University of Massachusetts Dartmouth

Department of Computer and Information Science

Detecting and Visualizing Knots and Slipknots in Protein Chains

A Thesis in

Computer Science

by

Shashank Mucheli Sukumar

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

January 2016

I grant the University of Massachusetts Dartmouth the non-exclusive right to use the work for the purpose of making single copies of the work available to the public on a not-for-profit basis if the University's circulating copy is lost or destroyed.

____

Shashank Mucheli Sukumar

Date __________________________________

We approve the thesis of Shashank Mucheli Sukumar

Date of Signature

Firas Khatib
Assistant Professor, Department of Computer and Information Science
Thesis Advisor

Haiping Xu
Associate Professor, Department of Computer and Information Science
Thesis Committee

Xiaoqin Zhang
Associate Professor and Graduate Program Director
Department of Computer and Information Science
Thesis Committee

Jan Bergandy
Chairperson, Department of Computer and Information Science

Robert E. Peck
Dean, College of Engineering

Tesfay Meressi
Associate Provost for Graduate Studies

ABSTRACT

Proteins are made up of chains containing a large number of residues and are fundamental entities of all biological creatures. Protein chains very rarely have knots or slipknots and these should generally be avoided when predicting protein structures, as they make it more difficult to reach the correct fold. This project focuses on finding and visualizing knots and slipknots in protein structures. A protein is said to be knotted if the chain loops through itself and not back out again through the same loop. Slipknots occur when the polypeptide chain goes through a loop but the chain eventually doubles back on itself, removing the knot. As slipknots are not mathematically knots, the standard protein knot detection algorithms do not report them.

In this project we improve one such knot detection algorithm, Knotfind, to find slipknots in proteins and we have established a server for everyone to use. Our new algorithm iteratively searches subsections of a protein chain and checks each of them for knots. If a knotted region is found, we search for the residue that unties the knot resulting in a slipknot. In order to visualize the knotted or slipknotted region, our algorithm generates a simplified protein chain stored in a text file that is loaded using JSmol (a Java/HTML5 based molecular visualization tool). Using the JSmol API we then highlight the slipknotted region of the protein chain, along with the residues that untie the knot, clearly defining the slipknot.

ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis advisor, Dr. Firas Khatib for his continuous support throughout my master's program. He has undoubtedly been my biggest source of inspiration and support over the past two years. Dr. Khatib guided me in each and every step of my program though weekly meetings, quick responses to emails and any sort of support that I would need to finish up my thesis. I was also fortunate to be one of his research assistants and enjoyed working on every problem that we had to solve. He was very organized in terms what had to be done and what the goal exactly was. He started with basic issues and gradually scaled up. I could not have asked for anyone better than Dr. Khatib as my advisor.

I would also extend my gratitude to my committee members Dr. Haiping Xu and Dr. Xiaoqin Zhang for giving me their time and for being a part of my thesis committee. Both of them have been a great form of support. Their suggestions have been valuable to me.

I would also like to thank my parents for giving me such a healthy life and their never ending love and care. They have been the source of my strength and they have, over the years made me the man I am today, by teaching me morals and ethics in every stage of my life.

To conclude, I would like to thank that one person, without whom I would not have been here today, my brother. He has rescued me every time I get into trouble. Without his encouragement I surely could not have been here today. I would also like to thank my friends and colleagues for their feedback and support.

TABLE OF CONTENTS

TABLE OF FIGURES

1. Introduction

Proteins are fundamental entities for all living beings. Proteins aid in almost every function that is carried out inside the body, mostly in the cells. They are large and complex molecules. Proteins are made up hundreds or thousands of amino acids which are connected to each other making them long chains. Different amino acids sequences form different proteins. Every protein has a unique structure. The amino acid sequence determines the structure and the function of any given protein.

Proteins are naturally not available in the form that we often see in visualizations, every protein structure is determined by using various methods like x-ray crystallography, Cryo EM or NMR spectroscopy. The amino acid sequence obtained by these methods is stored in a special text file called PDB – protein data bank and has a file extension .pdb. A PDB file usually contains the metadata and the structural information of the protein. Every protein has a unique PDB ID with which it is referenced.

Proteins are visualized using specialized visualization tools that allow the user to check every amino acid in the sequence. Every visualizing tool requires a PDB file of the protein. The popular protein visualization tools available today are PyMol, Jmol/JSmol, RasMol, etc. Using these tools and their supporting libraries we can tweak the visualizations to support our view.

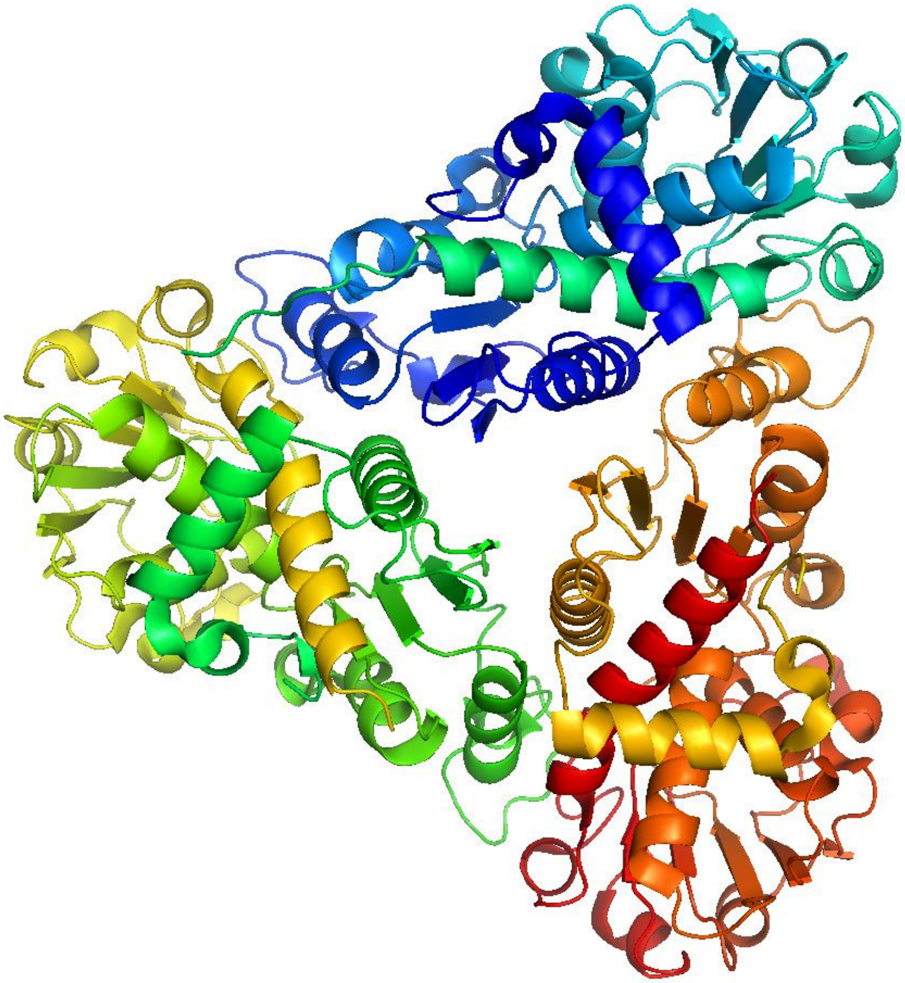


Figure 1: PyMol illustration of knotted protein 1JS1.

Knots and slipknots in protein chains make the protein folding process complicated and should generally be avoided. Biochemists have discovered many different types of knots in proteins with varied number of crossings in them. Reaching the native fold or the state with the most minimum energy in these types of proteins is very difficult and hence we need computational support to analyze the protein structures and determine if they are knotted or not.

The motivation behind developing this algorithm was to simplify the protein folding process in general by making sure that any given amino acid sequence does not have a knot or slipknot in it and also simplifying them rapidly and effectively. Biochemists around the world can use this algorithm to check if their protein models have knots and slipknots in them and can avoid knots and slipknots in their models without spending too much time to check for them manually. In most cases visualizing knots and slipknots is a tedious task, in this report we also present an integrated script to support knot and slipknot visualizations.

1. We provide a supporting script for PyMol to highlight knots and slipknots in the analyzed chain.
2. We have established a server to analyze the protein chains and give a clear visualization of the knot or slipknot using JSmol library.

The rest of this thesis is organizes as follows: Section 2 introduces to all the required background knowledge to help understand knotfind and slipknotfind. Section 3 talks about the related work that has been done by other institutions to work with knots and computing the knot types. Section 4 starts by formally introducing knotfind and explaining it in depth, section 4 then progresses to slipknotfind which is an extension of knotfind and describes slipknots and how to detect them and we conclude section 4 by explaining varies types of visualizations that we came up with to highlight knots and slipknots in protein chains. Finally, in section 6 we conclude our discussion and propose possible future ideas to extend this project.

2. Background Knowledge

2.1 Proteins and Protein Folding

2.1.1 Proteins

Proteins are fundamental entities for all living beings. Proteins aid in almost every function that is carried out inside the body, mostly in the cells. They are large and complex molecules. Each protein is associated with a specific function in the body. Few of those functions include:

- DNA replication: Helps in producing two identical chains of DNA.
- Messenger: Transmits signals to co-ordinate biological processes.
- Transportation: Binds small atoms inside them and transport across the body.
- Structural components: They form the structure of many tissues in our body such as muscle, hair, etc.,

2.1.2 Structure of a Protein

From chemical point of view, proteins are the structurally most complex and functionally rich molecules ever known. Proteins are made up of amino acids, a sequence of twenty different amino acids are bonded in different configurations to form a long string of protein. Every amino acid is linked with its neighbor through a covalent peptide bond, proteins are therefore called polypeptides. Order in which these amino acids are bonded defines the shape and structure of the protein chain. The structure of the protein determines the function of a protein in a cell[1]. Each protein chain has a different sequence of amino acids, many hundreds or thousands of protein structures are known to us and each of them have different sequence of amino acids[1]. The sequence of connected atoms at the core of the chain is called as polypeptide backbone. The atoms attached to this chain are those amino acids which are not involved in making of peptide bond and they give each amino acid a unique property. The entire sequence of amino acids determines the function of the protein.

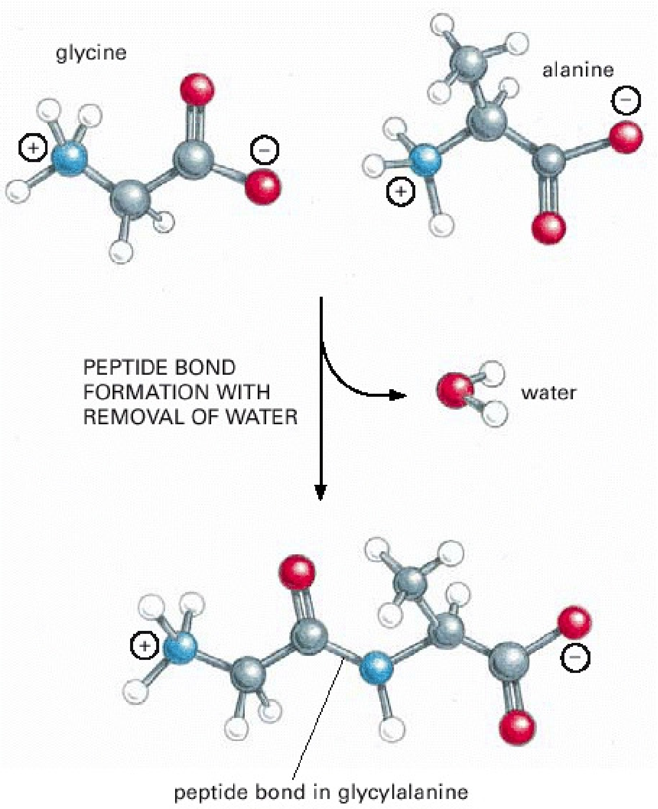


Figure 1: Formation of a peptide bond by removal of water.

