general idea of the dataset and the sources thereof, it is important to understand the explicit form this data takes when in use by the FGK algorithm. It has been noted that the FGK algorithm functions at a *local* level, meaning that each data member in the set consists of a protein segment, not the entire protein. This is where the idea of the aforementioned "sliding window" comes into play. The FGK algorithm adopts a window size of nine, such that each snapshot of the primary sequence generated by PISCES exhibits nine successive positions of that protein's primary sequence. This is done for each position in the protein, generating a new sequence segment at each position. The HSSP is used to, essentially, convert these segments into the useable frequency profile, generating a twenty (for each amino acid) by nine (for the window size) row for each segment generated by the sliding window. The secondary structure descriptors, written as 'H,' 'E,' and 'C,' for each of the nine positions are generated by DSSP appended to the end of each generated row, such that the secondary structure information corresponds to the nine positions described by the frequency profile. Finally, the tertiary structure information, generated by PISCES, is added for each row, describing the three-dimensional shape of each local sequence generated by the sliding window in terms of 36 distances from the center of the protein, defined in terms of mutual distances between each component amino acid. Further information is also added to identify the sequence segments, such as the protein name and the sequence number generated from that protein. One can see an example of what the data might look like in the following figure:



Protein Name	Protein Number	Pos 1-Amino Acid 1	Pos 1-Amino Acid 2	Pos 1-Amino Acid 2	Pos 1-Amino Acid 3	[…]	Pos 1-Amino Acid 20	Pos 2-Amino Acid 1	[…]	Pos 9-Amino Acid 20	Pos 1-Sec. Struct.	Pos 2-Sec. Struct.	[…]	Pos 9-Sec. Struct.	Tert. Struct. 1	Tert. Struct. 2	[…]
1W2W	1	0	5	3	0	[…]	10	0	[…]	20	H	E	[…]	C	1.304	3.43	[…]
1W2W	2	1	4	4	1	[…]	13	1	[…]	15	E	E	[…]	H	1.102	5.432	[…]
1W2W	3	1	9	3	4	[…]	8	2	[…]	12	E	H	[…]	E	1.934	4.43	[…]
1W2W	4	5	2	4	10	[…]	0	5	[…]	5	H	C	[…]	H	2.301	5.121	[…]
1W2W	5	6	0	2	11	[…]	0	3	[…]	0	C	C	[…]	H	1.123	3.123	[…]
1W2W	6	3	0	0	5	[…]	1	1	[…]	1	C	E	[…]	H	1.001	5.432	[…]
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Fig. 3: FGK Algorithm Data Structure

In this work, the dataset generated is based on 2,710 proteins obtained from PISCES, with the constraint that no proteins in the data share more than a 25% sequence identity. This is both for testing purposes, explained in the following sections, as well as for fulfilling the driving concept of the FGK algorithm, which relies on finding motifs that transcend protein family boundaries. If the sequence identity is too high amongst the data members, this implies that they all hail from the same protein family. If this is the case, that implies the motifs extracted will be representative of that particular family only (which is an approach used by purely homology-modeling algorithms). However, by enforcing this constraint, the FGK algorithm ensures the motifs it extracts transcend protein family boundaries, which expands coverage and accuracy of predictions. Granted this, each of the 2,710 proteins is run through the sliding window process, again with a window size of nine, generating more than 560,000 segments, where each segment is described in terms shown in Figure 3. This constitutes the training set, which is used to learn, or "train," the FGK algorithm as well as the full FGK-DF model. A similarly defined set, composed of the 2419 protein files excluded by the PISCES culling process, is used to test the FGK-DF model, such that its accuracy and coverage can be measured and defined based on the predictions made by the FGK-DF model. In the following sections, these two data sets are referred as the *training* and *testing* data sets, respectively.

2.2 Granulating and Clustering the Data: Fuzzy Greedy K-means

As previously mentioned, the FGK algorithm involves two major components: breaking the data into rough information granules and building finer clusters from those granules. This section discusses how the FGK algorithm makes use of the Fuzzy C-Means (FCM) and Fuzzy Greedy K-Means (FGK) to accomplish each of these respective tasks in the context of protein sequential motif extraction.

Granule computing, in a sentence, proposes one break a larger set of data into subsets, noted as "information granules," to allow for parallel execution [39]. This is required in the case of the FGK algorithm because of the rather large dataset (over 560,000 segments) described in the previous section. To generate these information granules, the FCM algorithm [19] is used. FCM works much like the popular clustering algorithm K-Means, only membership to each cluster is determined in a fuzzy manner rather than a static manner. FCM uses two primary equations: an equation (equation 1 below) for determining a degree of belonging to the cluster, and an averaging mechanism (equation 2) for determining the centroid of the cluster. These two equations, respectively, take the following forms:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (1)$$

$$C_j = \frac{\sum_{i=1}^{N} u_{ij*}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \quad (2)$$

Eq. 1 and 2: Fuzzy Degree of Belonging and Centroid Calculation

In these equations, U_{ij} is the degree of belonging of the data member x_i in the cluster j. c_j describes the centroid of the cluster j. C describes the number of clusters, N describes the number of data members, and m is the "fuzzification factor" [23], which determines the weight of the fuzzy logic as it takes place in the calculations. The "||…||" simply describes the distance formula that is used to determine the similarity/dissimilarity of a data member to a given centroid. The first formula, as mentioned, produces the degree of belonging for each data member for each cluster. The "degree of belonging" ranges from zero to one, with zero describing no belonging and one describing full belonging. In the same manner, the first equation generates a value ranging from zero to one, describing the degree of belonging for each data member in each cluster. This is used, in turn, for the second equation, which (as stated) determines the centroids for each cluster. As each data member has a variable belonging to the cluster, one can't simply add up all distances of data members in the cluster and then average it. Instead, one must account for the degree of belonging, and weight each distance appropriately. That is, for a data member with a high degree of belonging, more distance is contributed when determining the centroid. Those data members with a low degree of belonging contribute less distance. Just like with K-Means clustering, FCM begins with randomly selected points, determines membership to each of those points, calculates centroids, determines membership, recalculated centroids, and so on until the centroids no longer move or the change falls below a given threshold.

Thus, the FGK-DF uses FCM to produce distinct information granules, setting the fuzzification factor to 1.05 (m can range from 1.00 to infinity) and the number of information granules to ten, based on the results generated in previous work [5]. The distance formula (equation 3) utilized is based on that used by Han and Baker [12], described as the "city block metric." This formula is shown below:

$$Distance = \sum_{i=1}^{L} \sum_{j=1}^{N} |F_K(i,j) - F_c(i,j)|$$

Eq. 3: City Block Metric

This formula, when applied to the FGK data, basically states that distance is equal to the summation of the difference between the frequency value in data member F_k and F_c for each of the twenty amino acids (N) in each of the nine positions described by the window size (L). This distance formula is used in place of the "||…||" in the FCM formulas described above, as well as the distance formula for the FGK process described in the following passages. This formula generates the "distance," or difference that is exhibited between two frequency profiles for two given segments in the training data set. Using this formula, the distance threshold (i.e. the allowed difference between two protein sequence segments) is set to 13%, which roughly translates to the omission of 15% of the outlying data that could not be clustered to any of the centroids. This described setup results in ten information granules, whose main purpose is to reduce the running time (by producing granules that can be clustered in parallel) as well as to improve the quality of the data (by removing 15% of the data as outliers). Despite this, the information granules are still far too course for motif extraction, requiring a finer grained clustering algorithm to generate usable protein motifs.

In related work, Zhong et al. proposed an "improved" K-Means algorithm to resolve the initialization problem of traditional K-Means for protein sequence motif extraction [43]. The algorithm had two main steps for generating initial centroids: generate centroids by running traditional K-Means for a fixed number of iterations, then determine if those centroids can be added as viable initial centroids based on secondary structural similarity and then their distance to other initial centroids. This was run until the number of viable initial centroids was equal to the number of required clusters, *k,* which then traditional K-Means was run with those initial centroids. The distance measure was based on the "city block metric" formula described in the preceding passage, while the secondary structure similarity was based on the following equation:

$$\frac{\sum_{i=1}^{ws} \max(p_i H, p_i E, p_i C)}{ws}$$

Eq. 4: Secondary Structure Similarity Measure

Equation 4 uses the secondary structure of all items in a given cluster to determine the secondary structure similarity shared between all protein segments in that cluster. In the formula, P_iH describes the frequency of helices in the protein segments in the cluster at position *i* for each of the nine positions (*ws*). P_iE and P_iC describe the frequency of sheets and coils, respectively, in the same manner. Max() returns the maximum frequency of the three measures, as one would expect. Finally, this is all divided by the window size, ws, which is nine in this case. If one considers the simple example of three protein segments with a window size of three, the secondary structure similarity of three such proteins with 'HHH,' HEH,' 'HHH' as their secondary structures would result in roughly an 88% secondary structural similarity, using that formula.

Zhong et al.'s Greedy K-Means algorithm served as the basis for the FGK algorithm. In fact, the FGK is simply a "greedier" version (ignoring the addition of the FCM preprocessing step) of the Greedy K-Means algorithm [5]. More explicitly, the Greedy K-means aspect of the FGK follows the algorithm proposed by Zhong et al., except a dynamic threshold for required secondary structure similarity is implemented. The number of iterations was also fixed to five runs of traditional K-Means, where each respective run had a secondary structure similarity cutoff of greater than 80%, 75%, 70%, 65%, and finally 60%. These values were based on the idea that a cluster of protein segments with a secondary structure similarity of greater than 70% can be considered "structurally identical" [10], but also that those between 70% and 60% can be considered "weakly structurally homologous" [43]. This approach was indeed greedier, resulting, depending on the centroid distance threshold, on either too many centroids or not enough centroids. For instance, if the required distance threshold was set to 250 units, the algorithm "could always obtain more centroids" [5] than *k*. If there was a dearth of centroids, traditional K-means was run with a distance threshold of 800 to choose the rest of the initial centroids. Finally, after the five runs generated the initial centroid list, just as in Zhong et al.'s Greedy K-Means, a run of traditional K-Means was performed using the generated initial centroid list to produce the protein clusters.

This new "greedier" K-Means algorithm, combined with the aforementioned FCM setup, would result in the complete FGK algorithm [5]. In order to determine the number of clusters, *k*, for each information granule, the following equation (equation 5) was used:

$$C_k = \frac{n_k}{\sum_{i=1}^{m} n_i} \times p$$

Eq. 5: Granule Size

In this formula, C_k refers to the amount of clusters assigned to a given information granule 'k,' where n_k is the number of data members contained within said granule. M, in the bottom summation, refers to the number of clusters defined for the FCM run (which would be equal to the number of granules, ten). P refers to the total number of clusters, which is set, in this work, to 799, based on the research and results performed by Zhong et al. [43]. In effect, this formula balances the number of clusters for each granule based on the number of each granule's members. That is to say that if a given granule has fewer members, then the number of clusters it produces will also be lower. For instance, in this work, Granule 7 only has 4,583 members, thus the number of clusters it generates is only five. Conversely, Granule 0 has 136,112 members, such that its number of clusters is 151.

Taken altogether, this step in the FGK-DF model begins the isolation process for the protein sequence motifs, as each cluster generated represents a shared pattern amongst its member sequences. That is, each of the 799 generated clusters represents a unique motif that is contained within each of the protein segments that composed the cluster. The FGK-DF uses these motifs, in the prediction process, to match primary sequences of unknown proteins to a possible tertiary structure, but the clusters are generally too course and unrefined to jump straight to this step. Instead, the data requires one more refinement process to build the model such that it can generate predictions, that process being the development of the decision forest as the next chapter will explore.

III. Fuzzy Greedy K-Means Decision Forest (FGK-DF) Algorithm

3.1 Decision Tree Induction Processes

Consider a simple example in which a given person is deciding whether or not to take a walk. Clearly, the person would first run through the situation at hand, subconsciously asking questions surrounding, perhaps, the weather, what their schedule looks like, whether they need the exercise, etc. These could be arranged into a hierarchy, where the more important factors in determining if the person is take the walk would be asked first, and less important questions asked later. For instance, one of the first questions that one could ask is whether or not it is raining. This could also lead to branching of questions. For instance, one could ask if the temperature outside is "hot." If it is, then they could ask if they have a water bottle. If the temperature is hot and they have a water bottle, then they might either ask further questions, or conclude, after determining they have a water bottle, that they should go for a walk. The following figure demonstrates a possible structure for making this type of decision:

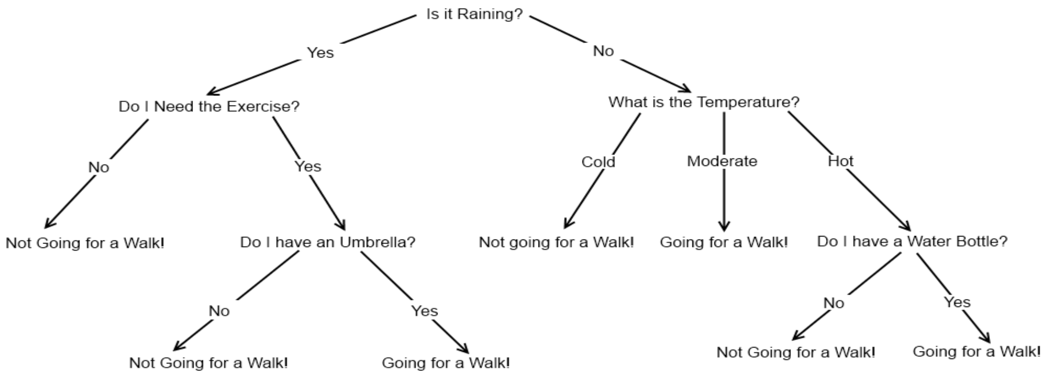


Fig. 4: Decision Structure for the Walk Scenario

Starting from the top of the figure, one can see the questions being raised, with each branch being a possible answer to that question. Each answer either leads to a conclusion ("Not Going for a Walk!") or to another question. Many of the questions above are binary, but the branching can include answers that fall within a range or set, such as the "What is the Temperature?" question, which can either be "cold," "moderate," or "hot." The ordering of the questions implies a relationship between the parent question (the one that precedes the current question) and the child question. For instance, "Do I have a Water Bottle?" clearly has a relation with a "hot" temperature. This relationship can be extended further up the graph, such as the "Do I have an Umbrella?" question, which most directly relates to the "Is it Raining?" question.

Of course, deciding on whether or not walking is a good idea on a given day has little to do with protein structure prediction. However, the underlying concept of this "decision tree" is very applicable, as the FGK-DF model employs decision trees to refine and make sense of the "rough" sequence motifs captured by the clustering step described in the previous chapter. Effectively, instead of telling one "yes, you should go for a walk," the FGK-DF model's decision trees will state whether or not a test protein from the testing set "belongs" to that cluster. If it does, the FGK-DF model can attempt to perform the prediction process on that protein. Each cluster developed by the FGK algorithm of the model will have one decision tree "trained" on it, resulting in a "decision forest" of 799 decision trees for each of the 799 clusters. Recalling that each cluster effectively implies an underlying, shared sequence motif among its members, each decision tree isn't so much deciding if a test protein belongs to the cluster as much as it is deciding if that test protein *shares that sequence motif*. This methodology is explained in full in the next section, but before one can truly delve into the prediction process, it is important to understand how one "trains," or, rather, inducts a decision tree, and how that process is used in the context of the FGK-DF model's training data. To understand this, one must understand the Itemized Dichotomizer 3 (ID3) decision tree algorithm.

