Development of a Computer Program for Prediction of Protein Structure and Function

Tom Wilson

jtwilson@mail.lipscomb.edu

HCI 5903 Capstone

Lipscomb University College of Pharmacy and Health Sciences

ABSTRACT

Determining the three-dimensional structure of proteins remains a computationally intensive and experimentally laborious challenge but comes with its own fair share of rewards. Elucidating a protein's structure and function holds incredible value for drug design and synthesis. This would allow the creation of more specific medications for established therapies and the creation of potential medications for disease states that are not currently treated adequately. It is generally regarded that a protein's sequence determines its structure which in turn leads to its function; however, in some cases, proteins are resistant to structural determination through traditional means, including x-ray crystallography, due to their inherent physical properties. It has been shown that structure and function can be predicted from a given sequence in a more computationally reasonable fashion while maintaining the predictive accuracy of other "brute force" methods of structure determination by utilizing a specific learning and clustering algorithm known as the K-Modes Attribute Clustering Algorithm (K-modes). When employing K-modes programmatically, a specific protein's sequence is compared across multiple organisms for which a sequence has been obtained. Specific locations within these sequences are clustered together based on the probability of their functional similarity, and these clusters can then be examined and interpreted. Work on this project is ongoing with the hope that a utility will be created for researchers to freely use for their proteins of interest.

INTRODUCTION

In its simplest form, this project entails writing software that predicts a protein's structure, and ultimately its function, through algorithmic comparison of a multiple sequence alignment of the same protein from multiple different organisms. Utilizing a machine learning approach as taken by Kirk Durston in his publication on the same topic (Durston, Chiu, Wong, & Li, 2012), the thought is that given enough sequences of the same protein, patterns would emerge that are either dispensable or indispensable to a protein's overall function and subsequent organismal necessity.

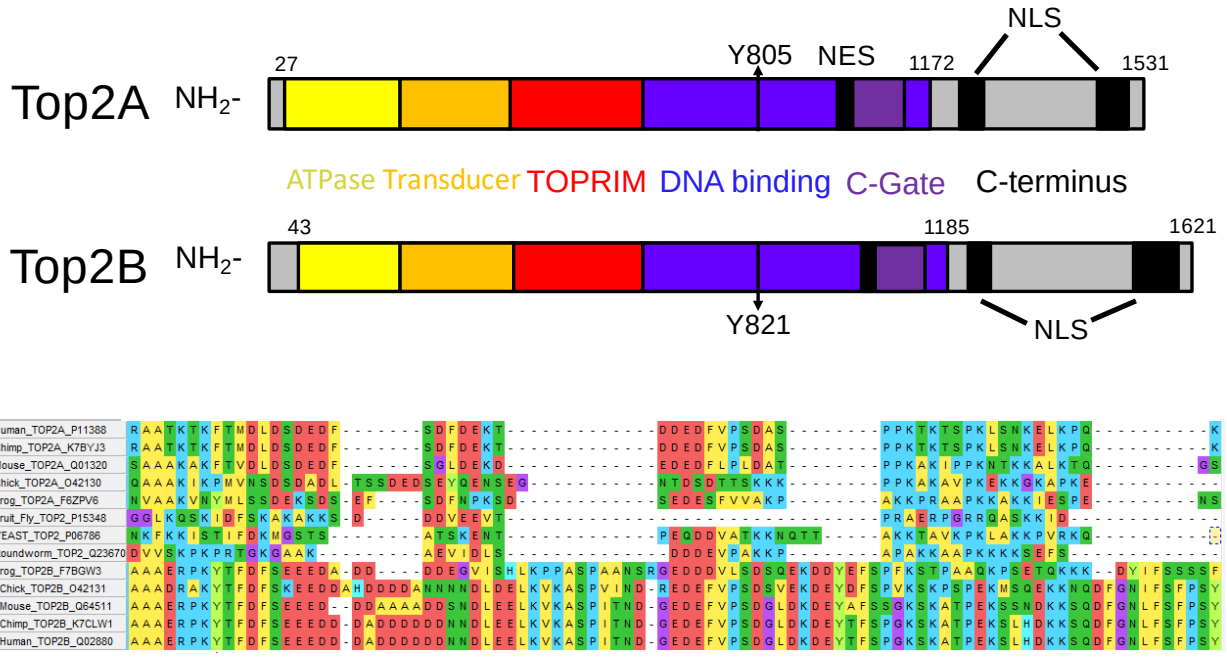
The importance of the scope of this project lies in the utility of proteins as medicinal targets for various therapeutic strategies. In the field's current state, protein sequencing vastly outnumbers crystallographic structural generation as is detailed further in this paper. Having the ability to predict structure and function of a protein simply from its sequence would mean having faster throughput of structural data that could potentially be put to use in disease pathophysiology, medication design, and even personalized medicine delivery (if an individual's proteome were sequenced and structured).

The importance of high throughput protein structuring is paramount to researchers today as medical technology evolves to have the ability to use this type of data. Currently the bottleneck is the process of structural determination which makes it a prime target for optimization.

Inspiration for this project stems from a question currently being asked by the laboratory of Dr. Joe Deweese at Lipscomb University. The Deweese lab studies the topoisomerase enzyme and its role in cancer therapy. Human topoisomerase is present in two isoforms: topoisomerase

IIα and topoisomerase IIβ. These two isoforms have 65.2% commonality in the core region comprised of approximately 1100 amino acids (Deweese et al., 2019) while in the carboxy terminus (C-terminus) region they share 30% commonality across approximately 400 amino acids. This can be seen in the following figure (Deweese et al., 2019):

Figure 1 - domain structures for topoisomerase IIα and topoisomerase IIβ along with a multiple sequence alignment of the C-terminal region from different organisms and isoforms.



A more detailed look at the differences between the isoforms can be seen in the appendix.

To date, the C-terminus has been resistant to x-ray crystallographic techniques for structural determination as it is generally thought to be largely mobile (Deweese et al., 2019). Additionally, the C-terminus is thought to be key to the functional differences seen between the two enzyme isoforms. It has been proposed that having the ability to target one isoform preferentially to the other would allow for improved anticancer efficacy while reducing

translocation events that potentially lead to the formation of secondary leukemias (Gibson, King, Mercer, & Deweese, 2016). If a computer program were to exist that could predict structure and function of the topoisomerase C-termini, work could begin on structure-activity relationship and drug targeting studies.

METHODS OF PROTEIN STRUCTURE PREDICTION AND COMPUTATIONAL COMPLEXITY

Determining the three-dimensional structure of proteins remains a computationally intensive and experimentally laborious challenge. In some cases, proteins are resistant to structural determination through traditional means, including x-ray crystallography, due to their inherent physical properties (Carpenter, Beis, Cameron, & Iwata, 2008; Lacapere, Pebay-Peyroula, Neumann, & Etchebest, 2007). The central dogma of proteomics is that a protein's sequence determines its structure which in turn leads to its function; therefore, structural determination of proteins allows researchers to elucidate otherwise unknown functions. Understanding the structure/function relationship of proteins holds promise for new treatment targets and medical research. The issue at hand is protein sequence generation far surpasses the ability and throughput of traditional means of structural determination. For example, the number of protein sequences in the National Institute of Health (NIH)'s "GenBank" has reached approximately 1 billion and doubles every 18 months as raw sequencing is computationally bound to computer processing power (Breuza et al., 2016; "GenBank and WGS Statistics," ; Wetterstrand, 2018). The number of actual three dimensional protein structures being generated from these sequences is much lower; the Protein Data Bank (PDB), which "has served as the single repository of information about the 3D structures of proteins, nucleic acids, and complex

assemblies" had only 11,000 structures uploaded in the last year (Berman, Henrick, & Nakamura, 2003).