In figure 1, This covalent bond forms when the carbon atom from the carboxyl group of one amino acid shares electrons with the nitrogen atom (blue) from the amino group of a second amino acid.

2.1.2.1 Levels of Protein Structure

- Primary structure: It consists of linear sequence of amino acids, covalent bonds and disulfide bonds in a protein structure.

- Secondary structure: It refers to the local structure of the backbone of the protein which is stabilized by intermolecular hydrogen bonding[2]. There are two types of secondary structures which are common, Alpha helices and Beta strands.
    - Alpha helix: It is a right handed spiral conformation in which every backbone N-H group denotes a hydrogen bond to the backbone C =O group of the amino acid[2].
    - Beta Strands: It is a polypeptide chain with 3-10 amino acids. Two or more parallel or antiparallel beta strands linked by hydrogen bonds form beta sheets.
- Tertiary structure: The three dimensional arrangement of secondary structures with a large number of non-covalent bonds between amino acids.
- Quaternary structure: Multiple polypeptide chains linking via non covalent bonds to form a single and larger protein[3].

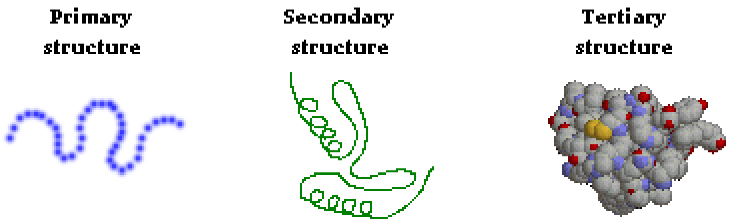


Figure 2: A simple visualization of primary, secondary, tertiary protein structures.

2.1.3 Protein Folding

Proteins fold and are held together by molecular interactions. The molecular interactions include thermodynamic stability, hydrophobic interactions and disulfide bonds formed inside the proteins[4].

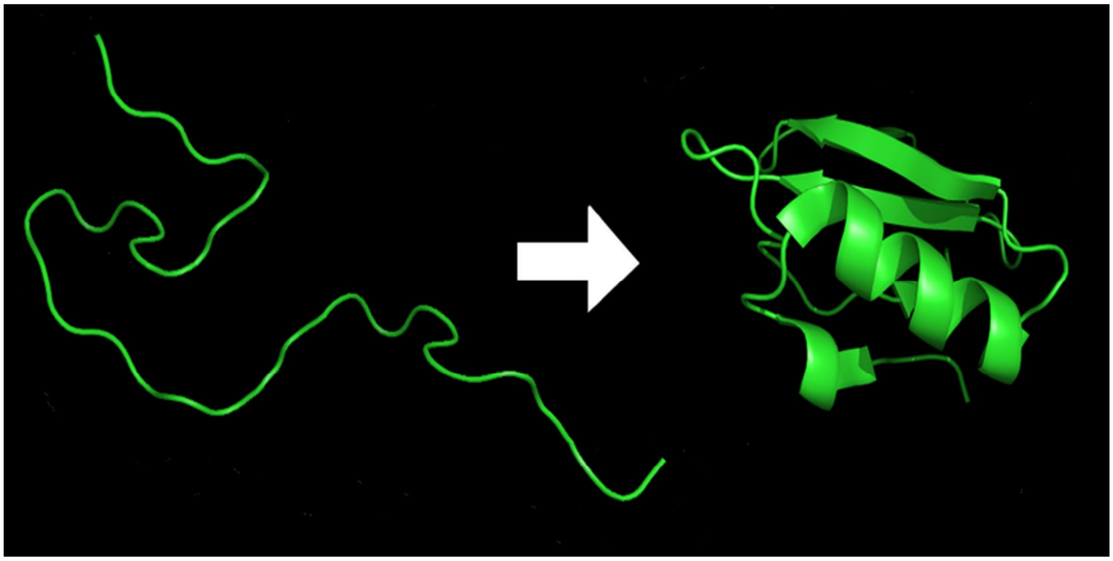


Figure 3: Protein folding sample.

2.1.3.1 Thermodynamic Interactions

Thermodynamics of a protein is plays a major role in protein folding. The interaction includes the inclination to form extended conformations, long range contact potentials depending on the residues and the formation of hydrogen bonds depending on the orientation[5]. The thermodynamic abilities inside the proteins are a major stabilizing force because if the protein is not in its lowest energy conformation it will continue to move and adjust until it finds its most stable state. The energy diagrams and maps are important to find out if the protein has reached its most stable state possible.

2.1.3.2 Hydrophobic Interactions

The hydrophobic model primarily relies on the short-range interactions of the secondary structure. Hydrophobic interactions have an impact not just on the primary structure but also the changes that happen in the secondary and tertiary proteins structures as well. Globular proteins get unique compact native conformations in water due to the hydrophobic effect. Even the proteins which are folded correctly have a hydrophobic core as a result of being hydrated by the water around it which is important because it creates a charged core which in turn leads to the creation of channels within the protein. The hydrophobic interactions have an impact on the protein even if the protein has found the most stable conformation on how they can interact with each other as well as fold themselves.

2.1.3.3 Disulfide Bonds

Disulfide linkages are another type of interactions that happen in protein folding.

The disulfide bond is a sulfur-sulfur chemical bond that results from an oxidative process that links nonadjacent cysteine's (amino acid) of a protein[6].

These are one of the major ways in which protein get folded. The disulfide bonds are cysteine-cysteine linkages that are a part of the final stable folded structure and those in which the pairs of cysteines alternate between the reduced oxidized states[6]. The bonds between the cysteines are quite stable once they are created.

2.2 Protein Data Bank

2.2.1 Introduction

They are the structural databases for the large molecules like proteins. Proteins are very tiny and are of microscopic size, determining their size and shape can be done using many modern day technologies, few of them are listed below.

1. X-ray Crystallography[7]: Is a method used to determine the arrangement of atoms in three dimensional spaces. This technique takes advantage of the interatomic spacing of most crystalline solids by employing them as a diffraction gradient for x-ray light, which has wavelengths on the order of 1 angstrom.
2. NMR Spectroscopy: Nuclear magnetic resonance spectroscopy exploits the magnetic properties of atoms to determine the physical and chemical properties of atoms. NMR can either be used to match against spectral libraries or the basic structure of the protein can be directly inferred [8].
3. Cryo-electron microscopy [9]: Cryo EM is becoming the primary technology in structural biology at a molecular resolution. In the past few years, Cryo EM has been used in a broad range of experimental methods. At the core, each of these is based upon the principle of imaging radiation-sensitive specimens in a transmission electron microscope under cryogenic conditions.

2.2.2 Standard PDB Format

The PDB's are populated with the data generated by biochemists after performing the above experiments on the protein structures. The crystallographic data of the proteins are stored in .pdb file format which are made available freely over the internet. The .pdb files can be downloaded from the websites like RCSB [10]. Every protein structure uploaded to any member organizations like RCSB, PDBe, PDBj is given a unique PDB ID to recognize the protein. Almost everywhere, the protein data is shared through a PDB (.pdb file).

A standard PDB file from RCSB is eighty columns wide and is always terminated by a end of the line indicator. The first six columns of every line contain a "record name" [10] and any PDB would have the following information in them:

Title and date: Name of the protein and the date when it was submitted and the PDB ID.

Experiment data, author information and remarks: The experiment through which the PDB data was generated followed by the name of the author and the journals it was published in and any remarks that the author wished to include.

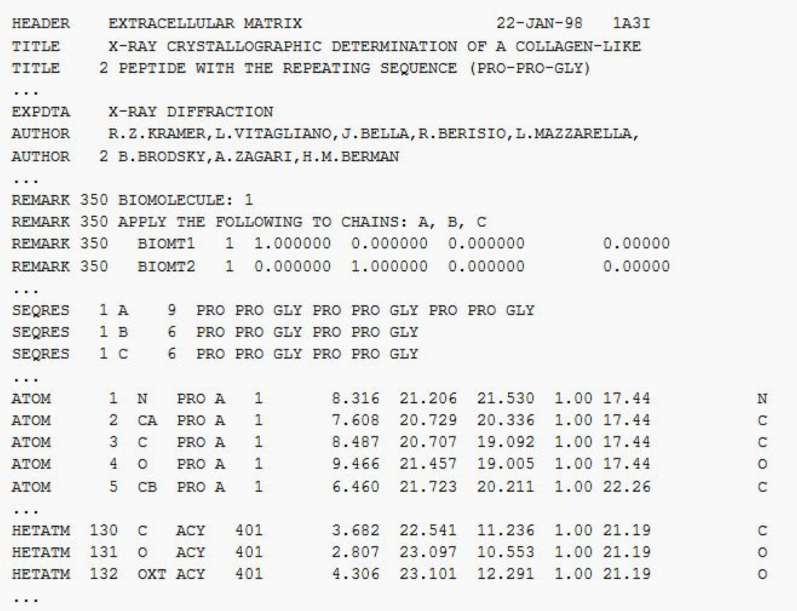
Physical structure: SEQRES, HELIX, SHEET etc.

> SEQRES is the list of the primary sequence of the polymeric molecules present in the entry [11].
>
> HELIX and SHEET: the data here corresponds to parts of the structure of the protein which forms the helical structures and the sheets in the protein. Visualizing softwares use data in this section to draw corresponding structures.
>
> Atom data:
>
> Every atom of every amino acid in the protein will have a row in this file which contains data like atom number, residue type, chain and most importantly its

structural data(x, y, z co-ordinates). The list of ATOM records for each chain starts with an N terminal and ends with the C terminal. Every polymer chain must be terminated with a TER record which defines the end of the particular chain. All the PDB files end with the line containing only the word END [10].

2.2.3 Role of PDB in Knotfind and Slipknotfind

In our algorithm, we consider only the cα atoms from the atom data in the PDB file. Cα's are alpha carbon atoms in the protein chain which form the backbone of the structure. Only the rows with cα atoms are filtered out of the protein chain and are stored in a Java ArrayList. Each row of cα corresponds to one atom row of the PDB and will have the corresponding cα's atom type, atom number, residue number, x-y-z co-ordinates and a few other columns of data. We save data from selected columns in the arraylist and discard everything else.



```
HEADER    EXTRACELLULAR MATRIX                    22-JAN-98   1A3I
TITLE     X-RAY CRYSTALLOGRAPHIC DETERMINATION OF A COLLAGEN-LIKE
TITLE    2 PEPTIDE WITH THE REPEATING SEQUENCE (PRO-PRO-GLY)
...
EXPDTA    X-RAY DIFFRACTION
AUTHOR    R.Z.KRAMER,L.VITAGLIANO,J.BELLA,R.BERISIO,L.MAZZARELLA,
AUTHOR   2 B.BRODSKY,A.ZAGARI,H.M.BERMAN
...
REMARK 350 BIOMOLECULE: 1
REMARK 350 APPLY THE FOLLOWING TO CHAINS: A, B, C
REMARK 350   BIOMT1   1  1.000000  0.000000  0.000000        0.00000
REMARK 350   BIOMT2   1  0.000000  1.000000  0.000000        0.00000
...
SEQRES   1 A    9  PRO PRO GLY PRO PRO GLY PRO PRO GLY
SEQRES   1 B    6  PRO PRO GLY PRO PRO GLY
SEQRES   1 C    6  PRO PRO GLY PRO PRO GLY
...
ATOM      1  N   PRO A   1       8.316  21.206  21.530  1.00 17.44           N
ATOM      2  CA  PRO A   1       7.608  20.729  20.336  1.00 17.44           C
ATOM      3  C   PRO A   1       8.487  20.707  19.092  1.00 17.44           C
ATOM      4  O   PRO A   1       9.466  21.457  19.005  1.00 17.44           O
ATOM      5  CB  PRO A   1       6.460  21.723  20.211  1.00 22.26           C
...
HETATM  130  C   ACY   401       3.682  22.541  11.236  1.00 21.19           C
HETATM  131  O   ACY   401       2.807  23.097  10.553  1.00 21.19           O
HETATM  132  OXT ACY   401       4.306  23.101  12.291  1.00 21.19           O
...
```

Figure 4: Protein data bank.