The ID3 algorithm, presented in 1975 by J.R. Quinlan [22], builds decision trees based on the minimization of entropy (i.e. the randomness of the dataset) at each branch in the tree. Within the context of the algorithm, there are three main concepts: labels, attributes, and values. Ultimately, what the ID3 algorithm produces is a label, which acts as the "decision" made by the decision tree. The attributes are effectively the "questions" asked at each level. In practice, these generally do not take the form of questions, but rather represent a discrete entity with any number of possible values. Values, themselves, are the "answers" to the attributes, or, in more explicit terms, simply the value that the attribute takes at that level. In the aforementioned Walk Decision Scenario, the two possible labels are "Yes" and "No" to the question "Am I going for a walk?" The attributes are "Need Exercise," "Temperature," "Have Umbrella," and "Have Water Bottle" (notice these changed from questions to discrete statements). The values for each, respectively, are {Yes, No}, {Cold, Moderate, Hot}, {Yes, No} and {Yes, No}. To determine the hierarchy of these attributes (and thus the structure of the tree), the ID3 uses a concept known as "information gain," which denotes the effectiveness of a given attribute in reducing the entropy of a dataset at a given branch. This requires both an equation for determining gain and entropy, as well as the presence of a *training set* to build the tree. To determine entropy, the following equations is used:

$$Entropy(s) = -\left(\frac{S_y}{S_C}\right)\log_2\left(\frac{S_y}{S_c}\right) - \left(\frac{S_N}{S_C}\right)\log_2\left(\frac{S_N}{S_c}\right)$$

Eq. 6: Entropy

For the entropy equation, *S* denotes a sub-collection of data of size S_c, where S_Y denotes the count of all items that belong to a class (i.e. have a "Yes" label), and S_N denotes the count of all items that do not belong to a class (i.e. have a "No" label). Log_2 simply refers to performing the logarithmic function with a base of two. Consider the following example: S is a collection of data with 14 members, such that 9 of the members have a "Yes" label, and the other 5 have a "No" label. The entropy would be equal to the following:

$$\mathbf{Entropy(S) = -\ (9/14)log_2(9/14) - (5/14)log_2(5/14)}$$

$$\mathbf{Entropy(S) = 0.940}$$

In this case, the data collection is very entropic, as an entropy of zero implies a perfectly classified set, an entropy of one implies one that is perfectly random. One can see if the counts for the "No" label members and the "Yes" label members were changed to 7 and 7 each, the entropy would be equal to one. Given that the goal of the ID3 algorithm is to reduce the entropy (and thus increase the information gain) at each branch, another equation is needed to determine "gain." This simple equation is shown below:

$$Gain(S,A) = Entropy(s) - \left(\frac{|s_v|}{s_c}\right) Entropy(s_v)$$

Eq. 7: Information Gain

S denotes the total collection of size S_c, *A* denotes an attribute in that data, S_V is the subset of *S* for which attribute *A* has a value *v*, and $|S_V|$ is simply the count of the items in subset S_V. As one can see, the gain equation simply takes the overall entropy of the collection, and determines how much a given attribute reduces that entropy by calculating the weighted entropy of that attribute. Using these two equations, and a training data set (which defines *S*), the ID3 algorithm can determine which attribute offers the greatest gain, and place it at the top of the decision tree hierarchy. In the “Walk Decision Scenario,” the question with the highest gain (and thus the question that most reduces the entropy of the data set) is the one that asks “Is it Raining?” That means that if there were a training data set (in this case, the experience of the person asking the subconscious question), most data members that had “yes” for the raining question would consistently correspond to “No” label for the walking decision. Conversely, most data members that had a “no” for the raining question would consistently correspond to the “Yes” label. Once the ID3 algorithm decides on the “root” of the tree, it begins recursively calling itself on the subsets of the data that fall to each branch. For example, once the “Is it Raining” question is set as the root of the decision tree, all data in the training set that has the “yes” value for the “raining” attribute go to one branch, and all data in the training set that has a “no” value goes to another branch. At this point, a new tree is to be built on these subsets that relies on another attribute (can't use the same attribute twice in a given branch) to reduce entropy. This continues until either one runs out of attributes, or until the subset of data is perfectly classified. For example, in Figure 4, all data members in the training data that have "yes" for "raining" and "no" for "need exercise" have a "no" label, and thus produce a "no" decision. If the data isn't perfectly classified, the decision is made on majority vote (i.e. if there are more "yes" labels than "no" labels, the decision is "yes").

Taken altogether, the ID3 algorithm is a simple but intuitive algorithm for producing decision trees, but now the question is how does it operate in context of the FGK-DF model? Just as in the above explanation, the decision trees in the FGK-DF require labels, attributes, and values, as well as training data to build the models. The training data is built from the clusters built on the training set described in the previous section. As was pointed out, each decision tree is trained upon the data in each cluster, such that each decision tree corresponds to exactly one cluster. The labels ultimately state whether or not a given protein, which can be run through the decision tree, belongs to the cluster that the decision tree is trained upon. To decide whether or not a data member in the training set produces a "yes" label or a "no" label, the secondary structure of that data member is compared to the average secondary structure of the cluster to produce an individual secondary structure similarity for that data member. If it is greater than a certain threshold, noted as a "label pivot" in this work, it is denoted with a "yes" label. If it is less than the label pivot, it is given a "no" label. The attributes used for the tree are based on the 180 possible positions in the frequency profile, such that the values are ranges of the possible values each position in the frequency profile. An example decision tree produced by the FGK-DF model might look like the following:

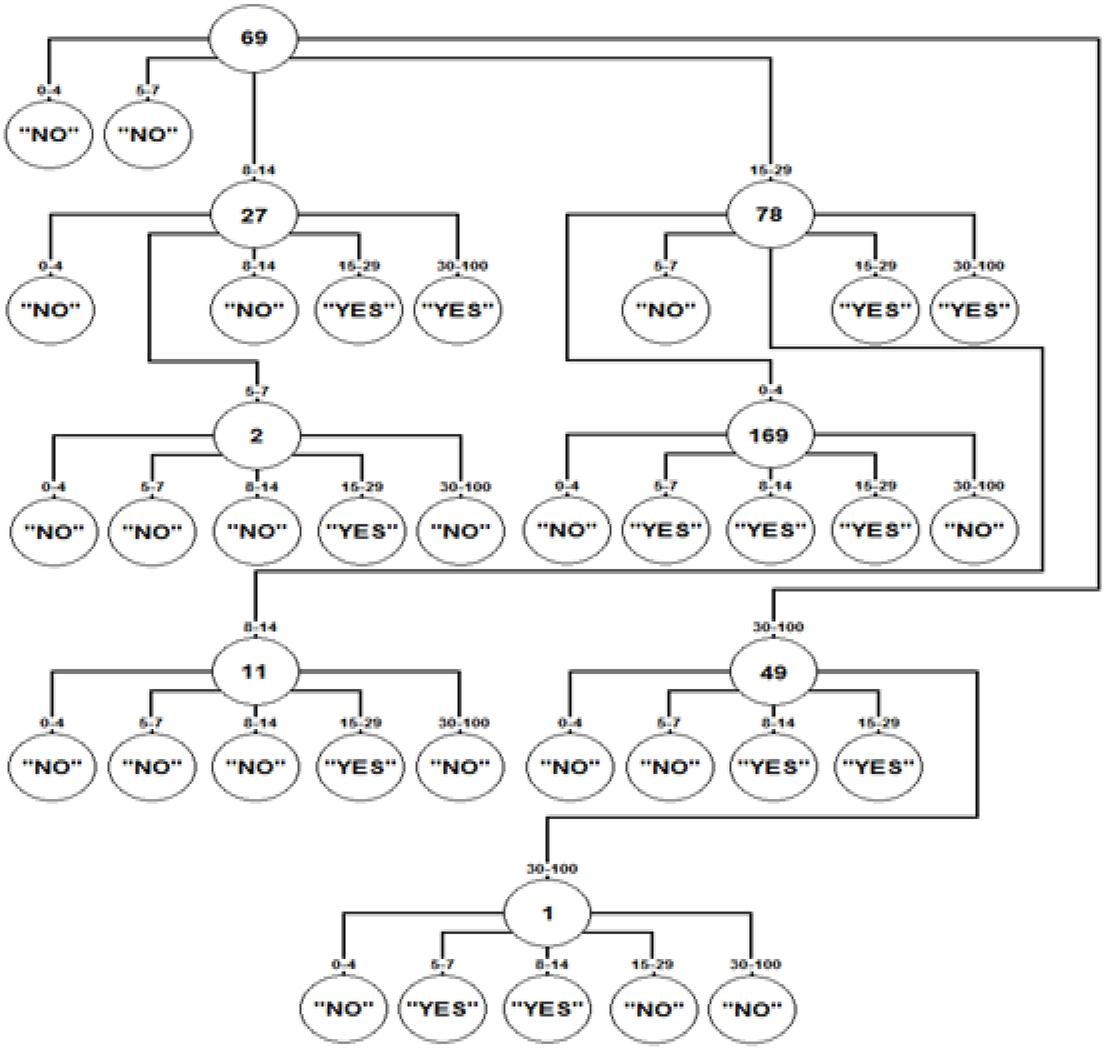


Fig. 5: Example FGK-DF Model Decision Tree

In the above figure, one can see bubbles with either a number or a decision in them. Each bubble with a number in it has a branch with a range of numbers at the very end. As was described in the previous passage, the bubbles with numbers in them refer to the positions (1-180) in the frequency profile of the given protein segment. The branches, and the ranges of numbers on each, represent the possible values each of those positions can take. One might note that these values range from 0 to 100, as one would expect. The “yes” and “no” bubbles denote the decision the tree makes for each possible branch. Thus, if an unknown protein segment was run through this tree, it would first start at the root, just like in the “Walk Decision Scenario.” Thus, one would first look in the unknown protein segment, into its frequency profile, and look at the value in position 69. Assume that the value at that position is 9, such that the decision tree would then look at position 27 in the unknown protein’s frequency profile. If the value was 6, the decision tree would look at position 2. Finally, if the value at position 2 was 16, the decision tree would decide that, “yes,” the unknown protein belongs to the cluster this decision tree was trained on, and that “yes,” this unknown protein shares the sequence motif that is present in the members of that cluster.

Granted all of this, the FGK-DF uses the protein segments in each cluster to train a decision tree using the ID3 decision tree algorithm, where the labels are determined by secondary structure similarity, and the branches and attributes determined by the values and positions in the frequency profile of those segments. The end label describes whether or not a protein segment run through the decision tree “belongs” to that tree, but more implicitly, whether or not that protein segment shares the sequence motif isolated in the cluster the decision tree is trained on. This produces a decision forest and, ultimately, the final form of the FGK-DF model. But, of course, this is not where the model ends. After all, the decision trees do not end in a possible tertiary structure, but just a simple statement of whether or not the inputted protein shares the sequence motif. The question then becomes how does one use the FGK-DF model to perform predictions? Put a different way, how does the FGK-DF model explore the decision forest to find the best prediction? The next section will not only explain this prediction process, but will also explain how this process represents and reflects the underlying concepts that support the FGK-DF model.

3.2 Producing Local Tertiary Structure Predictions using the FGK-DF Model

Granted the previous sections, it should be clear that the FGK-DF relies on two major concepts in its logic: there are sequential motifs that transcend protein family boundaries [12], and primary sequence determines the tertiary structure of a protein [15]. Combining the two concepts, one can see that if the conserved primary sequence motifs can be extracted, then those can be related to tertiary structures to effectively build up a database of "tertiary structure motifs." Since these "motifs" are conserved across protein family boundaries, this means that if an unknown protein was compared against this database, one could extract the sequence motifs that this unknown protein shared with those in the database, and determine is tertiary structure based on the corresponding structures to each motif. In other words, since primary structure and tertiary structure are linked, one can determine the tertiary structure of a protein by determining the presence of known sequential motifs. The FGK-DF model does just this, logically, when producing the predictions for the local tertiary structure of an unknown protein. This logic is assured by the setup described in the previous sections, as the clusters extract the motifs, and the decision trees refine it to a basic decision of whether or not an input protein also shares the motif. In the data section, it was pointed out that the data sets also contain tertiary data. For the training data set, this allows the FGK-DF model to relate the primary sequence to the tertiary structure (and, of course, for the testing data set, it is for prediction validation). Explicitly, the step taken to produce a local tertiary structure prediction for a protein segment from the testing data set is slightly more convoluted, falling through the twists and turns of exploring the vast decision forest for, effectively, the "best tree."

The process of performing a prediction first requires an input protein segment's frequency profile. Previously, this has been referred to as an "unknown protein," or a member of the "testing data set." This implies that, in the context of testing the FGK-DF model, this protein's secondary and tertiary structure is assumed to be unknown until after the prediction is made. Once the unknown/test protein is input, its frequency profile is scanned against the *representative frequency profile* of each decision tree. The representative frequency profile of each decision tree is determined by averaging the frequency profiles of all of the protein segments that compose the tree (rather than, for instance, all the protein segments that compose a branch in the tree). The distance formula is, of course, the city block metric formula used in the clustering step. All 799 decision trees' representative frequency profiles are compared against the unknown protein segment's frequency profile, where the "best tree" is determined to be the one with the lowest distance. Once this tree is decided upon, the unknown protein's frequency profile is run through the decision tree in a process similar to what is described in figure 5. If the protein segment is found to share the sequence motif represented by the decision tree (i.e. if the decision tree results in a “yes” decision for the unknown protein), then the tertiary structure of that protein segment is said to be equal to the *representative tertiary structure* of the branch the protein segment followed. Notice that the representative tertiary structure is not determined by averaging the tertiary structures of the entire protein segment set that composes the decision tree, but rather by averaging the tertiary structure of the subset of the protein segments that are represented by one branch of the decision tree. If the protein segment is not found to share the sequence motif, the next best tree is found, and so on until either the distance is greater than a given threshold, or a match is made. The accuracy of the prediction can be generated by comparing the predicted tertiary structure, and the “ground truth” tertiary structure extracted from the PDB.

Thus, taken altogether, to form predictions, the FGK-DF model explores the decision forest for the decision tree that is most similar, in terms of representative frequency profile, to the unknown protein. If, once run through the decision tree, the unknown protein is said to be a part of the tree, its tertiary structure is said to be equal to the representative tertiary structure of the branch of the decision tree the unknown protein corresponds to. To form the decision forest, the FGK-DF trains decision trees on each of the 799 clusters, such that the labels determine if a protein shares a certain sequence motif, the attributes refer to positions in the frequency profile, and values refer to the values the frequency profile positions can take. To build the clusters the decision trees are trained on, the FGK-DF uses the Fuzzy Greedy K-means algorithm, which is based on using the Fuzzy C-Means algorithm to break the data set into information granules, and using the “greedier” Greedy K-Means algorithm to form clusters on those granules based on the distances of frequency profiles, and the secondary structure similarity of the clusters. Finally, the data to form those clusters is extracted from DSSP, HSSP, and PISCES, which is based on the data that is found, centrally, in the PDB. This entire process is reiterated in Figure 6 below.

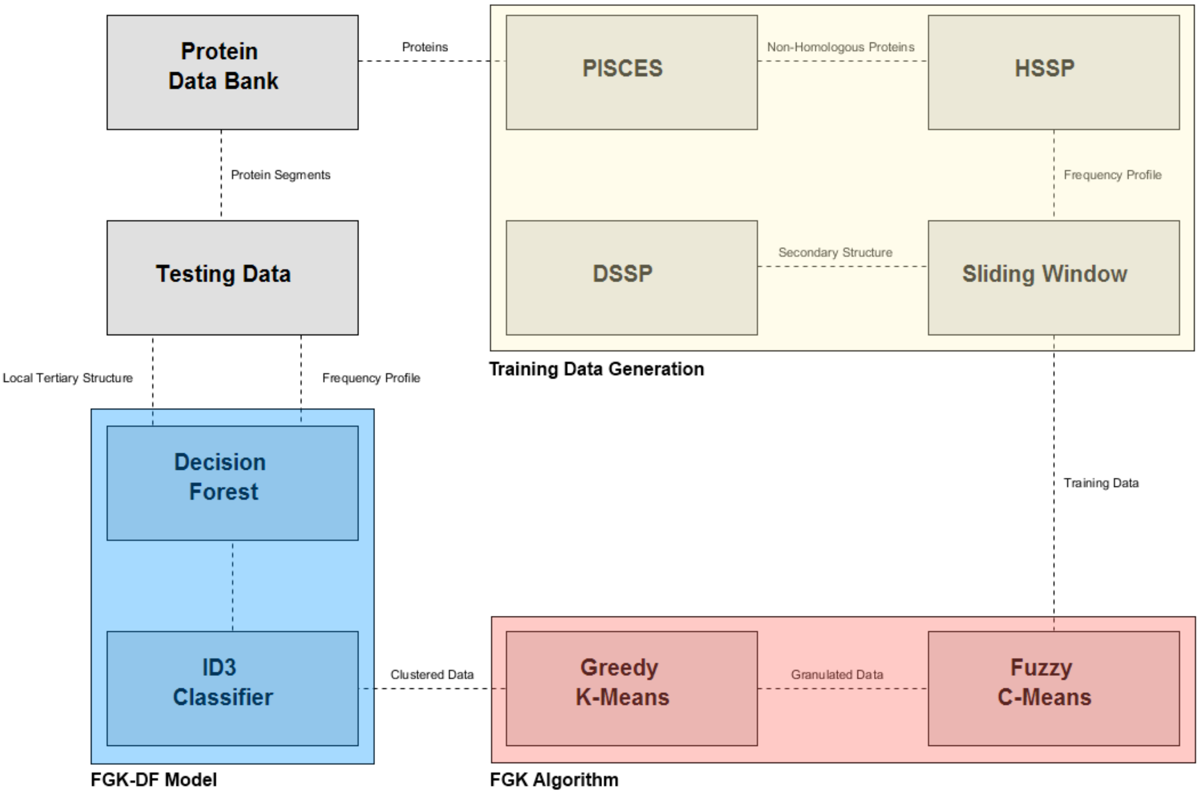


Fig. 6: The Fuzzy Greedy K-Means Decision Forest Model

Granted the outlined process, the following section will present the setup, execution, and results of an experiment to justify the use of the FGK-DF model, both in terms of producing protein tertiary structure predictions and in the context of improving the base work provided by the FGK algorithm described in the previous chapter.