Protein structures are usually determined by x-ray crystallography and, in some cases, nuclear magnetic resonance (NMR) spectroscopy. Ninety percent of the structures housed within the PDB database have been obtained through traditional means of x-ray crystallography (Kendrew et al., 1958). X-ray crystallography and NMR spectroscopy are time-intensive tasks as can be gathered from the discrepancy between available sequences and actual structures. Another method of structure determination falls within the computational simulation domain in lieu of laboratory methods. Computational protein structure prediction (PSP) typically utilizes enumeration or searching strategies with optimization which involve enormous probability spaces (Newman & Hart, 2001; Ngo, Marks, & Karplus, 1994). These methods involve calculating a protein's free energy and the global minimum of that energy as displayed in equation (1) (Ngo et al., 1994):

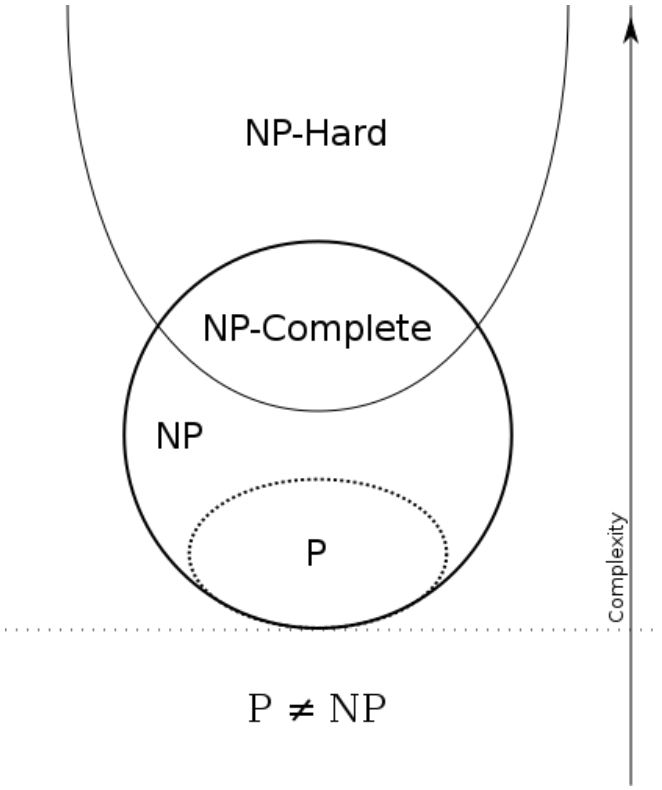
$$(1)\ U = \sum_b K_b^{bond}(l_b - l_b^0)^2 + \sum_a K_a^{angle}(\theta_a - \theta_a^0)^2 + \sum_t K_t^{torsion}(1 - \cos[n_t(\varphi_t - \varphi_t^0)]) + \sum_{i>j} K_{ij}^{nonlocal} f(r_{ij} / r_{ij}^0)$$

where equation (1) is the related optimization algorithm for the current method of PSP (Ngo et al., 1994) and belies the difficulty in computational PSP. PSP has been described by Ngo and Mark as being NP-Hard (Ngo et al., 1994). That is to say solving PSP through the method mentioned above will be as difficult as the most difficult non-deterministic polynomial time (NP) problem.

PRIMER ON P, NP, AND COMPUTATIONAL COMPLEXITY

Before moving further, a primer on P vs NP and complexity may be useful to at least conceptually grasp the computational nightmare that PSP involves and why much work is currently being done to develop new models of prediction not based on the raw processing of a protein's physical and chemical properties.

One of the unsolved fundamental questions of theoretical computer science is centered around P = NP where P is the set of all problems that can be solved in *polynomial time* and NP is the set of all problems that cannot be solved in polynomial time and therefore must be solved in *non-deterministic polynomial time* (Knuth, 1974; Leeuwen, 1998). As a drastic oversimplification, all problems in the set P can be thought of as "easy" while problems in the set NP can be thought of as "difficult". The relationship between P and NP can be seen in the following Euler diagram (Esfahbod, 2007):



As mentioned above, NP-Hardness problems are described as being as complex or hard as the most difficult problem in NP based on the Euler diagram. Another view of NP-Hardness is: "a problem *H* is NP-hard when every problem *L* in NP can be reduced in polynomial time to *H*; that is, assuming a solution for *H* takes 1-unit time, we can use *H*'s solution to solve *L* in polynomial time" (Knuth, 1974; Leeuwen, 1998). Since NP-Hard problems are not solvable in polynomial time (providing the convention $P \neq NP$ stands), they must be solved in a nondeterministic way. In a basic sense, solutions to NP problems are "guessed" by the computer (nondeterminate) and then verified in polynomial time. NP problems can therefore be thought of as easy to check but difficult to solve (Knuth, 1974; Leeuwen, 1998).

Applying these concepts to the field of computational PSP, it should be evident that these simulations become intractable rather quickly (Newman & Hart, 2001; Ngo et al., 1994). The issue at hand is the basis for the Levinthal paradox:

> In theory a protein is expected to require exponential time to fold, given an arbitrary starting configuration, whereas in practice proteins are observe to fold within seconds to minutes, independent of size… [where] exponential-time folding is expected because of the exponential size of the protein's conformational space (Ngo et al., 1994).

As stated by Newman and Hart:

> Exhaustive search of a protein's conformational space is clearly not a feasible algorithmic strategy. The number of possible conformations is exponential in the length of the protein sequence, and powerful computational hardware would not be capable of searching this space for even moderately large proteins (Newman & Hart, 2001).

The thought here is that even though proteins have the potential to fold in an exponentially large number of conformation (based on the protein's length), we only see a relative handful existing in nature.

One of the solutions being explored in this space currently, and the one that this project is based around, is utilizing a learning algorithm to take protein sequences and predict a structure in polynomial time based on probability and the mutual similarity of sequences between different groups of species. This algorithm will take no longer than $O(knp^2t)$ where *k* is the algorithm, *n* is the number of samples in our data set, *p* is the number of attributes in the sequence, and *t* is the number of iterations through which *k* is ran (Au, Chan, Wong, & Wang, 2005; Durston et al., 2012). This notation shall be explained in greater detail below.

K-MODES ATTRIBUTE CLUSTERING ALGORITHM

The aim of this project was the exploration and creation of a computer program that utilizes a learning algorithm in which protein structure and function can be predicted in a more computationally reasonable fashion while maintaining the predictive accuracy of other "brute force" methods of structure determination listed above. Current literature analysis shows that use of the K-Modes Attribute Clustering Algorithm (K-Modes) is being explored in the PSP field (Au et al., 2005; Durston et al., 2012; Gaud & Sharma, 2015; Zhexue & Ng, 1999; Zhou & Chan, 2015). The utility of the K-modes algorithm is multifold:

(1) K-modes is a learning algorithm and is an extension of the widely used K-means algorithm for data clustering (Au et al., 2005)

(2) K-modes is able to cluster categorical data based on their similarity in much the same way that K-means clusters numeric data based on Euclidean distance (Au et al., 2005; A. K. C. Wong & Li, 2008; Zhexue & Ng, 1999)

(3) K-modes is data-driven meaning regardless of the starting conditions, the same protein sequence will yield the same result as long as the number of clusters stays constant between conditions (Durston et al., 2012)

Before moving further, definitions and overall clarifications need to be proposed:

K-Means: K-means clustering is a type of unsupervised learning used when numeric data is unlabeled. K-means will iteratively group the data into *k* number of groups, or clusters, based on their similarity. K-means utilizes the Euclidean distance from the center of a defined cluster to determine group assignment. Once all data are assigned to clusters, the centroids are recalculated using the mean of all data points within the cluster (Trevino, 2016).

K-Modes Attribute Clustering Algorithm: K-modes clustering is an extension of K-means used for grouping categorical data. Instead of using the mode of all the values in a centroid's cluster to recalculate the centroids, the mode of the group is used as well as a frequency-based method for updating the mode of each centroid (Huang, 1998).