2.2.4 Visualize PDB's:

PDB's are plain text files which contain a lot of data. Given the increasing number of protein structures discoveries using faster mechanisms like Cryo EM, it gets very difficult and time consuming to visualize each and every protein manually and given the limited number of operations that we can perform on them, we need special and more sophisticated protein visualizing tools to visualize proteins effectively. There are a lot of them available today~~,~~ we use PyMol [12] and Jmol/JSmol [13] to visualize the proteins that we simplify using our algorithm. These tools also allow us to write our own custom scripts which will help any user to visualize structures as per their needs. PyMol comes with an API of its own and allows scripting in Python. Whereas Jmol/JSmol can be scripted using Java and JavaScript, using these tools and how we implement our scripts using them are explained in detail in the later sections.

2.3 Knots in Protein Structures

2.3.1 Knots

Knots in proteins are very rare and they should be avoided in the protein folding process. Proteins are considered knotted if their backbone is entangled to form a knot. Imagine pulling a protein by holding both its termini (N and the C terminals), if the sequence ends up having a loop which goes through itself, then we consider it as a knotted protein[14]. Knots in protein chains occur in much lesser frequency than would be expected from the polymers of the same length. In many cases the knots are formed near the terminals, these knots disappear when the short sections near the terminals of the polypeptide chain are trimmed. Deep knots in protein chain however occur very far from the terminals and are not often observed. Nearly 2% of the proteins deposited in the PDB are discovered to be knotted.

2.3.2 Discovery of Knots in Protein Structures

Relationship between folding pathways and it the native structure of the protein has led to the development of measures to predict the folding rates of a protein[15]. According to these measures proteins may not fold easily to its native states if their pathways have strange and unlikely configurations. These can happen particularly if the proteins have complex topological features such as knots. Although they are rare, a few of them have been found. Computational, structural, and biophysical investigations of these unusual proteins should lead to valuable insight into how proteins fold and how they maintain their folded configurations, often under extreme conditions[15].

Crippen in 1974[16] considered the case of how likely a linear protein chain would fold into a knot. None of the known structures back then had knotted protein backbones. Twenty years later Mansfield considered the situation computationally by taking the database of known protein structures and checking them for knots. In a mathematical sense, only a closed loop can get knotted, but ideas with open chains such as proteins were introduced which were considered knotted practically. The chain termini were extended from the body of the protein and joined them virtually[17]. In the first computational survey, approximately 400 known protein structures were examined and no deep knots were found. The concept of protein knots again changed in the year 2000 when Taylor[18] discovered a deeply knotted protein structure. This led to the reopening of the questions related to how unlikely knots are in proteins and emphasized on the importance of using computational methods to examine proteins with such topological features. Recent investigations have identified five or six distinct protein families having knots with significant depth[14],[19] and experiments to understand the effects of proteins have started.

2.3.3 Effects of Knots in Proteins.

In general a protein with no knots should get disentangled and it should not have any loops in it. Knots in proteins are tough to understand as the complexity to reach its native state becomes tougher and complicated to determine its function and proteins with knots are less efficient than the unknotted proteins. Structure Prediction methods do not model the protein folding process itself but rather try to seek only the native state. Consequently, the knot prediction mechanisms are not relevant to the protein modeling process, which means the possibility of finding knots are high in protein models.

2.4 Scripting Tools

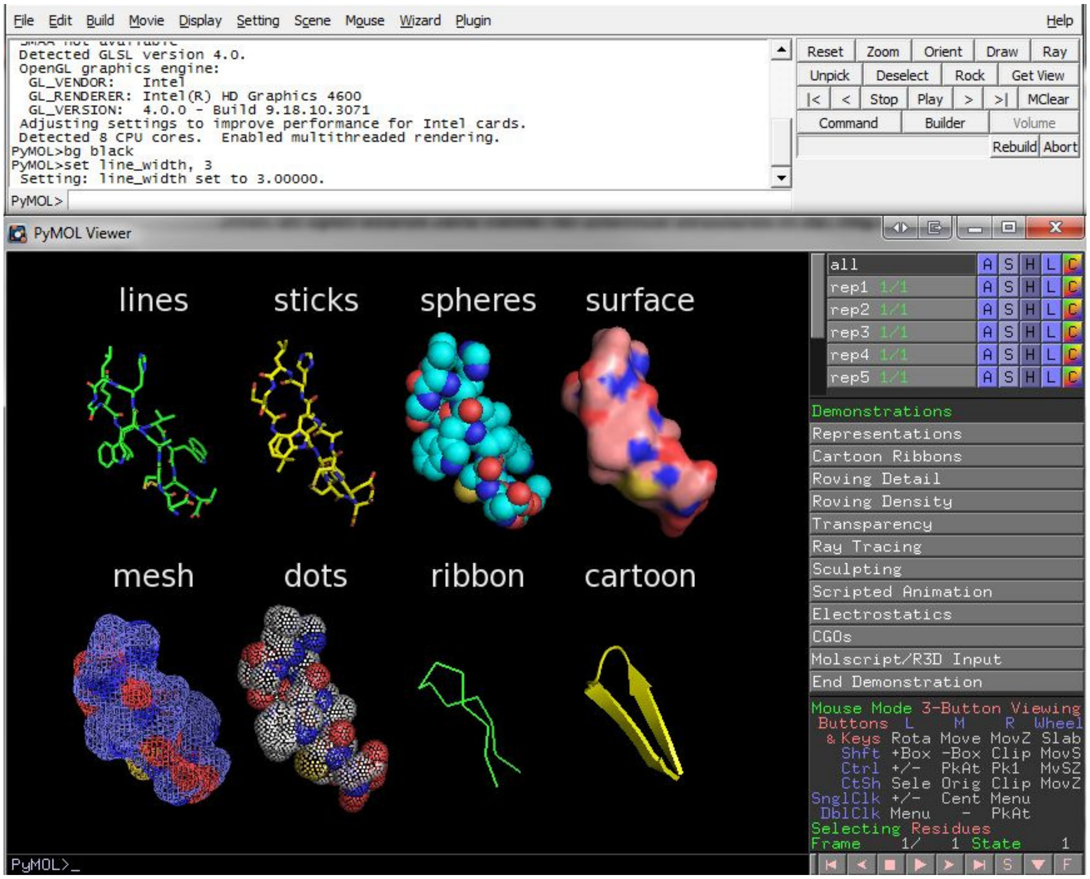
2.4.1 Need for Protein Visualization.

A tool to graphically visualize proteins and other types of molecules are required to see what atom data is encoded inside the protein data bank and to manipulate the images to view a protein from different perspectives. Without a visualizing tool, we cannot read a .pdb (Protein data bank) file only as a text file and it gets tedious to draw those co-ordinates in three dimensional spaces. Hence, researchers need a tool which is capable of displaying large amounts of protein data. Many visualization tools have been developed to visualize proteins whose structures are known and are available in .pdb files. Different visualizing tools were developed on different displaying capabilities like displaying the sequence, selecting a set of amino acids in a given structure, display disulfide bonds, hydrogen bonds, hydrogen bond distances and display different types of protein surfaces[20]. We have used two such freely available protein visualization tools over the internet with scripting capabilities which will help us customize loading and customizing protein visualizations for our research. They are PyMol and JSMol. PyMol is a python based protein visualizer which runs as a standalone application on a cross platform environment whereas JSMol is based on java and javascript, it can used to render three dimensional protein visualizations over an internet browser. Both of which are explained in detail below.

2.4.2 PyMol

PyMol is a free, modifiable and redistributable molecular visualization tool, which is frequently used by structural biologists and crystallographers. Images generated using PyMol can be seen in several research papers. PyMol users get a high resolution images and animations of biological structures such as proteins[21]. PyMol is an open source tool which is scripted using python. PyMol comes with a library of functions of its own. With simple knowledge of python and PyMol, its users can write custom scripts of their own to improve or highlight their visualizations.

PyMol accepts .pdb files to generate images and animations of macromolecules such as proteins which are available from the protein databank. .pdb files are generated by techniques like x-ray crystallography. By using this experimental data, structural biologists determine the location of each atom relative to each other in a molecule like protein.



```
 SMAA not available
 Detected GLSL version 4.0.
 OpenGL graphics engine:
  GL_VENDOR:   Intel
  GL_RENDERER: Intel(R) HD Graphics 4600
  GL_VERSION:  4.0.0 - Build 9.18.10.3071
 Adjusting settings to improve performance for Intel cards.
 Detected 8 CPU cores.  Enabled multithreaded rendering.
PyMOL>bg black
PyMOL>set line_width, 3
 Setting: line_width set to 3.00000.
PyMOL>
```

```
Mouse Mode 3-Button Viewing
 Buttons  L    M    R  Wheel
  & Keys Rota Move MovZ Slab
    Shft +Box -Box Clip MovS
    Ctrl +/-  PkAt Pk1  MvSZ
    CtSh Sele Orig Clip MovZ
SnglClk +/-  Cent Menu
 DblClk Menu   -  PkAt
Selecting Residues
Frame     1/    1 State     1
PyMOL>_
```

Figure 5: Different types of rendering offered by PyMol molecular visualizer.

2.4.3 Jmol

Jmol is a free and open source molecule viewer developed for myriad of different users. It is cross platform and runs on Windows, Linux and Mac OS X systems. It is a multi-language tool which is translated into multiple languages like Chinese, French and German. There are three different ways of using Jmol.

1. JmolApplet: It is a web browser applet that can be included in the web pages[13]. Jmol buttons are used to invoke scripts which can be read by the applet. The scripts can alter the behavior of the default illustrations of important structural features.
2. Jmol Application: It is a standalone java application that runs as an application on desktops.
3. Jmol Viewer: It is a development tool kit that can be integrated in any java application[13].

2.4.3.1 JSmol

JSmol is a JavaScript version of Jmol that allows developers to create web pages with Java or use HTML5 to create pages without Java. This enables Jmol to display three dimensional molecular structures on devices which does not support Java (such as a few smart phones and tablet computers).

JSmol with HTML5 is identical to JSmol with Java in terms of rendering. Java version gives a much smoother rotation than the HTML5 version. This is usually not noticeable, it is noticed for molecular structures with more than 20,000 atoms.