3.3 Experimental Setup for Tertiary Structure Prediction using the FGK-DF

In order for proper execution of the FGK-DF model on a data set, several parameters have to be carefully set based on analysis of the data and expert opinion. These parameters include the fuzzification factor for Fuzzy C-Means (set to 1.05 [5]); the window size for the frequency profile (set to 9 [43]); the secondary structure similarity thresholds for Greedy K-means (set to 100-80%, 79-75%, 74-70%, 69-65%, and 64-60%); and the number of information granules and clusters (set to 10 and 799, respectively). While this list is not exhaustive, it does outline the primary parameters for the Fuzzy Greedy K-means portion of the FGK-DF model. As these parameters have already been outlined and determined in previous research, they are not analyzed or justified any further in this work [5]. Rather, the parameters of interest in this section are those needed by the decision forest aspect of the FGK-DF model, those parameters being, primarily, entropy threshold, label pivot, and the attribute range list. The significance of each of these parameters and how they affect the model and results are outline in the following paragraphs.

As stated in the previous sections, entropy described the randomness of a particular dataset. The examples laid out suggested that the ID3 algorithm made decisions once the entropy of a subset of the data had been reduced to 0 (i.e. when the data had been perfectly classified). One can also force the ID3 algorithm to make decisions once the entropy had been reduced below a certain threshold, the so-called entropy threshold. This decision would be made based on majority voting according to the labels of the training data subset. For example, if the ID3 algorithm were making a decision on a particular branch in a given decision tree with a entropy threshold of 0.30, then if the entropy of the data set characterizing that branch were below 0.30, a decision would be made according to whether or not there were more instances with "yes" labels or "no" labels. If the entropy of the branch was above 0.30, the branch would continue to be grown with more branches to reduce the entropy until it fell below the threshold. This parameter essentially controls the "depth," so to speak, of the tree. A strict entropy threshold will cause a tree to be exceedingly deep, as it would force branches to be perfectly classified before a decision could be made. This may increase accuracy, as the tree would be highly specialized. However, as one might expect, this would vastly reduce coverage due to this specialization. This is a problem noted as "over learning," in which a model built by a classifier, such as the ID3 algorithm, has become specialized for making decisions on data that heavily reflects the training data that built the model. Granted that, it is easy to see that a higher entropy threshold removes the issue of over learning, granting much greater coverage at the loss of accuracy. In this work, the entropy threshold is set to 0.75, as previous research has shown it to offer the greatest tradeoff between potentially high quality results with reduced data complexity and reduced overlearning [6].

The other half of the decision making process lay with the label pivot. The label pivot, introduced and briefly explained in the previous sections, determines what is considered a "yes" or "no" label to the question of whether or not a protein segment is a member of a given cluster. The answer to this question is based on the secondary structure similarity of a protein segment with other segments in its cluster. In effect, each protein segment's secondary structure is compared against the average, or representative, secondary structure of the cluster. A score is generated based on the number of positions in which the protein segment has the same secondary structure (H, E, or C) as the representative structure. Thus, in context of the FGK-DF's setup, a maximum score would be nine (as the window size is nine), meaning that the protein segment in question has the exact structure as the representative secondary structure of the cluster. A score of zero suggests that the protein segment shares no structure similarity with the representative secondary structure of the cluster. At this point, these raw scores could be used as "decisions" for the decision trees, as it is completely possible to generate trees with non-binary decisions. However, to reduce overall complexity, both logically and computationally, the label pivot is introduced to reduce the ten (counting the score of zero) possible scores to two. Effectively, the label pivot is one number, in the range of zero to nine, such that all protein segments with a score of less than that pivot are given a "no" label. Those equal to or greater than the pivot are given a "yes" label. The adjustment of this parameter fundamentally changes the structure of the tree, as it determines what each instance is labeled as, and thus it can have extreme effects on the depth, width, coverage, and accuracy of any given tree. For instances, a label pivot of one would force almost all instances to be classified as "yes" proteins. This would allow the trees to quickly make decisions as the data would be nearly perfectly classified from the start. Of course, the tradeoff would be highly inaccurate trees. A label pivot of eight would, conversely, lead to highly accurate trees, but just like a low entropy threshold, the coverage would suffer greatly. In this work, the label pivot is said to be seven, based on prior experimental results [6].

The last parameter, the attribute range set is perhaps the most subtle in terms of its overall effects on the tree. In short, this parameter directly affects the width of the tree (whereas entropy affects the depth of the tree). In Figure 5, five ranges are used (0-4, 5-7, 8-14, 15-29, 30-100), but another possibility could be anywhere from one range (0-100) to one hundred ranges (0-0, 1-1, 2-2, etc.). The former extreme would result in a highly simple tree, but one that was extremely inaccurate. The latter extreme would result in a much more accurate tree, but one that would be impossibly complex and possibly over learned. In this work the range set used is a static range set including 0-4, 5-7, 8-14, 15-29, 30-100, determined by expert opinion [6].

Thus, given the above three parameters, an experiment is proposed and carried out that examines the justification of using the FGK-DF model to predict local tertiary protein structure by comparing results from predictions made with clusters generated by the Fuzzy Greedy K-means algorithm alone against results from the full FGK-DF model. In this experiment, the training and testing data includes a set of roughly 2,700 proteins from which nearly half a million protein segments are generated using the data preprocessing steps introduced in the previous chapter. Each data set was carefully constructed such that no protein shared more than 25% sequence identity with other any other protein in the dataset (to ensure that the FGK-DF model did not train for particular protein families). The aforementioned experiment, as the following sections will explore, generates results in terms of coverage (i.e. how many proteins segment structures were predicted using the model) and prediction accuracy (i.e. how well did the protein segment's real tertiary structure line up with the predicted tertiary structure).

3.4 Justification for Using the FGK-DF for Local Tertiary Structure Prediction

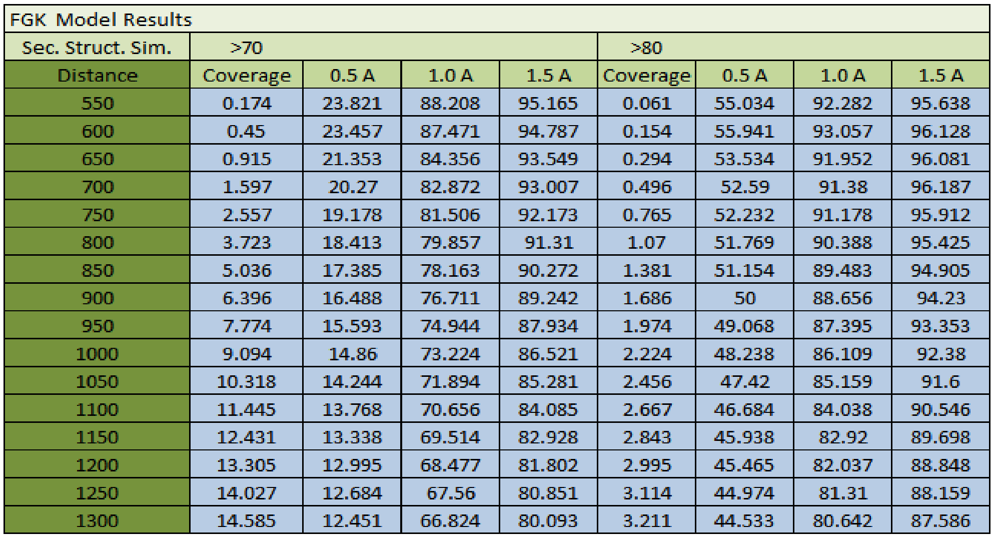
It should be noted, first and foremost, that one of the unfortunate aspects of the FGK-DF's novel approach to protein structure prediction is that there is no foreign or competing algorithm can be compared against it, directly. This is due to the fact that the FGK-DF model produces *local* tertiary structure predictions. That is why, throughout this work, the data in the data sets have not been regarded as proteins, but rather as protein segments. While this gives the FGK-DF model distinct advantages that have been outlined in other chapters, it does justify the use of the algorithm through direct comparisons with other leading algorithms all but impossible, as almost all other protein structure prediction algorithms perform global tertiary structure prediction (that is, the entire protein's structure, not just the segments' structures, are predicted). Fortunately, however, justifying the use of the FGK-DF model can still be accomplished in two different ways, the first of which is to use a prediction accuracy metric that lies on a 0-100% scale, such that it is clear, even lacking a direct comparison, when a model is effective. A score known as root-mean-square deviation, otherwise known as RMSD [40], is utilized to generate such a metric. The equation for RMSD is shown below:

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n}\left(x_{1,i} - x_{2,i}\right)^2}{n}}$$

Eq. 8: Root-mean-square Deviation

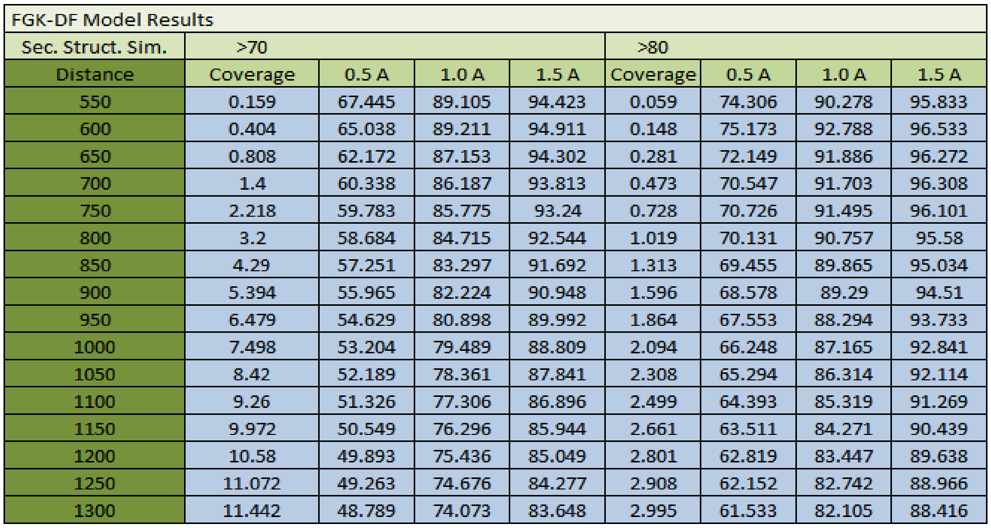
For the protein structure prediction, n would be 36, for each of the tertiary structure mutual distance calculations (as explained in the previous chapter). X_{1i} would refer to the ith position in the tertiary structure for protein x_1. Just the same, x_{2i} would refer to the ith position in the tertiary structure for protein x_2. These two "proteins," respectively, would refer to the predicted structure and the ground truth (i.e. the known structure extracted from the PDB). This equation would square the differences between each of the 36 positions for the predicted structure and the ground truth, heavily penalizing predictions that are highly incorrect. These differences are summed together and divided by n (or 36), and the overall result is scaled down by performing a square root, resulting in the final, weighted distance measured in angstroms which is a unit of length equal to one ten billionth of a meter (denoted by Å). Conventionally, acceptable predictions are those that are equal to or are below 1.5 Å, with good predictions are 1.0 Å [42]. This work introduces another tier at 0.5 Å, denoting exceptional prediction accuracy. This means that all 36 distances, once weighted and summed by RMSD, results in a distance less than or equal to 1.5 Å, 1.0 Å, or 0.5 Å (exclusively).

Granted this metric, one can then compare the results of running the full FGK-DF model against a model that uses only Fuzzy Greedy K-means to build its model. In other words, this experiment constitutes comparing the FGK-DF model, which builds decision trees on clusters in order to make tertiary structure predictions, against the FGK model, which uses only the clusters to generate tertiary structure predictions. This, again, is due to the extremely limited number of algorithms the FGK-DF model can be compared directly against. Despite this, the results of this comparison, using the static branching attribute range set and other parameters examined in the prior section, can be seen below in the following tables:



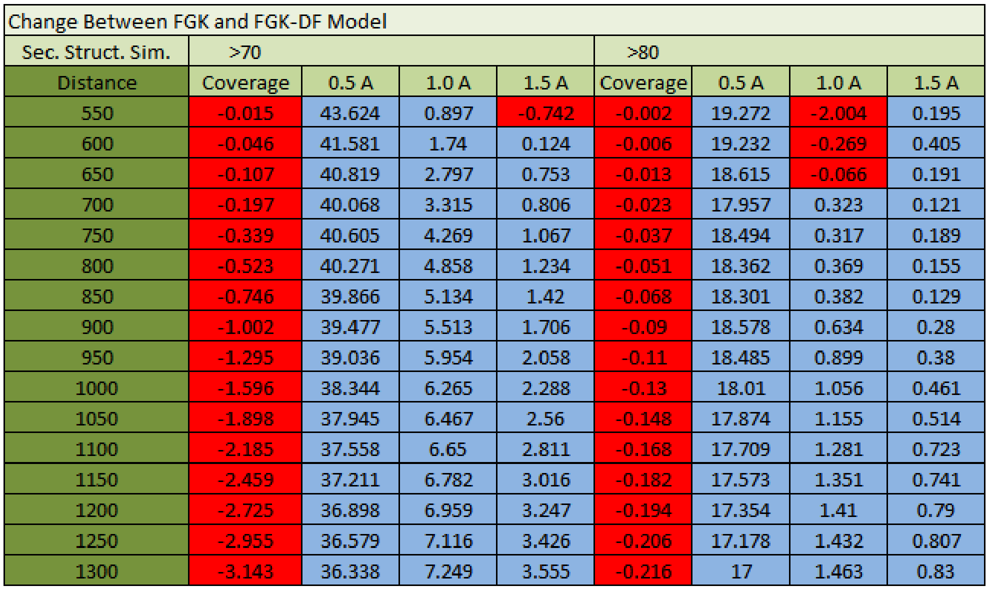
FGK Model Results								
Sec. Struct. Sim.	>70				>80			
Distance	Coverage	0.5 A	1.0 A	1.5 A	Coverage	0.5 A	1.0 A	1.5 A
550	0.174	23.821	88.208	95.165	0.061	55.034	92.282	95.638
600	0.45	23.457	87.471	94.787	0.154	55.941	93.057	96.128
650	0.915	21.353	84.356	93.549	0.294	53.534	91.952	96.081
700	1.597	20.27	82.872	93.007	0.496	52.59	91.38	96.187
750	2.557	19.178	81.506	92.173	0.765	52.232	91.178	95.912
800	3.723	18.413	79.857	91.31	1.07	51.769	90.388	95.425
850	5.036	17.385	78.163	90.272	1.381	51.154	89.483	94.905
900	6.396	16.488	76.711	89.242	1.686	50	88.656	94.23
950	7.774	15.593	74.944	87.934	1.974	49.068	87.395	93.353
1000	9.094	14.86	73.224	86.521	2.224	48.238	86.109	92.38
1050	10.318	14.244	71.894	85.281	2.456	47.42	85.159	91.6
1100	11.445	13.768	70.656	84.085	2.667	46.684	84.038	90.546
1150	12.431	13.338	69.514	82.928	2.843	45.938	82.92	89.698
1200	13.305	12.995	68.477	81.802	2.995	45.465	82.037	88.848
1250	14.027	12.684	67.56	80.851	3.114	44.974	81.31	88.159
1300	14.585	12.451	66.824	80.093	3.211	44.533	80.642	87.586