The value for *k* is integral to the algorithm being run as it determines the number of clusters generated. Depending on the application of the K-modes algorithm, different methods are used to determine the optimal number of clusters that should be generated. For the purposes of this use of the algorithm, *k* will start at a value of $n - 1$ where *n* is the number of attributes (aligned columns) and the algorithm will stop when $k = 2$ leaving two clusters. The clusters generated

during each iteration of the algorithm will then be visualized using a cluster tree graph to promote pattern identification and identification of interdependency among clusters.

MULTIPLE SEQUENCE ALIGNMENTS

The problem being addressed can be distilled down into a simple sequence: obtain multiple-alignment data for a protein/protein family of interest, calculate interdependency of the sequences in the multiple-alignment set, cluster the sequences based upon their comparable interdependence, and output the results.

Multiple sequence alignments are necessary for probabilistic purposes. A multiple sequence alignment is exactly as it sounds: multiple sequences (in this case, protein sequences) are aligned for comparison, taking into consideration substitution, insertion, and deletion events within the coding DNA sequence. The method employed in this project is the alignment of the same protein from multiple different species. According to Durston, et al., the minimum number of non-redundant sequences necessary for clustering is five; however, the more sequences aligned and clustered, the better the output resolution will be (Durston et al., 2012).

Multiple sequence alignments for proteins and protein families are relatively easy to obtain. Many open-source solutions exist if one were to perform the alignments themselves; however, a number of websites have these alignments available for download (Altschul, Gish, Miller, Myers, & Lipman, 1990). Refer to the appendix for a list of resources used to create multiple sequence alignments. Many of these tools have been aggregated on the site https://www.ebi.ac.uk/Tools/msa/. The website http://pfam.xfam.org/ was used in the publication by Durston et al. to obtain multiple sequence alignments for the ubiquitin protein family. The alignment is stored in a text file with one protein sequence per line. An example of a multiple

sequence alignment is shown as example in Figure 2 (Laboratory, 2019) where the left shows the species identifier and the right shows the sequence. Each letter represents an amino acid while a period represents a gap which is formed during the alignment process due to differences in genetic conservation between species.

Figure 2 – Truncated Multiple Sequence Alignment of Ubiquitin

```
1  Q9VEC8_DROME/3-75          ITIKV..LKGK.....DCTI.EVAPTSTILEVKHQIEAEL.Q...ISATNQ.KLLLLGRPL..NNEQTIASYPNIKEG.TKLNLVVIKP
2  UBL4A_MOUSE/3-74           LTVKA..LQGR.....ECSL.QVAEDELVSTLKHLVSDKL.N...VPVRQQ.RLLFKGKAL..ADEKRLSDY.NIGPN.SKLNLVVKPL
3  R7SL1_ARATH/3-74           VYIDT..ETGS.....SFSI.TIDFGETVLEIKEKIEKSQ.G...IPVSKQ.ILYLDGKAL..EDDLHKIDY.MILFE.SRLLLRISPD
4  Q9ZQZ4_ARATH/3-74          MTVEN..ESGS.....TFSI.DIGLQDTVLTFKRKIEMTQ.R...IPVSRQ.TIFFQGKLL..EDHLDIFEW.DILQN.PLLHLSISPD
5  Q9SYF2_ARATH/71-142        IFIKT..LTGR.....TNTY.EVKGSDTIRELKAKHEEKE.G...IPVEQQ.RLIFQGRVL..EDSKAISDY.NIKHE.STLHITLHQC
6  RUB1_YEAST/3-74            VKVKT..LTGK.....EISV.ELKESDLVYHIKELLEEKE.G...IPPSQQ.RLIFQGKQI..DDKLTVTDA.HLVEG.MQLHLVLTLR
7  NEDD8_HUMAN/3-74           IKVKT..LTGK.....EIEI.DIEPTDKVERIKERVEEKE.G...IPPQQQ.RLIYSGKQM..NDEKTAADY.KILGG.SVLHLVLALR
8  RUB3_ARATH/3-74            IKVKT..LTEK.....QIDI.EIELTDTIERIKERIEEKE.G...IPPVHQ.RIVYTGKQL..ADDLTAKHY.NLERG.SVLHLVLALR
9  B0BLT8_XENTR/30-101        LFIET..LTGT.....CFEL.RVSPYETVASVKSKIQRLE.G...IPVAQQ.HLIWNNMEL..EDECSLSDY.NISEG.CTLKMVLAMR
10 Q9TZ84_CAEEL/185-255       FNVI...LHGS.....RIPM.ELDSLDTIYTVGMALLKHG.G...LALHRQ.RLMLNDKEL..QYNQTLEEA.GIKDG.SVIKQHTDEK
11 P4KG4_ARATH/37-108         IYLT...LPGS.....VIPM.RVLESDSIESVKLRIQSYR.G...FVVRNQ.KLVFGGREL.ARSNSNMRDY.GVSEG.NILHLVLKLS
12 P4KG2_ARATH/37-108         VFLS...VSGS.....TMPM.LILESDSIAEVKLRIQTCN.G...FRVRRQ.KLVFSGREL.ARNASRVKDY.GVTGG.SVLHLVLKLY
13 UBQ8_ARATH/81-152          IFVQT..LTGK.....TITL.EVKSSDTIDNVKAKIQDKE.G...ILPRQQ.RLIFAGKQL..EDGRTLADY.NIQKE.STLHLVLRLC
14 UBQ8_ARATH/471-549         IFVKTFSFSGETPTCKTITL.EVESSDTIDNVKVKIQHKV.G...IPLDRQ.RLIFGGRVL..VGSRTLLDY.NIQKG.STIHQLFLQR
15 UBQ8_ARATH/240-316         IFVKN..LPYNSFTGENFIL.EVESSDTIDNVKAKLQDKE.R...IPMDLH.RLIFAGKPL..EGGRTLAHY.NIQKG.STLYLVTRFR
16 UBQ8_ARATH/157-235         IFVST..FSGKNFTSDTLTL.KVESSDTIENVKAKIQDRE.G...LRPDHQ.RLIFHGEELFTEDNRTLADY.GIRNR.STLCLALRLR
17 UBQ8_ARATH/395-466         IFVKL..FGGK.....IITL.EVLSSDTIKSVKAKIQDKV.G...SPPDQQ.ILLFRGGQL..QDGRTLGDY.NIRNE.STLHLFFHIR
18 R5RDZ3_9PROT/23-94         IFVKT..LTGK.....HITL.EVEPTDRIEDVKAKIQDKE.G...IPPDQQ.RLIFAGKQL..EDGNTLQDY.SIQKD.STLHLVLRLR
19 UPL5_ARATH/97-169          IFVRM..MSGG.....KTIVIHAEKYDTVEKLHQRIEWKT.K...IPALEQ.RVIYKGKQL..QRENSLTYY.SIEQD.ASLQLVARMQ
20 P91050_CAEEL/144-214       LCIAV.SMPGR.....LFSI.GANKMESVEQLKMKIECQT.G...IPRTKF.WLRLHGKPL..YDDKKLADY...KWD.TSVELLVRAS
21 Q9SV28_ARATH/11-79         FFVRL..LDGK.....SLTLSFSSPLAYGEQIKQRIFEQT.K...IPTHLQ.RLISGGYQI..SDGSAISQP.....D.ATVNLVLSLR
22 UHRF1_MOUSE/3-76           IQVRT..MDGK....ETHTVNSLSRLTKVQELRKKIEEVF.H...VEPQLQ.RLFYRGKQM..EDGHTLFDY.DVRLN.DTIQLLVRQS
23 OASL_HUMAN/436-507         VFVKN..PDGG.....SYAY.AINPNSFILGLKQQIEDQQ.G...LPKKQQ.QLEFQGQVL..QDWLGLGIY.GIQDS.DTLILSKKKG
24 YKA4_CAEEL/343-415         ILIKLKFMNDT.............EKNTYASLEDTVAKFKVDHFTNLANQVIRLIYQGQLL.REDHRTLEEY.GLQPG.SIVHCHISTT
25 H9L0X6_CHICK/393-463       VLVK...DSNK.....TTVY.TVRPTDTVKQLKQQIYACQ.H...VPVEQQ.RLTYETKEL..ENHHTLEHY.HVQPR.STIYLLLRLR
```

METHODOLOGY

Once the multiple sequence alignment data has been obtained, it needs to be parsed through and manipulated for prediction calculations. The structure of the multiple sequence alignment data lends itself well to array manipulation by any standard programming language available today where an array is defined as a comma-separated list of values (integers, strings, objects, etc.) enclosed by brackets. This multiple sequence alignment is then viewed in terms of the “attributes” it contains. An attribute in this case is a column of amino acids within the aligned sequence array. An example of an attribute can be seen in Figure 3.