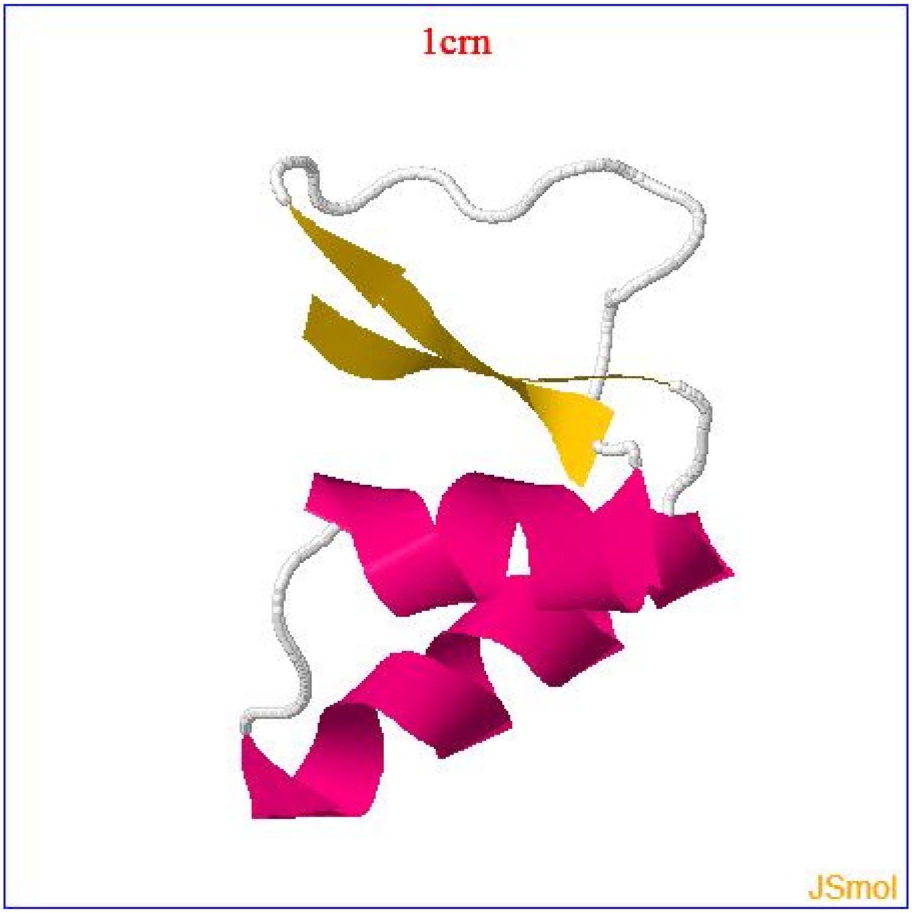


Figure 6: JSmol example.

3. Related Work

3.1 KnotProt Database

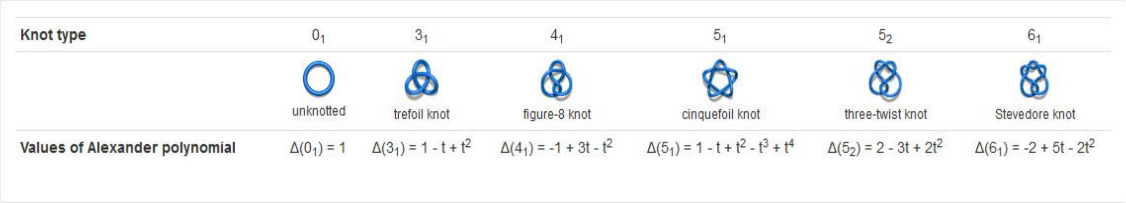
3.1.1 Introduction

The KnotProt database collects information about topologically non-trivial proteins, i.e. proteins with knots and slipknots and represents them in the form of a “knotting fingerprint”[22],[23], and presents many statistics based on the obtained results. It is based on the proteins deposited in the protein data bank and has a database of 900 proteins with knots and slipknots[22],[23]. Proteins form knots and slipknots in which backbone as a whole is unknotted.

The details about every entanglement in a protein are stored in the KnotProt database and are presented in the form of a knotting fingerprint. The knotting fingerprint has the information about the type of the knot in each sub chain of the protein and represents it with a matrix diagram. The KnotProt database also provides extensive statistics about proteins with knots and slipknots based on their biological features and geometrical data such as depth of the knot and type of fold, length of the knot etc. KnotProt analysis reveals proteins having slipknots and knots can be classified into distinct topological motifs, represented by a few patterns in the matrix. This data can be used to find proteins with slipknots or knots based on the homological sequence, a similar structure, or something with particular biological function. The KnotProt website is updated after new proteins are deposited to the PDB every Wednesday.

3.1.2 Knot Detection

Many different types of knots have been found in proteins so far. A few of them are denoted as follows: trefoil knot (3_1), figure-8 knot (4_1), (5_2) and Stevedore's knot (6_1). Unknotted loops are known as a trivial knot, or unknot and are denoted as 0_1. Knots are uniquely defined in closed chains. To define them in open chains such as proteins (which have loose ends), we choose how to connect the loose ends such that a loop is formed[18]. Making this choice optimally is a difficulty which has to be overcome while analyzing proteins. Once this choice has been made and once an open chain is transformed into a loop, the type of knot can be detected by calculating a knot invariant polynomial knows as Alexander polynomial [22],[23]. It can be calculated by planar projection (two dimensional) of a knot. Alexander polynomial is different for all the knots with eight or fewer crossings, which is enough to detect knots in protein chains (The most complicated knot found in proteins so far has only six crossings).



Knot type	0_1	3_1	4_1	5_1	5_2	6_1
	unknotted	trefoil knot	figure-8 knot	cinquefoil knot	three-twist knot	Stevedore knot
Values of Alexander polynomial	$\Delta(0_1) = 1$	$\Delta(3_1) = 1 - t + t^2$	$\Delta(4_1) = -1 + 3t - t^2$	$\Delta(5_1) = 1 - t + t^2 - t^3 + t^4$	$\Delta(5_2) = 2 - 3t + 2t^2$	$\Delta(6_1) = -2 + 5t - 2t^2$

Figure 7: Different types of knots and their associated Alexander Polynomials.

Defining knots in an open chain is nontrivial [18]. Knots can be uniquely determined only if the given protein chain is closed, and the classification of the knot depends on how one chooses to close the open chain [24]. KnotProt uses random closure method to close the open chains i.e. protein endpoints are connected several times to two randomly chosen points from a set of vertices of a truncated icosahedron placed on a sphere enclosing the given chain and these two points are connected by an arc on the circumference of the sphere. The knot associated with the given chain is then analyzed.

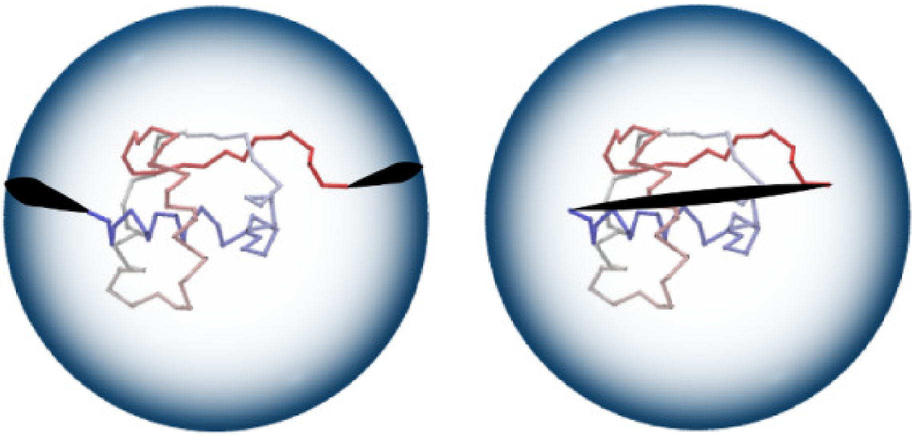


Figure 8: Random closure method (left). Direct closure method (right)

The knot types that are obtained by connecting the open ended chains are determined by calculating the polynomial knot invariants. For quick calculations the Alexander polynomials are used.

Calculating knot polynomials for short protein chains is fast, but it might consume significant amount of time to compute long chains (ex: for protein chains with more than 500 amino acids). Therefore before computing the Alexander polynomial the length of the chain is reduced by applying the KMT algorithm [25]. The algorithm processes all the triangles formed by three consecutive amino acids and removes the middle atoms if there the no line segment cutting through it. After completing a number of iterations only the core of the protein which constitutes the knot remains and all other amino acids are removed. A similar approach is later explained while describing the knotfind algorithm.

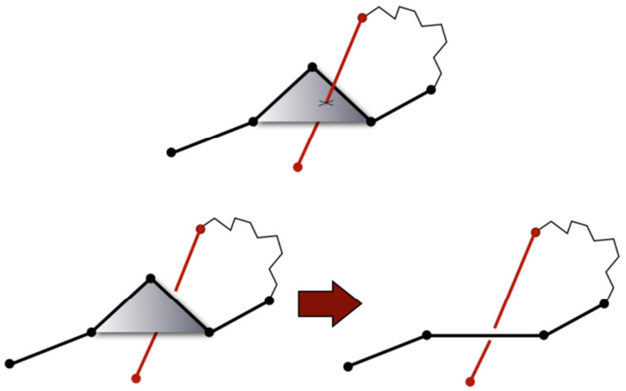


Figure 9: Illustration of the KMT algorithm.

The KnotProt database not only checks if the chain is knotted but also analyzes all the subchains of given protein. For any protein this information is presented as the Knotting data with a matrix diagram [22], [23].

3.2 Protein Knot Server – knots.mit.edu

Knots.mit.edu [26] is a server that processes protein structures and detects knots. The server accepts a protein structure either by its PDB ID or the user can upload the three dimensional structure in PDB or mmCIF format. The server uses Alexander polynomial to determine the type of the knot formed by the given protein structure. A visualization focusing the knot is presented and the server maintains a list of known proteins with knots.

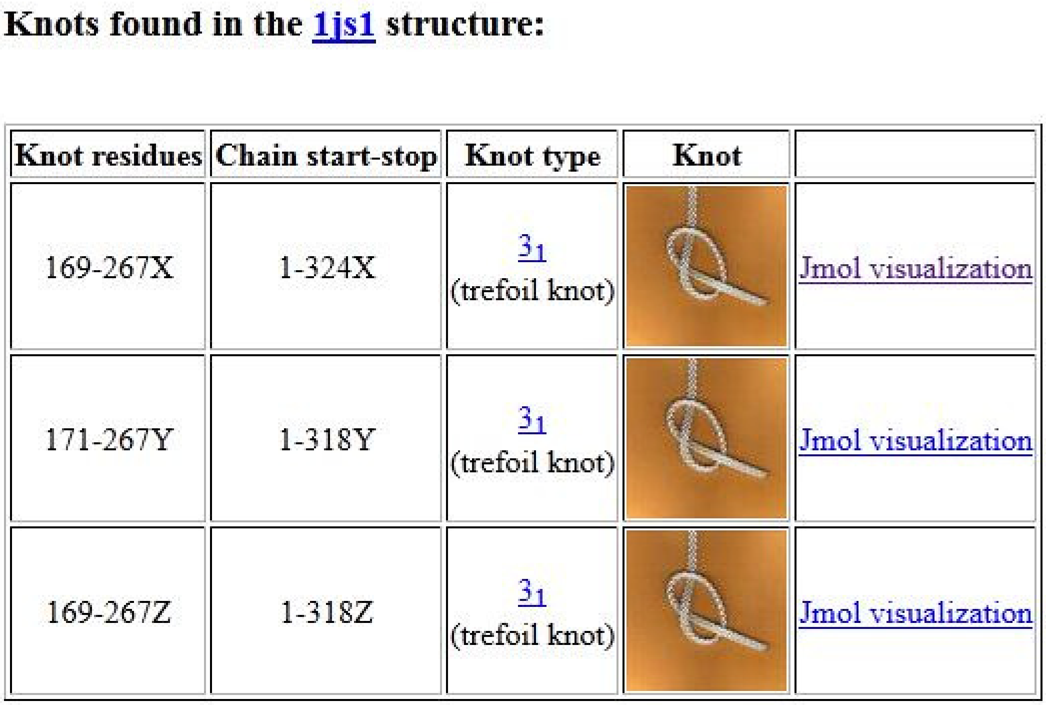
3.2.1 How Knots are Determined

Mathematically, knots are not well defined in open chains [27] such as proteins. Both the termini of the chain (N and C terminals) of open protein chains must be connected to make the entire chain a closed loop. To get a closed loop, the protein chain is reduced to its backbone and two lines are drawn from the termini towards the connection line between the center of mass of the backbone and the respective terminals. The two lines are then connected by a loop which makes the chain a closed loop and the Alexander polynomial [27], [28] can be applied on the resulting loop to classify the type of the knot formed by the loop. To calculate the size of the knot core, amino acids are removed successively from the N-terminus until the chain gets unknotted[19]. The procedure is again repeated from the C-terminus starting from the last deletion in the N-terminus of the chain which contained the knot. After every deletion an outward pointing line parallel to the respective lines are drawn for the entire chain. The size of the knot calculated by this method is not always precise and is approximate.

