

University of the Philippines Manila  
College of Arts and Sciences  
Department of Physical Sciences and Mathematics

STRIDE Protein Topology  
Cartoon Generator and Database  
(SPTCGaD)

A Special Problem in Partial Fulfillment  
of the Requirements for the  
Degree of Bachelor of Science in Computer Science

by

John Patrick S. La-anan

2008-39655

April 2012

## ACCEPTANCE SHEET

The Special Problem entitled “STRIDE Protein Topology Cartoon Generator and Database (SPTCGaD)” prepared and submitted by John Patrick S. La-anan in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

\_\_\_\_\_  
**Ma. Sheila A. Magboo, M.S.**  
 Adviser

**EXAMINERS:**

	<b>Approved</b>	<b>Disapproved</b>
1. Gregorio B. Baes, Ph.D. (candidate)	_____	_____
2. Avegail D. Carpio, M.S.	_____	_____
3. Richard Bryann L. Chua, M.S.	_____	_____
4. Aldrich Colin K. Co, M.S. (candidate)	_____	_____
5. Vincent Peter C. Magboo, M.D., M.S.	_____	_____
6. Geoffrey A. Solano, M.S.	_____	_____
7. Bernie B. Terrado, M.S. (candidate)	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

\_\_\_\_\_  
 Avegail D. Carpio, M.S.  
 Unit Head  
 Mathematical and Computing Sciences Unit  
 Department of Physical Sciences and  
 Mathematics

\_\_\_\_\_  
 Marcelina B. Lirazan, Ph.D.  
 Chair  
 Department of Physical Sciences  
 and Mathematics

\_\_\_\_\_  
 Reynaldo H. Imperial, Ph.D.  
 Dean  
 College of Arts and Sciences

## Abstract

STRIDE Protein Topology Cartoon Generator and Database (SPTCGaD) is a system which converts 3D representation of protein domains to its 2D form based on the knowledge-algorithm STRuctural IDentification (STRIDE). 2D representations are used since it gives better visualization of the structures that builds up the protein unlike its 3D form which may pose complexity especially when dealing with multiple and architecturally complex protein domains. The classification of protein domains used in the system is Class, Architecture, Topology and Homologous Structures (CATH) which describes a more defined domain definition focusing only on the two main secondary structure: alpha helix and beta strand.

## Table of Contents

CHAPTER 1: INTRODUCTION.....	1
A. Background of the Study .....	1
B. Statement of the Problem.....	9
C. Objectives of the Study .....	11
D. Significance of the Study.....	13
E. Scope and Limitations.....	14
CHAPTER 2 : REVIEW OF RELATED LITERATURE .....	16
CHAPTER 3: THEORETICAL FRAMEWORK .....	21
A. Protein Topology .....	21
B. Class, Architecture, Topology and Homologous Structure .....	21
C. Structural Protein Motifs .....	24
a) Bundles.....	24
b) Trefoils .....	25
c) Barrels .....	25
d) Rolls .....	26
e) Sandwiches.....	27
D. Protein Databank File .....	27
E. Protein Cartoon Generator .....	29
F. STRIDE Algorithm .....	31
G. Existing System of STRIDE Protein Topology Cartoon Generator and Database.....	35
H. Database Management System .....	41
I. Information System.....	42
J. Definition of Terms .....	42
CHAPTER 4: DESIGN AND IMPLEMENTATION.....	45
A. Context Diagram .....	45
B. Use-Case Diagram .....	46
C. Activity Diagrams.....	48
D. Flowcharts .....	54
E. Process Explosion .....	57
F. Entity-Relationship Diagram.....	63
G. Data Dictionary .....	63
H. Technical Architecture .....	64

CHAPTER 5: RESULTS .....	66
CHAPTER 6: DISCUSSION .....	90
Trefoil.....	91
Rolls .....	93
Alpha Barrel.....	95
Other Architectures Existing in SPTCGaD.....	97
Two-Layer Sandwich .....	97
Three-Layer Sandwich (aba).....	99
Orthogonal Bundle .....	101
Up-Down Bundle .....	103
Alpha Solenoid.....	105
Alpha Beta Barrel.....	107
Ribbon .....	109
Single Sheet.....	111
Orthogonal Prism.....	112
3-Propellor.....	114
5-Propellor.....	115
7-Propellor.....	116
CHAPTER 7: CONCLUSION .....	119
CHAPTER 8: RECOMMENDATION .....	120
CHAPTER 9: BIBLIOGRAPHY .....	122
CHAPTER 10: APPENDIX.....	<b>Error! Bookmark not defined.</b>
CHAPTER 11: ACKNOWLEDGEMENT .....	258

## CHAPTER 1: INTRODUCTION

### A. Background of the Study

Because of the advancement made in protein studies, researchers have been relying on the 3D representation of protein structures. [1] These types of representations are often quite complex to understand especially when the amino acid sequences are long or the protein structure itself is composed of many secondary structure elements. These elements involve coils, sheets, strands and turns which are known to be the building blocks of protein structures.

Study of protein secondary structure element is important because it is used for visualization, structure comparison and classification and homology modeling. [2] Also, the geometry of these secondary structure elements contributes to the complexity especially when the said structures tend to group close to each other and oriented into different directions. Figure 1 shows an example of a complex protein fold consisting of different secondary structure elements oriented in many directions.

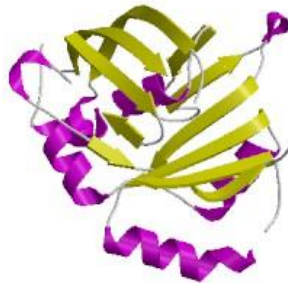


Figure 1. 3D representation of 2g3w Protein fold

A solution to the problem of analyzing a protein structure is to present the 3D representation data into its 2D form. 2D representations show connectivity between protein motifs and the direction of the structures from the amino (N) terminus to the carboxy (C) terminus. Figure 2 shows an example of converting a 3D to its 2D representation. [3] These information are used for protein searching and fold comparison to show structural patterns and spatial arrangement of protein architectures. These processes can be used in the field of medicine to determine protein structure, possible binding sites as well as other protein-protein interactions. For example water-soluble proteins tend to have their hydrophobic residues buried in the middle of the protein, whereas hydrophilic side-chains are exposed to the aqueous solvent. Thus, hydrophobic and hydrophilic interactions can affect also a protein's shape.

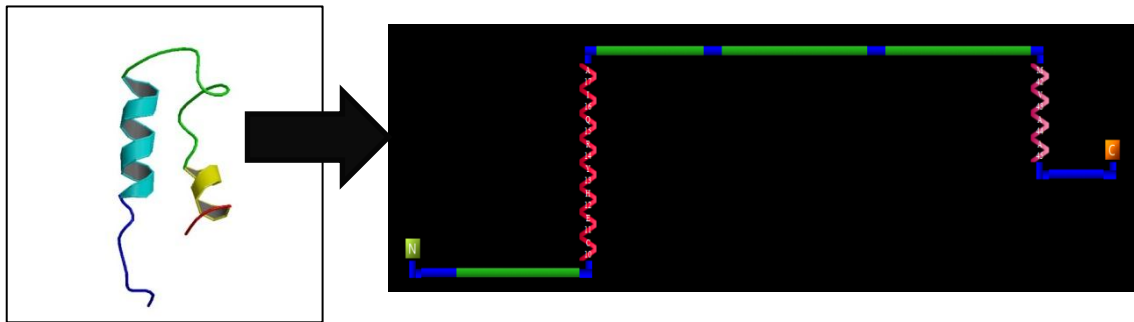


Figure 2. Conversion of 3D to 2D representation of a protein domain

Protein topology is defined to be features of a protein structure, the secondary structure elements, which are not changed when a protein is distorted. [4] The definition suggests that if a protein fold will be represented in 2D from its 3D representation, protein topology is the one that is to be considered. It is because the protein will be projected in many ways while the structural elements, their connectivity and their orientations, will be preserved. It is important to know the topology of a protein as it describes the spatial

adjacency within protein folds and it approximates orientation, neglecting details like loops and length of secondary structure elements, which may be difficult to be observed in 3D representation.

Of the simplified representations of converting 3D to 2D, protein topology cartoon provides a powerful way of presenting the relative arrangements of secondary structure elements and sequence-based features within a two-dimensional approximation of protein structure. It is because it offers a simplified diagrammatic representation in perpendicular view, looking side-on to the secondary structure elements. It uses shapes such as circles and triangles to represent secondary structure elements. Topology cartoons are commonly used in comparisons between structurally related proteins. [5]

To generate protein topology cartoons, knowledge-based algorithms are used. These algorithms are used to produce assignments of protein data, particularly the secondary structure sequence, which will describe the composition of the protein structure. For example, they can identify as to which part of the protein sequence depicts a helix structure. The algorithms use the 1D or the linear amino acid sequence as an input to generate the output secondary structure sequence. [6]

There are many knowledge-based algorithms that were used to analyze protein structures. STRucture IDentification (STRIDE) is one of those. This algorithm considers the hydrogen bonding energy and statistically derived backbone angle, which is the main factor for protein geometry, based on a given protein's secondary structure. The backbone angles provide the position and geometry of the protein secondary structure elements which identifies their orientation when the structures are folded in its 3D form. The hydrogen



bonding energy tells if two secondary structure elements are affected by forces such as repulsion and attraction. It helps in describing the spatial neighborhood of the protein fold in 3D. [2]

The information about the linear sequence of protein (amino acid sequence) to which the 2D representation is based-on can be obtained by accessing a database which consists of Protein Databank (PDB) format files. The PDB files containing the protein information is uploaded in the central repository. PDB files are named based on an identification code, the PDB id. It will then be parsed using the STRIDE algorithm. The output will be the file used to generate a simpler and detailed model of the protein structure in 2D representation. [6]

There are a lot of protein folds that are already been discovered and interpreted into linear sequences of amino acids.[8] These are then grouped based on structural similarity. Class, Architecture, Topology and Homologous structures (CATH) is one of the many protein classifications system. It classifies a diversity of protein folds that are grouped using protein structure descriptors such as a protein's class, architecture, topology and homologous structures. [9] Figure 3 shows a list of protein domains based on CATH. The classification mechanisms are interpreted to a code to identify the protein fold which then corresponds to a particular PDB id. CATH also uses its representative protein domain to group diversity of protein folds. [8]

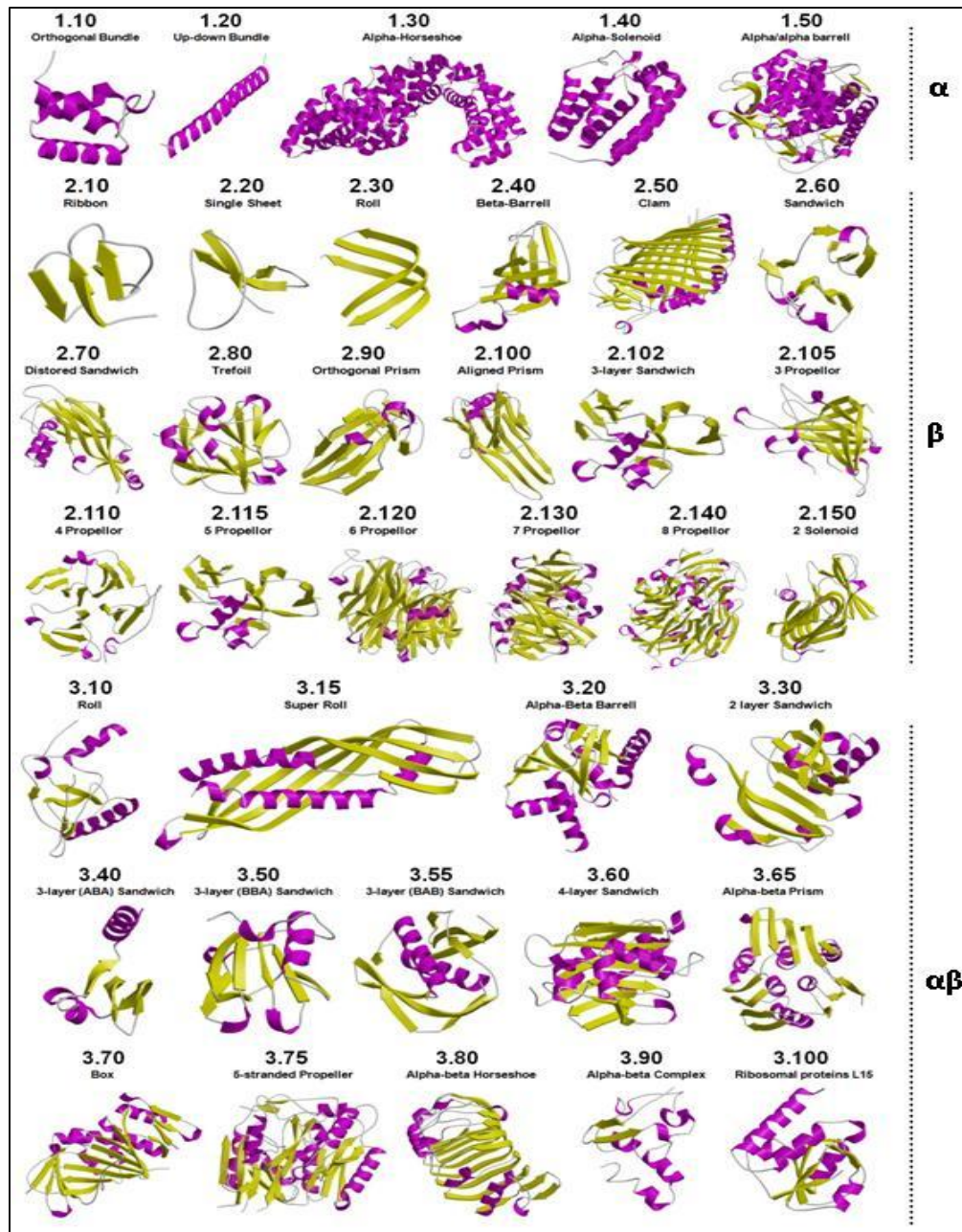


Figure 3. Representative protein domains of CATH

Topology of Protein Structures (TOPS) is a system which generates a 2D representation of protein domains using cartoon diagrams but its website has been down for maintenance since 2008. Figure 4 shows the current offline status of the TOPS website which can be accessed by <http://www.tops.leeds.ac.uk/>.

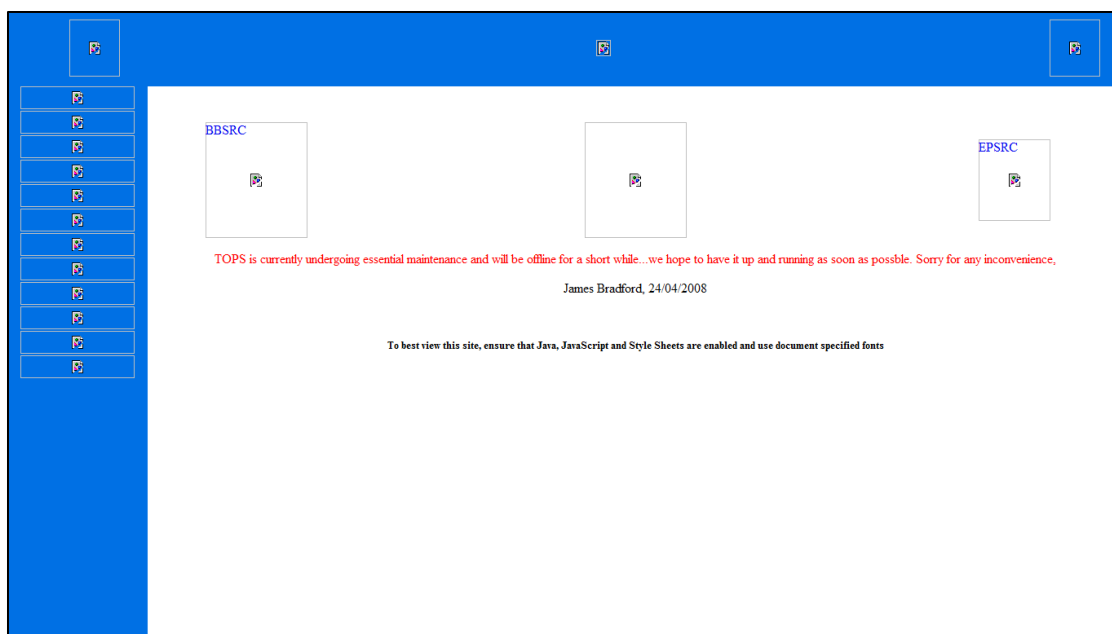


Figure 4. TIPS website down for maintenance

There is already an existing system for the STRIDE Protein Topology Cartoon Generator and Database (SPTCGaD), created by BS Computer Science students enrolled in Software Engineering class of Prof. Sheila A. Magboo and Dr. Vincent Peter C. Magboo, which supports protein classification on CATH and are grouped based on their representative protein domains. Figure 5 shows the existing system of SPTCGaD in the Agila server which can be accessed through link <http://agila.upm.edu.ph/~mmanlangit/protein/>. The STRIDE algorithm being used in the existing system is implemented using an external application written in C language.

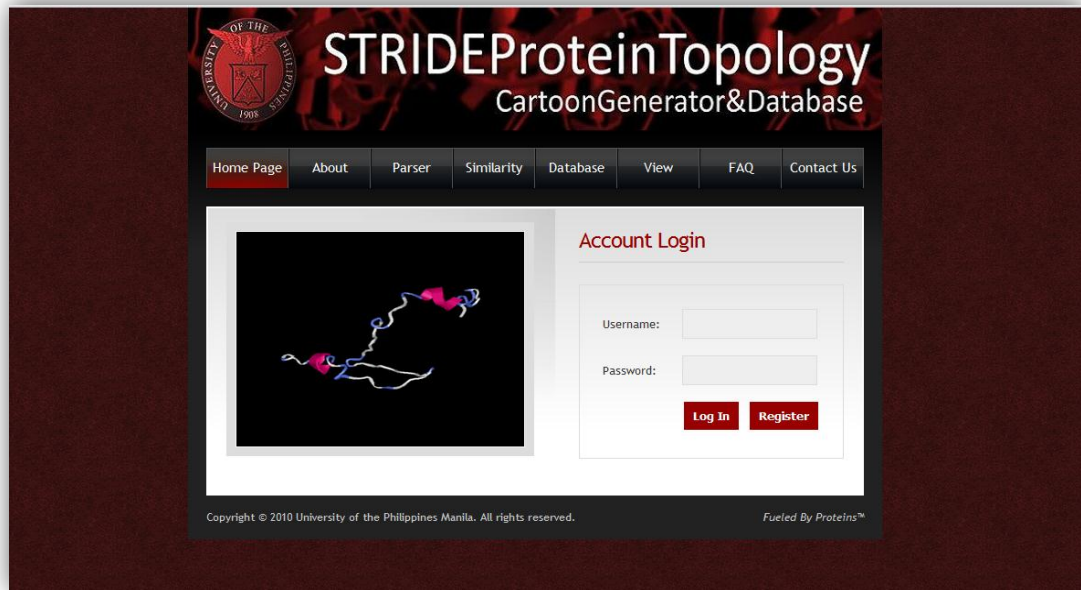


Figure 5. Existing STRIDE Protein Topology Cartoon Generator and Database

There are three types of users of the system: the non-registered user, the registered user and the system administrator. Below are the user functionalities of the existing system of SPTCGaD:

1. Allows all the users to
  - a. View Protein Topology Cartoon Diagrams
  - b. Compare Protein Sequences in terms of structural alignment such as Needleman-Wunsch, Smith-Waterman, ClustalW, FSA, MSA, and FASTA
  - c. Send Messages
2. Allows the registered users to
  - a. Login to the system
  - b. View STRIDE output after parsing a PDB file
  - c. Download STRIDE output files after parsing a PDB file
  - d. Upload STRIDE output to parse

The problem in the existing system is that there are only a limited number of protein folds based on representative protein domain that can be viewed in its 2D topological cartoon representation. Shown in table 1 is the updated list of architectures of representative protein domains based on the CATH Database website <http://www.cathdb.info/class.html>, and whether an algorithm already exists to generate the corresponding 2D representation. Available architectures are the ones that have algorithms to display protein in 2D in the existing system. Non-available architectures are the ones whose algorithms to generate 2D representation of protein are not yet available in the existing system. Protein domains that do not fall under the list of representative protein domains are classified as irregular.