Figure 3 – An attribute within this multiple sequence alignment is highlighted

```
data = [
    ['C', 'A', 'R', 'C', 'A', 'W', 'A', 'A'],
    ['C', 'G', 'K', 'C', 'G', 'Y', 'G', 'G'],
    ['C', 'N', 'M', 'C', 'N', 'F', 'N', 'N'],
    ['C', 'D', 'I', 'C', 'D', 'V', 'D', 'D'],
    ['C', 'A', 'L', 'C', 'A', 'H', 'A', 'A'],
    ['C', 'G', 'R', 'C', 'G', 'Q', 'G', 'G'],
    ['C', 'N', 'K', 'C', 'N', 'E', 'N', 'N'],
    ['T', 'D', 'M', 'T', 'D', 'P', 'D', 'D'],
    ['T', 'A', 'I', 'T', 'A', 'W', 'A', 'A'],
    ['T', 'G', 'L', 'T', 'G', 'Y', 'G', 'G'],
    ['T', 'N', 'R', 'T', 'N', 'F', 'N', 'N'],
    ['T', 'D', 'K', 'T', 'D', 'V', 'D', 'D'],
    ['S', 'A', 'M', 'S', 'A', 'H', 'A', 'A'],
    ['S', 'G', 'I', 'S', 'G', 'Q', 'G', 'G'],
    ['S', 'N', 'L', 'S', 'N', 'E', 'N', 'N']
]
```

Each column is an attribute, and these attributes are being compared to one another for clustering. Each row is the linear protein sequence from a single organism.

Historically clustering algorithms like K-means and K-modes, in its earlier uses, would first need to calculate the similarity of the attributes in order to then assign them to one cluster or another. That assignment would be based either on the Euclidean distance from the nearest centroid (K-means) or on the value of the mode of centroids (K-modes). This approach is sufficient for categorization of most data, numeric or categorical; however, it will not yield correct categorization of clusters based on interdependence of the attributes. “Clustering algorithms that group according to similarity between attributes are not appropriate for clustering in terms of interdependency, since dissimilar attributes may be interdependent. (For example, two aligned sites may be composed of dissimilar amino acids, yet be interdependent)” (Durston et al., 2012).

Similarity calculations in the literature include Euclidean distance as previously discussed as well as Pearson's correlation coefficient. Pearson's correlation coefficient is similar to Euclidean distance as a measure of centroid distance calculations for clustering. Pearson's correlation coefficient was proposed as a replacement for Euclidean distance for interdependence calculations; however, it does not handle outliers well in calculations and may relate dissimilar genes when they are, in fact, independent (Au et al., 2005).

In place of similarity calculations, interdependence calculations have been employed in the literature. The idea of Mutual Information (MI) has been proposed (Durston et al., 2012; Zhou & Chan, 2015) and used (Durston et al., 2012) to categorize genetic information and protein sequence alignments. Mutual information, when normalized by the joint entropy of the attributes being compared, works well for discrete data. Another option in place of Normalized Mutual Information (NMI) is the Maximal Information Coefficient (Zhou & Chan, 2015) which is more flexible toward the data it is processing though the outcomes are largely the same once data has been standardized between the two. This project utilized NMI for clustering because a greater amount of literature is available for NMI.

MUTUAL INFORMATION

As stated earlier, mutual information (more specifically normalized mutual information) is being used to determine the interdependency relationship between attributes in a protein's sequence. That is to say calculations are being performed on attributes in a list of sequences in order to determine how related and dependent they are on one another. Adding more definitions at this point may be of use going forward:

Mutual Information (MI): MI comes from the fields of probability and information theory and is the measure of the amount of dependence two random variables have on one another. Values for MI range from 0 to $+\infty$. This range is a caveat for the use of MI as comparison of numbers reaching infinity is unintuitive and difficult to interpret (Lange & Grubmuller, 2006). MI is calculated as shown in Equation (2) (Durston et al., 2012; A. K. C. Wong & Li, 2008; Andrew K. C. Wong, Liu, & Wang, 1976):

$$(2):\ I(X_i, X_j) = \sum_{x \in X_i} \sum_{y \in X_j} p(x,y) log(\frac{p(x,y)}{p(x)p(y)})$$

As can be seen by Equation (2), a joint probability = 1 would lead to a division by zero in the logarithmic statement. Thus, according to L'Hôpital's rule in differential calculus, $+\infty$ would be derived based on the divide-by-zero indeterminant. This is addressed with normalization; however, the only way to achieve a joint probability = 1 is by having perfect conservation between the sequences of different organisms which is rarely, if ever, seen in nature.

Normalized Mutual Information (NMI): NMI is the normalization of MI to a range from 0 to 1 inclusive. An NMI score of 1 entails complete dependence of one variable on another while an NMI score of 0 entails complete independence of the variables being compared. NMI is calculated by normalizing MI by the joint entropy of the compared variables. This is also known as Shannon Entropy and can be seen in Equation (3) (Durston et al., 2012; A. K. C. Wong & Li, 2008; Andrew K. C. Wong et al., 1976):

$$(3):\ H(X_i, X_j) = -\sum_{x \in X_i} \sum_{y \in X_j} p(x,y) log(p(x,y))$$

NMI, therefore, can be defined as seen in Equation (4) (Durston et al., 2012):

$$(4): R_{ij} = \frac{I(X_i,X_j)}{H(X_i,X_j)}$$

where R_{ij} is known as the interdependency redundancy measure (Durston et al., 2012; A. K. C. Wong & Li, 2008; Andrew K. C. Wong et al., 1976).

"To estimate the overall significant dependence of an attribute Xi with other attributes in a cluster of sites, the sum of normalized interdependency redundancy, *SR(i)*, of X_i with other attributes in the data array is calculated…" (Durston et al., 2012). *SR(i)* can be evaluated by Equation (5):