KMT algorithm is used to speed up the calculations [25]. This algorithm iteratively deletes amino acids from the chain which are not a part of the knot core. After analysis of the entire chain only the knot core remains. And the resulting sequence can be used to precisely visualize the knot.

There are a set of criteria that a protein chain has to satisfy in order to be considered as knotted [29]:

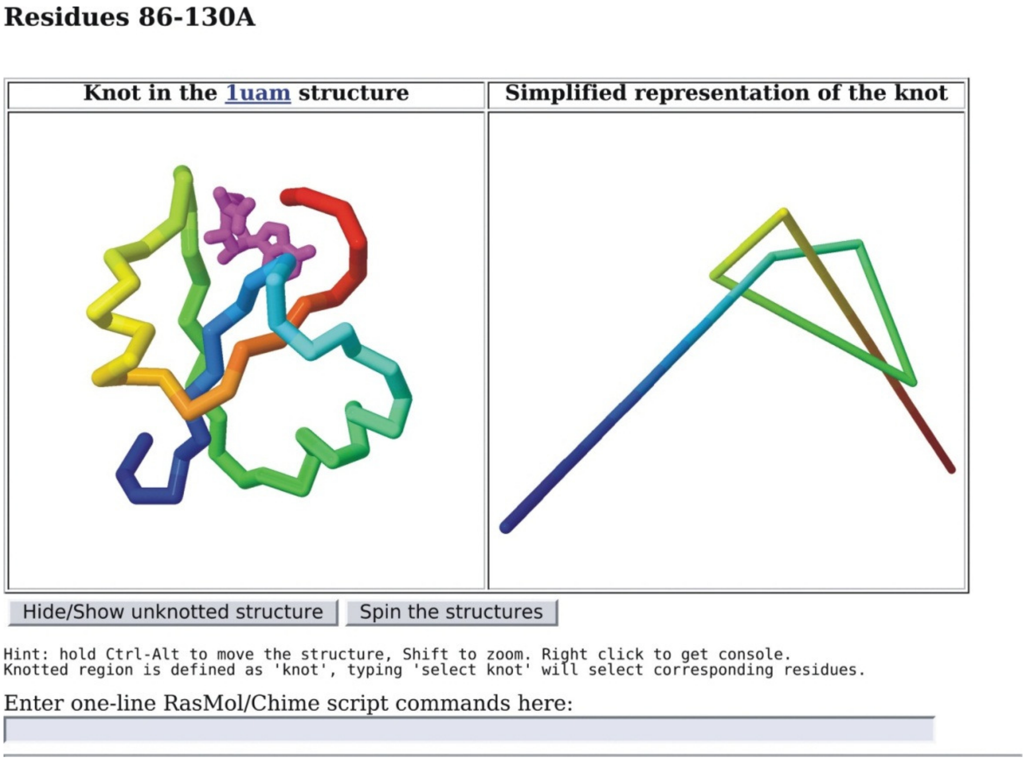
1. The calculated alexander polynomial should result in a knot.
2. The polypeptide chain should not be broken or have any gaps in between.
3. The knot should exist even if two amino acids are removed from the terminals (This avoids shallow knots or the knots which appear due to the closure method of the protein chain. The resulting knots are only deep knots.)



Knots found in the 1js1 structure:

Knot residues	Chain start-stop	Knot type	Knot	
169-267X	1-324X	3_1 (trefoil knot)		Jmol visualization
171-267Y	1-318Y	3_1 (trefoil knot)		Jmol visualization
169-267Z	1-318Z	3_1 (trefoil knot)		Jmol visualization

Figure 10: 1js1 protein knot visualized by knots server .



Residues 86-130A

Knot in the 1uam structure | Simplified representation of the knot

```
Hint: hold Ctrl-Alt to move the structure, Shift to zoom. Right click to get console.
Knotted region is defined as 'knot', typing 'select knot' will select corresponding residues.
```

Enter one-line RasMol/Chime script commands here:

Figure 11: RasMol views.

4. Knotfind and Slipknotfind.

4.1 Knotfind.

We have extended the knotfind algorithm in this project to detect slipknots and visualize both knots and slipknots. Knotfind is an efficient knot prediction algorithm that analyzes every residue in a protein structure and searches for knots in them. The chain is continuously simplified by removing residues until only the termini remains. Knotfind then returns a simplified PDB file that contains only the residues which form the simplified chain.

4.1.1 Knot Detection Using Knotfind

Knotfind uses only the alpha carbon atoms (cα) in a protein chain to detect the knots. The PDB is parsed into a Java method to extract only the data of the alpha carbon atoms and is stored onto a list. We only need the atom number and co-ordinates of the each cα atom. The atom data is later passed onto different methods for further processing of the chain.

Knotfind uses an iterative approach to simplify and eliminate atoms from the residue chains. It initially starts with 'n' atoms (total number of cα's in the given protein chain) and goes on until the size of the chain is two (for unknotted proteins) and proteins with knots will have two plus the unsimplified atoms.

Cα atoms arranged in the increasing order of i-1 to i+1 cartesian distance. Sets of three consecutive cα atoms i-1, i, i+1 are considered in each iteration. If there is no line segment, j, j+1 cutting through the triangle formed by connecting i-1, i, i+1, then cα i, is removed from the residue chain. If the line segment defined by j, j+1 is cutting through the triangle defined by i-1, i, i+1 then the cα i, is not simplified and the next set of i-1, i, i+1 is considered for simplification. This procedure is repeated until the last set of atoms are selected and simplified.

When the algorithm terminates it should only have N and C terminal cα atoms such that the chain has been simplified into a straight line. If the chain is not fully simplified, as in, there are other cα atoms remaining with the N and C terminals, and then those atoms define the knotted region of the protein [14].

When a knot is detected, to double check and verify the knot an alternate method is also used, Where the area of the triangle formed by connecting i-1, i, i+1 is considered, If area covered by the triangle is being interested by any line j, j+1 which or also in the same plane then the i'th is not simplified else, the i'th atom is simplified and the process is repeated until the all the residues are simplified or checked. A tolerance of 0.0003 Å is used to round off errors. This is considered as a possible line width of the line connecting j, j+1 [14].

The algorithm keeps a log of all the simplified residues and the unsimplified residues. When the algorithm terminates the remaining residues are stored back onto a new PDB file. This is then visualized.

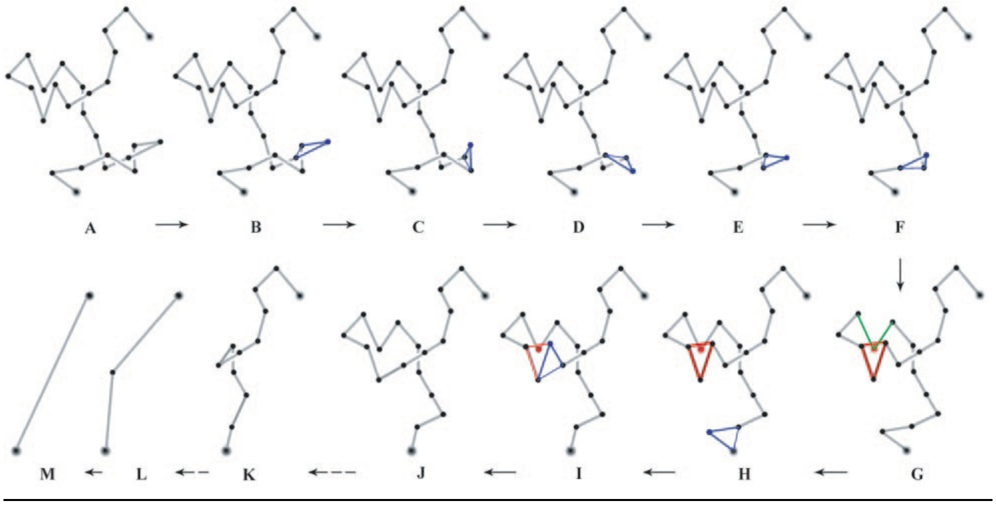


Figure 12: Simplification using Knotfind.

In figure 12 [14], we can see the changes in the state of the protein chain when the algorithm is running. Image A is the original structure of the backbone of the protein formed by connecting the cα atoms. The algorithm considers i-1, i, i+1such that the distances between them is the shortest. In image B, since no line segment is intersecting the lines connected by i-1, i, i+1 (The triangle marked in blue) the cα atom i is eliminated from the chain. The same process is repeated in the steps C through F. In the step G, an atom is intersecting the lines connected (The triangle marked in red), so i'th atom here is ignored and the next set i-1, i, i+1is considered for simplification. After iteration in steps H, one of the atoms with intersects in step G gets simplified and all the following atoms will get simplified in the later steps. When the algorithm terminates, since no knot is detected, it will have only the two terminal atoms left in the protein chain.

Knotfind algorithm can be used to benefit many structure prediction approaches, especially screening final models in a set of decoys and avoid a knotted model. Such screening methods are particularly important in automated methods like Robetta, where an expert will have to go in to check the final predictions manually. The speed of Knotfind makes it suitable not only for filtering of decoys to eliminate knotted structures but also during the protein structure prediction process as a filter or a part of the scoring scheme for optimization. Introduction of chain breaks in the modeling process is increasing the probability of knot formation. The location of chain breaks and the size of the gap generated by the chain breaks are important factors as well for the formation of knot [14]. To conclude, Knotfind algorithm is not only applicable to the homology based methods, but in any protein modeling process that introduce chain breaks during, including de novo prediction methods that have recently proved to have accuracy better than one second for small proteins with very few amino acids [30].

4.2 Slipknotfind

Slipknots are a different class of knots which rarely occur in the protein models. They are different when compared to the traditional knots and detecting them is a tedious task. It is also different from detecting the commonly found knots.

Slipknots are knots in which the knotted protein chain doubles back on itself and mathematically gets untied. If we assume to hold both N and C terminals and pull from both the ends [31], the knotted chain would get unknotted and would simplify into a straight line. But, a knot is still present in the chain and is affecting the energy of the protein model, even these should be processed and removed to have a protein with most minimum energy or a protein with the correct fold. Slipknots are not mathematically knots and hence the normal knot detection algorithms cannot report them, even Knotfind cannot report them when the entire backbone is passed to it at once. The Knotfind algorithm can simplify all the atoms in the chain other than N and C termini and would report no knots when the entire chain is passed. However, proteins with slipknots have a knot in them which when processed in subsections can be reported using knotfind.