Representative Protein Domain	Representative Protein Domain Subfamily	Architecture Availability
1. Bundles	1.1 Orthogonal Bundle	Available
	1.2 Up-Down Bundle	Available
2. Barrels	2.1 Alpha-Beta Barrel	Available
	2.2 Alpha Barrel	Not Available
	2.3 Beta Barrel	Not Available
3. Ribbon	Ribbon	Available
4. Single Sheet	Single Sheet	Available
5. Sandwiches	5.1 Sandwich	Not Available
	5.2 2-Layer Sandwich	Available
	5.3 3-Layer Sandwich	Not Available
	5.4 3-Layer (aba) Sandwich	Available
	5.5 3-Layer(aab) Sandwich	Not Available
	5.6 3-Layer(bba) Sandwich	Not Available
	5.7 3-Layer(bab) Sandwich	Not Available
	5.8 4-Layer Sandwich	Not Available
	5.9 Distorted Sandwich	Not Available
6. Prisms	6.1 Orthogonal Prism	Available
	6.2 Aligned Prism	Not Available
	6.3 Alpha-Beta Prism	Not Available
7. Propellers	7.1 3-Propellor	Available
	7.2 5-Propellor	Available
	7.3 7-Propellor	Available
	7.4 4-Propellor	Not Available
	7.5 6-Propellor	Not Available
	7.6 8-Propellor	Not Available
	7.7 5-stranded Propeller	Not Available
8. Solenoids	8.1 Alpha-Beta solenoid	Available
	8.2 2-Solenoid	Not Available
	8.3 3-Solenoid	Not Available
9. Horseshoes	9.1 Alpha Horseshoe	Not Available
	9.2 Alpha-Beta Horseshoe	Not Available

10. Rolls	10.1 Roll 10.2 Super Roll	Not Available Not Available
11. Clams	Clam	Not Available
12. Trefoils	Trefoil	Not Available
13. Complexes	13.1 Beta Complexes 13.2 Alpha-Beta Complexes	Not Available Not Available
14. Box	Box	Not Available
15. Ribosomal Protein	Ribosomal Protein L15	Not Available
16. Irregular	Irregular	Not Available

Table 1. List of representative protein domains for SPTCGaD based on CATH Groupings

One of the goals of this project is to continue the generation of 2D representation of other protein folds using the STRIDE algorithm.

## B. Statement of the Problem

There are many methods to study proteins. One method is to describe the protein's secondary structures based on their sequences. These protein sequences are obtained using human experiments which will be translated into data, in the form of a PDB file, and stored in various databases online. [7] The data must then be analyzed in order to have a deeper understanding of the various characteristics of the protein. This study of protein secondary structures can be used in structure comparison, classification and modeling which is a major step in characterization especially in a newly determined protein structure. [2]

Visual representation of a given protein's architecture can be quite complex and may be difficult to be extracted with some important details. In relation to visual representation of proteins, a 3D representation would not be enough or may impose complexity in describing a specific part of a given protein. A 2D representation is instead taken into consideration as it

simplifies and describes important characteristics of protein parts like its connections and directions based on protein fold orientation as opposed to the 3D representation especially when given a complex-structured protein. These connections and directions are used for protein searching and fold comparison to show structural patterns and spatial arrangement of protein architectures. [3] The 2D representations also identify a protein's structural motifs which cannot easily be identified when viewing in 3D. [10] The 1D representation, on the other hand, gives only the linear sequence particularly the amino acid and secondary structure sequences.

There is already an existing central repository system that uses STRIDE algorithm for generating 2D protein motif representations online. The structures there are based on CATH groupings. CATH is used instead of other protein domain classification such as Structural Classification of Proteins (SCOP) because the former defines smaller protein domains which are better in visualization than the latter having to define large protein domains which may be difficult to analyze. The problem is that there are only limited protein structure motifs that can be generated in that system. An objective of this project is to continue what has been made with the existing system by creating the algorithms using STRIDE output that would generate the other 2D protein structural motifs.

The available protein architecture algorithms in the existing system of SPTCGaD do not have documentations on as to how theoretically and programmatically they are defined and created. This poses an issue of having no historical resource, which may be a basis of other architecture algorithms that will be created in the future.

In the existing system also of SPTCGaD, some of the functionalities of the three types of users are not complete. For example, the send message to the system administrator functionality for all users does not completely work. It is because the inbox interface for the system administrator to read the messages sent by the users does not exist.

### **C. Objectives of the Study**

The main goal of the Stride Protein Topology Cartoon Generator and Database (SPTCGaD) system is to provide protein topology information by parsing uploaded PDB files using the knowledge-based algorithm of STRIDE (STRuctural IDentification) and use that to create 2D representations of protein folds based on CATH groupings. The topology information of protein provided in the website includes structural assignments of protein secondary structure elements which were derived from atomic coordinates from the STRIDE output file generated.

Using the atomic coordinates generated by the STRIDE output, a visualization of the 2D structure is displayed for a given protein. This would be able to aid the users in better understanding of proteins as opposed to the 3D structure.

The main objective of this project is to create functionalities and to continue existing functionalities that will define each type of user of the system. Another objective is to contribute to the main goal of the SPTCGaD system, completing the architecture algorithms of the protein domains.



The three types of users of the existing SPTCGaD system are the non-registered user, registered user and system administrator. In this project, the three types of users are retained and additional functionalities are added. Below is the list of additional functionalities that are implemented:

1. Allows the non-registered users to
  - a. Create an account
2. Allows the system administrator to
  - a. View messages
  - b. Delete messages
  - c. Reply to messages
  - d. Approve an account
  - e. Disapprove account

Trefoil, Roll and Alpha Barrel are the architectures made in this version of SPTCGaD. Existing architectures will still be available when parsing STRIDE output files to generate cartoon diagrams.

Along with the creation of new architecture algorithms, the existing architecture algorithms in the SPTCGaD system are documented as defined by theoretically (the expert Mr. Jerome Panibe) and programmatically (using a pseudo code of the algorithm).

## **D. Significance of the Study**

The SPTCGaD is a system which provides architecture-specific proteins and their 2D graphical representations. It is a web-based system that will display protein topology information based on a user-inputted protein STRIDE output file.

3D representation is converted to 2D because the former is quite complex to understand specially when dealing with multiple or architecturally complex chains of protein. 2D representation gives the user a better visualization of the structure and connectivity of proteins.

This site would be beneficial to educators, researchers and students who are concerned in the study of protein analysis based on their secondary structures which can be represented in a 2D cartoon.

STRIDE algorithm is used since it considers both the hydrogen bonding and the geometry of the amino acids in generating an output file as opposed to other algorithms which consider only one. [2] This would be useful in the deeper understanding of protein structures as the obtained information from the algorithm can lead to extents of variability in the study of a given protein especially in protein sequence and protein fold comparisons.

Since this version of the project continued the SPTCGaD system by adding three protein architecture algorithms namely trefoil, alpha barrel and roll; it contributes to the main goal of the system, to complete all architecture algorithms. The documentations of all available architectures (those which have already algorithms to generate its 2D representation) are created which will establish the historical resource for the whole

SPTCGaD project as it may be the basis for the creation of other protein architecture algorithms.

## **E. Scope and Limitations**

The STRIDE Protein Topology Cartoon Generator and Database (SPTCGaD) site provides information about the topology and architecture of the secondary structure of proteins. An existing source file, in this case STRIDE output and PDB files, must be uploaded in the database in order for such information of a specific protein structure to be displayed in the site. The SPTCGaD does not connect to central repositories such as the Research Collaboratory for Structural Bioinformatics (RCSB) site to run the architecture algorithms and obtain the cartoon diagrams of those PDB files uploaded in their system.

Uploaded files which are not yet parsed using STRIDE algorithm must be in PDB format which can be downloaded in sources such as in the RCSB Protein Databank. Other file formats will not be accepted as an input in the PDB file to STRIDE output file parsing algorithm.

Representative protein domains that are provided in the database are the ones included in the list of CATH groupings. The other hierarchical classifications such as sequence family, orthologous family, “like” domain, identical domain and domain counter (SOLID) will be of less concern as CATH will be the major classification rule that will be used in this project. [11] Also, PDB id will be used to identify a protein fold instead of the whole CATH code in the system.

The parsing algorithm of PDB files to STRIDE output files has already been implemented and thus, will not be the focus of the project. Instead, the implementation of generating the other protein motif families' 2D representation provided by the CATH groupings would be the main concern.

STRIDE output files that will be uploaded are assumed to be in text format (.txt) and contain a correct content which is based on the output of the parsing algorithm from PDB to STRIDE output file for the parsing algorithm to generate cartoon diagrams. Other file formats will not be accepted as an input in the STRIDE output file to cartoon file parsing algorithm.

The representative domains from CATH that were developed by the proponent in this project are: alpha barrels, trefoils and rolls. These are the architectures suggested by the expert, Mr. Jerome Panibe, to be done by the proponent.

Documentations, specifically the pseudo code and the theoretical description on as to how to model the protein domains in 2D, of all the available architecture algorithms in the existing system of SPTCGaD are included in the project.

Also, because all the information contained in the database is subject to change, the administrator is solely responsible for maintenance and for updating any obsolete data that might be uploaded.

The STRIDE algorithm and all of the structural alignment algorithms are all downloaded external applications and assumed to work properly based on the existing system.

## CHAPTER 2 : REVIEW OF RELATED LITERATURE

One of the most important parts of protein studies is identifying a protein's biological functions. Understanding a protein's 3D structure can contribute to that domain. But focusing only on using the 3D structure for protein analysis may lead to the some components or details of the protein ignored. It is because of the form complexity of protein structure.

Mono-dimensional descriptions such as its secondary structure may be able to simplify coarsely the information about the 3D form of a protein. [6] Secondary structures, because they allow a simple and intuitive description of 3D structures, are widely employed in a number of structural biology applications. [12]

Protein topology is one of the elements in characterizing proteins especially in providing their secondary structure sequence. It can be defined as the relationship between the sequential ordering of secondary structure elements and their spatial organization. [13] Protein topology is one of the principal properties by which protein structures can be classified, categorized and compared. It can be used to describe more complex local 3D motifs, for example, the Greek keys.

2D representation of protein topology structures can be useful tools in protein searching and fold comparison. [9] However, the main application of protein topology representation is, of course, to straightforwardly show the structural pattern and spatial arrangement of protein architectures. Protein topology representations can be classified into two styles: cartoon diagram and topology diagram. Cartoon diagram utilizes symbols such as triangles and circles which illustrates the composition of  $\alpha$  helices and  $\beta$  sheets. Topology

diagrams utilize arrows and cylinders which can be parallel or anti-parallel oriented. Figure 5 shows the comparison between topology and cartoon diagrams based on their symbol usage.

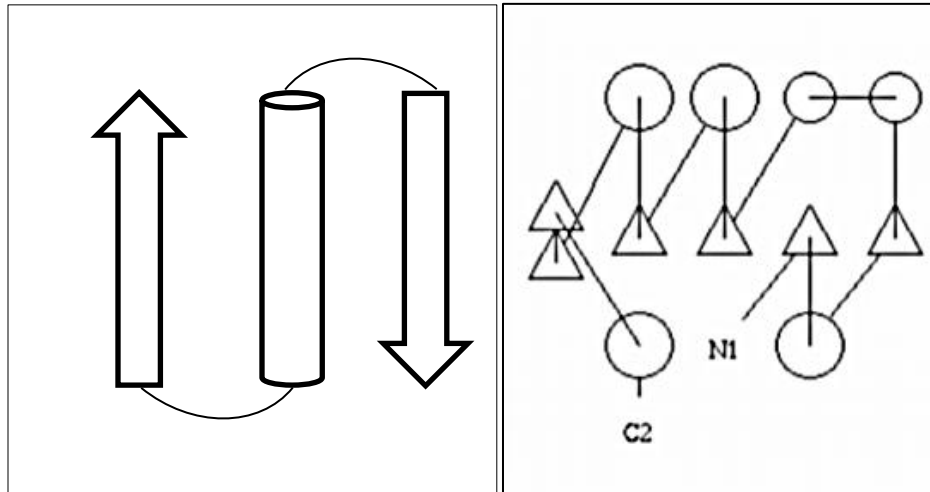


Figure 6. Topological diagram (left) and a cartoon diagram (right)

Protein 3D structures are often described as a succession of repetitive secondary structures, protein motifs which are consists of mainly alpha-helices and beta-sheets. [6] In 1951, Pauling and Corey predicted the existence of two periodic motifs in protein structures: the  $\alpha$ -helix and the  $\beta$ -sheet which turned out to be major features of protein architecture. [12] The said protein motifs are then found in lower level structures such as in the secondary and supersecondary structure of proteins.

Information about known and studied proteins is stored in an archive where people interested in protein studies can access it online. The Protein Data Bank (PDB) is the central worldwide repository for three-dimensional (3D) structure data of biological macromolecules. [7] The Research Collaboratory for Structural Bioinformatics (RCSB) is currently the main repository online providing the said protein information. One of its

features is the integration and searchability of data from over 20 other sources covering genomic, proteomic and disease relationships. [14]

The content of data stored in PDB is harvested using data curation from different sources such as source organisms, SWISS-PROT and GenBank data, enzyme commissions and PubMed data. These data are then classified using public domains CATH (Class, Architecture, Topology and Homologous Superfamily) and SCOP (Structural Classification of Proteins). These public domains use a hierarchical structural classification in assessing the data. [15]

CATH 3.2 is the latest version of CATH which contains 114215 domains, 2178 Homologous superfamilies and 1110 fold groups. [16] The classification procedure of CATH that has been done includes implementation of manual and automated techniques (using computational algorithms) wherein each protein has been chopped into structural domains and assigned into homologous superfamilies (groups of domains that are related by evolution). [11]

STRIDE is secondary structure alignment method which implements a knowledge-based algorithm that makes combined use of hydrogen bond energy and statistically derived backbone torsional angle information and is optimized to return resulting assignments in maximal agreement with crystallographers' designations. [2] It is a more precise algorithm in describing the protein secondary structure compared to others because other algorithms such as Define Secondary Structure of Proteins (DSSP) which uses single hydrogen bond energy terms to assign eight states of secondary structure elements (SSEs) to three-dimensional coordinates. [17]

Protein structures are often visualized in a simplified form, with the so-called ribbon diagram with secondary structures shown as helices and arrows being the most popular. [18] Topology of Protein Structure (TOPS) is one algorithm that automatically generates protein cartoon diagrams. [9,19] It is first devised by Westhead *et al* (1998, 1999). The diagrams include the sequence of secondary structure elements (SSEs) and two sets of relationships defined between pairs of SSEs, which are hydrogen bonds (hbonds) and chiralities. Figure 6 shows an example of a cartoon diagram generated by TOPS which is composed of lines, triangles and circles. The existing system which implements TOPS algorithm for protein cartoon diagram representation is currently not available.

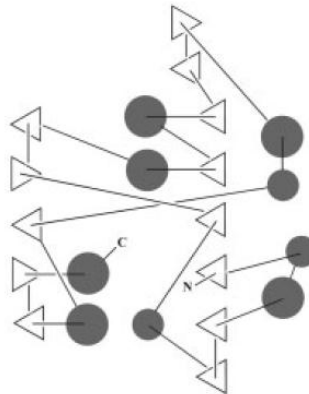


Figure 7. Example of a cartoon diagram generated by TOPS

Pro-origami is a system for automatically generating protein structure cartoons. The cartoons are intended to make protein structure easy to interpret by laying out the secondary and super-secondary structure in two dimensions in a manner that makes the structure clear. The cartoons are drawn in a style similar to those that might be drawn manually using a tool such as Charles Bond's TopDraw program, rather than the purely topological style such as those drawn by TOPS. [20]



Other protein secondary structure assignments are SECSTR, which is a development from DSSP, and XTLSSTR, which uses backbone atoms compute dihedral angles and distances. [6] Still, DSSP and STRIDE are the most popular to be used since they produce nearly identical assignments.

## **CHAPTER 3: THEORETICAL FRAMEWORK**

### **A. Protein Topology**

Protein Topology, on the fundamental level, is a sequence of protein secondary structure elements like a primary topology string. [21] It defines the relationship between sequential ordering of secondary structure elements and their spatial organization and is considered to be the one of the principal properties by which proteins can be classified, categorized and compared. [13]

Super-secondary structure motifs, such as Greek-key or  $\alpha - \beta - \alpha$  motifs, describe the interaction and position of secondary structure elements. The arrangements of secondary structure element motifs open the possibility for a topological description of protein structures. The first theoretical work on protein topologies refer to  $\beta$  structure topologies or  $\alpha$  topologies. [13]

The simplest representations of protein topologies are schematic diagrams of protein folds illustrating the secondary structure elements and their spatial neighborhoods. The representative protein topologies are given by Class, Architecture, Topology and Homologous Structures (CATH); Structural Classification of Proteins (SCOP) and Topology of Secondary Structures (TOPS). [13]

### **B. Class, Architecture, Topology and Homologous Structure**

Class, Architecture, Topology and Homologous Structures (CATH) is a manually curated classification of protein domain structures. Each protein has been chopped into

structural domains and assigned into homologous superfamilies (groups of domains that are related by evolution). This classification procedure uses a combination of automated and manual techniques which include computational algorithms, empirical and statistical evidence, literature review and expert analysis. The hierarchical domain classification used by CATH is based on the protein structures of the Protein Databank. [11]

CATH overall gives information based on a given structure classified in the database on the structure and the known functions of that protein. Evolutionary relationships involving the structure of interest and other proteins in the database can also be determined. The database of CATH gives an overall view of the known protein structure universe to date. [11]

The latest version of CATH, CATH 3.4 which is released in November 2010, contains 24,232 newly assigned domains, 163 new homologous superfamilies and 49 new folds (topologies). [11] Table 2 shows a summary of classified CATH clusters.