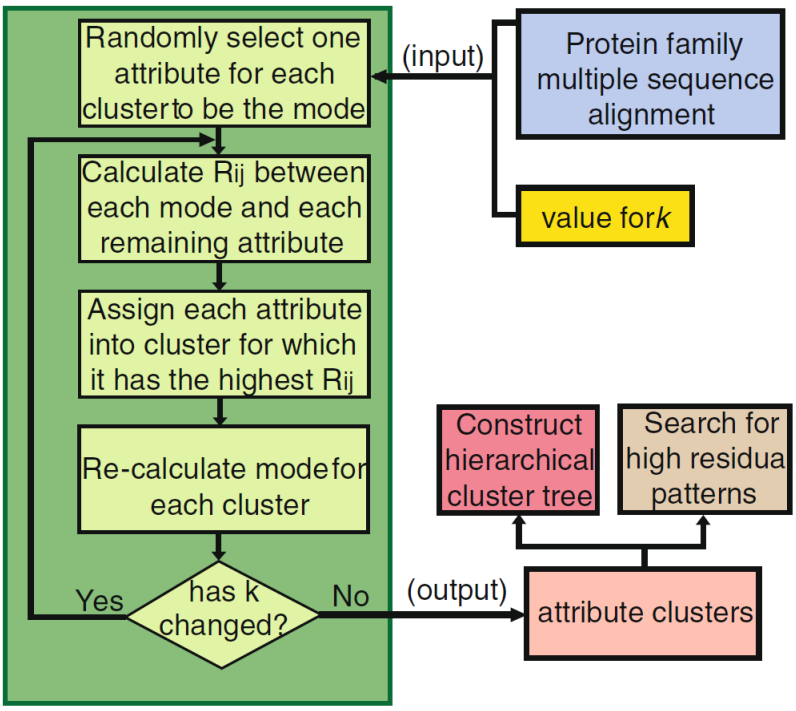
$$(5): SR(i) = \sum_{(i,j)\in N*} R_{i,j}$$

where *N** is the set of *(i, j)* attribute pairs (Durston et al., 2012; A. K. C. Wong & Li, 2008; Andrew K. C. Wong et al., 1976; Zhou & Chan, 2015). The mode (what would be the centroid in K-means) is calculated by dividing *SR(i)* by *N** to reach the average normalized interdependency redundancy value of that cluster (Durston et al., 2012). This mode is used for clustering with the K-modes algorithm and is integral to the machine learning process.

ALGORITHMIC APPROACH

Now that a foundation has been laid for understanding the scope and intricacies of the project, an algorithmic approach can be devised to run through the steps and calculations mentioned earlier. This project was created using Python version 3.7, Scikit-Learn version 0.20.3, and NumPy version 1.16.2 (Pedregosa et al., 2013). Durston, et al. created a figure displaying the K-modes algorithm as can be seen below (Durston et al., 2012):

Figure 4 – Overview of algorithm for attribute clustering and pattern discovery



Once a multiple sequence alignment has been input into the program, a number of steps take place in a loop-wise fashion until the prescribed value for *k* has been reached. In this case, the algorithm will run for a number of iterations from $k - 1$ to $k = 2$. The output of each iteration is saved to a text file and called upon for the next iteration. Once $k = 2$ has been reached, the file containing output from each iteration is used to construct a cluster tree graph depicting the relationship between the clusters as they form. The formation and explanation of the cluster tree will be covered in more depth in the OUTPUT section of this paper.

As mentioned earlier, the time complexity of this algorithm is no longer than $O(knp^2t)$ (Au et al., 2005; Durston et al., 2012) where *k* is the algorithm, *n* is the number of samples in our data set, *p* is the number of attributes in the sequence, and *t* is the number of iterations through which *k* is ran. Given this time complexity, the algorithm as it has been presented should run on any modern-day computer given decent specifications. If larger proteins or more samples are

required for study, time and space requirements will increase. In that situation computational clusters would be the preferred method of evaluation, if available, as run time would be lower depending on the machine's specifications.

At this point it may be beneficial to walk through a test-case multiple sequence alignment to solidify the concepts and algorithm explained above:

1. Input multiple sequence alignment and compute the value of *k*.

Figure 5 – hard-coded array containing a test-case multiple sequence alignment

```
data1 = np.array([
    ['C', 'A', 'R', 'C', 'A', 'W', 'A', 'A'],
    ['C', 'G', 'K', 'C', 'G', 'Y', 'G', 'G'],
    ['C', 'N', 'M', 'C', 'N', 'F', 'N', 'N'],
    ['C', 'D', 'I', 'C', 'D', 'V', 'D', 'D'],
    ['C', 'A', 'L', 'C', 'A', 'H', 'A', 'A'],
    ['C', 'G', 'R', 'C', 'G', 'Q', 'G', 'G'],
    ['C', 'N', 'K', 'C', 'N', 'E', 'N', 'N'],
    ['T', 'D', 'M', 'T', 'D', 'P', 'D', 'D'],
    ['T', 'A', 'I', 'T', 'A', 'W', 'A', 'A'],
    ['T', 'G', 'L', 'T', 'G', 'Y', 'G', 'G'],
    ['T', 'N', 'R', 'T', 'N', 'F', 'N', 'N'],
    ['T', 'D', 'K', 'T', 'D', 'V', 'D', 'D'],
    ['S', 'A', 'M', 'S', 'A', 'H', 'A', 'A'],
    ['S', 'G', 'I', 'S', 'G', 'Q', 'G', 'G'],
    ['S', 'N', 'L', 'S', 'N', 'E', 'N', 'N']
])
```

Figure 6 – Computing the length of the data array to assign to k

```
num_rows = len(data)
num_cols = len(data[0])
```

2. Select a random attribute to begin, calculating the interdependency redundancy (Equation (4)) of the random attribute to all other attributes.

Figure 7 – Selecting a random attribute with NumPy's built-in random generator function

```
# Random attribute selection for first pass
rand = np.random.randint(0, num_cols)
```

3. Find the attribute that has the highest interdependency redundancy value when compared with the randomly chosen attribute in step 2 and store the subsequent value and attribute.

Figure 8 – Looping through the data array, calculating the interdependency redundancy of the random attribute and all other attributes

```
# First, random clustering
for i in range(num_cols):
    if(rand != i):
        # A[:,i] returns column at index i of array A for NumPy arrays
        rii = nmis(data[:, rand], data[:, i])
```

```
        RAND_OUTPUT_DICT[rand, i] = rii
        if(rii > max_R):
            max_R = rii
            # Index of the attribute with the highest Rii score
            starting_attribute = i
```

4. Calculate the normalized interdependency redundancy (Equation (5)) as well as the average normalized interdependency redundancy of each cluster. Store the calculated values in a text file.

Figure 9 – Calculate normalized interdependency redundancy (NIR) and average normalized interdependency redundancy (ANIR)

```
for cluster in clusters:
    NIR += rii[cluster]

ANIR = NIR / len(clusters)
```

5. Repeat steps 2 through 5 without selecting a random attribute. Continue iterative calculations and value output until only two clusters remain.

Figure 10 – Perform clustering over all attributes in the data array after the random clustering of step 2

```
# Taking output of random clustering to form a true clustering
max_R = 0
for i in range(num_cols):
    if(starting_attribute != i):
        rii = nmis(data[:, starting_attribute], data[:, i])
```

```
        FIRST_RUN_DICT[starting_attribute, i] = rii
        if(rii > max_R):
            max_R = rii
```

OUTPUT

The main expected output of this program includes a standardized text file and a cluster tree graph created from the output file.

Figure 11 – Example output text file from test run

```
RUN = 1
K = 9
Clusters: (1,3);
SR(1,3) = 0.9;
SR(mode(1,3)) = 0.9;
---
RUN = 2
K = 8
Clusters: (1,3); (5,6);
SR(1,3) = 0.9; SR(5,6) = 0.899;
SR(mode(1,3) = 0.9; SR(mode(5,6)) = 0.899;
---
```

Given the output file's standard format, it is trivial to programmatically extract data from that file for use in other iterations (i.e. data from run one informs data from run two) and the creation of data visualizations such as the cluster tree. The following figures show cluster trees for the ubiquitin and transthyretin proteins (Durston et al., 2012):