Table 1: FGK Model Results (Cluster Only)



FGK-DF Model Results								
Sec. Struct. Sim.	>70				>80			
Distance	Coverage	0.5 A	1.0 A	1.5 A	Coverage	0.5 A	1.0 A	1.5 A
550	0.159	67.445	89.105	94.423	0.059	74.306	90.278	95.833
600	0.404	65.038	89.211	94.911	0.148	75.173	92.788	96.533
650	0.808	62.172	87.153	94.302	0.281	72.149	91.886	96.272
700	1.4	60.338	86.187	93.813	0.473	70.547	91.703	96.308
750	2.218	59.783	85.775	93.24	0.728	70.726	91.495	96.101
800	3.2	58.684	84.715	92.544	1.019	70.131	90.757	95.58
850	4.29	57.251	83.297	91.692	1.313	69.455	89.865	95.034
900	5.394	55.965	82.224	90.948	1.596	68.578	89.29	94.51
950	6.479	54.629	80.898	89.992	1.864	67.553	88.294	93.733
1000	7.498	53.204	79.489	88.809	2.094	66.248	87.165	92.841
1050	8.42	52.189	78.361	87.841	2.308	65.294	86.314	92.114
1100	9.26	51.326	77.306	86.896	2.499	64.393	85.319	91.269
1150	9.972	50.549	76.296	85.944	2.661	63.511	84.271	90.439
1200	10.58	49.893	75.436	85.049	2.801	62.819	83.447	89.638
1250	11.072	49.263	74.676	84.277	2.908	62.152	82.742	88.966
1300	11.442	48.789	74.073	83.648	2.995	61.533	82.105	88.416

Table 2: FGK-DF Model Results (Cluster and Decision Trees)



Change Between FGK and FGK-DF Model								
Sec. Struct. Sim.	>70				>80			
Distance	Coverage	0.5 A	1.0 A	1.5 A	Coverage	0.5 A	1.0 A	1.5 A
550	-0.015	43.624	0.897	-0.742	-0.002	19.272	-2.004	0.195
600	-0.046	41.581	1.74	0.124	-0.006	19.232	-0.269	0.405
650	-0.107	40.819	2.797	0.753	-0.013	18.615	-0.066	0.191
700	-0.197	40.068	3.315	0.806	-0.023	17.957	0.323	0.121
750	-0.339	40.605	4.269	1.067	-0.037	18.494	0.317	0.189
800	-0.523	40.271	4.858	1.234	-0.051	18.362	0.369	0.155
850	-0.746	39.866	5.134	1.42	-0.068	18.301	0.382	0.129
900	-1.002	39.477	5.513	1.706	-0.09	18.578	0.634	0.28
950	-1.295	39.036	5.954	2.058	-0.11	18.485	0.899	0.38
1000	-1.596	38.344	6.265	2.288	-0.13	18.01	1.056	0.461
1050	-1.898	37.945	6.467	2.56	-0.148	17.874	1.155	0.514
1100	-2.185	37.558	6.65	2.811	-0.168	17.709	1.281	0.723
1150	-2.459	37.211	6.782	3.016	-0.182	17.573	1.351	0.741
1200	-2.725	36.898	6.959	3.247	-0.194	17.354	1.41	0.79
1250	-2.955	36.579	7.116	3.426	-0.206	17.178	1.432	0.807
1300	-3.143	36.338	7.249	3.555	-0.216	17	1.463	0.83

Table 3: Change From FGK Model to FGK-DF Model

In Tables 1-3, the column listed as “Distance” refers to the city-block metric distance (Equation 3) between the frequency profile of the unknown or target protein being input to a given model, and the representative frequency profile of a given cluster. These distances act as thresholds, such that for a prediction to be made on a given protein, the distance between its frequency profile and a given’s clusters representative frequency profile must be less than the threshold. The row labeled “Sec. Struct. Sim.” is shorthand for secondary structural similarity, a measure introduced in the previous chapter (Equation 4). In this case, the measure refers to the overall secondary structure similarity of the proteins contained in the decision tree/cluster the prediction is being made on. Again, just like distance, this acts as a threshold, such that only the predictions made on trees or clusters of at least 70% or at least 80% secondary structure similarity are reflected in the above charts. Naturally, coverage and the 0.5 Å – 1.5 Å columns refer to the aforementioned prediction coverage and prediction accuracy measures, such that the values that exist in the columns reflect percentages out of 100%.

Granted this, Table 1 reflects the results of using only the FGK model to perform tertiary structure predictions, while Table 2 reflects the results of using the FGK-DF model to perform tertiary structure predictions. Table 3 summarizes the differences between the two result sets, noting areas that decreased with the use of the FGK-DF model with red cells, and areas that increased with blue cells. A brief glance at Table 3 will note that most of the table is populated by blue cells, indicating an overall increase in result quality by using the FGK-DF model over that of the FGK model. The only consistent drop in quality is a minor decrease in coverage, with overall decreases averaging less than 1%. This minor drop in coverage is more than adequately offset by the massive increases in prediction accuracy, especially in terms of the exception prediction accuracy tier (0.5 Å). This, in itself, justifies at least the use of the FGK-DF model over that of the FGK model for the purpose of predicting local tertiary structure. In order to justify the use of the FGK-DF model in terms of pure coverage and prediction accuracies, one need only look at Table 2. Considering that these results are based on comparing molecular distances between experimentally determined protein structures and predicted protein structures, the top prediction accuracies of 75.173%, 92.788%, and 96.533% for the prediction accuracy tiers 0.5 Å, 1.0 Å, and 1.5 Å, respectively, are astounding. Of course, one would note that the coverage at such levels is extremely low (0.148%). Instead, one can regard other distance thresholds and secondary structure similarity thresholds which, while they often times have lower prediction accuracies, they have much higher coverage. Consider bottom row (distance of

1300) on Table 2, with a secondary structure similarity threshold of greater than 70%. While its prediction accuracies are lower (though still acceptable in both the good and adequate categories), its coverage of 11.442% means that roughly 55,000 protein segments were predicted with the noted prediction accuracies. Again, given the scale at which these predictions are being made, the success here cannot be understated. As such, these objective results also justify the use of the FGK-DF model for predicting local tertiary structure.

Thus, the conclusion for this experiment, which asks if the use of the FGK-DF for predicting local tertiary structure of proteins is justified, is based on two conditions: the FGK-DF model outperforms the FGK model in terms of prediction coverage and accuracy, and the FGK-DF model, using purely objective and stand-alone metrics performs adequately given the challenges the problem of tertiary structure prediction presents. From the above analysis and the results shown in Tables 1-3, it is very clear that the FGK-DF model not only outperforms the FGK model, but that it produces results with acceptable coverage and outstanding prediction accuracies. Granted that the FGK-DF can justifiably be utilized for tertiary structure prediction, it should be noted that the model itself has a considerable weakness, which is an assumed motif size. This assumed motif size is a consequence of relying on a set window size (introduced in the previous chapter) for determining what constitutes a given 'protein segment.' A set window size limits motif extraction quality in two potential ways. First, the window size may be too restrictive to adequately encompass all potential motif lengths. For instance, the window size is nine in this work, but it is very possible than a motif in the data could be larger than nine positions, resulting in extracted motifs that are disjointed or cut off. The other possibility is that the motif in the data is shorter than the window size, meaning that any extracted data would be accompanied by considerable noise. Given that these protein family transcending motifs are often weaker and more subtle, this could adversely affect the results. As the FGK-DF model is heavily reliant on properly extracted motifs that transcend protein families in order to make proper tertiary structure predictions, this work proposes the Hierarchically-Clustered Hidden Markov Model (HC-HMM) approach for discovering and extracting protein motifs in a manner that makes no assumption on the side of the motif. In this approach, each protein sequence, defined in terms of a frequency profile, is modeled as a Hidden Markov Model and hierarchically clustered according to the minimum distance achievable between given HMMs. Once all HMMs are clustered, those regions with greater than a given threshold of clustered HMM nodes are to be considered protein motifs. No assumption is made on the size of the protein motif, as each sequence is treated as a separate HMM, and the approach can detect protein motifs that transcend protein family boundaries as the model does not rely on protein homologies. The next chapter will explore this approach in greater depth, explaining what a Markov Model is, extending this to Hidden Markov Models, connecting this structure to the representation of primary sequences of proteins, laying out the HC-HMM algorithm, and finally exploring the results of using the HC-HMM algorithm to extract primary sequence motifs.

IV. Hierarchically Clustered-Hidden Markov Models Algorithm

As the previous chapters would explore, one of the greatest short comings of the FGK-DF model, and many other algorithms that rely on extracting primary sequence motifs (such as PROSITE [9], PRINTS [36], MEME [37], DREME [13], etc.), is that there are assumptions on the maximum or minimum length of the extracted sequential motifs. In the FGK-DF model, this assumption is explicitly set by the window size, such that the training and testing data is segmented into individual window size-length residues and then merged using a sliding window technique explained in previous chapters. The limitation of using an assumed motif size can be overcome by two distinct steps: (A) changing the way that the data is represented such that the protein segments are not needlessly segmented into residues and (B) changing the way the data is clustered and the way sequential motifs are extracted such that each data point does not need to the same assumed size to perform the distance calculations for clustering and extraction. The following sections will explore how the Hierarchically Clustered-Hidden Markov Models (HC-HMM) algorithm solves these two problems, respectively, by representing the primary sequence of the proteins as Hidden Markov Models, and by hierarchically clustering those HMMs along sections of greatest similarity and extracting those sections with high numbers of clustered HMMs as sequential motifs. As such, it is pertinent to understand what a Markov Model is, what a Hidden Markov Model is, and how these data structures can be used to represent protein primary sequences.

4.1 Markov Models, Hidden Markov Models, and Proteins

A Markov Model, and by extension a Hidden Markov Model (HMM) is based on a system of states and probabilities that exist between those states. An example of this is shown below:

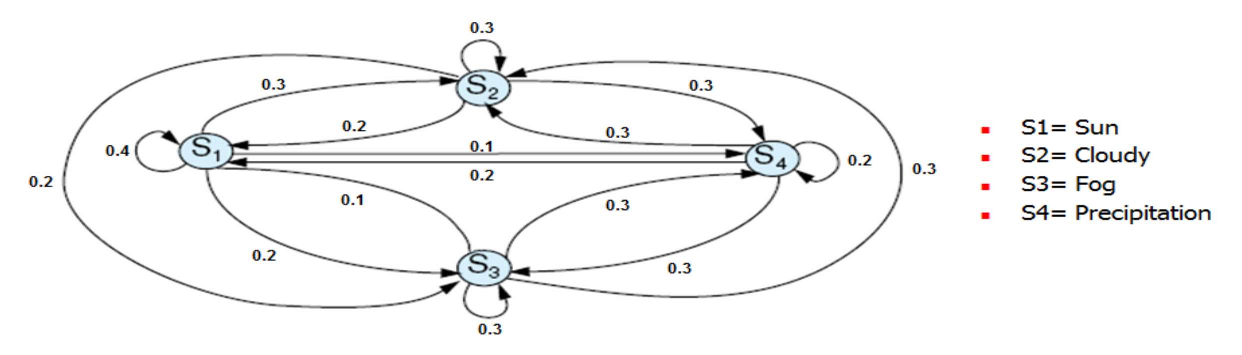


Fig. 7: Markov Model Example

In Figure 7, the states in question are weather and the probabilities suggest the likelihood that one weather pattern will be replaced by another one in the following day. In other words, regard state S1, which corresponds to sunny weather. One can follow the model and note that the probability that the sunny weather will transition to S2, which is cloudy weather, is 0.3, or 30%. From there, the probability that the cloudy weather will transition back to sunny weather, assuming it can measured based solely on the previous day's weather, is 0.2, or 20%. The rest of the transitions and states follow suit, however one should note that this is a simple Markov Model, both in terms of the transition equation (which can get quite complex), and that it isn't an

HMM. HMMs make use of *hidden states*, and is more complex in concept that the Markov Model. It is best exemplified by the following figure:

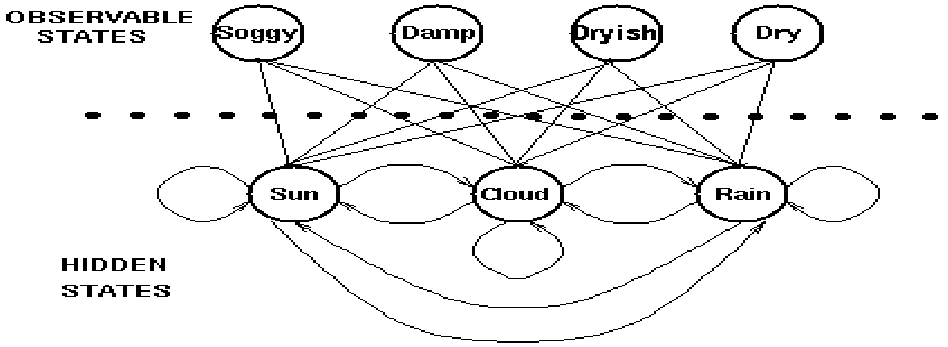


Fig. 8: Hidden Markov Model Example

In Figure 8, one will note that there is a division between those states that are "observable" ("soggy," "damp," "dryish," and "dry") and the "hidden" states of the model ("sunny," "cloudy," "rainy"), which are used to build the model and determine, or output, the observable states. The states "sun," "cloud," and "rain," are "hidden" because the sequence these states are fired in in order to produce the observable states is unknown; only the output, the "observable" states, can be seen. The HMM can contain multiple hidden levels, where there are probabilities to go from one level to the next, as well as probabilities to output an observable state, making it very flexible and much more representative of how processes in the world actually work [26].

Granted this, the question now becomes how does one use a Hidden Markov Model to more adequately and accurately represent a protein primary sequence? The answer lies in a work by Baldi et al., titled “Hidden Markov models of biological primary sequence information,” in which the HMM is structured with five primary states: the start state, terminal state, the emission state, the insert state, and the delete state, following the evolutionary behavior explained in Figure 2, Chapter 1 [30], such that the traversal each node of the HMM produces an amino acid (or is mute) to build up and represent the overall primary sequence of a protein. A graphical representation of this structure is shown in the figure below:

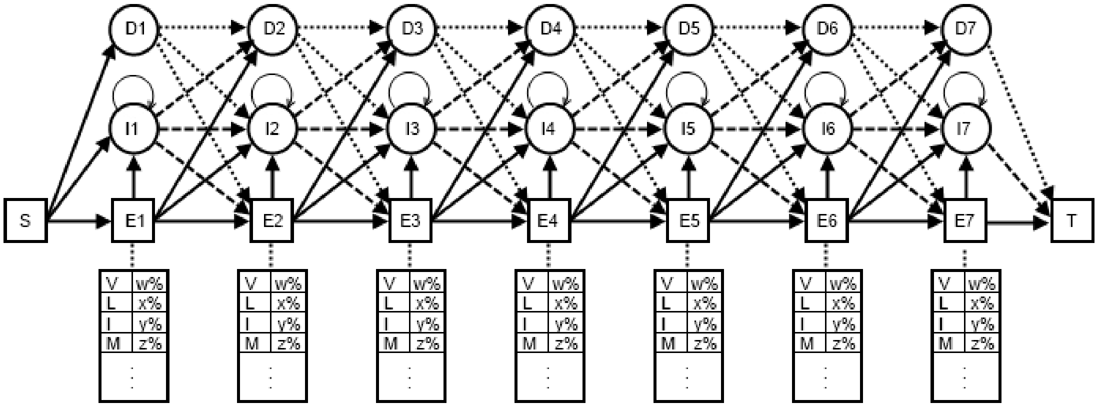


Fig. 9: Protein Primary Sequence as a Hidden Markov Model

In Figure 9, state *S* refers to the aforementioned starting state of the HMM. It produces no output and its transitional probabilities are defined by first node in the protein sequence, where “node” refers to the collection of transitional probabilities {$p(D_i)$, $p(I_i)$, $p(E_i)$} and states {D_i, I_i, E_i} which describe the behavior and characteristics of the *ith* position in a protein sequence. For each node, the state D_i refers to the delete state, which outputs no amino acids. $p(D_i)$ refers to the transitional probability that a given state in $node_{i-1}$ will transition to D_i. State I_i refers to the insertion state and outputs an amino acid based on the frequency profile of $node_i$, where frequency profile refers to a probability distribution of each possible amino acid appearing at a given position within a given protein sequence. *$p(I_i)$* refers to the transitional probability that a given state in node_{i-1} will transition to I_i as well as the probability that I_i will transition to itself again (which can be repeated to an arbitrary degree based on said probability). State E_i refers to the emission state, which outputs a single amino acid based on the frequency profile of node_i. $\text{p}(E_i)$ refers to the transitional probability that a given state in node_{i-1} will transition to E_i. Finally, state T refers to the terminal state, which marks the end of the Markov chain and produces no output.