Class	Architecture	Topology	Homologous Superfamily	Domains
1	5	376	839	32396
2	20	228	514	39140
3	14	577	1082	79038
4	1	101	114	2346
Total	40	1282	2549	152920

Table 2. Summary of the number of clusters within each of the four classes in CATH

Each protein structure is decomposed into one or more chains which in turn are split into one or more domains before being classified into homologous superfamilies according to both structure and function. Table 3 shows the full hierarchical classification of CATH SOLID. At the Class, or C-level, the domains are classified simply on the basis of their secondary structure content [whether they are mostly  $\alpha$ -helical (Class 1) or  $\beta$ -sheet (Class 2), contain a significant percentage of both secondary structure elements (Class 3) or contain very little secondary structure (Class 4)]. The domains within each class are then sorted

according to their architecture—that is similarities in the arrangements of secondary structures in 3D space. Each architecture (A-level) is further broken down into one or more topology, or fold, groups (T-level), where the connectivity between these secondary structures are taken into account. The domains are then classified into their respective homologous superfamilies (H-level) according to similarities in sequence, structure and/or function. [16]










Depth	Letter	Name	Clustering criteria
1		Class	Secondary structure content
2		Architecture	General spatial arrangement of secondary structures
3		Topology	Spatial arrangement and connectivity of secondary structures (fold)
4		Homologous Superfamily	Manual curation of evidence of evolutionary relationship (at least two criteria from sequence/structure/function must be observed)
5		Sequence Family (S35)	$\geq 35\%$ sequence similarity
6		Orthologous Family (S60)	$\geq 60\%$ sequence similarity
7		“Like” domain (S95)	$\geq 95\%$ sequence similarity
8		Identical domain (S100)	100% sequence similarity
9		Domain counter	Unique domains

Table 3. Full hierarchical classification of CATH SOLID

Codes are used to represent protein structures in the CATH database. Table 4 shows an example of CATH coding of 1nr3. The first major level, Class, clusters proteins based on their general secondary structure content and are represented by the first number in the CATH code.[11]

Domain	CATH code	C	A	T	H	S	O	L	I	D
1nr3A00	3.30.1190.10.1.1.1.1.1	3	30	1190	10	1	1	1	1	1

Table 4. CATH coding example for 1nr3

A domain identifier is assigned to every classified domain in the CATH database. It consists of a 4-character PDB code, for example 1kcm, followed by the chain name, denoted by a letter, and a two-digit domain number. If there is only one chain, it will be assigned the letter A in the same way as the first chain in a multi-chain structure. If there is only one domain in the chain then 00 is used for the domain number. The structure 1kcm has only a single domain in a single chain; the domain identifier will therefore be 1kcmA00. [11]

There has been a change of domain identifiers in the latest version of CATH due to the emergence of protein structures with more than nine domains. This is mainly because of the improve determination of protein structures by means of experimental techniques. [11]

### C. Structural Protein Motifs

#### a) Bundles

Bundles are motifs which belong in the mainly alpha class of the CATH classification. They are usually treated as a component of a large protein folding units. They are composed of helices that are adjacent with each other. The helices are then directed by turns (up-down). [11] Figure 8 shows an example of a 3D representation of a bundle, 1qqv.



Figure 8. 3D representation of 1qqv classified as mainly alpha and belongs to orthogonal bundle architecture based on CATH

### b) Trefoils

Trefoils are structures that assume an internal threefold symmetry. They are in the mainly beta class which makes them composed of beta strands that form parallel and anti-parallel orientations. [11] Figure 9 shows an example of a 3D representation of a trefoil, 1qqv.

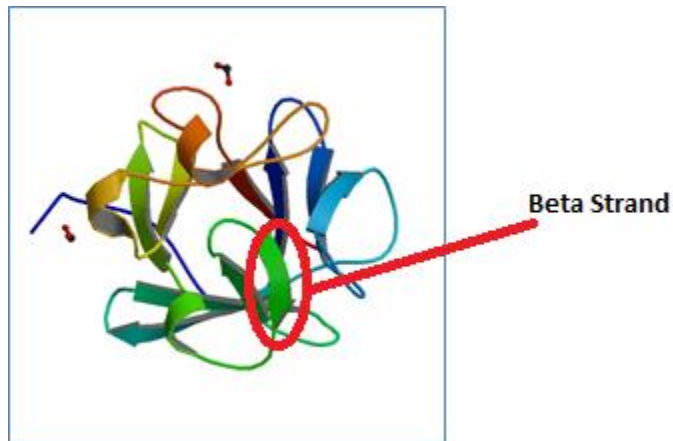


Figure 9. 3D representation of 1ipo classified as mainly beta and belongs to trefoil architecture based on CATH

### c) Barrels

It is a protein motif that is composed of parallel and anti-parallel alpha or beta strands that form a “closed barrel”. The alpha or beta strands can form Greek keys which can be a part of barrel depending on the architecture. Barrels can be identified by a ring of hydrogen bonds in the derived secondary structure information. The information includes the number of strands which composes the barrel. [22] Figure 10 shows an example of a 3D representation of a barrel, 1qqv.

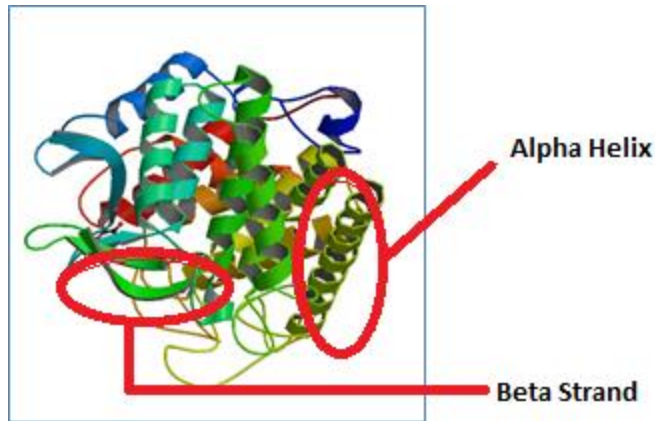


Figure 10. 3D representation of 1h12 classified as mainly alpha and belongs to the barrel architecture based on CATH

**d) Rolls**

Rolls are protein structures that can be composed of strands of alpha helices or beta. The said structures can be either parallel or anti-parallel oriented. [11] Figure 11 shows an example of a 3D representation of a roll, 3fxy.

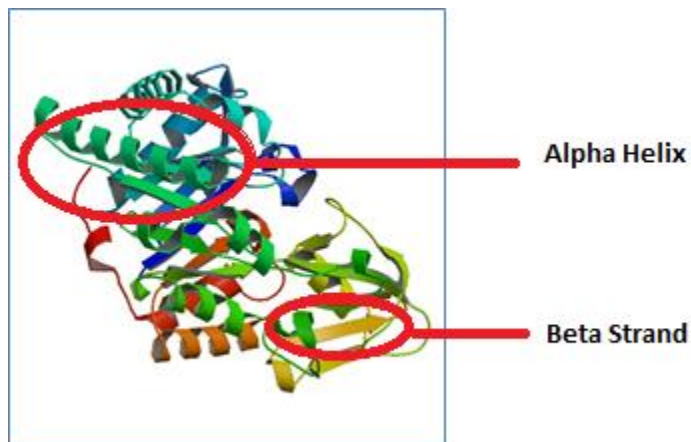


Figure 11. 3D representation of 3fxy classified as mainly beta and belong to the roll architecture based on CATH

### e) Sandwiches

Sandwiches are protein motifs that can be composed of strands, Greek keys, rolls, sheets and folds. The structures can be alpha, beta or mixed and can be in parallel or anti-parallel orientations. [11] Figure 10 shows an example of a 3D representation of a sandwich, 1kvi.

The structures are arranged in layers just like how a sandwich looks like.

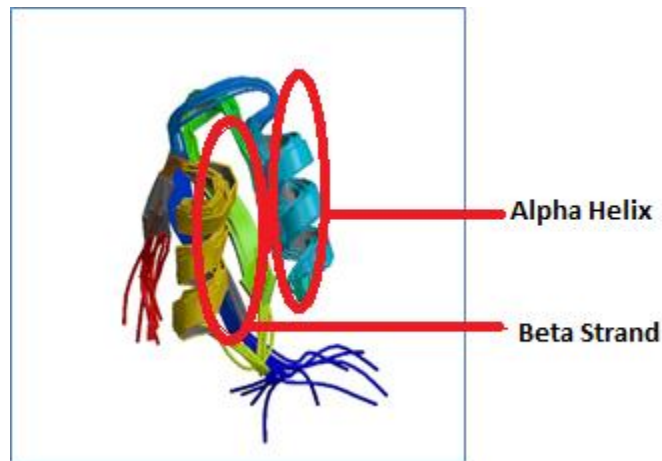


Figure 12. 3D representation of 1kvi classified as mixed alpha and beta, and belong to the sandwich architecture based on CATH

## D. Protein Databank File

Protein structures are determined by experimental methods which include NMR, X-ray and Electronic Microscopy. The experimentally solved protein structures are archived at the publicly available databases such as, Protein Data Bank (PDB). Protein Data Bank (PDB) is an internationally referred protein structure database, it comprises of the atomic coordinates of the three dimensional (3D) structure of protein. [7, 23] The coordinate file is organized in a common file format, called the PDB format.



The PDB file includes the following information in table 5. [24]

HEADER	Date entered into Data Bank; identification code
OBSLTE	Identifies entries which have been replaced
COMPND	Name of molecule and identifying information
SOURCE	Species, organ, tissue, and mutant from which the molecule has been obtained, where applicable
EXPDTA	Experimental technique of structure determination
AUTHOR	Names of contributors
REVDAT	Revision date; identifies current modification level
SPRSDE	Identifies entries which have replaced others
JRNL	Literature citation that defines coordinate set
REMARK	General remarks
SEQRES	Residue sequence
FTNOTE	Footnotes relating to specific atoms or residues
HET	Identification of non-standard groups or residues (heterogens)
FORMUL	Chemical formulas of non-standard groups
HELIX	Identification of helical substructures
SHEET	Identification of sheet substructures
TURN	Identification of hairpin turns
SSBOND	Identification of disulfide bonds
SITE	Identification of groups comprising the various sites
CRYST1	Unit cell parameters, space group designation
ORIGX	Transformation from orthogonal Å coordinates to submitted coordinates
SCALE	Transformation from orthogonal Å coordinates to fractional mcystallographic coordinates
MTRIX	Transformations expressing non-crystallographic symmetry <sup>4</sup>
TVECT	Translation vector for infinite covalently connected structures
MODEL	Specification of model number for multiple structure models in a single data entry
ATOM	Atomic coordinate records for "standard" groups
HETATM	Atomic coordinate records for "non-standard" groups
SIGATM	Standard deviations of atomic parameters
ANISOU	Anisotropic temperature factors
SIGUIJ	Standard deviations of anisotropic temperature factors
TER	Chain terminator
ENDMDL	End-of-model flag for multiple structure models

	in a single data entry
CONNECT	Connectivity records
MASTER	Master control record with checksums of total number of records in the file, for selected record types
END	End-of-entry record

Table 5. List of PDB file components

All PDB files follow a format. For a file to be considered a PDB file, the following format should be seen in the file: [25]

- \* Each line is 80 columns wide and is terminated by an end-of-line indicator.
- \* The first six columns of every line contain a "record name". This must be an exact match to one of the stated record names described in detail below.
- \* The list of ATOM records in each polymer chain must be terminated by a TER record.
- \* ATOM records for polymer atoms must include non-blank chain ID fields.
- \* Each file should terminate with a line containing only the word END.

## E. Protein Cartoon Generator

Experts derived the first protein topology diagrams as a cartoon representation of a biological point of view. These representations include schematic diagrams of protein folds illustrating the secondary structure elements and their spatial neighborhoods. [13] Figure 13 shows the cartoons used in the existing system of SPTCGaD.

Some of the secondary structure elements involve in the diagrams are:

1. Beta Strand

A beta strand refers to the building blocks of a beta sheet. It is constituted by several amino acids, usually 3-10, which are bound to one another in an almost extended configuration. [26]

## 2. Turns

Turns indicate the direction of the protein motif from N-terminal to C-terminal.

## 3. Coils

Coils are motifs which consist of series of spirals or rings.

## 4. Alpha-Helix

Alpha-helices are the major secondary structure occurring in proteins. It is a rod-like structure, stabilized by hydrogen bonds between the carboxyl group of each amino acid residue of the chain and the amino group of the amino acid situated four residues ahead in the linear sequence. [26]

## 5. $3_{10}$ Helix

$3_{10}$  helices are rare motifs in a protein wherein unlike alpha-helices, they consist of 3 residues per turn.

## 6. N-Terminal

The n-terminal refers to the start of the protein terminated by a free amine group ( $-\text{NH}_2$ ).

## 7. C-Terminal

The c-terminal refers to the end of an amino acid chain terminated by a free carboxyl group ( $-\text{COOH}$ ).

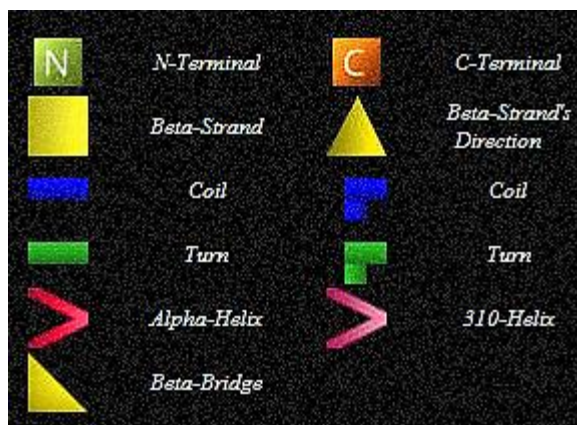


Figure 13. Topological cartoons used in the existing system of SPTCGaD

## F. STRIDE Algorithm

STRIDE is a knowledge-based algorithm for protein secondary structure assignment from atomic resolution protein structures. It was developed by Dmitrij Frishman and Patrick Argos. It considers elements such as the protein's atomic coordinates and hydrogen bonding in generating an output. [2]

In using the STRIDE algorithm, a PDB file is required which serves as the input. The generated output will contain the following output format:

Position	Description
1-3	Record code
4-5	Not used
6-73	Data
74-75	Not used
75-79	Four letter PDB code

Table 6. General format of STRIDE output file

Each line in the STRIDE output begins with a code representing a record type. Each record type has a format on as to how they are written and what information it contains in the output. Table 7 shows a detailed description of the components of the STRIDE output file.

<b>Record Type</b>	<b>Record Type Meaning</b>	<b>Description</b>	<b>Format</b>
REM	Remarks	Remarks and blank lines	Free
HDR	Header	Protein name, date of file creation and PDB code	Free
CMP	Compound	Full name of the molecule and identifying information	Free
SRC	Source	Species, organ, tissue, and mutant from which the molecule has been obtained	Free
AUT	Author/s	Names of the structure authors	Free
CHN	Chain Identifier	File name and PDB chain identifier	File name beginning from position 6 followed by one space and one-letter chain identifier
SEQ	Sequence	Amino acid sequence	6-9 First residue PDB number 11-60 Sequence 62-65 Last residue PDB number
STR	Structure	Secondary structure summary	11-60 Secondary structure assignment
LOC	Location	Location of secondary structure elements	6-17 Element name 19-21 First residue name 32-26 First residue PDB number 28-28 First residue chain identifier 36-38 Last residue name 42-45 Last residue PDB number 47-47 Last residue chain identifier
ASG		Detailed secondary structure assignment	6-8 Residue name 10-10 Protein chain identifier 12-15 PDB residue number 17-20 Ordinal residue number 25-25 One letter secondary structure code (**) 27-39 Full secondary structure name 43-49 Phi angle

			53-59 Psi angle 65-69 Residue solvent accessible area
DNR	Donor	Donor residue	6-8 Donor residue name 10-10 Protein chain identifier 12-15 PDB residue number 17-20 Ordinal residue number 26-28 Acceptor residue name 30-30 Protein chain identifier 32-35 PDB residue number 37-40 Ordinal residue number 42-45 N..O distance 47-52 N..O=C angle 54-59 O..N-C angle 61-66 Angle between the planes of donor complex and O..N-C 68-73 angle between the planes of acceptor complex and N..O=C
ACC	Acceptor	Acceptor residue	6-8 Acceptor residue name 10-10 Protein chain identifier 12-15 PDB residue number 17-20 Ordinal residue number 26-28 Donor residue name 30-30 Protein chain identifier 32-35 PDB residue number 37-40 Ordinal residue number 42-45 N..O distance 47-52 N..O=C angle 54-59 O..N-C angle 61-66 Angle between the planes of donor complex and O..N-C 68-73 angle between the planes of acceptor complex and N..O=C

Table 7. Detailed format of STRIDE output file

HDR, CMP, SCR and AUT records are directly copied from the PDB file, if supplied by the authors. If only the secondary structure summary is requested, only CHN, SEQ, STR and LOC records will be output. Hydrogen bond information (records DNR and ACC) was made very redundant to facilitate human reading and will not be reported by default. [27]

The cartoon generator main concerns the STR record type which holds the information about the secondary structure summary. The STR record type is composed of

different structure code which identifies the elements of the secondary structure of a protein. Table 8 shows a list of codes of secondary structure elements used in a STRIDE output file, in its STR record type, that can be present in a protein domain.

H	Alpha helix
G	3-10 helix
I	PI-helix
E	Extended conformation
B or b	Isolated bridge
T	Turn
C	Coil (none of the above)

Table 8. List of codes of secondary structure elements used in a STRIDE output file that can be present in a protein domain

The ASG record type holds the detailed secondary structure information (see Figure 13) which will be used in creating a protein domain's 2D representation. Columns 2 (residue name), 3 (domain letter), 4 (residue number) and 6 (secondary structure) are the ones needed in the process of creating the representation.