We leverage this property of slipknots to find them and report them. We extend the already existing knotfind algorithm to detect slipknots. We divide the entire protein chain into small subsections and iteratively run the knotfind algorithm on these subsections and increase the size of the chain after every pass until all the residues are checked. While processing, slipknots are mathematically knots until it finds that one atom (which we mark as the k1 atom) which unties the whole chain and makes the chain unknotted.

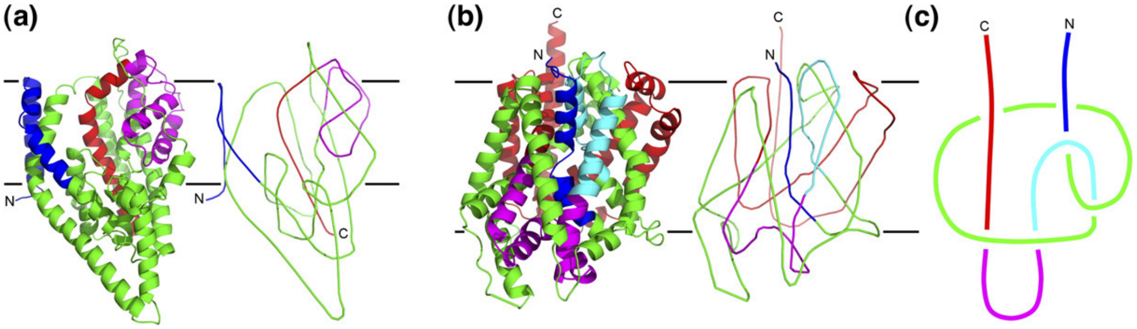


Figure 13: Simplifying the protein chain to highlight slipknot.

The transition from image A to C in the above figure [15] represents the simplification of the protein chain to have just the alpha carbon atoms after smoothening the backbone of the initial structure.In the above image, in section C, we can notice that pulling both N and C terminals would untie the chain and make the whole protein unknotted. These types of knots go undetected when we run traditional knot find algorithms on them.

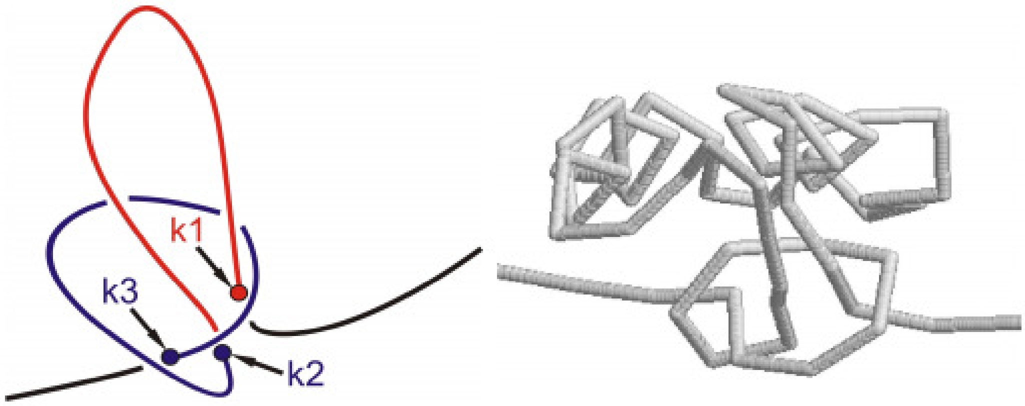


Figure 14: The atoms which define the slipknotted region.

The figure 14 [23] defines the points which form a slipknot. Atom k3 to k2 form the knot and continues to be knotted until k1 is found which unties the whole chain and makes the protein chain unknotted.

4.2.1 Slipknot Classification

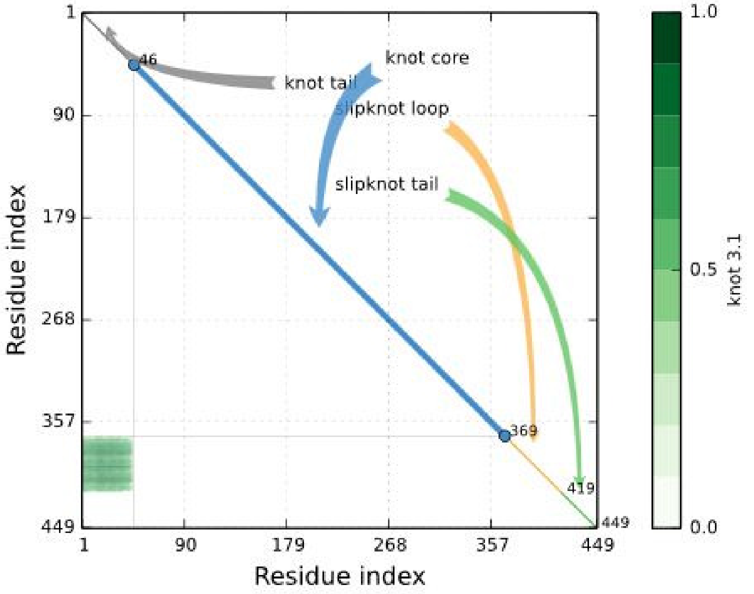


Figure 15: KnotProt for 1ALK chain A.

A slipknotted protein chain is classified into four different regions and each of them represents a part of the knot. Following are its details:

- Knot Core [32], [33]: The shortest sub chain with a knot.
- Knot Tail: A segment between one terminal and the Knot core.
- Slipknot Tail: The longest segment starting at one terminal, for which no change in topology is detected.
- Slipknot Loop: The segment between knot core and the slipknot tail.

4.2.2 Extending Knotfind to Detect Slipknots

We extend the knotfind algorithm to find slipknots, we run knotfind algorithm on subsections of the protein chains. We start the processing with only three atoms in the chain, Knotfind would return false with a chain size of three as no line segment could possibly intersect a line drawn to connect these three atoms. After every attempt of knotfind we increase the size of the chain by one, as in we append the next available cα atom in the pool to the sub chain which we are considering to process slipknots, and run the knotfind algorithm on it.

For any knotted or slipknotted sample, we first focus on getting the k3 and k2 atoms which define the knot core for any knotted sample. If no knot core is found during the entire process it is evident that the sample is not knotted or slipknotted. If the knotfind algorithm returns a knot for any given subsection of the protein we report the k3 and k2 atoms and continue increasing the size of the chain till ‘n’.

Once a knot is found, the algorithm checks until what length the algorithm would return true. If at any point the algorithm returns false, then we mark that atom which removes the knot as k1 atom and report it along with k3 and k2. And we mark the entire chain as slipknotted. The regions from k3 to k2 would form the knot core and k2 to k1 would be the slipknot loop. Atoms from one to k3-1 would be marked as knot tail and atoms from k1+1 to ‘n’ would be marked as slipknot tail.

The algorithm logs all the residues which are unsimplified cα atoms from k3 to k1 onto a PDB file along with all the atoms which constitute the knot tail and slipknot tail. With the knot tail and slipknot tail we can visualize the slipknots better.

At the end of the analysis, based on what type of knot the protein has, we create PDB files with the corresponding atom data in it. The new PDB files will have only the cα data in it for knotted structures and knot tail and slipknot tail along with the unsimplified cα atoms for the slipknotted protein structures.

4.2.3 Visualizing Knots using Pymol

The new PDB's are loaded into PyMol for visualization. PyMol has an API of its own written in Python that we have used to highlight knots and slipknots. We wrote a python script to connect all the atoms and visualize them serially.

Here is an example of how a knotted protein would look like before and after applying the script:

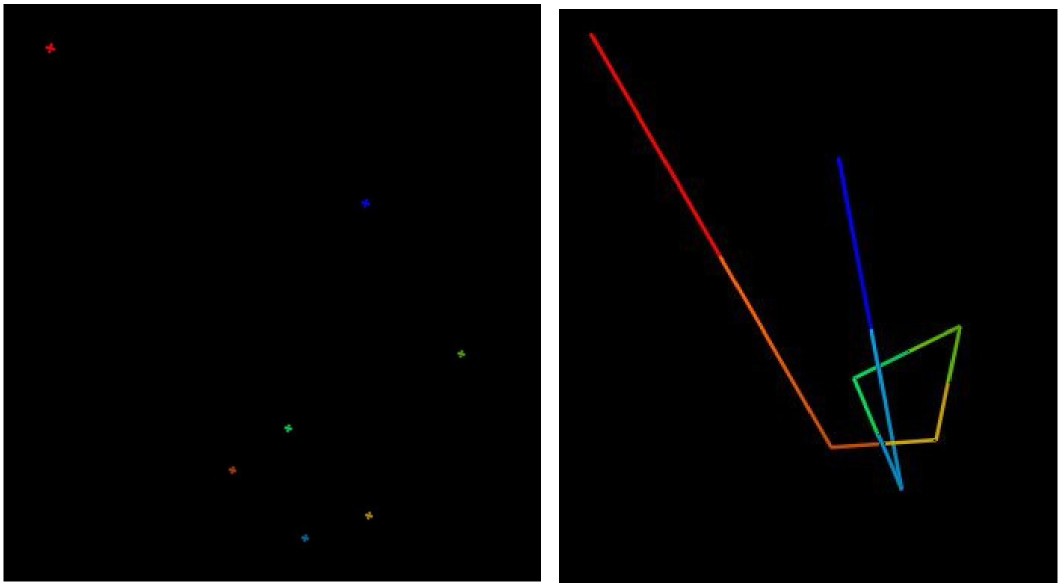


Figure 16: Connecting atoms to highlight knotted region.

The above images are the unsimplified atoms of the protein 1js1. The knot can be visualized clearly in the second image after connecting the atoms and color them with spectrum coloring. Once the PDB is loaded into PyMol, the script has to be loaded along with it to connect cα atoms.

4.2.4 Visualizing Slipknots using PyMol

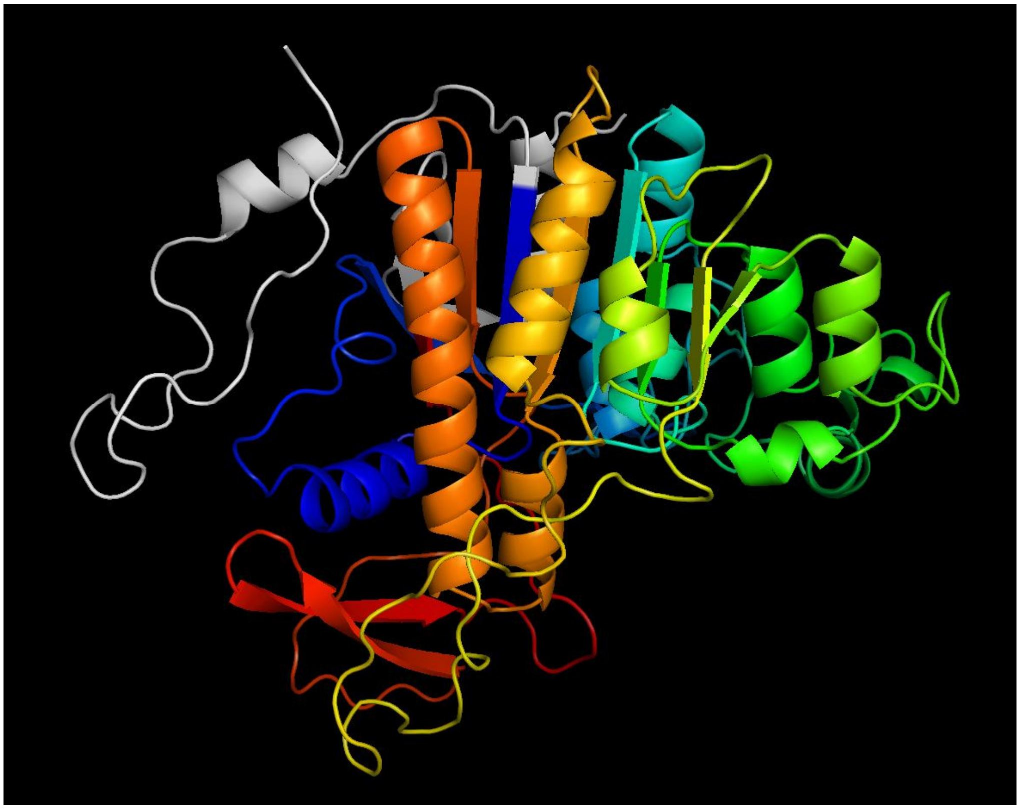


Figure 17: Knot Core of 1ALK.