Using this structure, any number of protein primary sequences can be easily represented, both structurally and behaviorally, by simply defining the probabilities of each of the three primary states (emission, insertion, and deletion) for each amino acid position in the protein sequence. However, while representing a protein primary sequence using its behaviorally probabilities does more accurately describe the sequential structure and makes no assumptions on motif size (as the protein sequence is in no way segmented), simply representing a protein primary sequence as a HMM does not resolve the problem of being able to *extract* primary sequence motifs without an assumed protein motif size. The solution this work explores to resolve this issue of extracting motifs without an assumed size is to perform hierarchical clustering on the produced HMMs by aligning and clustering two or more HMMs along nodes of highest similarity based on distance calculations and extracting areas with at least *m* aligned HMMs as sequential motifs. This process is noted as the Hierarchically-Clustered Hidden Markov Model algorithm, as the next section will explore in greater depth.

4.2 HC-HMM Methodology

As mentioned in the previous section, in order to make no assumptions on the size of the protein sequential motifs that are to be extracted, the HC-HMM uses hierarchical clustering, which builds a *hierarchy* of clusters rather than treating all clusters as distinct, equal entities, such as in K-Means clustering. A simple example of hierarchical clustering is shown in Figure 10 below.

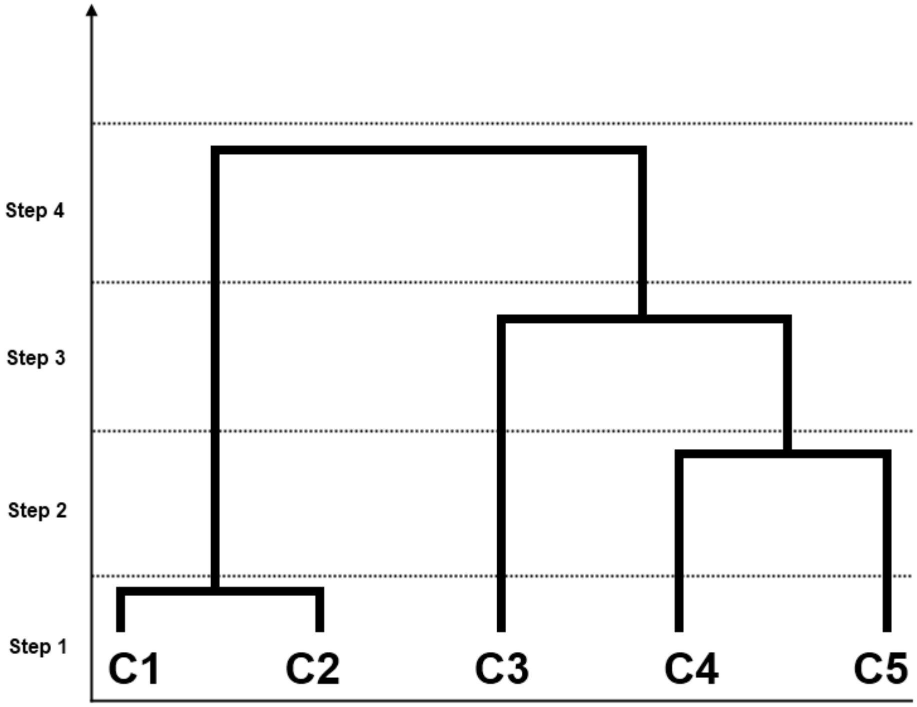


Fig. 10: Hierarchical Clustering Example

The process of hierarchical clustering begins like any other clustering process, with distinct, un-clustered data elements. In Figure 10 above, these data elements constitute a set containing C1, C2, C3, C4, and C5. The clustering process begins in Step 1, such that, using a distance equation or other comparable similarity metric, hierarchical clustering determines the first two data points that are most similar to each other. In the example above, the first two data elements that are most similar to each other are C1 and C2, which are clustered together as the first level of the cluster hierarchy. The clustering process continues by determining the next two most similar data points, which in this example include C4 and C5. Just as with C1 and C2, these are clustered and added to the hierarchy. This same process is carried out in Step 3, with C3 being determined to be most similar *to the cluster* generated by C4 and C5, creating a new cluster containing a lower level cluster and a data point. This process of determining the similarity between a single data point and cluster can be carried out a great number of ways, one of the more common including averaging all of a cluster's data points into one representative data point and comparing it against the single data point. Finally, the clustering is completed in step 4 when only one, last cluster is possible to be generated, the one encompassing clusters {C1, C2} and {C3, C4, C5}, which is added at the third and final level of the hierarchy. The process of hierarchical clustering can be terminated prematurely based on a given threshold or by reaching a certain level in the hierarchy. For instance, the example in Figure 10 could have been terminated after a certain step (such as Step 3) or once the similarity measures being generated were beyond a given threshold.

Granted the process of hierarchical clustering, HC-HMM attempts to build a hierarchy by comparing each node of a HMM chain against another node in another HMM chain based on weighted distance calculations utilizing each nodes' emission state, insert state, and delete state probabilities. Those HMM chains containing the nodes that are considered the most similar are clustered as a level in the hierarchy. The clustering process begins with the shortest HMM chain and terminates when all HMM chains have been clustered into one root cluster. The pseudocode for this approach is shown in Figure 11 below:

```
$\alpha$ = List of generated HMM models
$\beta$ = List of processed HMM models

WHILE length($\alpha$) > 0:
   $\alpha_i$ = Find_And_Remove_Shortest_Model($\alpha$)

   minDistance, curDistance, offset = 0
   leastModel = NULL

   FOR each $\alpha_j$ in $\alpha$:
      FOR each $node_k$ in $\alpha_j$:
         FOR each $node_l$ in $\alpha_i$:
            curDistance += **Dis**($node_{k+l}$, $node_l$)

         curDistance /= length($node_l$)

         IF curDistance <= minDistance:
            leastModel = $\alpha_j$
            minDistance = curDistance
            offset = k

   IF minDistance <= THRESHOLD:
      Add_Model_To_Cluster(leastModel, $\alpha_i$, offset)
   ELSE:
      $\beta \leftarrow \alpha_i$
```

Fig. 11: HC-HMM Algorithm

In Figure 11, 'α' refers to the list of HMM chains generated using the same source of protein primary sequence information described in previous chapters, the HSSP. 'β' refers to the list of processed HMM chains, containing those models that have failed to achieve the minimum distance threshold. Ultimately, all chains will be placed in list 'β' due to the traversal of the chain size hierarchy. The function 'Find_And_Remove_Shortest_Model()' removes the HMM chain with the fewest number of nodes from the list α and stores the removed value in α_i. The local variables 'minDistance,' 'curDistance,' and 'offset' respectively refer to the minimum distance between two HMM chains that has been achieved thus far, the current distance of the current chains being examined, and the number of empty nodes to be inserted at the beginning of chain 'α_i' to achieve the proper clustering with the currently examined chain. The local variable 'leastModel' holds a pointer to the HMM chain that currently has the shortest cluster distance with chain '$\alpha_{i.}$' The function 'Dis($node_k$, $node_l$)' determines the distance between two input nodes using one of the following three equations:

$$\boldsymbol{Naive}(k,l) = |p(D_k) - p(D_l)| + |p(I_k) - p(I_l)| + \boldsymbol{FPD}(k,l)$$

Eq. 9: HC-HMM Naïve Distance Calculation

$$\boldsymbol{Mult}(k,l) = (|p(D_k) - p(D_l)| + 1) * (|p(I_k) - p(I_l)| + 1) * \boldsymbol{FPD}(k,l)$$

Eq. 10: HC-HMM Multiplicative Distance Calculation

$$\boldsymbol{Add}(k,l) = |p(D_k) - p(D_l)| * \boldsymbol{FPD}(k,l) + |p(I_k) - p(I_l)| * \boldsymbol{FPD}(k,l) + \boldsymbol{FPD}(k,l)$$

Eq.11: HC-HMM Additive Distance Calculation

Where 'k' and 'l' refer to two nodes from two different HMM chains, '$p(D_k)$' refers to the deletion state transitional probability of node 'k,' '$p(I_k)$' refers to the insertion state transitional probability of node ‘k,’ and ‘FPD’ returns the frequency profile distance between two nodes, defined by the following equations:

$$\boldsymbol{FPD}(k,l) = \sum_{i=1}^{20} |Freq_k(i) - Freq_l(i)|$$

Eq. 12: Frequency Profile Distance

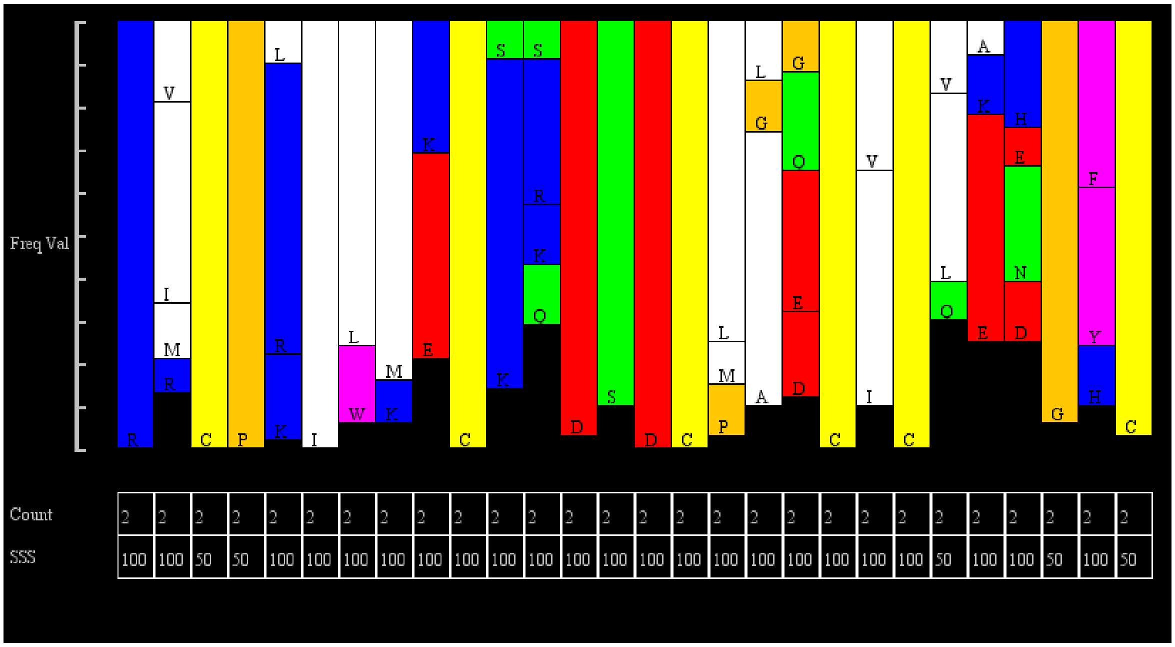
Where ‘$Freq_k(i)$’ refers to the probability that amino acid ‘i’ will be emitted by node ‘k.’ Equations 9, 10, and 11 are referred to, respectively, as the Naïve, Multiplicative, and Additive distance equations. The Naïve distance equation lightly penalizes the cluster distance by adding the absolute difference between the insertion and deletion transitional probabilities of node ‘k’ and node ‘l’ to the frequency profile difference. The Multiplicative distance equation heavily penalizes the cluster distance by multiplying the absolute difference of each node’s deletion and insertion transitional probability plus one (such that if the transitional probabilities are equal, the distance is not penalized at all) with the frequency profile distance. Finally, the Additive distance equation penalizes the cluster distance by separately multiplying insertion transitional probability absolute difference and deletion transitional probability absolute difference with the frequency profile distance.

Once the distance is found for a particular clustering attempt, it is compared against the ‘minDistance.’ If it less than the ‘minDistance,’ the leastModel, minDistance, and offset are all updated appropriately. This is repeated for all possible clusters for a given chain, for all chains.

Once the chain with the minimum clustering distance is found, its distance with chain 'α_i' is compared against a set value stored in 'THRESHOLD.' If the distance is less than the threshold, the function 'Add_Model_To_Cluster()' is called, which averages the transitional probabilities and frequency profiles of each node clustered in the chains 'leastModel' and 'α_i.' Each averaged value is weighted by the number of proteins represented by that node, which is extracted from the HSSP data.

This process of removing the smallest HMM chain and attempting to cluster it with the remaining chains is performed until the list 'α' is empty. At this point, the list 'β' will contain all remaining models, including those that have clustered with other chains as well as chains that failed to cluster with any chains. The latter of these are ignored in the final step of the HC-HMM approach, which constitutes the sequential motif extraction.

The final and most pivotal step in the HC-HMM method, motif extraction, is conceptually simple: extract all local sequences with at least *m* HMM chains clustered at a given position and declare each one to be a sequential motif. This takes advantage of the fact that the HC-HMM compares and clusters HMM chains along their most similar nodes, generating what is effectively an alignment. In a given hierarchy generated by the HC-HMM, there can be a large number of prominent alignments composed of two or more HMM chains overlapping over several nodes. These overlapping alignments composed of at least *m* HMM chains, again, are to be considered sequential motifs. The process of extracting these motifs can be autonomously performed by iterating over all produce HMM clusters and flagging any contiguous sequences within a HMM cluster that meet the above criteria. To verify that a flagged sequence is a potential motif, visual inspection through a HMM cluster visualizer utility can be performed. An example of the output of the visualizer using a sample HMM cluster is shown in Figure 12 below:



Count	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
SSS	100	100	50	50	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	50	100	100	50	100	50

Fig. 12: HC-HMM Sequential Motif Visualizer

In the above output the average frequency profile (Freq Val), the number of clustered HMMs (Count), and the secondary structure similarity (SSS) per node are shown for HMMs that have been successfully clustered. The average frequency profile per node is shown in terms of single, multi-colored bar denoting values between 0 and 100%. Each color corresponds to a set of amino acids: amino acids V, L, I, M, A are white, F, W, Y are magenta, G, P are orange, S, T, Q, N are green, C is yellow, H, R, K are blue, and E, D are red. Note that as certain amino acids share colors in the visualizer, some contiguous blocks of color (such as the R and K or V, I, and M blocks in Figure 12) are separated by black lines to denote individual amino acid frequencies. For the sake of clarity, amino acids with frequencies of less than 8% are not shown.

In addition to the amino acid frequencies, the count for each node is provided, shown as a number in the first row below the frequency profile data in Figure 12, which denotes the number of chains that were clustered on each node. In the example shown in Figure 12, there are two chains clustered on each node, meaning the resulting "alignment" is composed of two HMM chains. Finally, the secondary structural similarity, shown as a number between 0.0 and 1.0 (with 1.0 denoting complete structural homology) on the bottom row below the frequency profile data in Figure 11, refers to the overall homology of the secondary structure of each node in a given cluster, computed using equation 4 introduced in Chapter 2.

Thus, all together, the Hierarchically Clustered-Hidden Markov Model method first takes in protein primary sequence information and generates, for each protein sequence, a Hidden Markov Model. Each of these generated HMM chains are then removed, starting with the smallest chain, and clustered with other HMM chains or HMM chain clusters based on largest nodal similarity utilizing one of the three weighted distance equations listed above. The clustering process terminates once all HMM chains are clustered, at which point sequential motifs can be extracted based on discovering and flagging contiguous sequences of at least *m* clustered chains. Therefore, given the process involved in the HC-HMM method, the following section will explore the effectiveness of the method in extracting sequential motifs from a set of protein primary sequences, and examine notable motifs extracted by the process.