Figure 14 shows an example of a STRIDE output file, specifically by 1cem, and the columns needed in creating the 2D representation.

```

REM                                     1CEM
REM ----- Detailed secondary structure assignment----- 1CEM
REM                                     1CEM
REM |---Residue---| |--Structure--| |-Phi-| |-Psi-| |-Area-| 1CEM
ASG ALA A 33 1 C Coil 360.00 143.26 93.4 1CEM
ASG GLY A 34 2 T Turn 75.15 -166.44 6.3 1CEM
ASG VAL A 35 3 T Turn -53.98 126.30 15.9 1CEM
ASG PRO A 36 4 T Turn -86.69 132.25 45.3 1CEM
ASG PHE A 37 5 T Turn -68.40 -29.83 13.6 1CEM
ASG ASN A 38 6 C Coil 57.93 27.25 144.0 1CEM
ASG THR A 39 7 C Coil -100.91 172.93 23.5 1CEM
ASG LYS A 40 8 C Coil -128.32 132.10 177.0 1CEM
ASG TYR A 41 9 T Turn -67.59 139.61 11.9 1CEM
ASG PRO A 42 10 T Turn -67.73 -18.58 91.6 1CEM
ASG TYR A 43 11 T Turn -132.67 146.77 105.6 1CEM
ASG GLY A 44 12 T Turn 70.02 -170.41 41.0 1CEM
ASG PRO A 45 13 T Turn -74.35 151.19 32.7 1CEM
ASG THR A 46 14 T Turn -85.91 -33.35 32.7 1CEM
ASG SER A 47 15 T Turn -146.52 143.08 26.4 1CEM

```

Figure 14. Example of a STRIDE output file

An online STRIDE server is available in this link <http://webclu.bio.wzw.tum.de/stride/>, which offers services such as basic and visual assignment generation, contact map rendering and Ramachandran plot given a PDB file, a PDB identifier or a PDB data. Figure 15 shows an output generated in the said link. The server only generates a 1D representation as a visual assignment output.

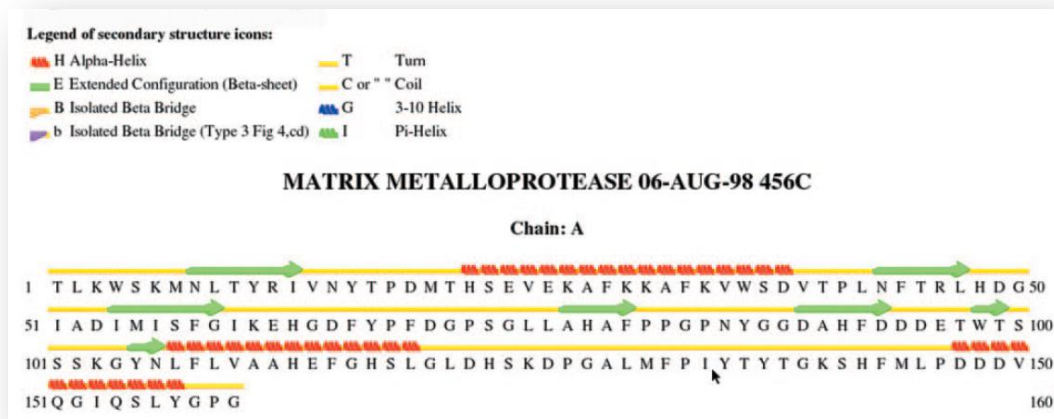


Figure 15. 1D graphical representation of a STRIDE output file in the website <http://webclu.bio.wzw.tum.de/stride/>

### G. Existing System of STRIDE Protein Topology Cartoon Generator and Database

The existing system of SPTCGaD which can be accessed online, <http://agila.upm.edu.ph/~mmanlangit/protein/>, has three types of users: a registered user, the system administrator and a non-registered user. Each has different privileges on as to what page can functionalities can be accessed in the site. The account registration is currently not available in the existing system.



The existing system has implemented 12 architectures of specific representative protein domain cartoons. It also has external programs like the STRIDE output parser which accepts PDB files as an input.

Other external program algorithms which are used for comparing protein domains are Needleman-Wunsch, Smith-Waterman, CLUSTALW, FSA, POA-MSA and FASTA are also implemented in the site.

### 1. Login Page

It is the page where registered users can start sessions in the system.

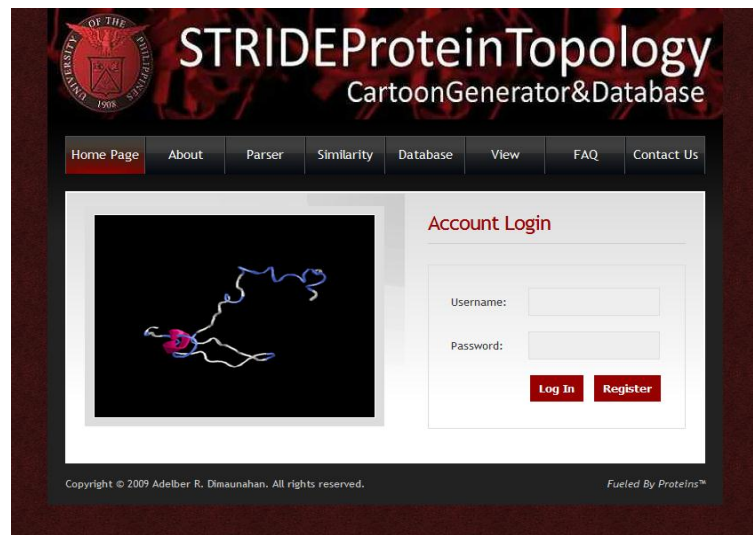


Figure 16. Login Page in the existing system of SPTCGaD

## 2. About Page



Figure 17. About page in the existing system of SPTCGaD

## 3. Compare Similarity of Protein Sequences page using Pairwise and Multiple alignments.

Alignment algorithms include Needleman-Wunsch, Smith-Waterman, CLUSTALW, FSA, POA-MSA and FASTA. The user can upload a file or choose an already uploaded file for the protein domain to be compared. The user will then choose the algorithm to be used to compare the protein domains and then click *Start Comparing* button to start comparing.

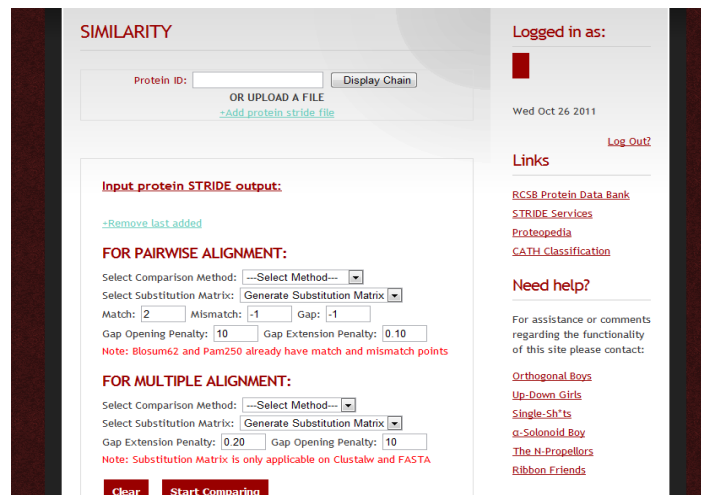


Figure 18. Compare Similarity Page of the existing system of SPTCGaD

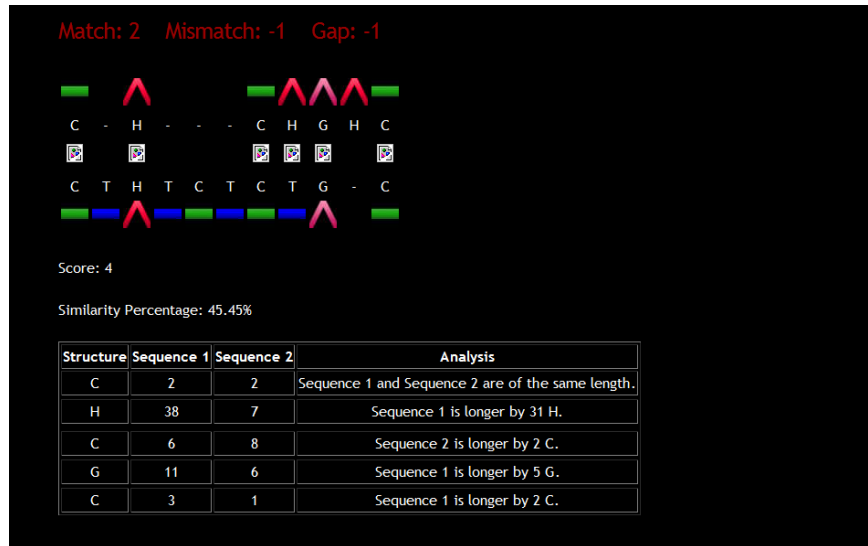


Figure 19. Example of Protein Comparison Result of 1YBZ and 1EHS (both Up-Down Architecture) using Needleman-Wunsch Algorithm

#### 4. Database Page

It is the page where a protein domain code is inputted to view its 2D topology cartoon representation.

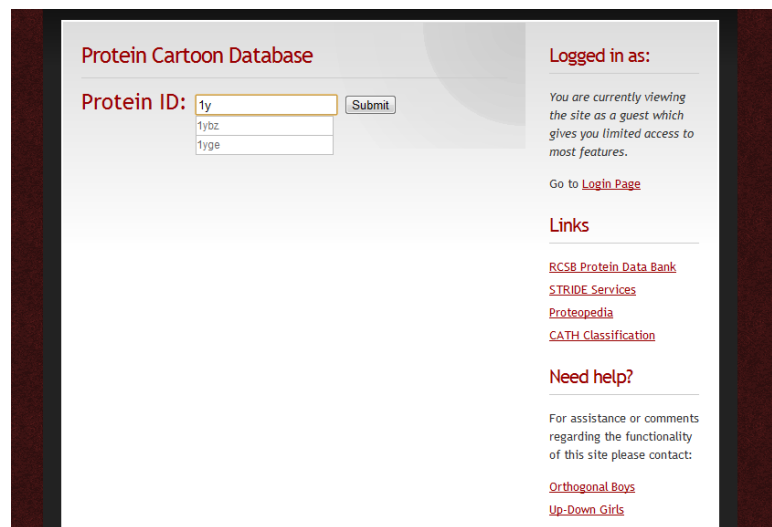


Figure 20. Search Protein Domain Page of SPTCGaD

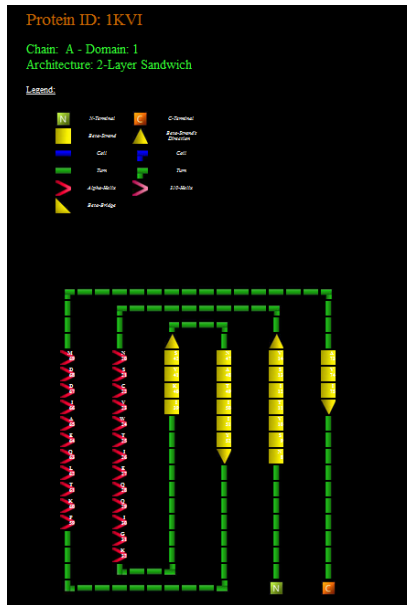


Figure 21. Example of 2D representation of 1aps (2-Layer Sandwich Architecture) in the existing system of SPTCGaD

### 5. View Sequences Page

It displays the basic information (chain, domain, architecture and sequence) of protein domains stored in the database.

View Sequences

PROTEIN ID: 1aps

CHAIN	DOMAIN	ARCHITECTURE	SEQUENCE
A	1	2-Layer Sandwich	1 - 98

[Back](#)

Logged in as:

You are currently viewing the site as a guest which gives you limited access to most features.

Go to [Login Page](#)

Links

- [RCSB Protein Data Bank](#)
- [STRIDE Services](#)
- [Proteopedia](#)
- [CATH Classification](#)

Need help?

For assistance or comments regarding the functionality of this site please contact:

- [Orthogonal Boys](#)
- [Up-Down Girls](#)
- [cinthia ch'ez](#)

Figure 22. View Sequences Result page for 1aps in the existing system of SPTCGaD

## 6. Contact Page

It is the page to send a message to the system administrator.

The screenshot shows a web page titled "Contact". On the left, there is a form with fields for "Your Name:", "Your Email:", "Subject:" (with a dropdown menu set to "Enquiry"), and "Message:". Below the form are "Reset" and "Send" buttons. On the right, there is a "Logged in as:" section showing "You are currently viewing the site as a guest which gives you limited access to most features." and a link to "Go to Login Page". Below that is a "Links" section with links to "RCSB Protein Data Bank", "STRIDE Services", "Proteopedia", and "CATH Classification". At the bottom right is a "Need help?" section with text about assistance and links to "Orthogonal Boys", "Up-Down Girls", "Single-Sh'ts", "α-Solonoid Boy", "The N-Propellers", and "Ribbon Friends".

Figure 23. Contact Page in the existing system of SPTCGaD

## 7. Parser Page

It is the page where a registered user can upload a PDB file to generate STRIDE output file and topology cartoon if architecture is implemented, or upload a STRIDE output file to generate topology cartoon if architecture is implemented. Only registered users can access this page.

The screenshot shows a web page titled "Protein STRIDE Parser". On the left, there is a form with a "Choose File" button (showing "No file chosen"), "Clear" and "Start Parsing" buttons, and a note: "\* Supported file types include txt, rtf, and doc. Files should be in STRIDE output format." On the right, there is a "Logged in as:" section showing "Jpanibe" and "System Administrator Wed Oct 26 2011" with a "Log Out?" link. Below that is a "Links" section with links to "RCSB Protein Data Bank", "STRIDE Services", "Proteopedia", and "CATH Classification". At the bottom right is a "Need help?" section with text about assistance and links to "Orthogonal Boys", "Up-Down Girls", "Single-Sh'ts", and "α-Solonoid Boy".

Figure 24. STRIDE Output Parser Page in the existing system of SPTCGaD

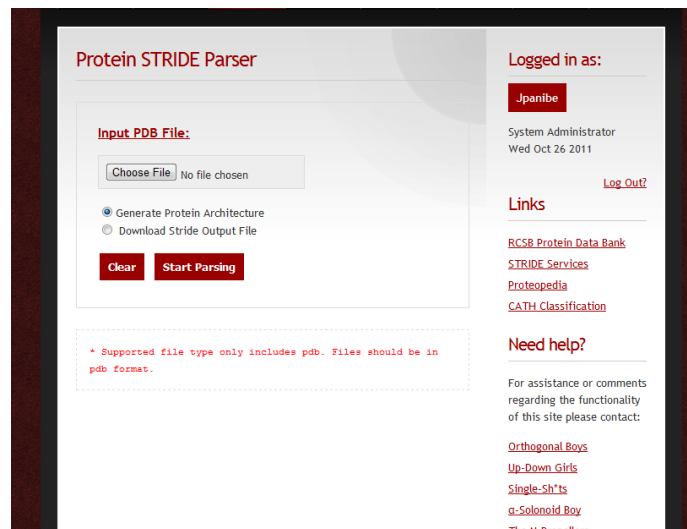


Figure 25. PDB File Parser Page in the existing system of SPTCGaD

## H. Database Management System

A Database Management System (DBMS) is simply can be referred to as a database manager. [15] It controls the organization, storage and retrieval of data (fields, records and files) in a database. The data referred to involves fields, records and files. [28]

A DBMS also controls the security of the database which means that only those with access privileges can access the data. DBMS also controls the integrity of the database which focus on handling user requests that the data in it can be continuously accessed and consistently organized and maintained. [29]

Microsoft's SQL Server, MySQL, Oracle Database, Microsoft Access and IBM's DB2 are some example of DBMSs. [29]

## I. Information System

Information Systems (IS) are collections of technical and human resources. Their goal is to provide storage, computing, distribution and communication for information required by a computer user or to meet an objective. [30] Its main components are inputs, processing mechanisms and outputs. [31]

It usually varies from different levels of organizational hierarchy based also on different types of decisions required. [30]

## J. Definition of Terms

1. Secondary Structure – refers to the basic building blocks of protein motifs.
2. Fold – it is the assumed intricate three-dimensional shape of a protein molecule.

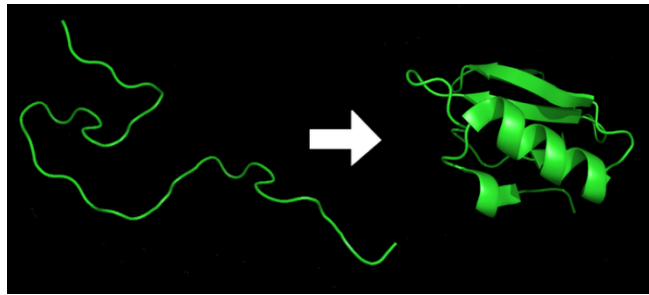


Figure 26. Protein before and after folding

3. Motif – it refers to a set of contiguous secondary structure elements that either have a particular functional significance or define an independently folded domain.
4. Greek key – it is an all-beta arrangement found in many different proteins and which topologically resemble a design found in Ancient Greek vases.

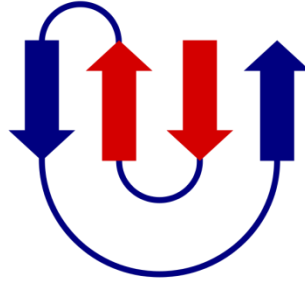


Figure 27. Greek key protein topology diagram

5. Anti-parallel – it describes the direction of two adjacent protein motifs to be opposite of each other, one upward and one downward.

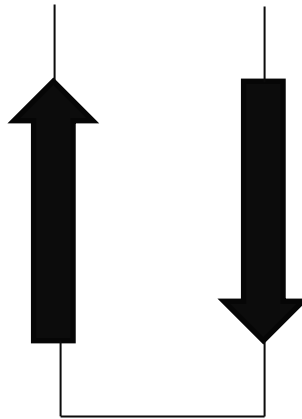


Figure 28. Topological diagram showing anti-parallelism

6. Domain – is a part of protein sequence and structure that can evolve, function, and exist independently of the rest of the protein chain. It also refers to the recurrent units of proteins. Same or similar domains can be found in different proteins.
7. Architecture – refers to the arrangement and orientation of secondary structure elements but not its connectivity.



8. Topology – refers to the manner in which the secondary structure elements are connected.
9. Chain – it is a group of amino acids which is the building block of a protein.

## CHAPTER 4: DESIGN AND IMPLEMENTATION

### A. Context Diagram

The existing system supports three types of users: system administrator, registered user and guest user. It will be also implemented in this project. The context diagram is shown in Figure 29.