The above image is an example of how a slipknotted region would look like. The colored region forms the knot core and the slipknot loop and the tails are marked in gray. It is very hard to figure out how the colored region would form a slipknot and is a tedious task. After we run the slipknotfind algorithm on the above protein, 1ALK, we get the following result:

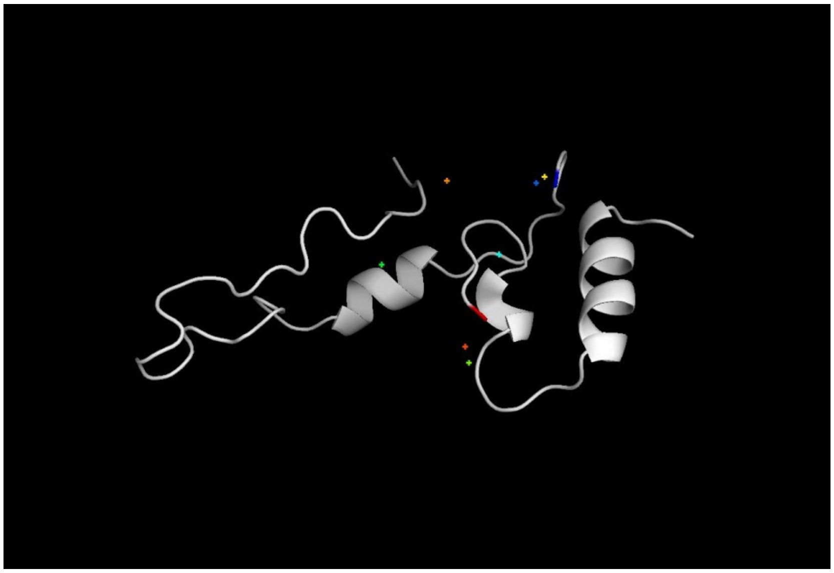


Figure 18: Simplified knot Core of 1ALK.

The above image has a few atoms which are not connected. These are the atoms which define the slipknot here is what we get by connecting these cα atoms.

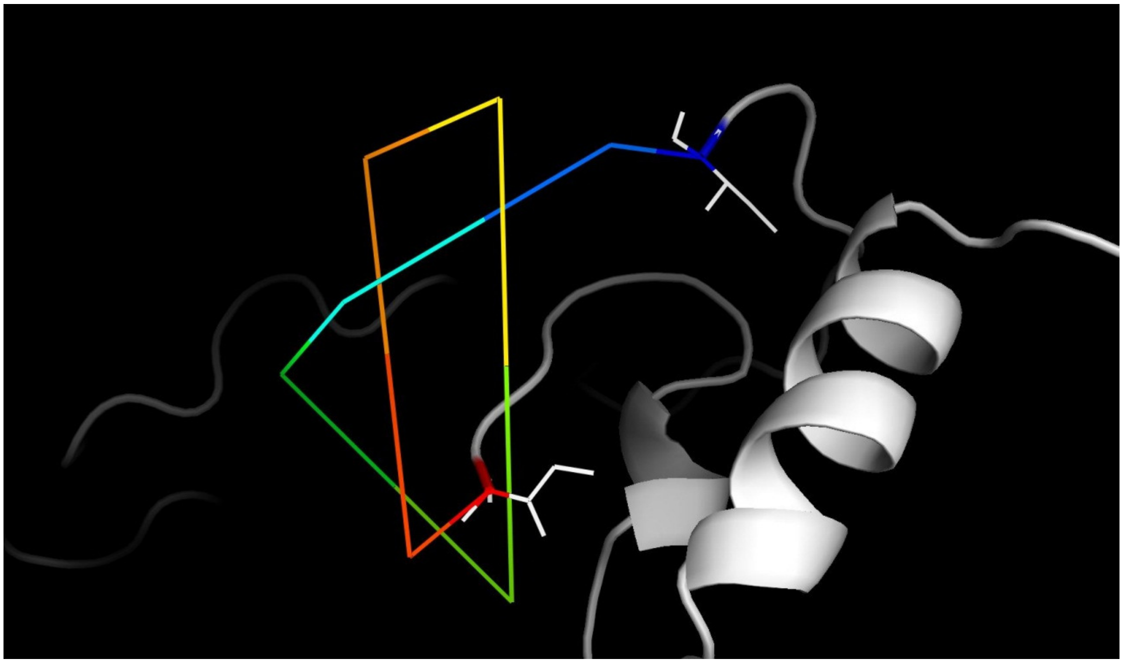


Figure 19: Atoms connected to highlight slipknot.

By connecting cα atoms, the slipknot in the above PDB becomes evident. And atoms from blue to through red define the knot core and the slipknot loop.

4.2.5 Visualizing Knots and Slipknots using JSmol Molecular Visualizer

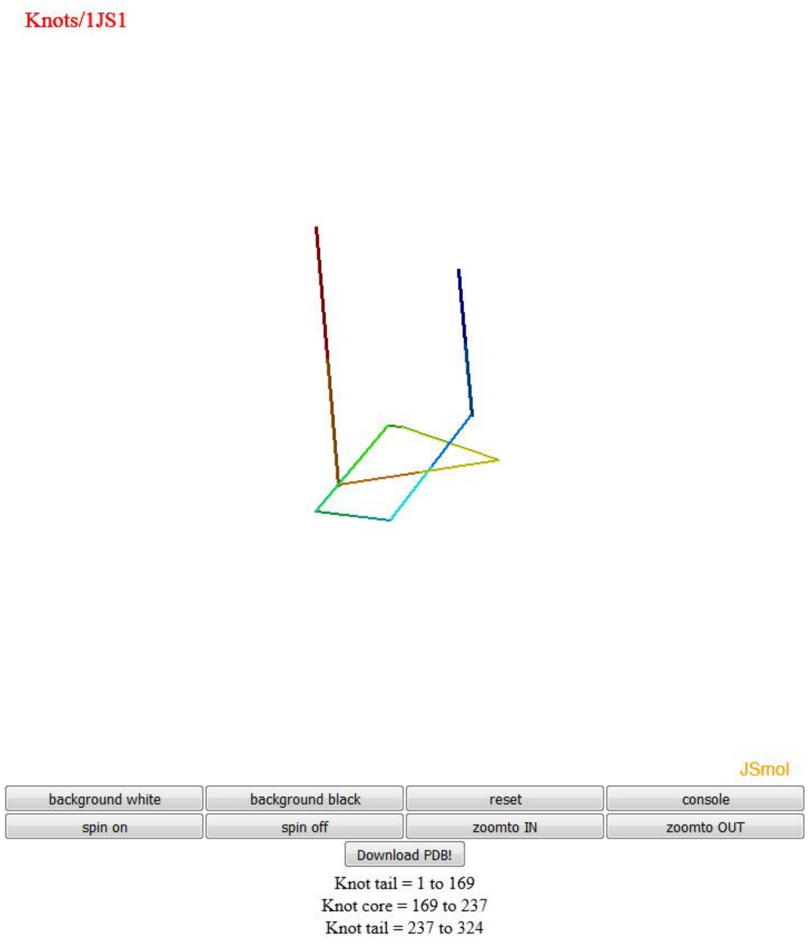
Knotfind and Slipknotfind were initially implemented using Java, using PyMol we could enable its reach only to PC's, but we wanted biochemists and structural biologists to use Knotfind and Slipknotfind to process their protein structures to detect knots from anywhere across the globe so we came up with an idea to extend the abilities of Knotfind and Slipknotfind to a website that can be accessed from anywhere in the world with a stable internet connection.

The java class that was used to find the knots has been extended using a JSP page to create an interface to upload a PDB file. The uploaded PDB file is then passed to the Knotfind and Slipknotfind class which runs on the apache tomcat server along with the chain ID that has to be processed. Once the analysis is complete, the server generates a simplified PDB and passes it to the browser along with details that are needed to highlight the knot.

The connection between the server and the client is maintained using sessions and browser cookies. The pdb ID, server path, atoms which constitute the knot or slipknot are sent to the browser using the browser cookies.

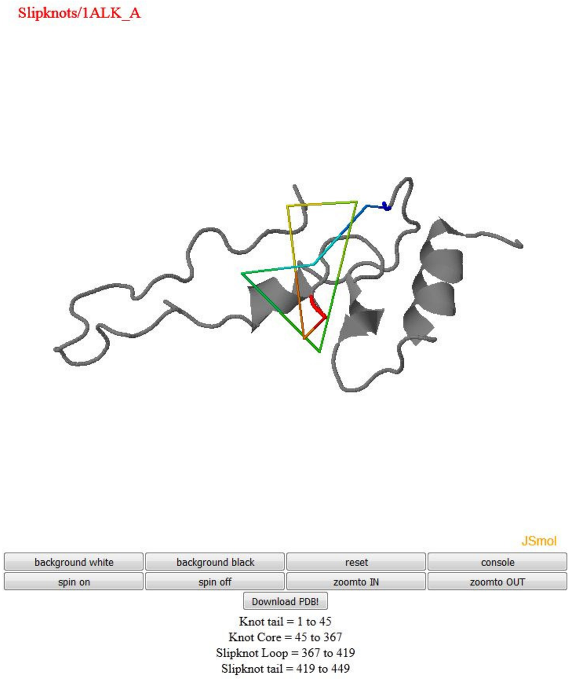
JSmol a Javascript molecular visualization tool equipped with a library of its own is used to highlight the knot on the browser. JSmol is an extension of Jmol a Java based molecular visualizer, JSmol supports all the commands that Jmol console accepts.

A JSmol script using JSmol API is loaded along with the PDB. The script initializes JSmol and connects the appropriate atoms based on the type of the knot in the given PDB and highlights the knot. The browser window also allows the users to call some basic JSmol functions to rotate or zoom into the PDB to have a closer look. The window also gives the knotting fingerprint of each protein after computing the knot.



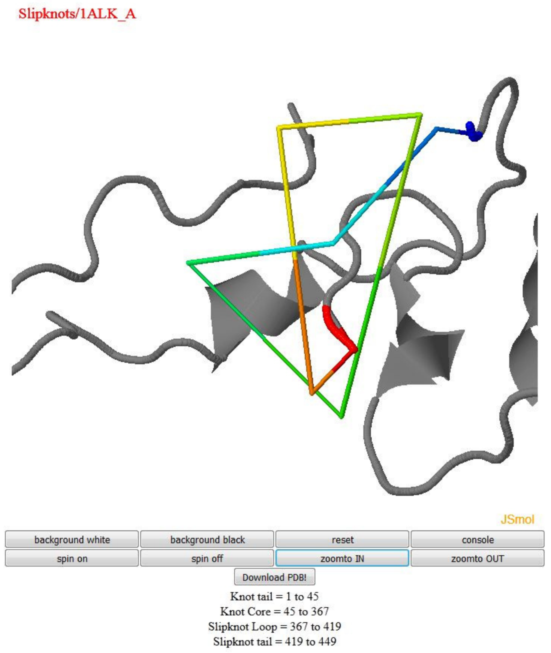
Knot tail = 1 to 169
Knot core = 169 to 237
Knot tail = 237 to 324

Figure 20: Highlighting the knot in 1JS1.