4.3 HC-HMM Motif Extraction Results (Data Trends)

In order to test the effectiveness of the HC-HMM for extracting sequence motifs that transcend protein family boundaries, 2,593 HSSP files representing proteins exhibiting less than 25% sequence identity were processed by the HC-HMM method utilizing each of the three distance formulas defined in the Methodologies section (Naïve, Multiplicative, and Additive) over a range of distance thresholds normalized between 0 and 1 and a step size of 0.01. Each HSSP file, which contains not only the frequency profile information but also the insertion and deletion probabilities for each amino acid position in the protein primary sequence, was converted into a distinct HMM chain using the structure described in the previous sections. The data was supplemented by the DSSP for secondary structure information strictly for the evaluatory purposes as outlined in the previous section. For each produced HMM cluster, motifs were extracted based on a minimum node cluster count, *m*, such that any contiguous sequence of HMM nodes with at least a node cluster count of *m* would be considered a motif. The count, average length, and average secondary similarity of the extracted motifs for each application of the HC-HMM method were recorded. This process was executed for values of *m* ranging between 3 and 5, the results of which are shown in Figures 13-18 below:

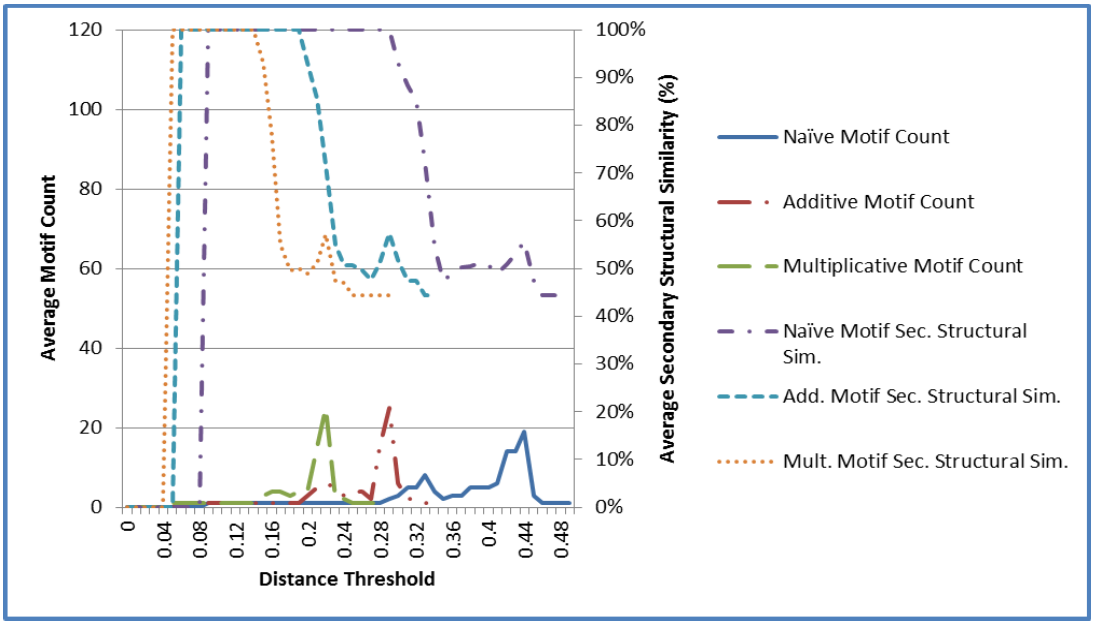


Fig. 13: Motif Count and Secondary Structural Similarity when *m* = 3

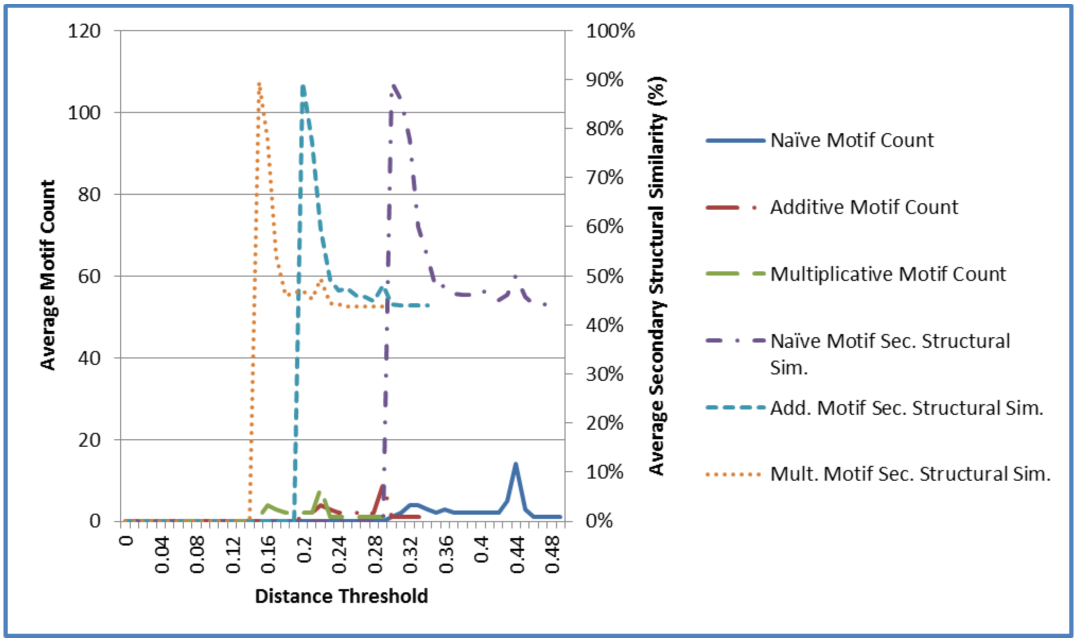


Fig. 14: Motif Count and Secondary Structural Similarity when *m* = 4

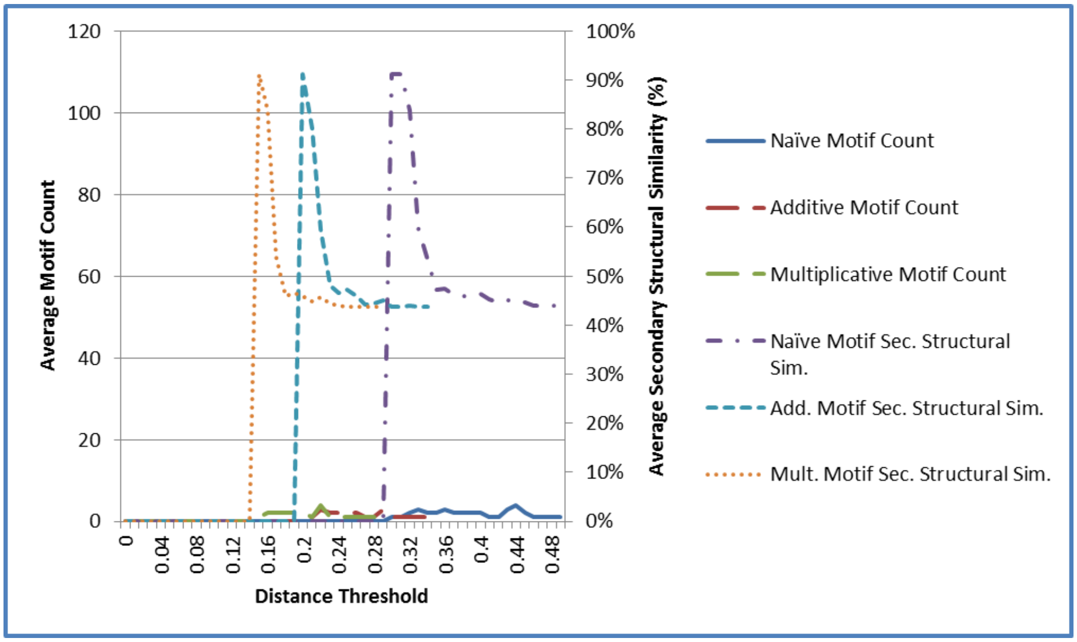


Fig. 15: Motif Count and Secondary Structural Similarity when *m* = 5

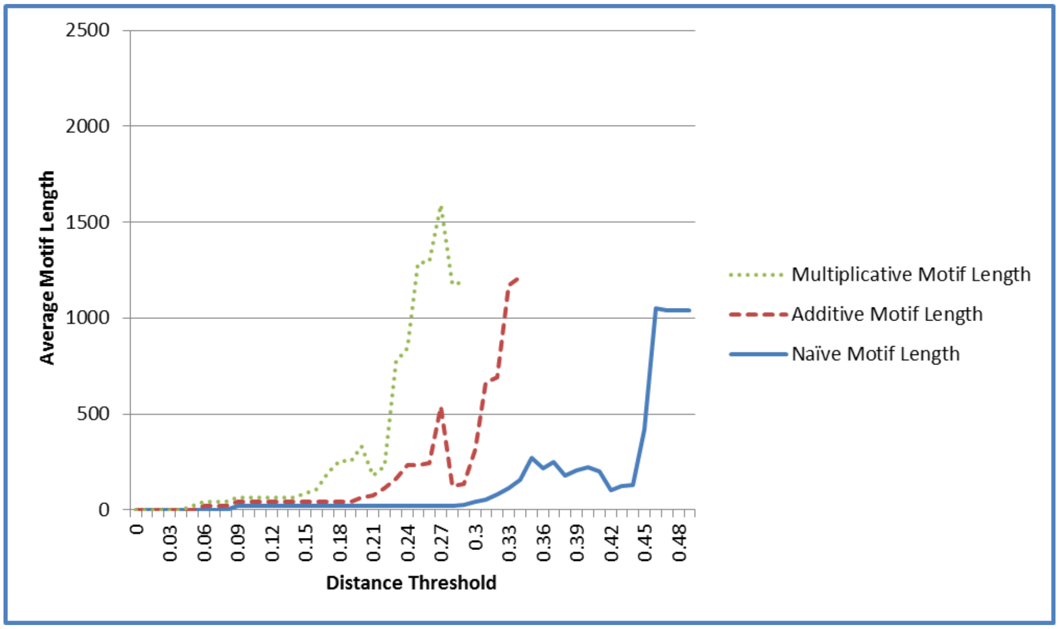


Fig. 16: Motif Count Length when *m* = 3

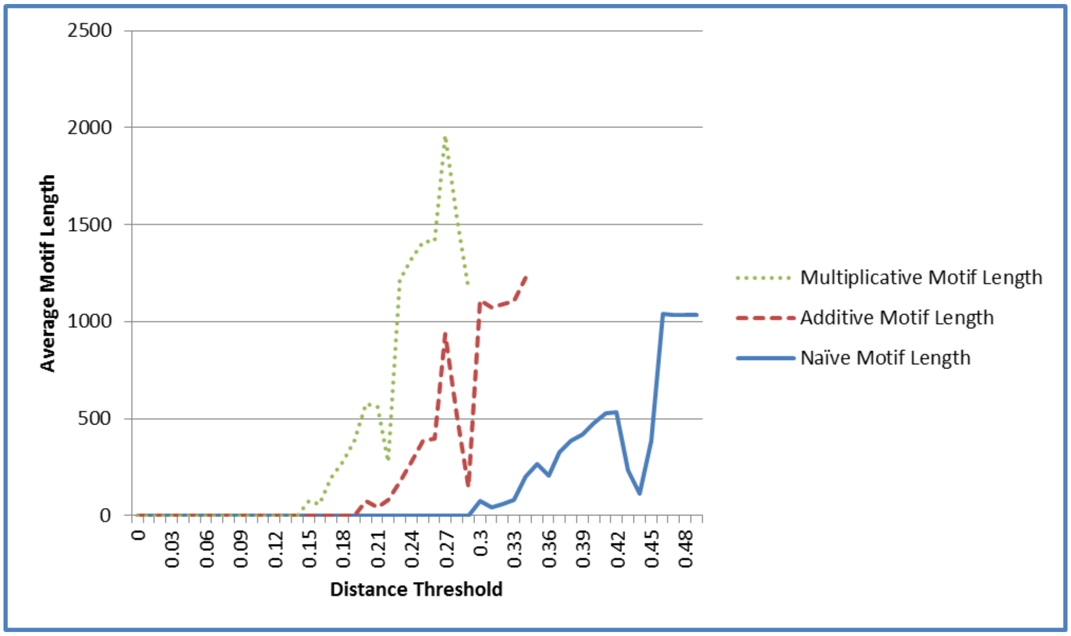


Fig. 17: Motif Count Length when *m* = 4

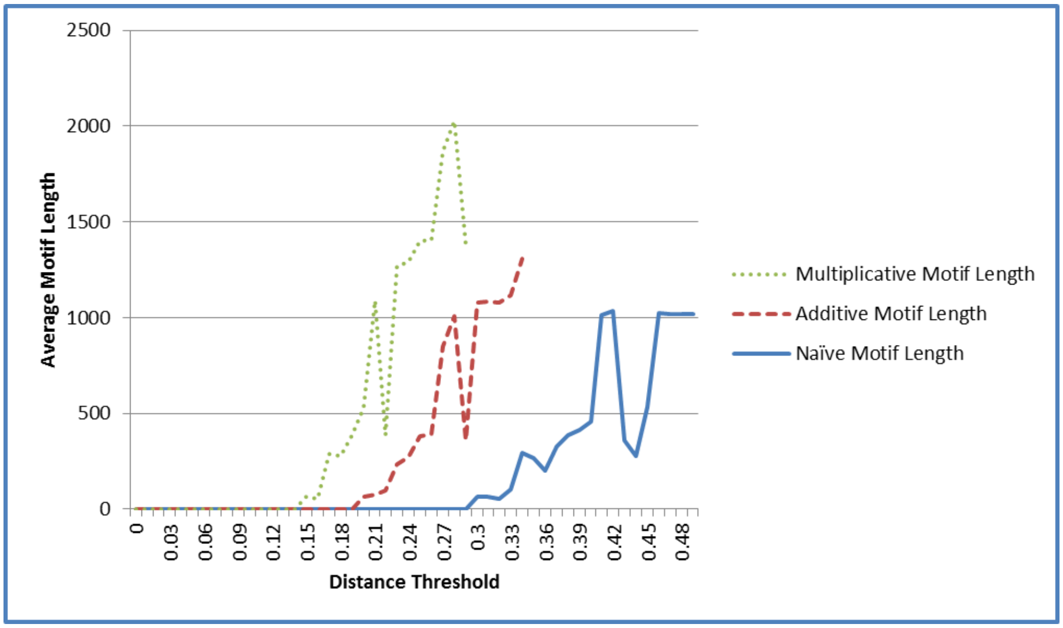


Fig. 18: Motif Count Length when *m* = 5

In Figures 13-15, the average motif count and secondary structural similarity of each HMM cluster produced by a given threshold (ranging from 0.0 to 0.50, omitting distance thresholds that do not produce HMM clusters) is shown for each of the three distance functions and increasing values of *m*. Note that in Figures 13-15, average secondary structure similarity is scaled by the right vertical axis while average motif count is scaled by the left vertical axis. A common trend for all values of *m* shown above is that as distance threshold increases (and thus becomes less restrictive) the motif count, in general, increases as secondary structure similarity decreases. This trend continues until a tipping point in the distance threshold is met, at which all protein data is clustered into one large cluster. At this point, the motif count and secondary structure similarity both spike, producing a significant local maximum for both count and secondary structure similarity. This trend is most apparent when $m = 3$, growing gradually more subtle as *m* increases.

A similar trend can be seen in Figures 16-18, showing the average length of each motif as distance threshold increases for each of the three distance functions. Motif length increases as the distance threshold increases. This is due to the less restrictive distance thresholds, again, causing the HMMs to cluster into one large cluster, increasing the possible length of contiguous sequences. Inverse to what Figures 13-15 exhibited, the motif length drops to a local minimum as the distance threshold tipping point is reached. It is notable that as *m* increases, the average length of the motifs also increases. This is most likely due to smaller values of *m* detecting shorter, sparser motifs, thus lowering the overall average length.

Interestingly enough, all three distance formulas produce roughly identical trends with varying distance threshold scales, suggesting that the primary difference in the three distance formulas is sensitivity, with Multiplicative being the most sensitive and Naïve being the least sensitive. Given that, the Naïve function will be the only function discussed any further.

4.3 HC-HMM Motif Extraction Results (Extracted Motifs)

Given the assumption that motifs generated with greater values of m indicate more prominent motifs, and given the assertion that there exists a local maximum for motif secondary structural similarity as distance threshold increases, this work extracts three notable motifs generated with the Naïve distance function with a distance threshold of 0.30 where m=3 and m = 5. These motifs, as depicted by the visualizer, are shown in Figures 19, 20, and 21. The proteins used to generate Figure 19 include 1qsu (chain A), 1q7d (chain A), and 1dzi (chain B). The proteins used to generate Figure 20 include 1cgd (chain A), 1ei8 (chain A), and 1bkv (chain A). The proteins used to generate Figure 21 include 1gqe (chain A), 1ic2 (chain A), 1gmj (chain A), 1uuj (chain A), and 1na3 (chain A).