Figure 12 – Cluster tree for ubiquitin

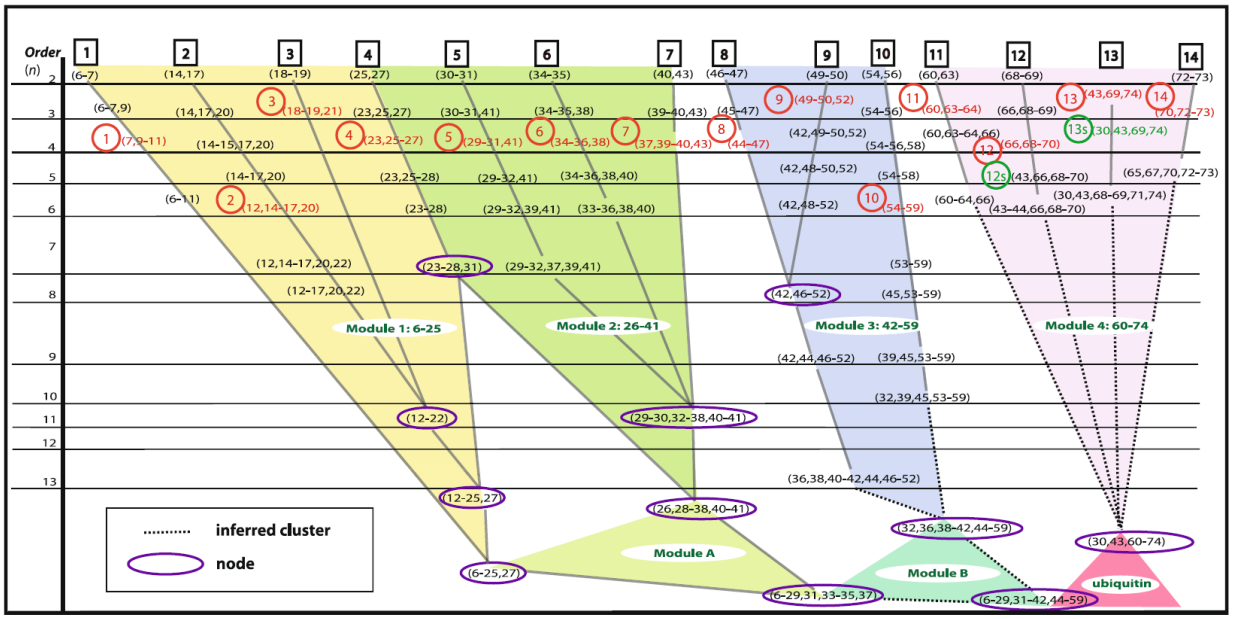
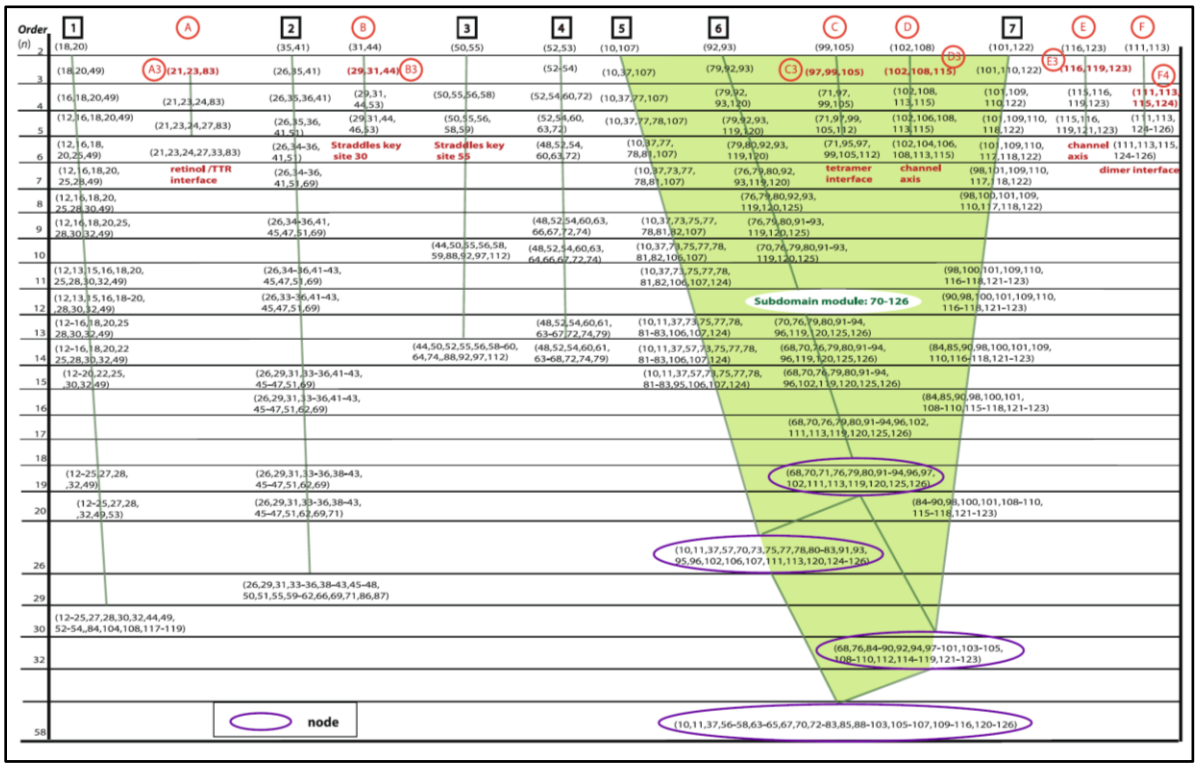

Order (n)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Module 1: 6-25
Module 2: 26-41
Module 3: 42-59
Module 4: 60-74
Module A
Module B
ubiquitin
inferred cluster
node

Figure 13 – Cluster tree for transthyretin



DISCUSSION

Due to the complex nature of the relationships and calculations at hand, verification and repeatability are of utmost importance, especially if this project is to become a useful tool for researchers in the future. I hesitate to overstate the impact of the implications of this project's success; however, I do believe that at the very least a tool with the capability of independently predicting a protein's structure based solely on its sequence will be useful for enzymologists, crystallographers, and drug designers to name a few. I wish to reiterate the importance of the scope of this project lies in the utility of proteins as medicinal targets for various therapeutic strategies. Having the ability to predict structure and function of a protein simply from its

sequence would mean having faster throughput of structural data that could potentially be put to use in disease pathophysiology and medication design, to name a few.

In its current state, this project needs more time to mature and be validated. As it stands, the focus was replication of proprietary probability-based software and subsequent modification to fit into a new domain, namely biochemistry. Next steps should focus on validating this algorithm against protein sequences with known and verified crystallographic structures. Successful structural prediction has occurred for ubiquitin and transthyretin (Durston et al., 2012); however, those proteins are relatively simple when compared to larger enzymes such as topoisomerase.

Scalability of this program is also an area that will require time and focus. As mentioned earlier, protein sequence and probability space grows exponentially with the length of the sequence. That leads to an intractable programming runtime if we were to simply add up all the values for angles, bonds, and torsion. While that exact situation is not an issue for this project, finding a balance in the number of organisms compared coupled with efficient programmatic methods is of the utmost importance if the desire is for this software to run on the laptops of researchers anywhere.

The benefit and personal utility of creating this project is almost unmatched in terms of other endeavors that I have undertaken. I was exposed to a foreign field of which I had only a cursory understanding before beginning. Information theory is both fascinating and head-spinning but seeing its applications to “wet” science with potential carryover into medical practice was phenomenal. I was able to meet with and learn from experts in many different fields: engineers, information scientists, computer scientist, and bioinformaticists; all of which shared invaluable knowledge and advice for this project and goals beyond.

In my opinion, diving head first into a field in which I knew practically nothing was the most beneficial quality of this entire process. I "learned how to learn", so to speak. I was able to take and disseminate information of which just a few months ago I did not have the capability to understand. The exposure and practice of diving into an unfamiliar field and learning it will prove to be one of the most useful skills I can take away from this as my career develops over the years.

I also was able to more deeply incorporate time management and the skill of goal-setting. Setting realistic, attainable goals is a practiced art that until recently I was not well-versed in. Developing the ability to meet personal and internal deadlines for time and project management is still an ongoing process personally; however, I have improved vastly throughout the time spent with this project.