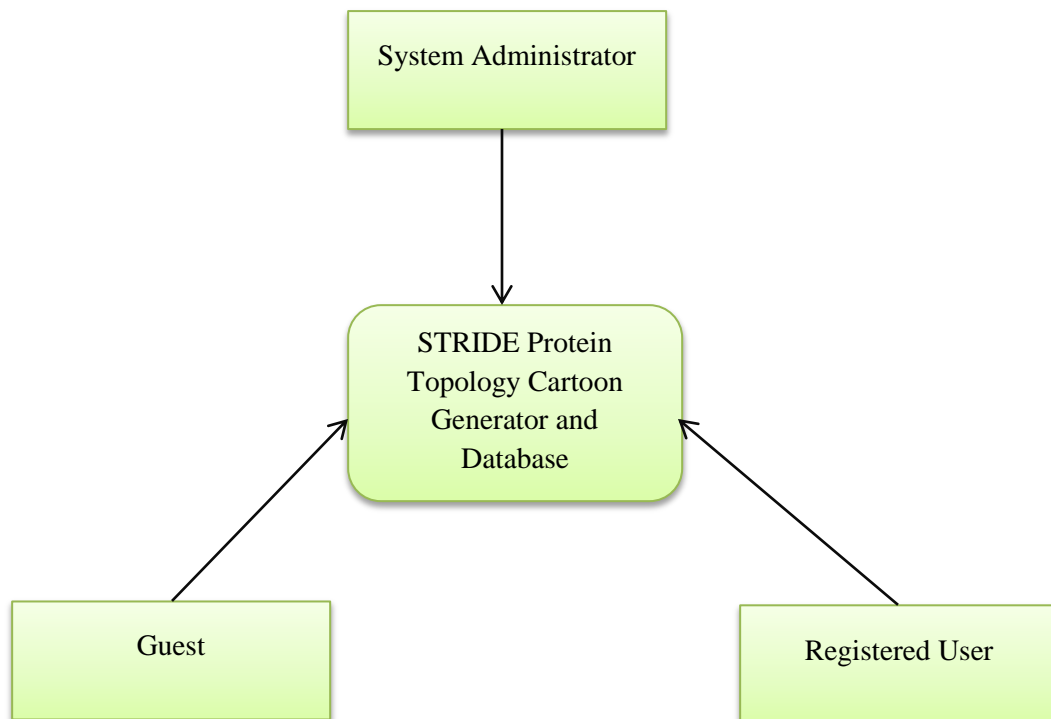


Figure 29. Context Diagram of SPTCGaD

## B. Use-Case Diagram

The existing system allows a non-registered user, can be referred to as an unregistered user, to request an account upon visiting the SPCTGaD. He/she can view protein topologies or starting and ending residues of a sequence; compare protein sequences; send messages; view STRIDE output; and download STRIDE output.

The system administrator and a registered user can do the same activities that a guest user can do but the former can log-in to the system; upload and parse a PDB file to generate a STRIDE output; upload STRIDE output to parse data.

In addition, the system administrator can view and delete messages which are sent usually to request an account or to report bugs in the system. He/she is the person who will create accounts for registered users.

Since he/she is the main person who manages the system, the system administrator is not allowed to send messages. Figure 30 shows the use-case diagram for SPCTGaD.

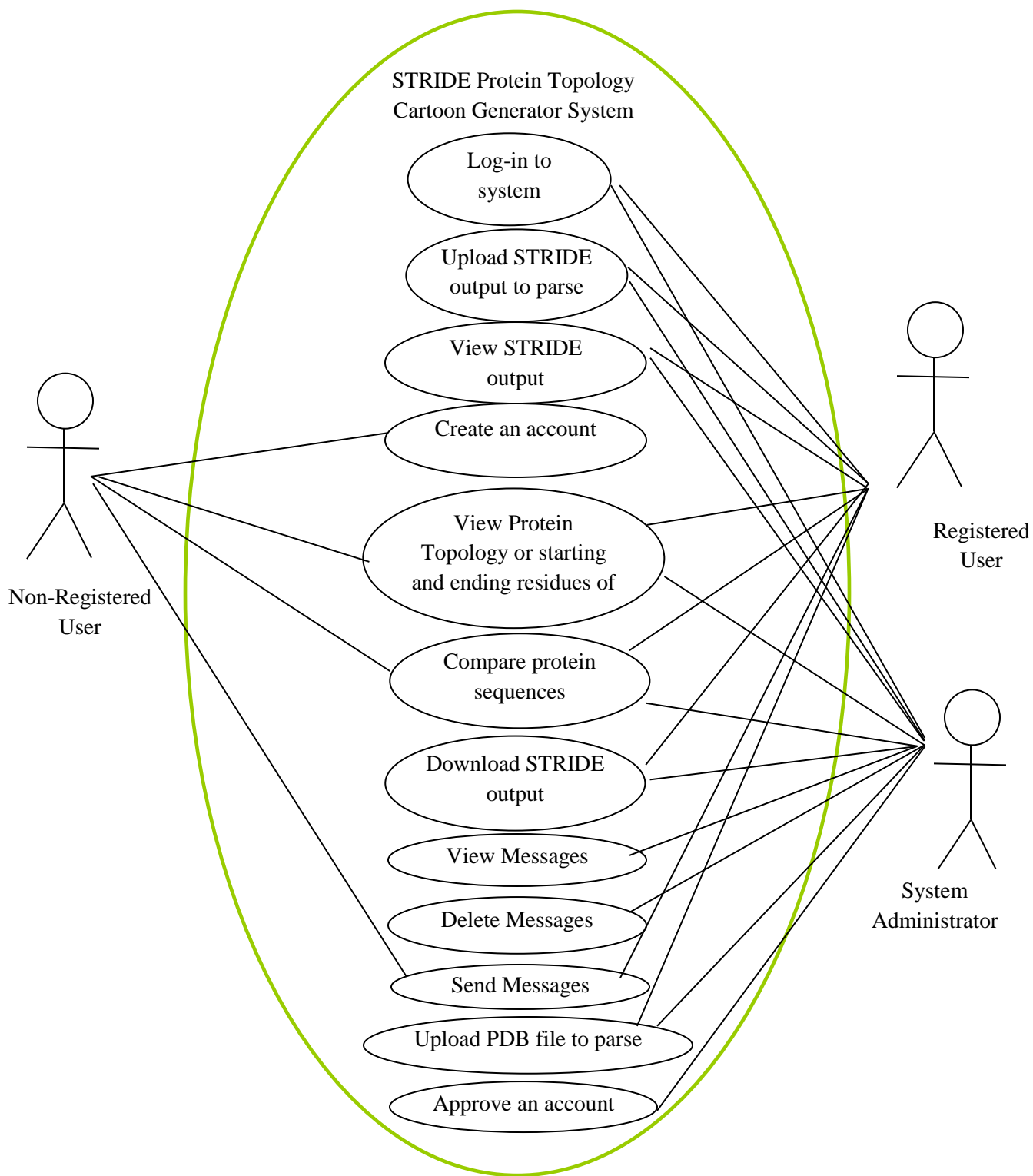


Figure 30. Use-Case Diagram for SPTCGaD

### C. Activity Diagrams

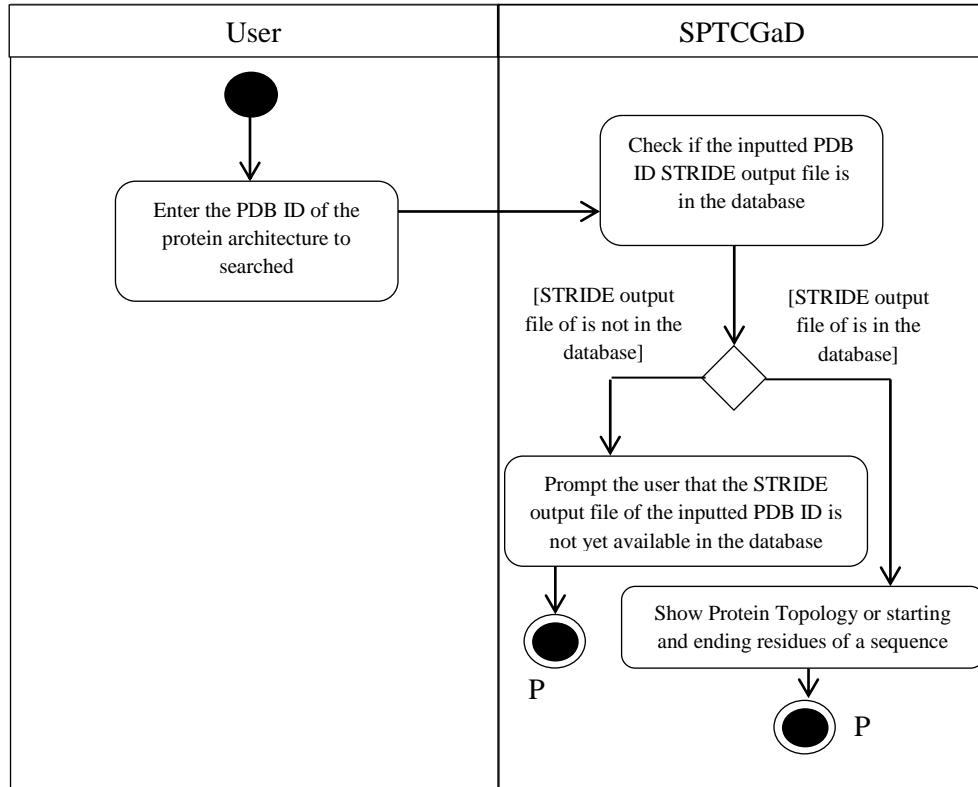


Figure 31. Activity diagram for View Protein Topology Cartoon Diagrams functionality for all users of SPTCGaD

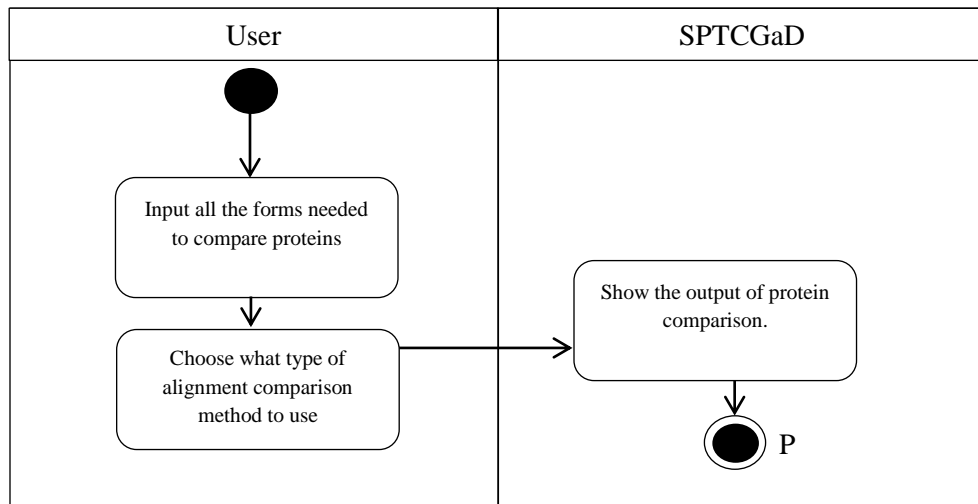


Figure 32. Activity diagram for Compare protein sequences (Using alignment methods such as Needleman-Wunsch, Smith-Waterman, CLUSTALW, FASTA, FSA and POA-MSA) functionality in SPTCGaD

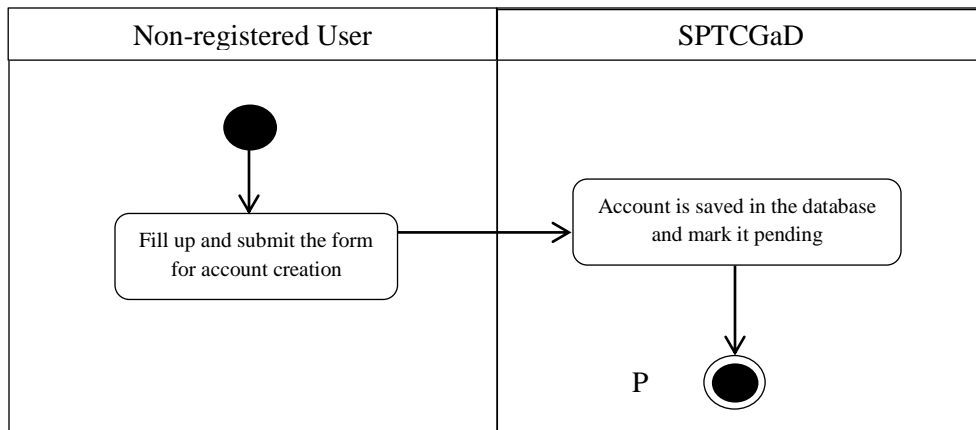


Figure 33. Create an account functionality for registered users in SPTCGaD

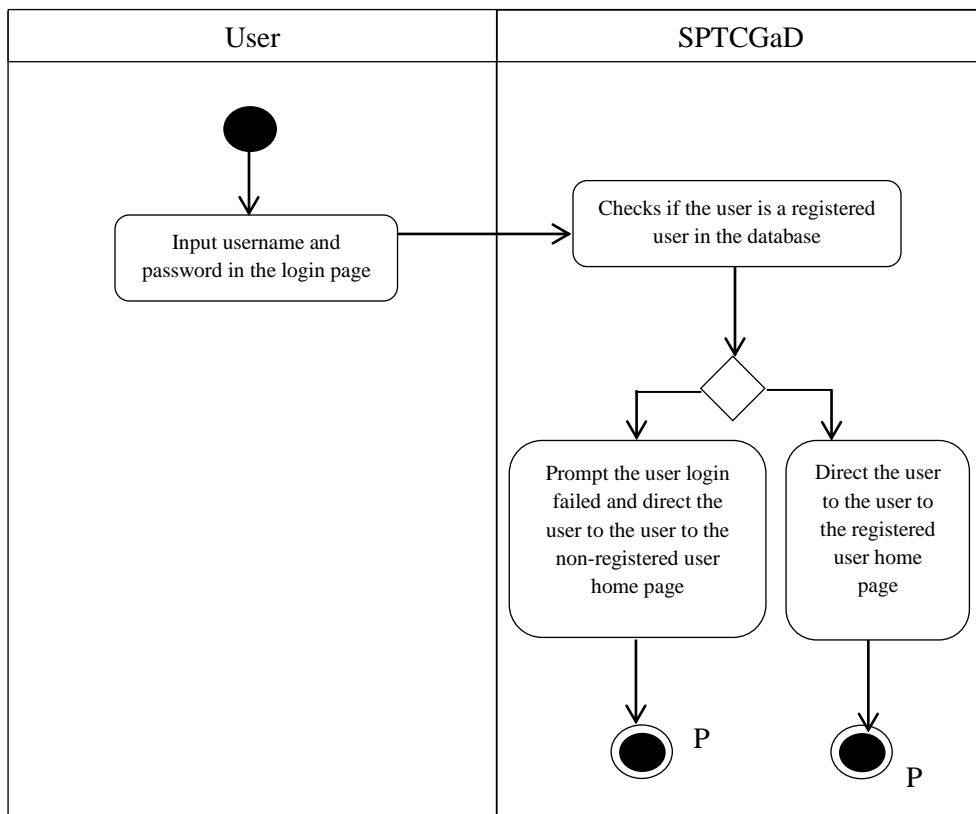


Figure 34. Login functionality for registered users in SPTCGaD

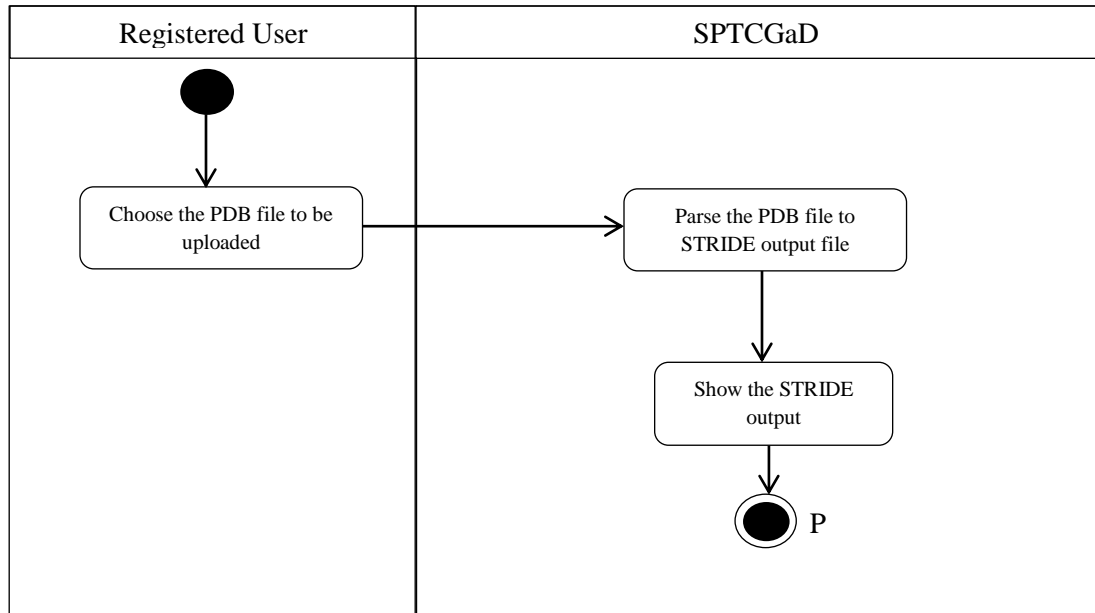


Figure 35. View Stride Output by Uploading a PDB file functionality for registered users in SPTCGaD

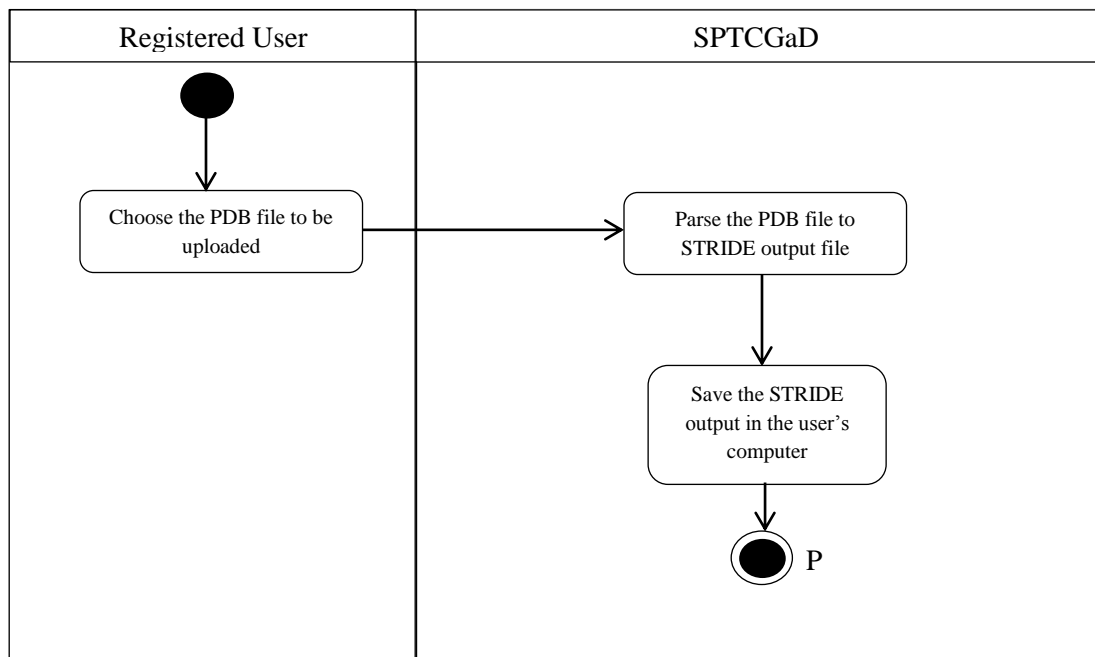


Figure 36. Download STRIDE output file by uploading a PDB file functionality in SPTCGaD