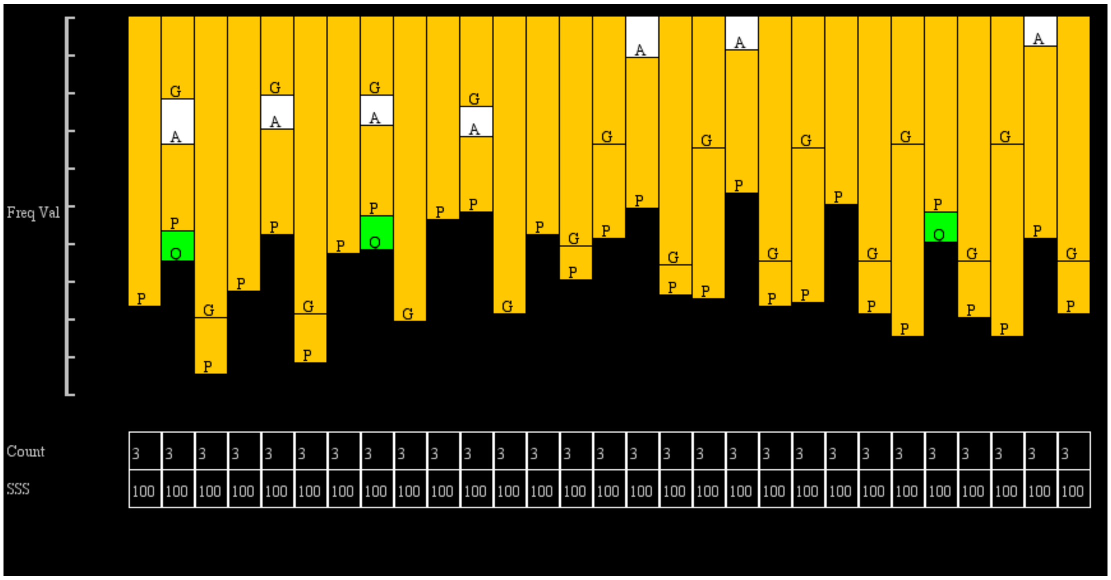


Fig. 19: Motif (29 Residues, 100% Secondary Structural Similarity) when *m* = 3

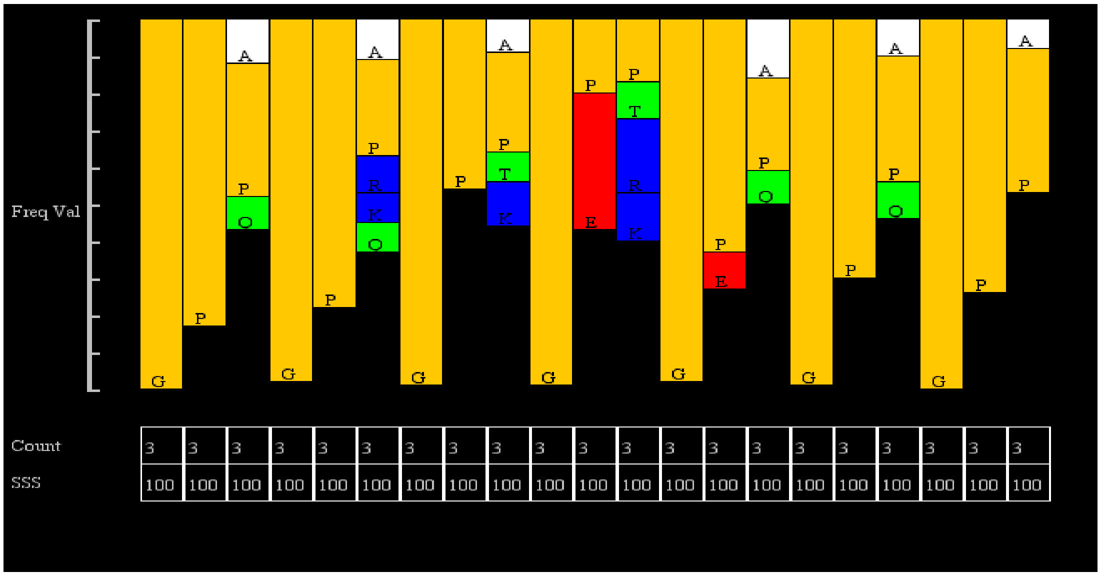


Fig. 20: Motif (21 Residues Long, 100% Secondary Structural Similarity) when *m* = 3

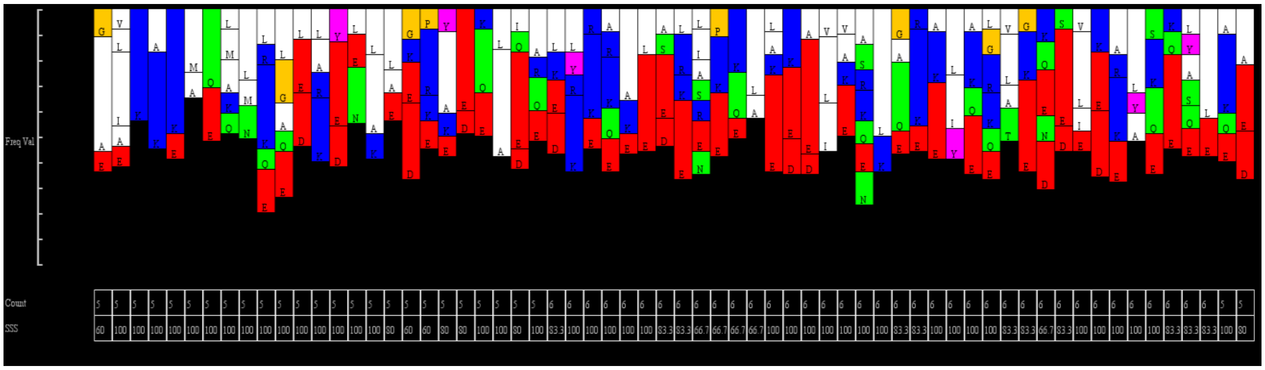


Fig. 21: Motif (64 Residues Long, 91.09% Sec. Struc. Sim.) when *m* = 5

In Figure 19, there is a clear and relatively consistent three residue pattern in the extracted motif, with a residue predominately composed of glycine followed by a residue typically composed of proline and ended by a residue composed of alanine, proline, and glutamine. This pattern is roughly repeated seven times in the extracted motif. Figure 19 holds a similar repeated pattern structure, with two clear patterns: proline-dominant residue followed by glycine, alanine, and proline-dominant residue followed by a glycine dominant residue, and a pattern defined by proline-dominant residue followed by a glycine-dominant residue followed by roughly equal parts proline and glycine. Each of these two patterns repeat themselves roughly five times within the motif. It is important to note that the average secondary structure similarity of these two motifs is 100%, which suggests that these motifs are significant not only for primary sequence analysis, but structural analysis as well.

Figure 21 denotes a much larger but less regular motif, extracted based on high overall secondary structural similarity (91.09%) as well as its high cluster count, which ranges from 5 to 6. While this motif does not contain any apparent repeating patterns, there are regularities to note. The motif, as a whole, generally exhibits a high frequency of glutamic acid with smaller but persistent traces of aspartic acid. Though not as consistently present, there is a notable frequency of lysine as well as leucine. Again, given the high cluster count and high secondary structure similarity, this motif has strong implications for both sequential and structural analysis. It is also possible that this motif, given its considerable length, is potentially composed of sub motifs, though further analysis would be required to test this assertion.

To explore the potential of this methodology for structural prediction and analysis, an average tertiary structure for the three motifs shown in Figures 19-21 was generated and visualized. To generate the tertiary structure information, the base protein models and chain for each extracted motif (described in the prior paragraphs) was used to perform a query on the PDB. The three dimensional positions for each alpha carbon atom for a given chain of a given protein was recorded, and a mutual distance matrix was calculated between each recorded vertex contained within the generated motif to remove any rotational, mirroring, etc. inconsistencies in the extracted tertiary information. Each mutual distance matrix was then averaged for each protein present in a given motif. The resulting tertiary structures for the motifs denoted by Figures 18-20 are shown, respectively, by Figures 22-24 below:

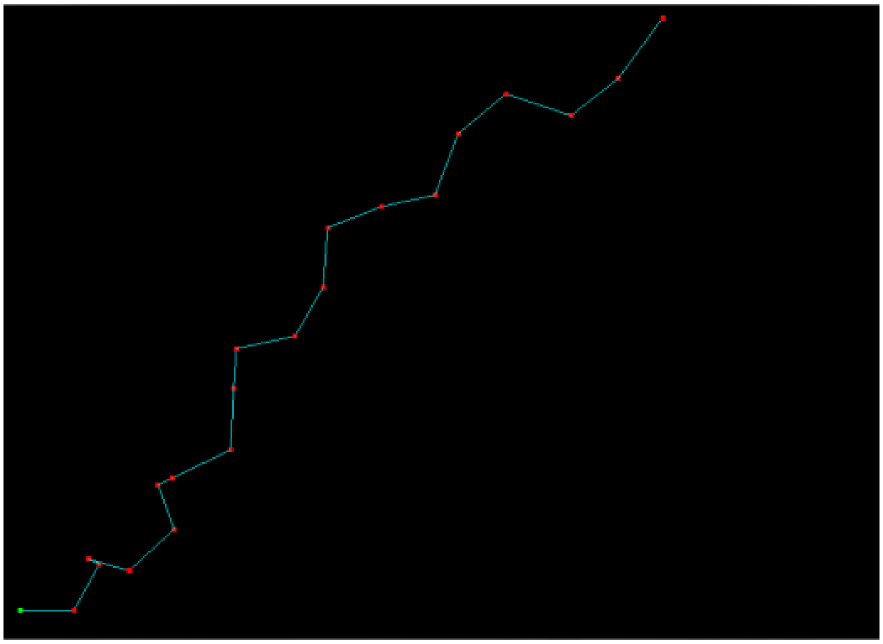


Fig. 22: Visualized Tertiary Structure of Motif Containing Proteins 1qsu, 1q7d, and 1dzi

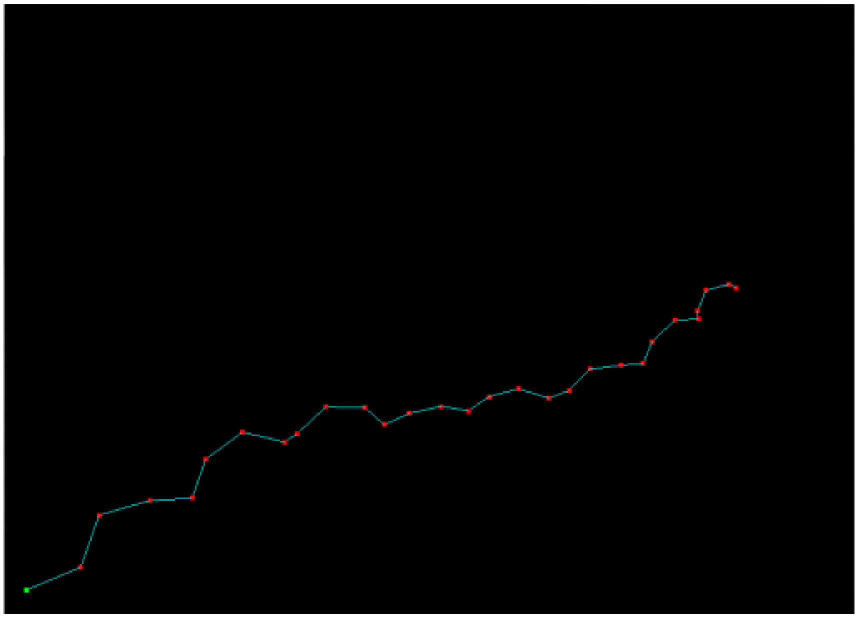


Fig. 23: Visualized Tertiary Structure of Motif Containing Proteins 1cgd, 1ei8, and 1bkv

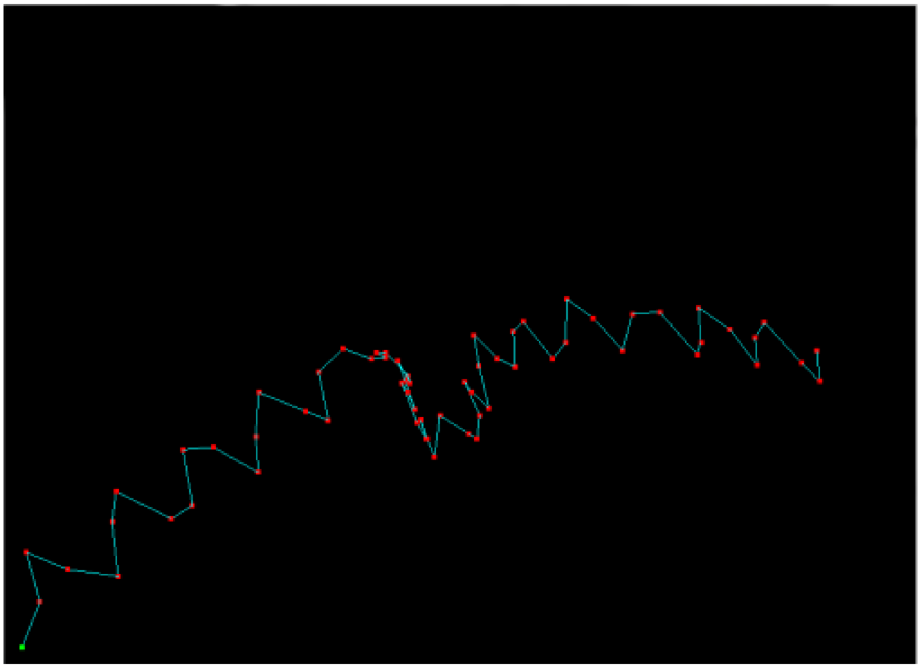


Fig. 24: Visualized Tertiary Structure of Motif Containing Proteins 1gqe, 1ic2, 1gmj, 1uuj, and 1na3

Thus, taken together, the limitations of the FGK-DF model, as well as many other motif extraction methodologies, is examined, with a focus on an assumed window size. This particular limitation is analyzed and overcome by utilizing the Hierarchically Clustered-Hidden Markov Model (HC-HMM) approach by representing protein data as Hidden Markov Models capturing protein behavior and metrics in terms of insertion, deletion, and amino acid probability nodes and hierarchically clustering the resulting HMM chains by minimizing distance between any two given chains. Motifs can then be extracted without any assumption on the length of the motif by analyzing the clusters and extracting contiguous sequences with a given threshold of clustered proteins. The effectiveness of this methodology and various parametric setups were critically examined in terms of the number, quality, and length of the resulting motifs. Furthermore, several example motifs generated by the HC-HMM approach were shown, examined, and visualized in terms of their averaged tertiary structure.

Granted the effectiveness of this approach for eliminating both outlined shortcomings, there is still much that can be improved upon. While the application of the HC-HMM on the outlined data is capable of generating over 100 distinct motifs from the generated clusters, those motifs typically only represent small contiguous segments where $m = 2$. While these motifs still contain valuable information, for the purposes of utilizing the HC-HMM for motif extraction and the FGK-DF for processing said motifs, further improvements are necessary, as the concluding chapter will touch on briefly.

V. Conclusion

Throughout this work, the methodology, results, merits, and drawbacks of the FGK-DF model for predicting protein local tertiary structure and the HC-HMM method for extracting primary sequence motifs have been laid out. Granted this, one must come back to why the work is important. What purpose is there in the FGK-DF and HC-HMM in the grand scheme of structural genomics? Even more abstractly, one must question the point of structural genomics, why finding the structure and thus function of proteins is important. Furthermore, it is pivotal to examine future works, particular focused on extending the functionality of the HC-HMM for extracting more informative and higher quality sequential motifs. As this work concludes, these two questions will be answered in the following sections, noting the importance of structural genomics (and by consequence the FGK-DF/HC-HMM methods) in cheap and effective drug design, as well as the future improvements on the HC-HMM that would generate both more numerous and higher quality motifs.