This project is an ongoing collaboration within the Deweese lab and as such has not reached full fruition. Much progress has been made in creating a program to undertake different probabilistic calculations and comparisons; however, replication of the Durston, et al. paper has not been fully achieved. Dr. Kirk Durston is a collaborator and has been a frequent contact during this process. He has been tremendous help in validating calculations and results with one caveat: the program that he used was proprietary and he was not allowed to see any of the code or manipulate it in any way. Progress and goals were initially set with the expectation that Durston could provide more support on development of the logic and actual program which would allow for more time to develop an interface and methods of data and cluster interpretation. This not being the case, much more time has been spent on recreating the proprietary software used by Durston for his paper. The algorithm and software have largely been finished at this stage along with verification of the performed calculations. What remains going forward is

generation of cluster tree graphs from the data output and developing methods of interpretation. Once the rest of the "core" future actions have been completed, I would like to create an interface for other researchers to use that would allow them to upload or reference a protein sequence or multiple sequence alignment. Once the sequences have been provided the program would run and output the data text file along with the cluster tree graph and interpretation for them to download and use in their own research.

Work on this project will continue if only for internal use in the Deweese lab as characterization of the C-terminus of human topoisomerase II continues.

CONCLUSIONS

Determining the three-dimensional structure of proteins remains a computationally intensive and experimentally laborious challenge. Typical methods involve x-ray crystallography for structural determination and multiple laboratory enzymatic assays for functional prediction and determination. Unlocking these attributes of different proteins holds promise as being medically significant as treatment strategies can be developed based on what is uncovered about certain proteins. The current issue in the growth of this field and approach is the slow throughput when transitioning from a protein's sequence to its structure and its function. There may, however, be a computational method of structure and function determination that can outperform the more traditional methods listed above while still maintaining a promising level of accuracy. Machine learning is one such computational method that is currently being explored.

Machine learning has immense potential for use in the biological and medical sciences, especially for process optimization and probabilistic prediction. I set out to utilize a machine learning approach to assist in the prediction of a protein's structure and subsequent function

based solely on the comparison of its sequence across its occurrence in several organisms. The K-modes clustering algorithm was used to evaluate and cluster attributes in a protein multiple sequence alignment whose relationships could then be visualized through the use of a cluster tree graph.

Inspiration for this project came from a question that Dr. Joe Deweese is attempting to answer regarding the two isoforms of human topoisomerase II. This enzyme has distinct functional regions that have been thoroughly crystallized and characterized experimentally; however, each isoform has a unique C-terminus that has so far been resistant to crystallization attempts due, in part, to the thought that the C-terminus is large and relatively unstructured. Up until this point research in the topoisomerase field has carried on without the need to know exactly what the C-terminus' structure and function are. The puzzling piece revolves around the difference in the function and roles each of the isoforms plays. As mentioned above, throughout the main enzymatic core there is a high level of sequence conservation. The real difference is seen when comparing the C-termini of the isoforms. This suggests that the C-terminus is required for specificity and functional differentiation. A more in-depth review of work and thoughts behind C-terminus functional directing can been seen in the appendix under the *Codon differences between human topoisomerase II isoforms* section. Having the ability to analyze the sequences of the two isoforms and predict a structure and potential function of each C-terminus would be a tremendous step forward in the research and understanding of the roles these enzymes play and how they could be exploited in medical therapies.

Creation of a tool with the capability of independently predicting a protein's structure and function based solely on its sequence, in my opinion, will be a large step forward in enzymology

and drug design. Once this has been realized, new medical therapies can be designed that, until now, have been impossible.

APPENDIX

Multiple Sequence Alignment Resources

1. The European Bioinformatics Institute has aggregated many of the following tools
    a. https://www.ebi.ac.uk/Tools/msa/
2. Software
    a. Clustal
        i. http://www.clustal.org
        ii. Developers include Des Higgins; Fabian Sievers; David Dineen; Andreas Wilm (all at the Conway Institute, UCD)
    b. T-Coffee
        i. http://www.tcoffee.org
        ii. Developers include Cédric Notredame, Centro de Regulacio Genomica (CRG) – Barcelona
    c. MAFFT
        i. http://www.mafft.cbrc.jp
        ii. Developers include Kazutaka Katoh
    d. MUSCLE
        i. http://www.drive5.com/muscle/
        ii. Developers include drive5
    e. JALVIEW
        i. http://www.jalview.org
    f. DIALING-TX
        i. http://www.dialing-tx.gobics.de

ii. Developers include Amarendran R. Subramanian

g. FSA (Fast Statistical Alignment)

i. http://fsa.sourceforge.net

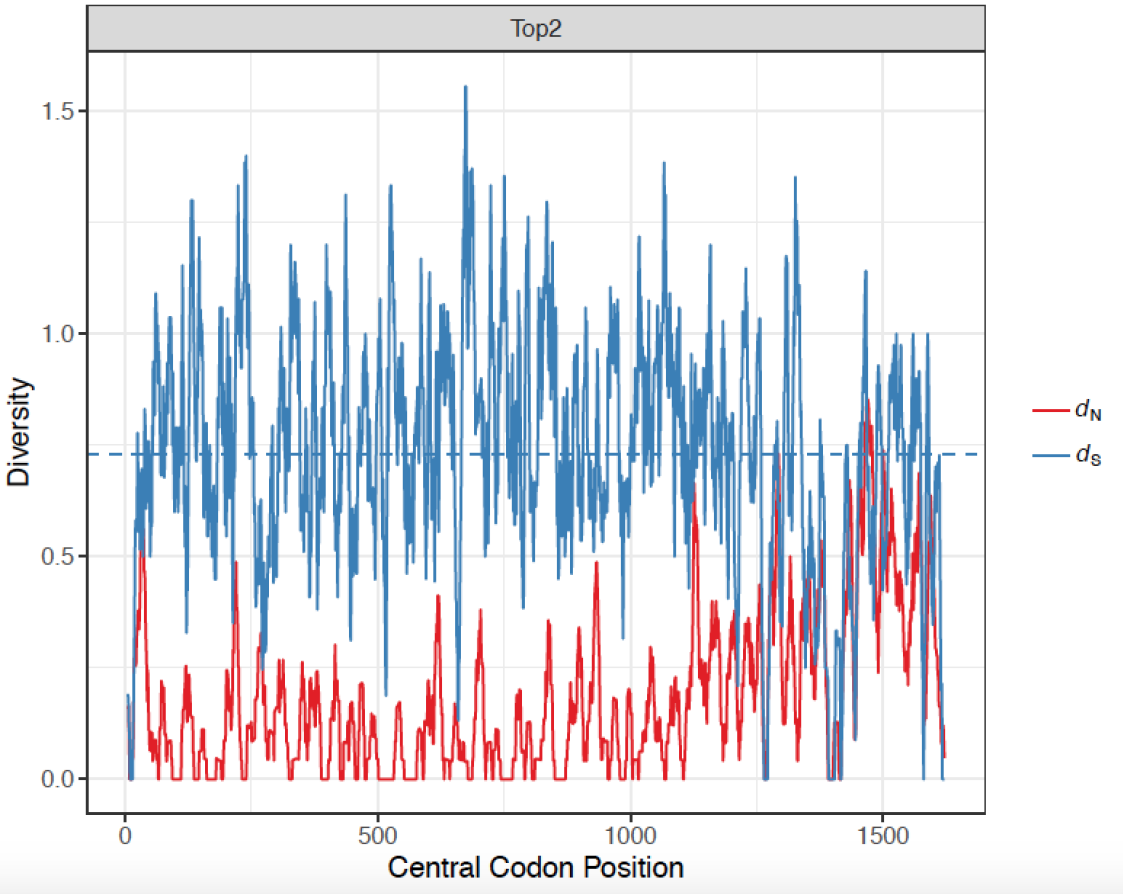
ii. Can align thousands of sequences

h. Pfam

i. http://pfam.xfam.org

ii. doi: 10.1093/nar/gky995

Codon differences between human topoisomerase II isoforms (Deweese et al., 2019)



As can be seen in Figure 1 at the beginning of this paper, the C-terminus of each isoform begins roughly around the 1200 amino acid position. The graph above plots diversity arising from synonymous and nonsynonymous codon sequences as a function of sequence position. Most changes in actual codon sequences up until the C-terminus region do not lead to a functional change in amino acid coding (i.e. multiple codons can code for the same amino acid). However, when the C-terminus region begins, a large amount of nonsynonymous coding starts to occur leading to functional and expressional differences between the two isoforms.