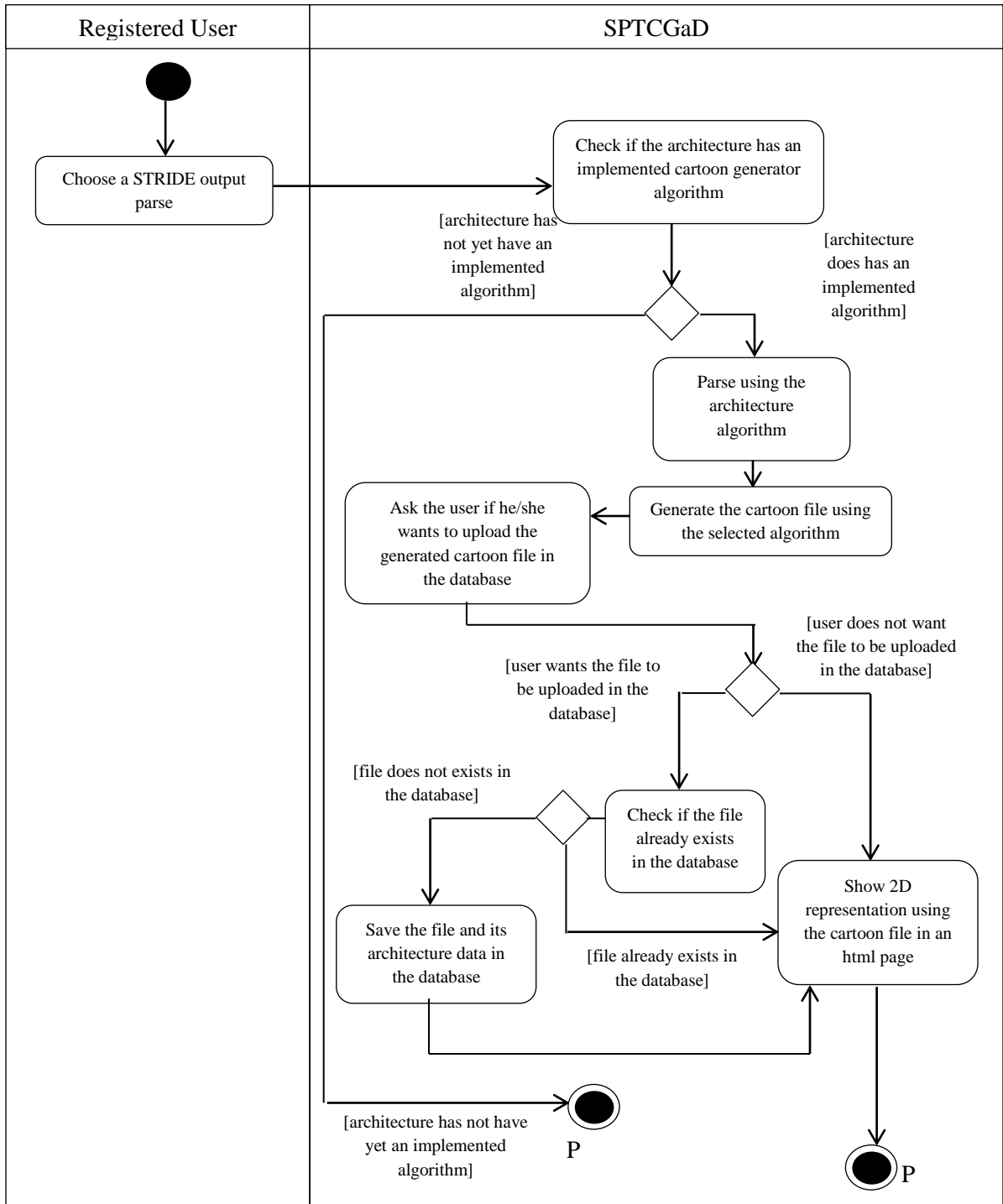


Figure 37. Activity diagram of Upload STRIDE Output File to Parse functionality for registered users in SPTCGaD



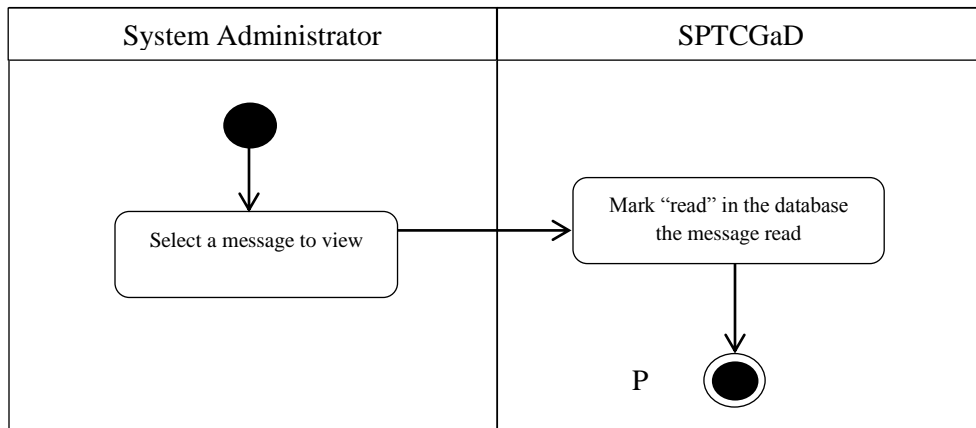


Figure 38. Activity diagram of View message functionality for the system administrator in SPTCGaD

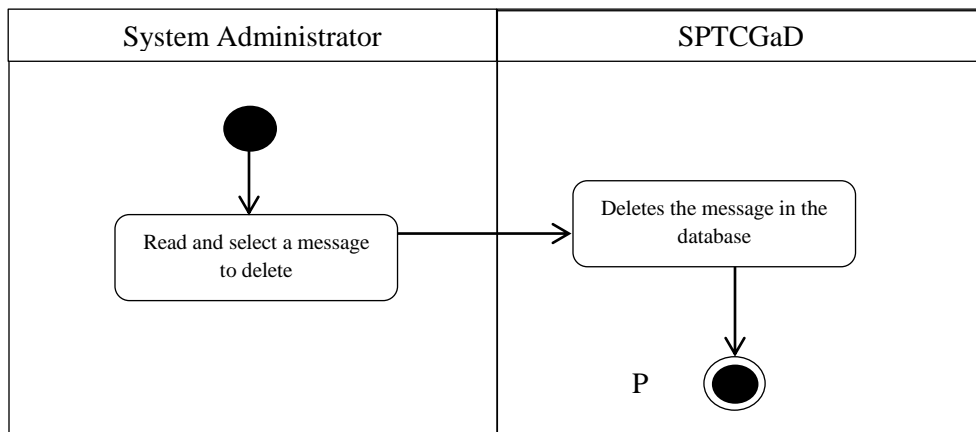


Figure 39. Activity diagram of Delete message functionality for system administrator in SPTCGaD

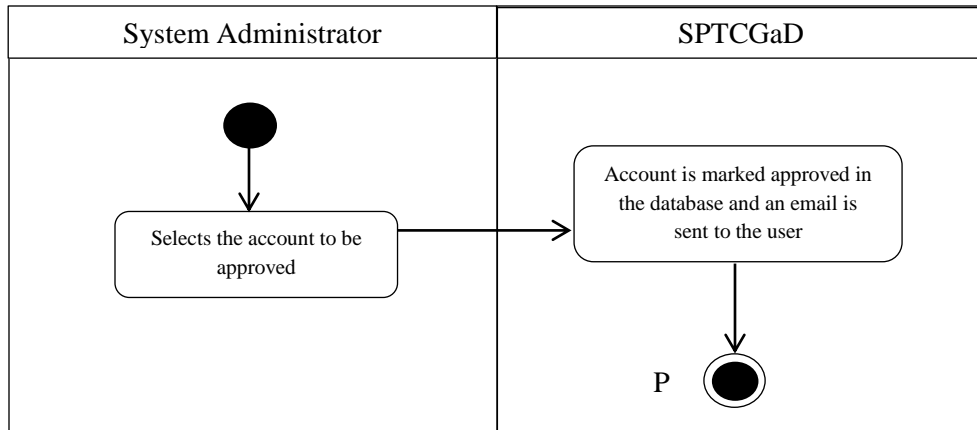


Figure 40. Activity diagram of Approve an account functionality for system administrator in SPTCGaD

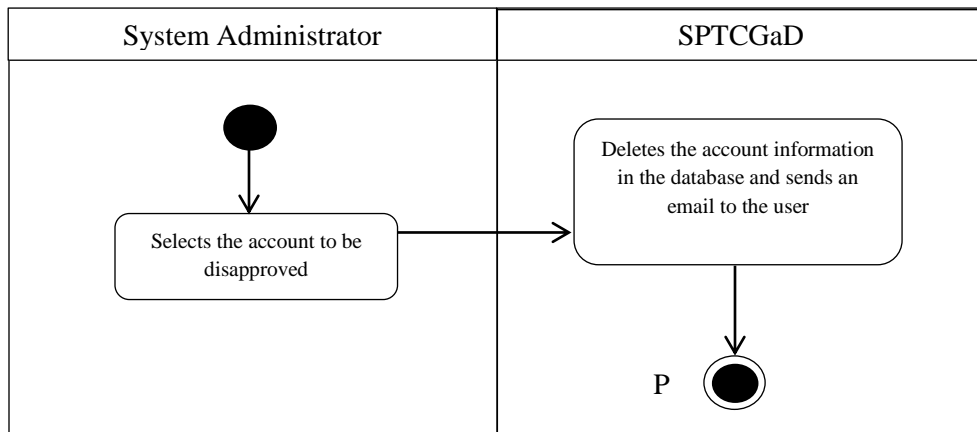


Figure 41. Activity diagram of Disapprove an account functionality for system administrator in SPTCGaD

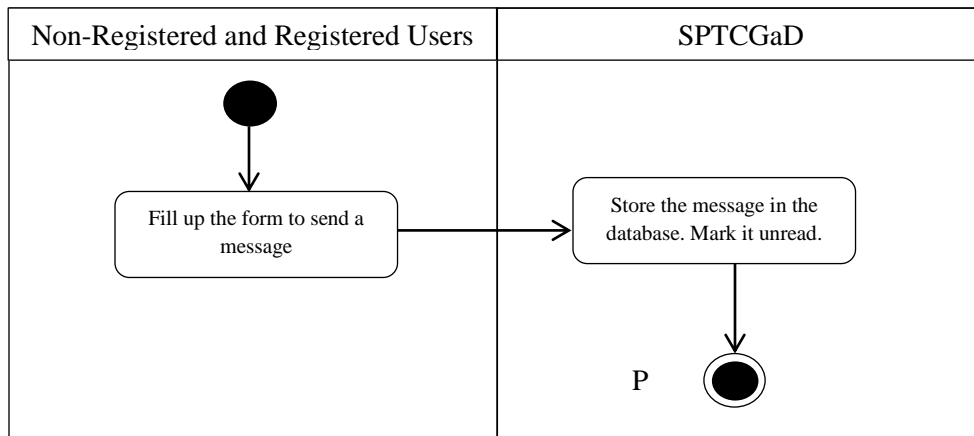


Figure 42. Activity diagram of Send a message functionality for all users of SPTCGaD

#### D. Flowcharts

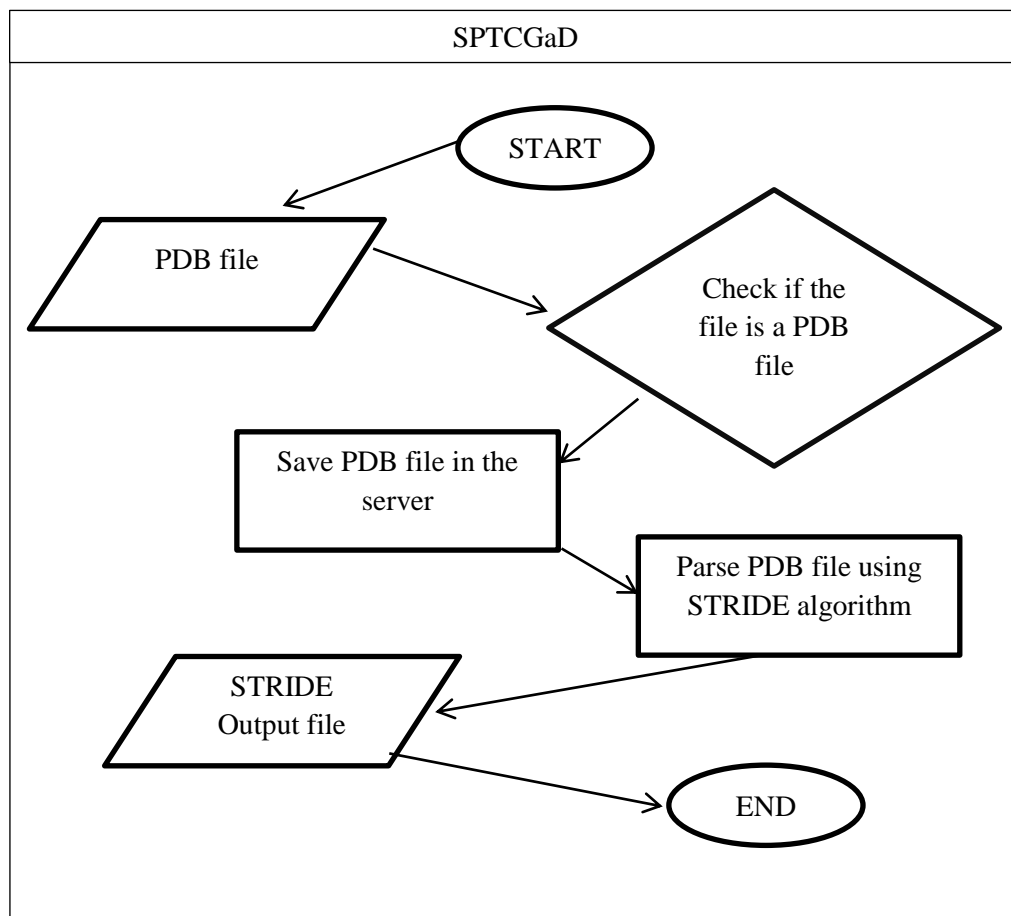


Figure 43. Flowchart of Parse PDB file into STRIDE output file process

The STRIDE algorithm implemented to parse PDB files is an external application written in C language. The program can be downloaded in <http://webclu.bio.wzw.tum.de/stride/>. Given input of a PDB file with the 3D coordinates of the protein domain, the external application of STRIDE generates the 2D secondary structure assignments exported in a text file following a specified format.

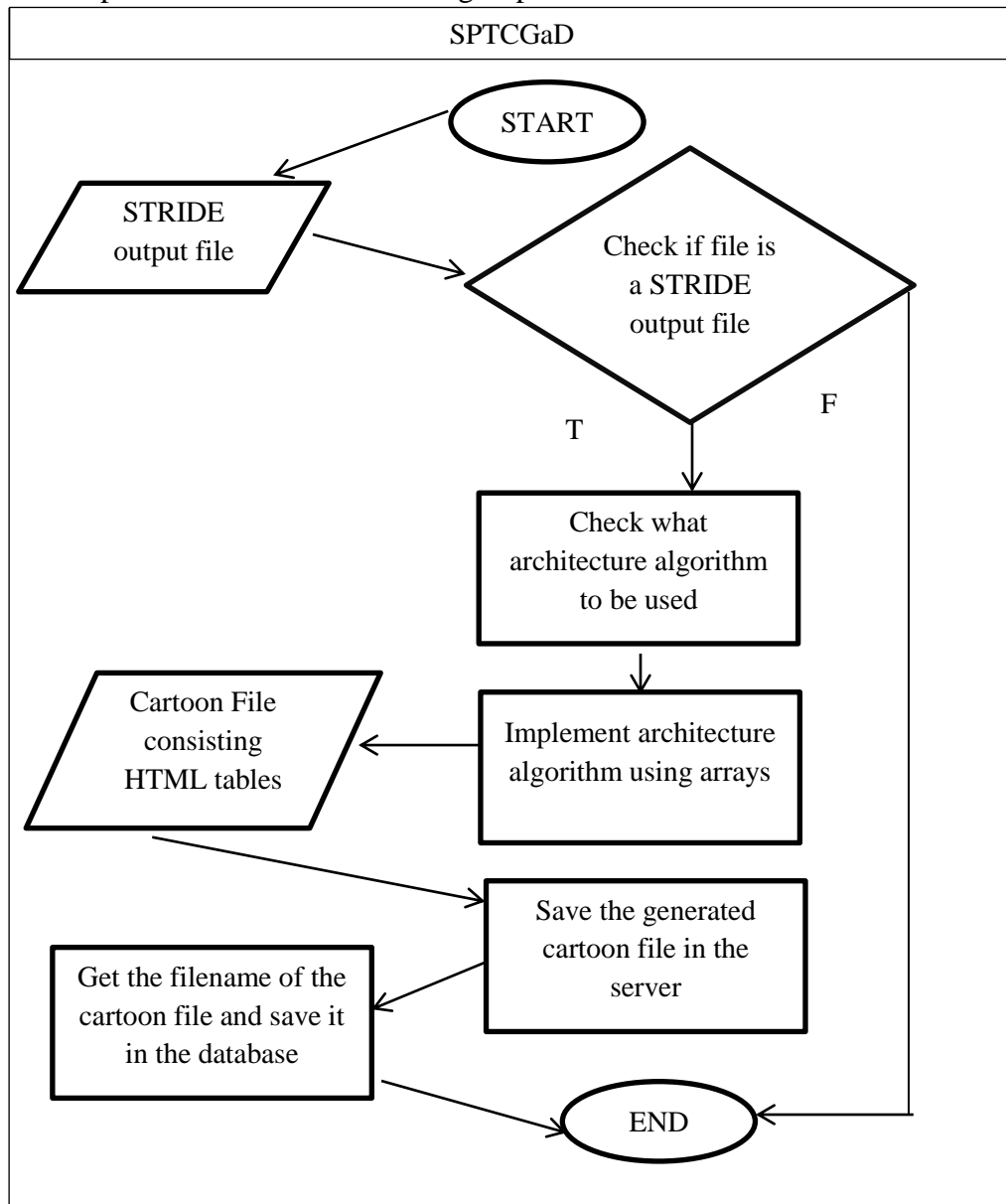


Figure 44. Flowchart of Generate the Cartoon file and save it in the database process