5.1 The Social Implications of Rapid Protein Structure Prediction

As noted in the Introduction, x-ray crystallography and NMR spectroscopy, the accepted historical approach to determining protein structures directly, were extremely time consuming, expensive, requiring expertise, etc., whereas models such as the FGK-DF model were incredibly cheap in terms of time and effort per prediction. Granted that, the question that is asked, now, is what are the implications using rapid protein structure prediction models, such as the FGK-DF model? How does protein structure prediction via an algorithm and a computer, rather than a lab brimming with biologists, time, and money, change or benefit society? The answer lies in the explicit goals of the field of structural genomics, and the potential it holds for pharmaceutical companies. Structural genomics, to reiterate, is best explained by the following passage: "Structural genomics (SG) is an international effort to determine the three-dimensional shapes of all important biological macromolecules, with a primary focus on proteins. A major secondary goal is to decrease the average cost of structure determination…" [20]. Clearly, this paper is well aligned with the concepts and goals of structural genomics, as not only has this work provided a method by which one can predict the local tertiary structure of a protein, it does so quickly (over 2,000 protein structures determined to some degree in less than an hour) and with an appreciable level of accuracy. But what connection does structural genomics and tertiary structure prediction share with pharmaceutical companies? As it has been heavily implied throughout this work, the structure of a protein determines its function [15]. Knowing the structure *and* function of a protein is invaluable to pharmaceutical companies as the following passage states:

> The long path from genomic data to a new drug can conceptually be divided into two parts. The first task is to select a target protein whose molecular function is to be moderation, in many cases blocked, by a drug molecule binding to it. Given the target protein, the second task is to select a suitable drug that binds to the protein rightly, is easy to synthesize, is bio-accessible and has no adverse effects such as toxicity. The knowledge of the three-dimensional structure of a protein can be of significant help in both phases [and] affords well-founded hypotheses of the function of the protein. [38]

Put a different way, knowing the structure of a protein allows drug designers to create a drug that directly tackles the problem with significantly reduced risk of adverse side effects, as the drug would be designed specifically to bind to the target protein. This potential has been noted by pharmaceutical companies, which can spend millions in testing and design of new drugs (whose benefits can be outweighed by the aforementioned side effects). This has led to a larger exploration of what can be safely gleaned from predicting protein structures, in lieu of experimentally determined structures. The information can be as broad as the type of protein (such as enzyme or hormonal protein), to much more detailed information concerning the binding sites, and how other molecules interact with them [38]. This information, depending on the resolution, can lead directly to more beneficial drugs, with the ultimate goal of the research being the advent of truly personalized medicine, a goal structural genomics is making increasingly viable [14].

Of course, the previous passage, as well as this work as a whole, suggests that this research is already well on its way, that the FGK-DF model for local tertiary structure prediction, or the HC-HMM for motif extraction, has nothing new to add to the brimming potential of structural genomics. But this is far from true, as the FGK-DF model and HC-HMM method holds implications for the field of structural genomics itself. One of the primary concerns is that to actually reach the goals set forth by structural genomics, such that the research is viable for pharmaceutical companies, the structures predicted by the various models must be a novel protein structure. That is to say, the structure produced by the models must be unique [38] [20]. Clearly, this is not an easy obstacle for methods that are based on homology-modeling, which rely, at their core, on predicting structure based on the lack of uniqueness. Homology-modeling and experimental structure deduction can be used in tandem to overcome this weakness, but it begs the question: why bother? While the FGK-DF model cannot produce completely unique sequences, the fact that it A: relies on sequence motifs that transcend protein families and B: predicts tertiary structure at a local level, meaning that the structures the model produces are much more dynamic and have a much higher possibility of reflecting the true tertiary structure of the protein in question. This leads one to another problem with most current methods of structural prediction, which generally work on a global level and thus loss the aforementioned resolution. In fact, research suggests that "the molecular function of the protein is tied to *local* structural characteristics pertaining to binding pockets on the protein surface" [38]. The FGK-DF model provides high resolution predictions on exactly that, providing an excellent basis for determining not just binding sites in particular proteins, but patterns shared by all binding sites due to its basis in protein family-transcending sequence motifs. And of course, the HC-HMM method has the prominent benefit that it makes no assumptions on the motif size, allowing it to outperform, in regards to accurate motif extraction and depiction, most popular sequential motif extraction methods that currently exist in the field of structural genomics.

Thus, the main social implication of the FGK-DF model combined with the HC-HMM method, as well as other such models that further the aims of structural genomics, is to make drug design cheaper, more efficient, and the end drugs much more beneficial and safer to use. If the end goal is realized, personalized medicine, side effects are expected to all but disappear, as well as many debilitating ailments that can be remedied through gene therapy (supported also by structural genomics research) and personalized drugs. It is easy to see that this should lead to longer, more fulfilling lives to those otherwise doomed to struggle through life with either symptoms or side effects disrupting their daily routine. Though this is a very idealistic situation, it should be clear that the FGK-DF model and HC-HMM method, amongst its sister algorithms and competing approaches, are leading the way to such a situation one step at a time.

5.2 Extending the HC-HMM Method

As mentioned in the previous chapter, one of the most prominent areas requiring improvement is the HC-HMM's ability to extract meaningful motifs both in numerous quantities and higher quality. To reiterate, the HC-HMM generated over 100 distinct motifs from the produced clusters, such that the motifs were usually only short, contiguous segments where m = 2. This could be due to a great number of reasons, most prevalent possibly being that a given HMM chain clusters with another HMM chain based on only one node without the possibility of introducing of gaps. This implies than between two HMM chains, there can only exist one motif, which is not a correct assumption given that two or more protein sequences can exhibit more than one motif at a given time.

Therefore, one of the possible improvements to the HC-HMM method is to allow for the introduction of gaps. This can be done in a variety of ways, but the proposed method in this work is to create a mutual distance matrix examining the difference, in terms of the distance equations set forth in Chapter 4, of all of the nodes for all of the HMM chains being processed. Those node pairs that exhibit below a given dissimilarity would then be flagged as what would effectively be motif "seeds." These "seeds" could then be grown, from left to right on their respective chains based on a diminishing similarity threshold, such that each subsequently added node onto a given seed would have less stringent similarity requirements.

With this approach, multiple motif "seeds" can appear in any given HMM chain pair, allowing multiple motifs to be extracted from only one HMM pair (and thus, implicitly, the introduction of gaps). This would allow for more numerous motif extractions and, ideally, the motifs extracted, given the fine grained, node-based similarity measures, would be of much higher quality. Granted such success, this new method could completely replace the FGK portion of the FGK-DF model, in so far as protein sequential motif extraction is concerned. This would allow the newly improved FGK-DF model, trained with extremely accurate and high quality motifs that transcend protein family boundaries, to perform even higher quality tertiary structure predictions with, ideally, higher coverage. With increased coverage, the FGK-DF model could be extended to begin predicting global tertiary structure as well as protein folding (known as quaternary structure). With this in hand, the complete three-dimensional model of the protein can be produced, and thus its function elucidated. This, of course, is too far in the distance to adequately discuss with any true accuracy without first ascertaining the effectiveness of the extended HC-HMM method for protein sequential motif extraction.

5.3 Final Remarks

In this work, the problem of determining/predicting local tertiary structure of proteins cheaply, accurately, and quickly was examined, with the proposed solution being the FGK-DF model. The competing models, which ranged from the wet-lab experiments of x-ray crystallography and NMR spectroscopy, to the bioinformatics approaches that included homology-modeling, de novo approaches, and threading approaches, each had their caveats, whether that be the extreme slowness, inherent limitations, or overt complexity. Instead, this work proposed one approach the problem not by explicitly determining the structure or looking for homologues, but rather by using conserved primary sequence motifs that transcend protein family boundaries. This approach allows users of the model to predict proteins with no known homologues, and to predict proteins of any size (feats homology-based modeling and de novo approaches can't currently accomplish). The FGK-DF, even in it's somewhat naïve state, can produce thousands of local tertiary structure predictions in less than an hour, with acceptable accuracy and coverage.

The work extends the FGK-DF model by tackling the limitation that is the model's assumed window size by introducing the HC-HMM method, which utilizes hierarchical clustering to allow for the extraction of motifs of different lengths. The HC-HMM's ability to extract motifs in terms of the number, quality, and length of the resulting motifs was examined, with focus on select motifs extracted by the method. The results and the examined motifs both strongly support utilizing and extending the HC-HMM as a tool for extracting sequential motifs without an assumed window size. Given this, while the FGK-DF model and HC-HMM method are in no way finished or near completion, granted the possibility of dynamic sequence representation as well as much further changes, such as global tertiary structure prediction, the combined strengths of the FGK-DF model and HC-HMM method already hold powerful implications not only for the aforementioned drug research and development, but for structural genomics itself.

Thus, in the end, does this work provide the end all for structural genomics, protein tertiary structure prediction, and protein sequential motif extraction? No. Does it provide exceedingly cheap and effective personalized medicine? No. Does the model and results shown in this work provide justification for further research and development on this model? Yes. And does this work provide the necessary steps towards providing a model that can provide strong implications for structural genomics in terms of tertiary structure prediction that produces novel folds and sequential motif extraction that makes no assumption on the motif size? Yes, yes it does.

REFERENCES

[1] A. Andreeva, “Data growth and its impact on the SCOP database: new developments,” Nucleic Acids Research, vol. 36, pp. 419-425, Nov. 2007.

[2] A. E. Torda, “Perspectives in protein-fold recognition,” Current Opinion in Structural Biology, vol 7, pp. 200-205, 1997.

[3] A.L. Spek, “Structure validation in chemical crystallography,” Acta Crystallographica, Section D, vol. 60, no. 4, pp. 148-155, 2004.

[4] A. S. Nair, “Computational Biology & Bioinformatics: A Gentle Overview,” Communications of the Computer Society of India, pp. 1-13, 2007.

[5] B. Chen et al., “Novel efficient granular computing models for protein sequence motifs and structure information discovery,” International Journal Computational Biology and Drug Design, vol. 2, no. 2, pp. 169-186, 2009.

[6] B. Chen et al., “Protein Sequence Motifs Extraction Using Decision Forest,” Proceedings of International Conference on Bioinformatics & Computational Biology, pp. 96-102, 2011.

[7] C. Chothia and A.M. Lesk, “The relation between the divergence of sequence and structure in proteins,” EMBO Journal, vol. 5, no. 4, pp. 823-826, 1986.

[8] C. Hardin et al., “Ab initio protein structure prediction,” Current Opinion in Structural Biology, vol. 12, pp. 176-181, 2002.

[9] C.J.A. Sigrist et al., "New and continuing developments at PROSITE," Nucleic Acids Research, pp. 1-4, Nov 2012.

[10] C. Sander and R. Schneider, “Database of homology-derived protein structures and the structural meaning of sequence alignment,” Proteins Struct. Funct. Genet., vol. 9, no. 11, pp. 56-68, 1991.

[11] C. Sander and R. Schneider, “Database of similarity derived protein structures and the structure meaning of sequence alignment,” Proteins: Struct. Flunct. Genet., vol. 9, no. 1, pp. 56-68, 1991.

[12] D. Baker et al., “Protein Structure Prediction and Structural Genomics,” Science, vol. 294, no. 5540, pp. 93-96, 2001.

[13] "DREME: Motif discovery in transcription factor ChIP-seq data," Bioinformatics, vol. 27, no. 12, pp. 1653-1659, 2011.

[14] G.C. Kennedy, “Impact of genomics on therapeutic drug development,” Drug Development Research, vol. 41, pp. 112-119, 1997.

[15] G. Karp, Cell and Molecular Biology: Concepts and Experiments, 6th ed., New York: John Wiley & Sons Inc, 2009, pp. 52-66.

[16] G. Wang and R. Dunbrack Jr., “PISCES: a protein sequence culling server,” Bioinformatics, vol. 19. no.12, pp. 1589-1591, 2003.

[17] H.M. Berman, “The Protein Data Bank: a historical perspective,” Acta Crystallographica Section A: Foundations of Crystallography, vol. 64, no. 1, pp. 88-95, 2008.

[18] I. Eidhammer et al., “Pairwise Global Alignment of Sequences,” in Protein Bioinformatics: An Algorithmic Approach to Sequence and Structure Analysis, Chichester, England: John Wiley & Sons Ltd, 2004, ch. 1, sec. 1, pp. 3-23.

[19] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Maryland, USA: Kluwer Academic Publishers Norwell, 1981.

[20] J. Chandonia and S.E. Brenner, The Impact of Structural Genomics: Expectations and Outcomes, California, USA: Lawrence Berkeley National Laboratory, Dec. 2005.

[21] J. K. M Sanders et al., Modern NMR Spectroscopy: A Guide for Chemists, New York: Oxford University Press, 1998.

[22] J.R. Quinlan, “Induction of Decision Trees,” Machine Learning, vol. 1, pp. 81-106, 1986.

[23] L.A. Zadeh, “Fuzzy sets,” Information and Control, vol. 8, no. 3, pp. 338-353, 1965.

[24] L.E. Baum and T. Petrie, “Statistical Inference for Probabilistic Functions of Finite State Markov Chains,” The Annals of Mathematical Statistics, vol. 37, no. 6, pp. 1554-1563, 1966.

[25] L. Kelly et al., "Recognition of remote protein homologies using three-dimensional information to generate a position specific scoring matric in the program 3D-PSSM," Proceedings of RECOMB'1999, pp. 218-225, 1999.

[26] L.R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286, Feb. 1989.

[27] M. Gribskov et al., "Profile Analysis," Methods Enzymol, vol. 183, pp. 146-159.

[28] N. Campbell et al., "The Structure and Function of Macromolecules," in Biology, San Francisco, California: Benjamin Cummings, 1999, ch. 5, sec. 1, pp. 58-82.

[29] N.C.W. Goonesekere et al., "Context-specific amino acid substitution matrices and their use in the detection of protein homologs, Proteins: Structure, Function, and Bioinformatics, vol. 71, pp. 910-919, 2008.

[30] P. Baldi. et al., "Hidden Markov models of biological primary sequence information," Proceedings of Natural Academy of Science, USA, vol. 91, pp. 1059-1063, 1994.

[31] R. Bonneau and D. Baker, "Ab Initio Protein Structure Prediction: Progress and Prospects," Annual Review of Biophyics and Biomolecular Structure, vol. 30, pp. 173-89, 2001.

[32] R.H. Lathrop, "The protein threading problem with sequence amino acid interaction preferences is NP-complete," Protein Engineering, vol. 7, no. 9, pp. 1059-1068, 1994.

[33] R. Hooft et al., "The PDBFINDER database: a summary of PDB, DSSP and HSSP information with added value," CABIOS, vol. 12, no. 6, pp. 525-529, 1996.

[34] S.R. Eddy, "What is dynamic programming?," Nature Biotechnology, vol. 22, pp. 909-910, 2004.

[35] S.Y. Chung and S. Subbiah, "A structural explanation for the twilight zone of protein sequence homology," Structure, vol. 4, pp. 1123-1127, 1996.

[36] T.K. Attwood et al., "The PRINTS database: a fine-grained protein sequence annotation and analysis resource—its status in 2012," Journal of Biological Databases and Curation, vol. 2012, 2012.

[37] T.L. Bailey "MEME SUITE: tools for motif discovery and searching," Nucleic Acids Research, vol. 37, no. 2, pp. 202-208, 2009.

[38] T. Lengauer and R. Zimmer, “Protein structure prediction methods for drug design,” Briefings in Bioinformatics, vol. 1, no. 3, pp. 275-288, Sept. 2000.

[39] T.Y. Lin, “Data mining and machine oriented modeling: a granular computing approach,” Journal of Applied Intelligence, vol. 13, pp. 113-124, 2002.

[40] V. Maiorov et al., "Significance of Root-Mean-Square Deviation in Comparing Three-dimensional Structures of Globular Proteins," Journal of Molecular Biology, vol. 235, pp. 625-634, 1994.

[41] W. Kabsch and C. Sander, “Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features,” Biopolymers, vol. 22, pp. 2577-2637, 1983.

[42] W. Zhong et al., “Clustering support vector machines for protein local structure prediction,” Expert Systems with Applications, vol. 32, pp. 518-526, 2007.

[43] W. Zhong et al., “Improved K-Means Clustering Algorithm for Exploring Local Protein Sequence Motifs Representing Common Structural Property,” Nanobioscience, vol. 4, no. 3, pp. 255-265, Sep. 2005.

[44] Y. Zhang, “Progress and challenges in protein structure prediction,” Current Opinion in Structural Biology, vol. 18, no. 3, pp. 342-348, 2008.