> The relative rates of synonymous (d_S) and nonsynonymous (d_N) DNA substitutions between the isoforms suggests that these sequences are under purifying selection for functional constraint ($d_N/d_S = 0.25$; $P < 0.0001$, *Z* test, 1,000 bootstrap replicates). The 434 C-terminal codons, however, had a relatively high d_N/d_S ratio of 0.59 ($P < 0.0001$), reflecting elevated amino acid diversity (Deweese et al., 2019).

It is hypothesized by the Deweese laboratory that the C-termini control the specific function of each isoform. Topoisomerase IIα and topoisomerase IIβ both maintain DNA topology within the cell during the cell cycle, and the method by which they maintain topology is by reducing torsional strain via DNA strand cleavage, strand passage, and ligation. The difference lies in where the enzymes focus. This has been shown by Linka, et al. through C-terminus exchange between the two isoforms (Linka et al., 2007). In an over-simplified explanation, if a β C-terminus is placed on a topoisomerase IIα protein, it assumes more β-like functions and roles. The same is true if a topoisomerase IIα C-terminus is placed on a topoisomerase IIβ protein. If definitive structural and functional determination could take place, it may allow for drug targeting of one isoform over the other which would greatly reduce adverse drug events from

cross-enzyme specificity.

REFERENCES

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology, 215*(3), 403-410. doi:10.1016/s0022-2836(05)80360-2

Au, W. H., Chan, K. C., Wong, A. K., & Wang, Y. (2005). Attribute clustering for grouping, selection, and classification of gene expression data. *IEEE/ACM Trans Comput Biol Bioinform, 2*(2), 83-101. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/17044174. doi:10.1109/TCBB.2005.17

Berman, H., Henrick, K., & Nakamura, H. (2003). Announcing the worldwide Protein Data Bank. *Nat Struct Biol, 10*(12), 980. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/14634627. doi:10.1038/nsb1203-980

Breuza, L., Poux, S., Estreicher, A., Famiglietti, M. L., Magrane, M., Tognolli, M., . . . UniProt, C. (2016). The UniProtKB guide to the human proteome. *Database (Oxford), 2016*. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/26896845. doi:10.1093/database/bav120

Carpenter, E. P., Beis, K., Cameron, A. D., & Iwata, S. (2008). Overcoming the challenges of membrane protein crystallography. *Curr Opin Struct Biol, 18*(5), 581-586. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/18674618

https://www.sciencedirect.com/science/article/pii/S0959440X08000997?via%3Dihub. doi:10.1016/j.sbi.2008.07.001

Deweese, J., Hoang, K., Menzie, R., Fief, C., Ayes, C., Wilson, J., . . . Nelson, C. (2019, 2019). *The variable C-terminal domain of human type II topoisomerases as a functionally*

relevant therapeutic target. Paper presented at the American Society for Biochemistry and Molecular Biology, Orlando, FL.

Durston, K. K., Chiu, D. K., Wong, A. K., & Li, G. C. (2012). Statistical discovery of site inter-dependencies in sub-molecular hierarchical protein structuring. *EURASIP J Bioinform Syst Biol, 2012*(1), 8. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/22793672. doi:10.1186/1687-4153-2012-8

Esfahbod, B. (2007). In N. Euler diagram for P, NP-Complete, and NP-Hard set of problems. (Ed.). Wikimedia Commons.

Gaud, N., & Sharma, N. (2015). K-modes Clustering Algorithm for Categorical Data. *International Journal of Computer Applications, 127*(17), 1-6. doi:10.5120/ijca2015906708

Gibson, E. G., King, M. M., Mercer, S. L., & Deweese, J. E. (2016). Two-Mechanism Model for the Interaction of Etoposide Quinone with Topoisomerase IIalpha. *Chem Res Toxicol, 29*(9), 1541-1548. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/27533850. doi:10.1021/acs.chemrestox.6b00209

Huang, Z. (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery, 2*(3), 283-304. doi:10.1023/a:1009769707641

Kendrew, J. C., Bodo, G., Dintzis, H. M., Parrish, R. G., Wyckoff, H., & Phillips, D. C. (1958). A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. *Nature, 181*(4610), 662-666. doi:10.1038/181662a0

Knuth, D. E. (1974). Postscript about NP-hard problems. *ACM SIGACT News, 6*(2), 15-16. doi:10.1145/1008304.1008305

Laboratory, E. M. B. (2019). Family: ubiquitin (PF00240). Retrieved from https://pfam.xfam.org/family/PF00240#tabview=tab3

Lacapere, J. J., Pebay-Peyroula, E., Neumann, J. M., & Etchebest, C. (2007). Determining membrane protein structures: still a challenge! *Trends Biochem Sci*, *32*(6), 259-270. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/17481903

https://www.cell.com/trends/biochemical-sciences/fulltext/S0968-0004(07)00084-9?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS0968000407000849%3Fshowall%3Dtrue. doi:10.1016/j.tibs.2007.04.001

Lange, O. F., & Grubmuller, H. (2006). Generalized correlation for biomolecular dynamics. *Proteins*, *62*(4), 1053-1061. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/16355416. doi:10.1002/prot.20784

Leeuwen, J. v. (1998). Algorithms and complexity. In *Handbook of Theoretical Computer Science* (Vol. A): Elsevier.

Linka, R. M., Porter, A. C., Volkov, A., Mielke, C., Boege, F., & Christensen, M. O. (2007). C-terminal regions of topoisomerase IIalpha and IIbeta determine isoform-specific functioning of the enzymes in vivo. *Nucleic Acids Res*, *35*(11), 3810-3822. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/17526531. doi:10.1093/nar/gkm102

National Center for Biotechnology Information. Retrieved from https://www.ncbi.nlm.nih.gov/genbank/statistics/

Newman, A., & Hart, W. E. (2001). The Computational Complexity of Protein Structure Prediction in Simple Lattice Models. *CRC Press*.

Ngo, J. T., Marks, J., & Karplus, M. (1994). Computational Complexity, Protein Structure Prediction, and the Levinthal Paradox. *The Protein Folding Problem and Tertiary Structure Prediction*, 433-506. doi:10.1007/978-1-4684-6831-1_14

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2013). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Trevino, A. (2016). Introduction to K-means Clustering. Retrieved from https://www.datascience.com/blog/k-means-clustering

Wetterstrand, K. (2018). DNA Sequencing Costs: Data. Retrieved from https://www.genome.gov/27541954/dna-sequencing-costs-data/

Wong, A. K. C., & Li, G. C. L. (2008). Simultaneous Pattern and Data Clustering for Pattern Cluster Analysis. *IEEE Transactions on Knowledge and Data Engineering, 20*(7), 911-923. doi:10.1109/tkde.2008.38

Wong, A. K. C., Liu, T. S., & Wang, C. C. (1976). Statistical analysis of residue variability in cytochrome c. *Journal of Molecular Biology, 102*(2), 287-295. doi:10.1016/s0022-2836(76)80054-x

Zhexue, H., & Ng, M. K. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems, 7*(4), 446-452. doi:10.1109/91.784206

Zhou, P., & Chan, K. C. (2015). *An Unsupervised Attribute Clustering Algorithm for Unsupervised Feature Selection*. Paper presented at the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, France.