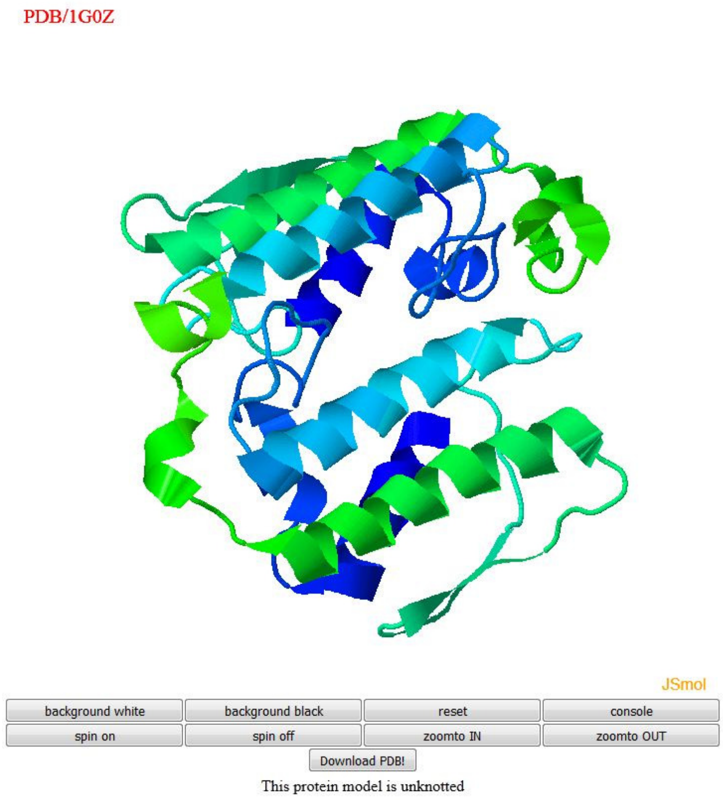
Knot tail = 1 to 45
Knot Core = 45 to 367
Slipknot Loop = 367 to 419
Slipknot tail = 419 to 449

Figure 21: Slipknot in 1ALK.



Knot tail = 1 to 45
Knot Core = 45 to 367
Slipknot Loop = 367 to 419
Slipknot tail = 419 to 449

Figure 22: Closer look of slipknot in 1ALK.



This protein model is unknotted

Figure 23: Closer look of slipknot in 1ALK.

If the uploaded PDB is not knotted, then the entire PDB is displayed in a three dimensional view with a message stating that the structure is not knotted. The on screen functions are to give better understanding of the knot in the structure

1. Spin: Spin on and off rotates the model 360 degrees enabling the user to inspect the knot from various views.
2. Zoomto: Zoomto in and out functions aid in getting a closer look of the protein structure. Each click zooms in or out 2X.
3. Download PDB: Enables the user to download the simplified protein structure.
4. Console: The user can execute any Jmol/JSmol command as per their need.

5. Conclusion and Future Work

In this thesis, we introduce slipknotfind, an extended version of already existing knotfind algorithm. We use the knotfind's approach of removing the atoms which do not intersect with the triangle formed by the chosen triplet of alpha carbon atoms and iteratively extend apply the knotfind algorithm to the increasing length of subsections of the protein chain starting from residual size of three atoms. When a knot is found initially for a subsection and it disappears in a larger subsection in the later iterations using longer subsections of the same protein chain then we report the chain to be slipknotted.

We then have created two different schemes of visualizing knots and slipknots, the first approach was done locally using standalone protein visualizer PyMol and we included a python script to connect the simplified atoms and generate a clean visualization highlighting knots or slipknots in the chain. The second approach was designed to reach limitation of the first approach, accessibility from anywhere. So, we came up with a web interface of analyzing and visualizing knots or slipknots. We have established a web server using Java sever pages to handle the processing of knots and slipknots, and JSmol to handle the visualizations.

In future, this system can be improved in a number of ways to add more functionality.

1. Compute the Alexander polynomial to determine the number of crossings in a given slipknot by connecting the ends of the chain strategically.
2. Have a database of known knots and slipknots and update the PDB list automatically every Wednesday after PDB website gets updated with new set of proteins, this saves time for biochemists who lookup already existing proteins.

REFERENCES

[1] et al. Alberts B, Johnson A, Lewis J, “The Shape and Structure of Proteins.,” *New York Garl. Sci.*, 2002.

[2] C. C. (1976) Levitt M, “Levels of Protein Structure,” *Nature 261 (5561)*. [Online]. Available: https://www.ebi.ac.uk/training/online/course/biomacromolecular-structures-introduction-ebi-reso/proteins/levels-protein-structure.

[3] B. Soediono, “The Structure of Proteins,” *Journal of Chemical Information and Modeling*, 1989. [Online]. Available: http://www.vivo.colostate.edu/hbooks/genetics/biotech/basics/prostruct.html.

[4] “Protein Folding.” [Online]. Available: http://chemwiki.ucdavis.edu/Biological_Chemistry/Proteins/Protein_Structure/Protein_Folding#Resources.

[5] M. Cieplak and S. Niewieczerzal, “Hydrodynamic interactions in protein folding,” vol. 130, no. 12, p. 124906, 2009.

[6] H. Kadokura, F. Katzen, and J. Beckwith, “Protein disulfide bond formation in prokaryotes.,” *Annu. Rev. Biochem.*, vol. 72, pp. 111–135, 2003.

[7] “X-ray Crystallography.” [Online]. Available: http://chemwiki.ucdavis.edu/Analytical_Chemistry/Instrumental_Analysis/Diffraction/X-ray_Crystallography.

[8] “What is NMR?” [Online]. Available: http://chem.ch.huji.ac.il/nmr/whatisnmr/whatisnmr.html.

[9] J. L. S. Milne, M. J. Borgnia, A. Bartesaghi, E. E. H. Tran, L. a. Earl, D. M. Schauder, J. Lengyel, J. Pierson, A. Patwardhan, and S. Subramaniam, “Cryo-electron microscopy - A primer for the non-microscopist,” *FEBS J.*, vol. 280, no. 1, pp. 28–45, 2013.

[10] “RCSB.” [Online]. Available: http://deposit.rcsb.org/adit/docs/pdb_atom_format.html.

[11] “pdb101.” [Online]. Available: http://pdb101.rcsb.org/.

[12] “PyMol.” [Online]. Available: https://www.pymol.org/.

[13] “Jmol/JSmol.” [Online]. Available: http://jmol.sourceforge.net/ .

[14] F. Khatib, M. T. Weirauch, and C. a Rohl, “Rapid knot detection and application to protein structure prediction.,” *Bioinformatics*, vol. 22, no. 14, pp. e252–9, 2006.

[15] N. P. King, E. O. Yeates, and T. O. Yeates, “Identification of Rare Slipknots in Proteins and Their Implications for Stability and Folding,” *J. Mol. Biol.*, vol. 373, no. 1, pp. 153–166, 2007.

[16] G. M. Crippen, “Topology of globular proteins.,” *J. Theor. Biol.*, vol. 45, pp. 327–338, 1974.

[17] M. L. Mansfield, “Are there knots in proteins?,” *Nat. Struct. Biol.*, vol. 1, no. 4, pp. 213–214, 1994.

[18] W. R. Taylor, “A deeply knotted protein structure and how it might fold.,” *Nature*, vol. 406, no. 6798, pp. 916–919, 2000.

[19] P. Virnau, L. A. Mirny, and M. Kardar, “Intricate Knots in Proteins: Function and Evolution,” *PLoS Comput. Biol.*, vol. 2, no. 9, p. e122, 2006.

[20] S. N. Ansari and S. Iliyas, “a Comparative Study of Protein Structure Visualization Tools for Various Display Capabilities,” *Biosci. Discov. An Int. J. Life Sci.*, vol. 2, no. 2, 2011.

[21] “Visualizing the Secondary Structure of a Protein.” [Online]. Available: vlab.amrita.edu,. (2012). Visualizing the Secondary Structure of a Protein. Retrieved 8 January 2016, from vlab.amrita.edu/?sub=3&brch=275&sim=1427&cnt=1 .

[22] O. J. and S. A. Sulkowska JI, Rawdon EJ, Millett KC, “Conservation of complex knotting and slipknotting patterns in proteins,” *Proc. Natl. Acad. Sci.*, 2012.

[23] J. I. Sułkowska, P. Sułkowski, and J. N. Onuchic, “Jamming proteins with slipknots and their free energy landscape,” *Phys. Rev. Lett.*, vol. 103, no. 26, pp. 2–5, 2009.

[24] K. C. Millett, E. J. Rawdon, A. Stasiak, and J. I. Sułkowska, “Identifying knots in proteins.,” *Biochem. Soc. Trans.*, vol. 41, no. 2, pp. 533–7, 2013.

[25] K. Koniaris, “Self-entanglement in ring polymers,” vol. 95, no. August, pp. 2873–2881, 1991.

[26] G. Kolesov, P. Virnau, M. Kardar, and L. a Mirny, “Protein knot server: detection of knots in protein structures.,” *Nucleic Acids Res.*, vol. 35, no. Web Server issue, pp. W425–8, 2007.

[27] C. C. Adams and T. R. Govindarajan, “The Knot Book: An Elementary Introduction to the Mathematical Theory of Knots,” *Phys. Today*, vol. 48, no. 4, p. 89, 1995.

[28] P. Virnau, Y. Kantor, and M. Kardar, “Knots in globule and coil phases of a model polyethylene,” *J. Am. Chem. Soc.*, vol. 127, no. 43, pp. 15102–15106, 2005.

[29] G. Kolesov, P. Virnau, M. Kardar, and L. A. Mirny, “Protein knot server: detection of knots in protein structures,” *Nucleic Acids Res.*, vol. 35, no. Web Server, pp. W425–W428, 2007.

[30] P. Bradley, K. M. S. Misura, and D. Baker, “Toward high-resolution de novo structure prediction for small proteins.,” *Science*, vol. 309, no. 5742, pp. 1868–1871, 2005.

[31] O. Nureki, M. Shirouzu, K. Hashimoto, R. Ishitani, T. Terada, M. Tamakoshi, T. Oshima, M. Chijimatsu, K. Takio, D. G. Vassylyev, T. Shibata, Y. Inoue, S. Kuramitsu, and S. Yokoyama, “An enzyme with a deep trefoil knot for the active-site architecture,” *Acta Crystallogr. Sect. D Biol. Crystallogr.*, vol. 58, no. 7, pp. 1129–1137, 2002.

[32] M. Jamroz, W. Niemyska, E. J. Rawdon, A. Stasiak, K. C. Millett, P. Sułkowski, and J. I. Sulkowska, “KnotProt: a database of proteins with knots and slipknots.,” *Nucleic Acids Res.*, vol. 43, no. Database issue, pp. D306–14, 2015.

[33] O. J. and S. A. Sulkowska JI, Rawdon EJ, Millett KC, “Conservation of complex knotting and slipknotting patterns in proteins,” *Proc. Natl. Acad. Sci.*, vol. E1715–

E172, 2012.