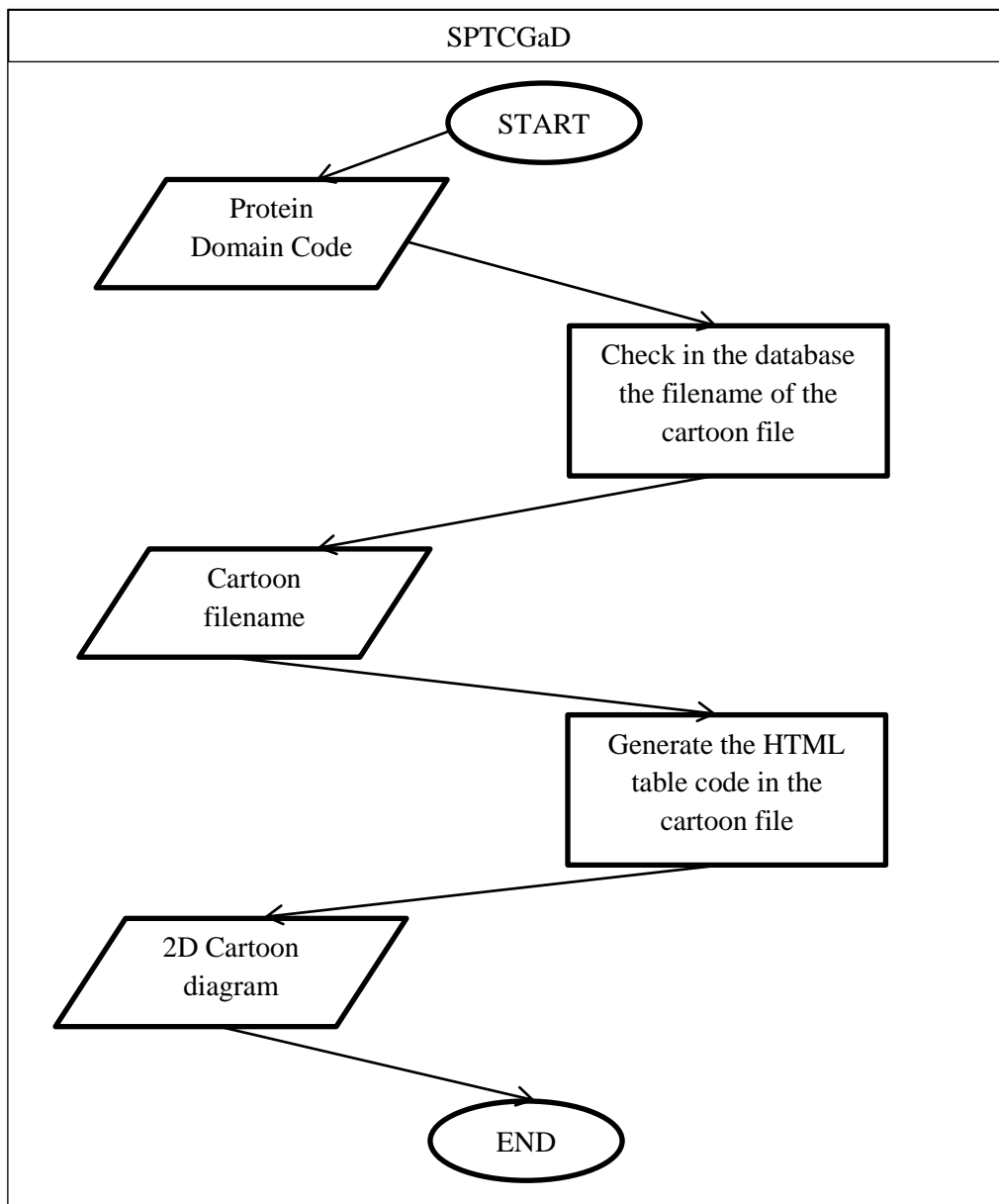


Figure 45. Flowchart of Show the architecture cartoon process

## E. Process Explosion

\* Implement architecture algorithm using arrays

The input for this process is the STRIDE output file which will be then extracted information that will be stored in an array. The following are the steps on the information extraction process on the file.

1. Open the STRIDE output file and get the information in the ASG lines of the file
  - The Alternative Splicing Gallery (ASG) lines contain the detailed information of the secondary structure assignments. Columns 2 (residue name), 3 (chain letter), 4 (residue number) and 6 (structure) are the necessary columns for the process. It will be then stored in an array. Figure 46 shows the ASG information of 1RG8 from its STRIDE output file.

```
REM ----- Detailed secondary structure assignment----- 1RG8
REM 1RG8
REM |---Residue---| |--Structure--| |Phi-| |Psi-| |Area-| 1RG8
ASG HIS A -3 1 C Coil 360.00 153.24 226.2 1RG8
ASG HIS A -2 2 C Coil -92.30 -11.08 128.7 1RG8
ASG HIS A -1 3 C Coil -139.77 158.53 22.8 1RG8
ASG HIS A 0 4 C Coil -61.93 176.03 55.2 1RG8
ASG PHE A 1 5 C Coil -134.46 31.17 114.4 1RG8
ASG ASN A 2 6 C Coil -76.63 122.32 101.4 1RG8
ASG LEU A 3 7 C Coil -106.15 146.43 65.5 1RG8
ASG PRO A 4 8 C Coil -75.72 162.56 18.9 1RG8
ASG PRO A 5 9 C Coil -89.22 -10.63 125.6 1RG8
ASG GLY A 6 10 C Coil -69.74 -169.11 57.9 1RG8
ASG ASN A 7 11 C Coil -150.57 177.88 77.5 1RG8
ASG TYR A 8 12 C Coil -111.66 15.65 69.4 1RG8
ASG LYS A 9 13 C Coil -69.07 -31.79 176.1 1RG8
ASG LYS A 10 14 C Coil -121.83 154.60 149.8 1RG8
ASG PRO A 11 15 C Coil -65.85 164.97 37.2 1RG8
ASG LYS A 12 16 E Strand -141.49 161.41 36.9 1RG8
ASG LEU A 13 17 E Strand -95.76 138.13 1.4 1RG8
ASG LEU A 14 18 E Strand -103.60 102.76 1.0 1RG8
ASG TYR A 15 19 E Strand -92.64 124.49 76.3 1RG8
ASG CYS A 16 20 E Strand -99.38 119.88 0.0 1RG8
```

Figure 46. ASG information of 1RG8 from its STRIDE Output file

2. Get the hydrogen bonding of the residues in the ACC and DNR lines

- The Acceptor (ACC) and Donor (DNR) identify the hydrogen bonding of secondary structures especially in beta strands which is essential in determining their orientation whether they are parallel or anti-parallel in some architecture. Columns 2 (residue name), 3 (chain letter), 4 (residue number), 6 (partner residue name), 7 (partner residue chain letter) and 8 (partner residue number) are the relevant columns in the parsing algorithm. All of the ACC and DNR lines will be covered. These information will be stored in an array.

Figure 47 shows the ASG information of 1RG8 from its STRIDE Output file

```

REM ----- Mainchain hydrogen bonds ----- 1RG8
REM
REM Definition of Stickle et al., J.Mol.Biol. 226:1143-1159, 1992 1RG8
REM A1 is the angle between the planes of donor complex and O..N-C 1RG8
REM A2 is the angle between the planes of acceptor complex and N..O=C 1RG8
REM 1RG8
HBT 136 1RG8
HBI 0 1RG8
HBC 68 /home/jpla-anan/public_html/stride/protein2/pdb/1RG8.pdb A 141RG8
HBC 68 /home/jpla-anan/public_html/stride/protein2/pdb/1RG8.pdb B 141RG8
REM 1RG8
REM |--Residue 1--| |--Residue 2--| N-O N..O=C O..N-C A1 A2 1RG8
ACC ASN A 2 5 -> LEU A 89 92 3.0 133.7 115.0 17.7 61.0 1RG8
DNR LYS A 12 15 -> LEU A 44 47 3.0 159.4 110.6 8.2 19.2 1RG8
ACC LYS A 12 15 -> LEU A 44 47 3.0 157.1 128.3 14.9 51.0 1RG8
DNR LEU A 13 16 -> LEU A 135 138 3.0 150.4 111.3 7.1 14.8 1RG8
ACC LEU A 13 16 -> LEU A 135 138 3.0 153.0 110.3 2.0 22.6 1RG8
ACC LEU A 14 17 -> LEU A 23 26 3.0 158.7 108.6 5.1 35.2 1RG8
DNR TYR A 15 18 -> LEU A 133 136 3.0 146.3 123.0 6.8 53.4 1RG8
ACC TYR A 15 18 -> LEU A 133 136 3.0 150.3 107.9 7.0 23.4 1RG8
DNR CYS A 16 19 -> HIS A 21 24 3.0 157.5 122.0 4.8 79.8 1RG8

```

Figure 47 shows the ACC and DNR information of 1RG8 from its STRIDE Output file

The output of this process will be the arrays from ASG, ACC and DNR lines. These arrays will be then used as an input of architecture algorithms.

\* Check what architecture algorithm to be used

The input files would be the STRIDE output file to be parsed; CATH Domain List text file which contains the list of protein domains and their respective classification (class and architecture numbers in the system's case) based on CATH; and the CATH Domall text file which contains the Domain boundaries for each PDB Chain necessary for identifying the architecture of each parts of a protein domain. The following are the steps on as to how to check on what algorithm to be used in the process:

1. Get the Protein Domain name

- The protein domain name is obtained in the HDR line of the STRIDE output file. Figure 48 shows the Header (HDR) line of the STRIDE output file.

```
REM Please cite: F.Eisenhaber & P.Argos, J.Comp.Chem. 14, 1272-1280, 1993 1RG8
REM           F.Eisenhaber et al., J.Comp.Chem., 1994, submitted      1RG8
REM
REM ----- General information ----- 1RG8
REM
HDR HORMONE/GROWTH FACTOR                11-NOV-03   1RG8      1RG8
CMP MOL_ID: 1;                            1RG8
CMP MOLECULE: HEPARIN-BINDING GROWTH FACTOR 1; 1RG8
CMP CHAIN: A, B;                          1RG8
CMP SYNONYM: HBGF-1, ACIDIC FIBROBLAST GROWTH FACTOR, AFGF, 1RG8
CMP BETA-ENDOTHELIAL CELL GROWTH FACTOR, ECGF-BETA; 1RG8
CMP ENGINEERED: YES                       1RG8
SRC MOL_ID: 1;                            1RG8
```

Figure 48. HDR line from the STRIDE output file of the Protein Domain 1RG8

2. Check on as to what architecture the protein domain belongs to

- It uses the CATH domain list text file. It searches the file on what is the class and architecture number of the protein domain it belongs to which will then be the output. The first column is the protein domain name; the second is the class of the protein domain; and the third is the architecture number of the



protein domain. These will only be the relevant columns for the parsing algorithm. Figure 49 shows 1RG8 having 2 protein chains having 1 domain each belonging to the trefoil architecture.

2qk7B00	2	70	240	10	2	1	1	2	1	287	2.40
1pvlA00	2	70	240	10	2	1	2	1	1	298	2.00
1n7vA03	2	70	250	10	1	1	1	1	1	232	2.20
1n7uA02	2	70	250	10	1	1	1	1	2	263	2.40
1rg8A00	2	80	10	50	1	1	1	1	1	141	1.10
1rg8B00	2	80	10	50	1	1	1	1	2	141	1.10
1jqzA00	2	80	10	50	1	1	1	1	3	141	1.65
1jqzB00	2	80	10	50	1	1	1	1	4	141	1.65
1z2vA00	2	80	10	50	1	1	1	1	5	140	1.90
1z2vB00	2	80	10	50	1	1	1	1	6	140	1.90

Figure 49. 1RG8 having 2 domains in the CATH Domain List text file

3. Get the number of the chains of architectures in the protein domain using the CATH Domall text file.
  - Since a protein can have many domains with different architectures, the CATH Domall text file is used to identify how many chains of architecture exists in a protein. Only columns 1 (chain name) and 2 (number of domains, in D%02d format) will be utilized in the process. Figure 50 shows the CATH Domall information for 1RG8 having 2 protein chains in the CATH Domall text file.

1rg5H	D02	F02	1	H	12	-	H	116	-	1	H	117	-	H	247	-	H	10
1rg5L	D02	F01	1	L	1	-	L	163	-	1	L	164	-	L	263	-	L	264
1rg5M	D02	F00	1	M	1	-	M	143	-	1	M	144	-	M	302	-		
1rg6A	D01	F00	1	A	5	-	A	71	-									
1rg7A	D01	F00	1	A	1	-	A	159	-									
1rg8A	D01	F00	1	A	-3	-	A	137	-									
1rg8B	D01	F00	1	B	-3	-	B	137	-									
1rg9A	D03	F02	2	A	108	-	A	135	-	A	270	-	A	382	-	2	A	10
1rg9B	D03	F02	2	B	108	-	B	135	-	B	270	-	B	382	-	2	B	10
1rg9C	D03	F02	2	C	108	-	C	135	-	C	270	-	C	382	-	2	C	10
1rg9D	D03	F02	2	D	108	-	D	135	-	D	270	-	D	382	-	2	D	10
1rgaA	D01	F00	1	A	1	-	A	104	-									


Figure 50. CATH Domall information for 1RG8

The output for this process will be the protein chains, with their domains, and their respective architecture where they belong to.

\*Save the file and its architecture data in the database

The input in this process is the STRIDE output file to be uploaded by the user.



1. Save the uploaded file in the  (pdb\_files) directory. It is the repository of all files uploaded in the system.

2. Save the following information about the data in the database:

- STRIDE output filename

- Topology

- Class

- Architecture Number
- Cartoon file name
- Uploaded By
- Protein
- Chain
- Domain Number
- Start residue number
- End residue number

Figure 51 shows the uploaded data information of 1rg8 in the SPTCGaD.

STRIDE_FILENAME	TOPOLOGY	CLASS	ARCHITECTURE_NUM	CARTOON_FILENAME	UPLOADED_BY	PROTEIN	CHAIN	DOMAIN_NUM	START	END
LSDVAXJ_stride_output.txt	TreFoil	2	80	1rg8_A_1.txt		1rg8	A	1	-3	137
LSDVAXJ_stride_output.txt	TreFoil	2	80	1rg8_B_1.txt		1rg8	B	1	-3	137

Figure 51. Protein domain 1rg8 information stored in the database

## F. Entity-Relationship Diagram

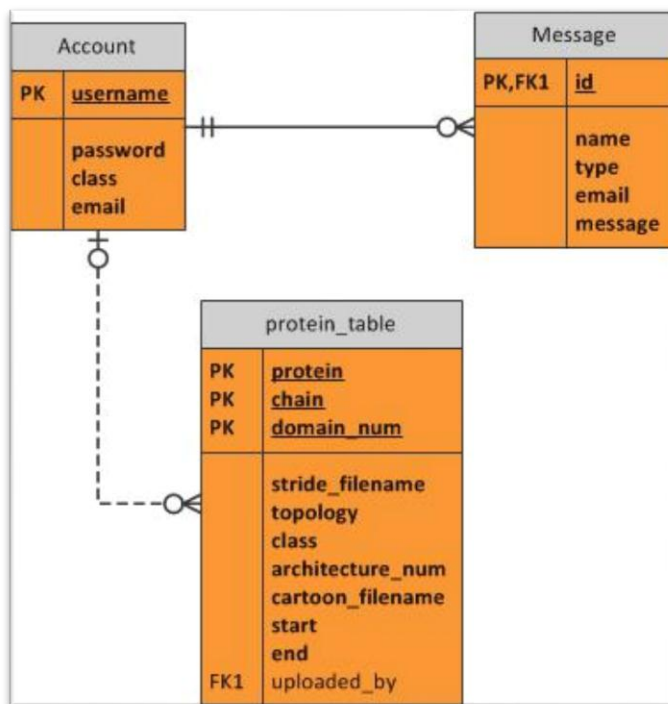


Figure 52. Entity-Relationship Diagram of SPTCGaD

## G. Data Dictionary

The existing STRIDE Protein Database is consists of 3 tables, namely: account, messages and protein\_table. Listed below are the list of tables and their respective data fields.

Primary keys are underlined.

\*account – contains the account details of a user

Field	Type	Description
<u>username</u>	VARCHAR(20)	Unique identifier of user; used for login
<u>password</u>	VARCHAR(32)	Password of user
class	VARCHAR(20)	Code assigned to a class
email	VARCHAR(50)	Email of user

Table 8. Data dictionary for account table

\*message – contains all the messages, read and unread, for the system administrator

<b>Field</b>	<b>Type</b>	<b>Description</b>
<u>id</u>	INT(4)	Unique identifier of a message
name	VARCHAR(50)	Name of the message sender
type	VARCHAR(30)	Type of the message
email	VARCHAR(50)	Email of the message sender
message	TEXT	Message of the user
date	DATE	Date when the message is sent

Table 9. Data dictionary for message table

\*protein\_table – lists all the information about a protein structure and together with the input and output filenames

<b>Field</b>	<b>Type</b>	<b>Description</b>
STRIDE_FILENAME	VARCHAR(30)	Filename of the STRIDE output for a given protein
TOPOLOGY	VARCHAR(30)	Topology information of protein
CLASS	VARCHAR(2)	Class classification of protein
ARCHITECTURE_NUM	VARCHAR(3)	Architecture number
CARTOON_FILENAME	VARCHAR(60)	Filename of the generated protein cartoon
UPLOADED_BY	VARCHAR(30)	Name of the protein structure uploader
<u>PROTEIN</u>	VARCHAR(4)	Protein name based on PDB id
<u>CHAIN</u>	VARCHAR(1)	Chain sequence of protein
<u>DOMAIN_NUM</u>	VARCHAR(2)	Domain number
START	VARCHAR(4)	Start residue
END	VARCHAR(4)	End residue

Table 10. Data dictionary for protein\_table table

## H. Technical Architecture

The SPTCGaD is a web-based system that can be accessed by the users using the Internet. The system is implemented on a Microsoft Windows 7 Professional Operating System. The system is tested using Mozilla Firefox, Internet Explorer and Goggle Chrome

browser. The database used is stored in the Agila web server. The following is the minimum requirements for the system to work:

Client Side:

1. Pentium IV processor or its equivalent
2. At 64mb video card
3. 128mb system of memory
4. Microsoft Windows 2000/XP/Vista/7, Linux
5. Broadband /DSL/internet connection
6. An internet browser installed preferably Mozilla Firefox, Google Chrome or Internet Explorer
7. JavaScript enabled

Server Side:

1. Pentium IV processor or its equivalent
2. 128mb system of memory
3. Linux Server System
4. Broadband /DSL/internet connection
5. XAMPP or LAMPP

## CHAPTER 5: RESULTS

The STRIDE Protein Topology Cartoon Generator and Database (SPTCGaD) is an online website that focuses on the 2D representation of protein architectures. Figure 53 shows the home page of the SPTCGaD website. Figure 50 shows the home page of SPTCGaD.

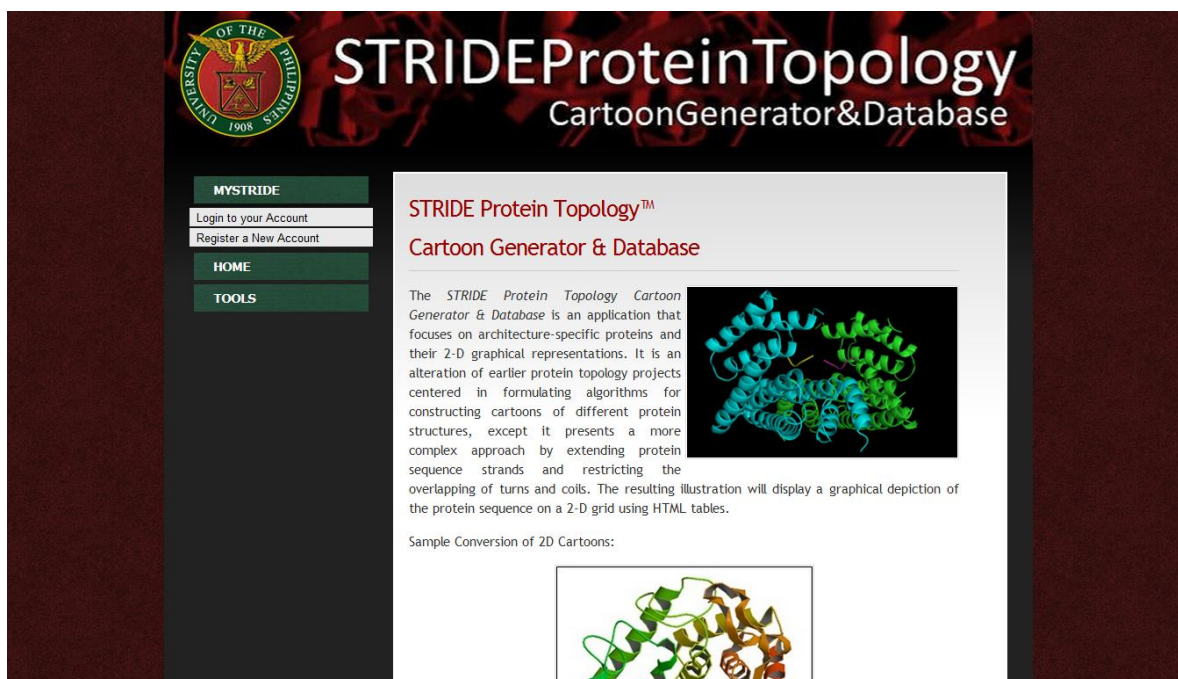


Figure 53. SPTCGaD Home Page

There are three types of users in the SPTCGaD website: non-registered user, registered user and system administrator. Non-registered users can submit their application for website account in the Registration page. Registration for system administrator is currently closed. The following are the pages existing in the SPTCGaD website and the respective roles to which they can be accessed.

Pages that can be accessed by all users:

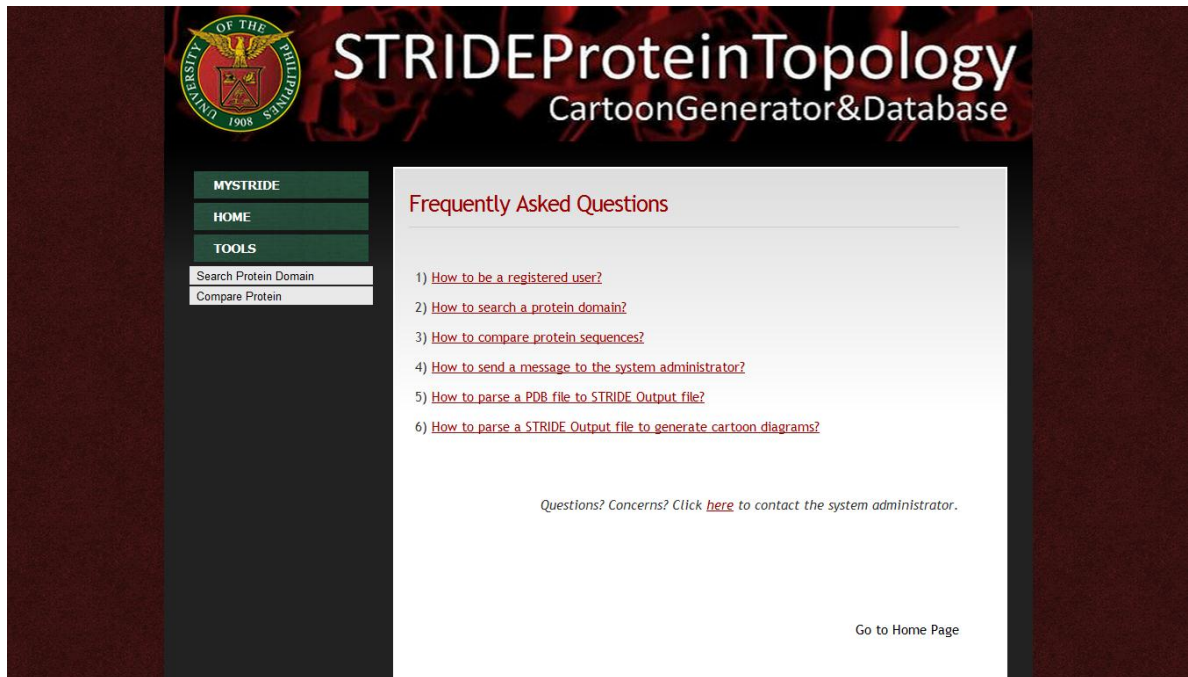


Figure 54. Website FAQ Page

To see the answer from the list of topic questions, the user must click the link corresponding it. Figure 55 shows an example of a page of a topic question.

If the user wants to ask a question or if the user wants to raise any concerns about the website, he/she must click the link *here* to go to the Contact Us page and send a message to the system administrator. Figure 56 shows the Contact Us page of the SPCTGaD system.

If the user wants to go back to the home page, he/she must click the link *Go to Home Page*.






Figure 55. How to parse a STRIDE Output file to generate cartoon diagrams page from the Website FAQ

The Architectures Available Page shows the list of architectures provided by CATH (see <http://www.cathdb.info/class.html>) with which available architecture algorithms are displayed in link form (see Figure 56).

The page where the links directs will display the name of the architecture; its theoretical description provided by the expert and by the CATH database; its algorithm on as to how it will be represented in 2D; and an example screenshot of a protein domain in 3D converted to its 2D form. Figure 57 shows the architecture page of the Ribbon architecture.



# STRIDE Protein Topology Cartoon Generator & Database

**MYSTRIDE**

**HOME**

Website FAQ

Architectures Available

About Us

Contact Us

Sitemap

External Links

**TOOLS**

## Architectures Available

The following are the representative domain architectures classified by Class, Architecture, Topology and Homology Modelling (CATH).

Representative Protein Domain	Representative Protein Domain Subfamily	Availability
1. Bundle	<a href="#">Orthogonal Bundle</a>	Available
	<a href="#">Up-Down Bundle</a>	Available
2. Barrel	<a href="#">Alpha Beta Barrel</a>	Available
	<a href="#">Alpha Barrel</a>	Available
	Beta Barrel	Not Available
3. Ribbon	<a href="#">Ribbon</a>	Available
4. Single Sheet	<a href="#">Single Sheet</a>	Available
	Sandwich	Not Available
	<a href="#">2-Layer Sandwich</a>	Available
	3-Layer Sandwich	Not Available
	<a href="#">2.1 Layer / alpha / Sandwich</a>	Available

Figure 56. Architectures Available page of SPCTGad

**MYSTRIDE**

**HOME**

Website FAQ

Architectures Available

About Us

Contact Us

Sitemap

External Links

**TOOLS**

## Ribbon

Ribbon is an architecture wherein the structure groups (alpha helix or beta strand) are grouped into two and will form a chain of structures. It belongs to class number 2 with architecture number 10.

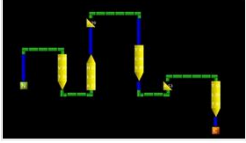
Algorithm:

Initialize a main array. While traversing the sequence array obtained in the parsing algorithm, check the type of structure and if the turn flag is activated.

While a beta strand is not encountered, put the structure in the array. If a beta strand is encountered, activate turn flag. Beta strand encountered will be placed in an array and directed downward.

If turn flag is activated, put two horizontal coils in the array. After that, deactivate turn flag and start putting the next set of structure group until a beta strand is again encountered. If a beta strand is encountered, activate again the turn flag and put in the array four coils horizontally for separation. Deactivate again the turn flag. This means that the algorithm has already encountered two beta strands and will proceed to the rest of the structures following the same pattern.

Example: 1h6p Chain A Domain 1  
Cartoon



3D

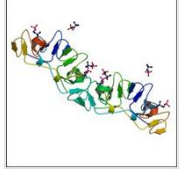


Figure 57. Ribbon Architecture page in the SPCTGad

The About Us page of the SPCTGaD (see Figure 58) shows the goals of the website and the link of the timeline to which displays the development of the SPTCGaD site (see Figure 59).

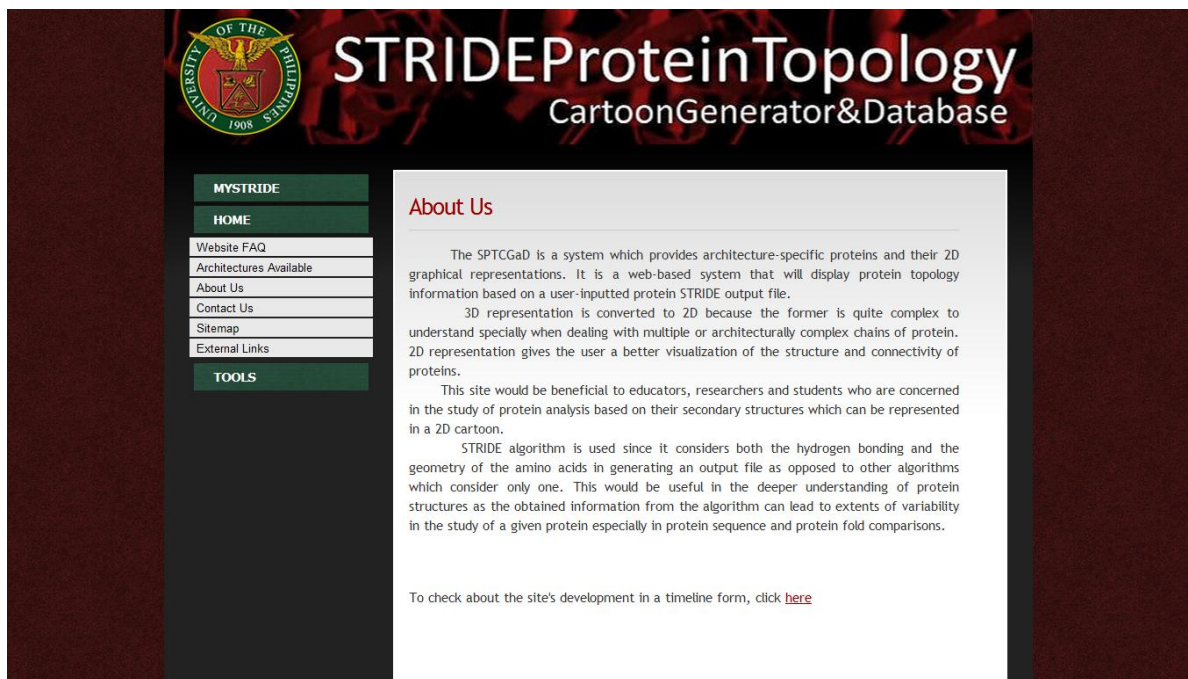


Figure 58. About Us page of SPTCGaD



Figure 59. Timeline of SPTCGaD website development that can be accessed through this link: <http://www.tiki-toki.com/timeline/entry/33743/STRIDE-Protein-Topology-Cartoon-Generator-and-Database/>

The Contact Us Page (see Figure 60) is the page where a website user can send message for the system administrator. It can be for inquiry, for reporting bugs, for support or for other subject of matter. The **Reset** button clears all the user input in the forms. The **Send** sends the form of application to the system administrator. All of the form fields are required to be filled-up.

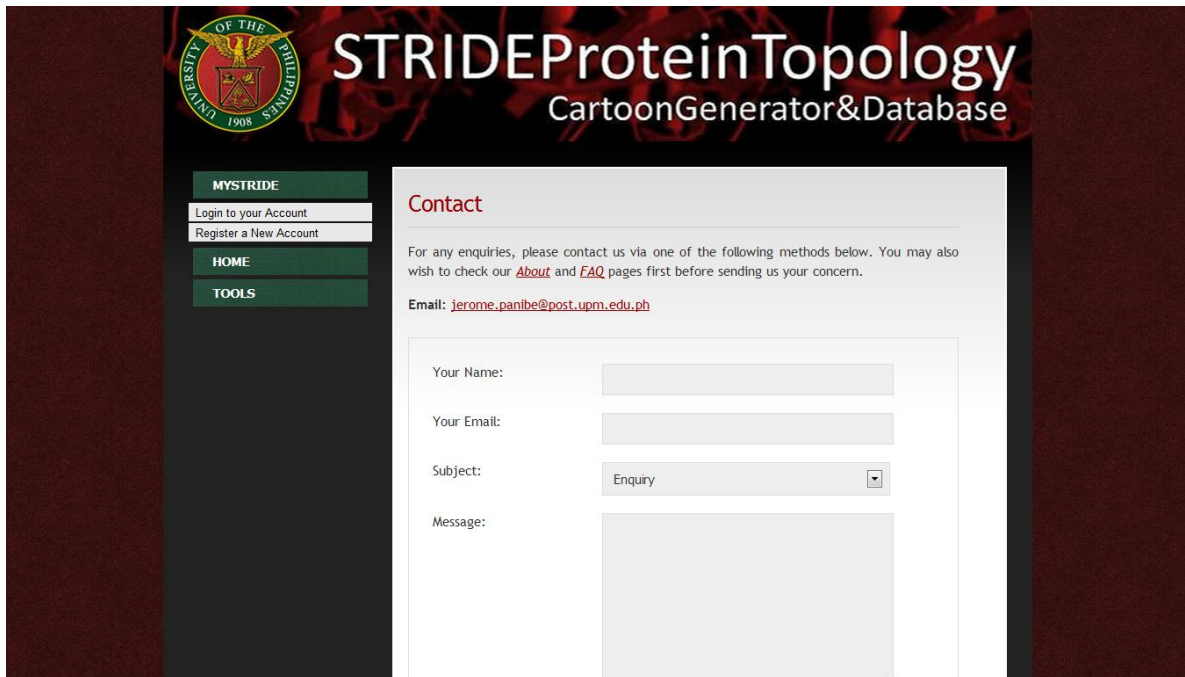


Figure 60. Contact Us Page

The Sitemap page shows the pages that a user may access in the website (see Figure 61). Some of the links can be only accessed by the registered users.

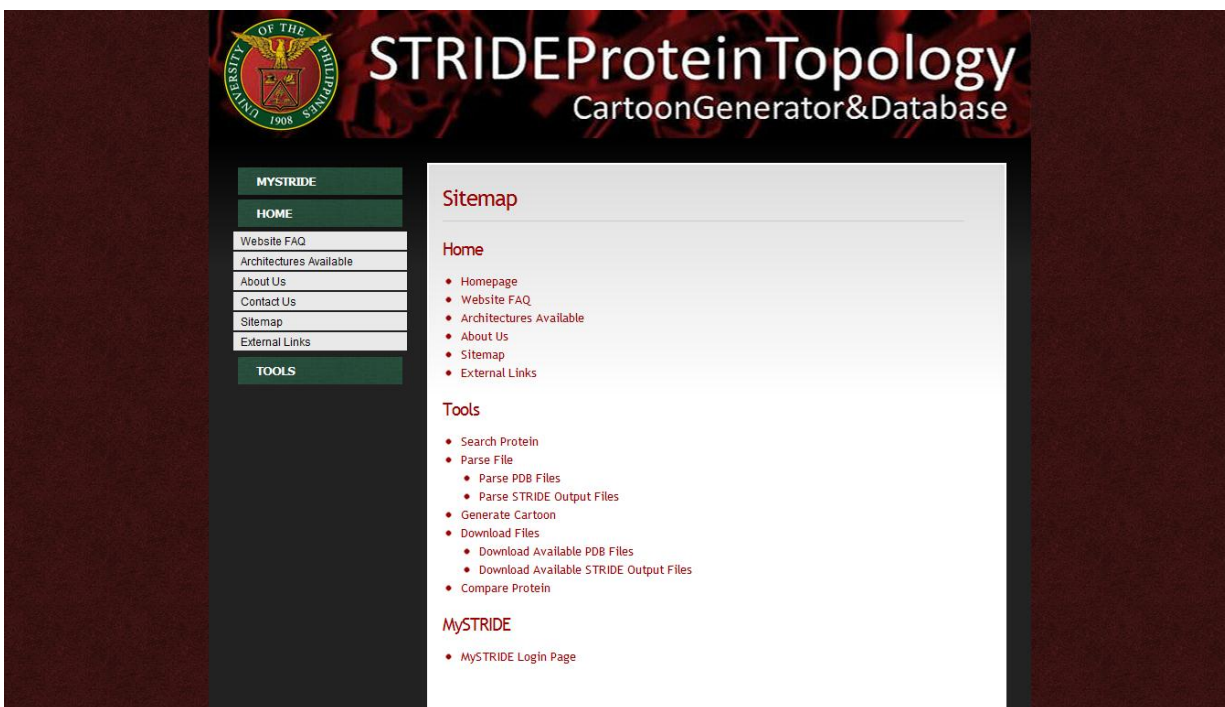


Figure 61. Sitemap page

The External Links page shows the links which may be useful for the user to access (see Figure 62). For example, the RCSB Protein Data Bank can be accessed by the user to download PDB files that can be parsed in the site.

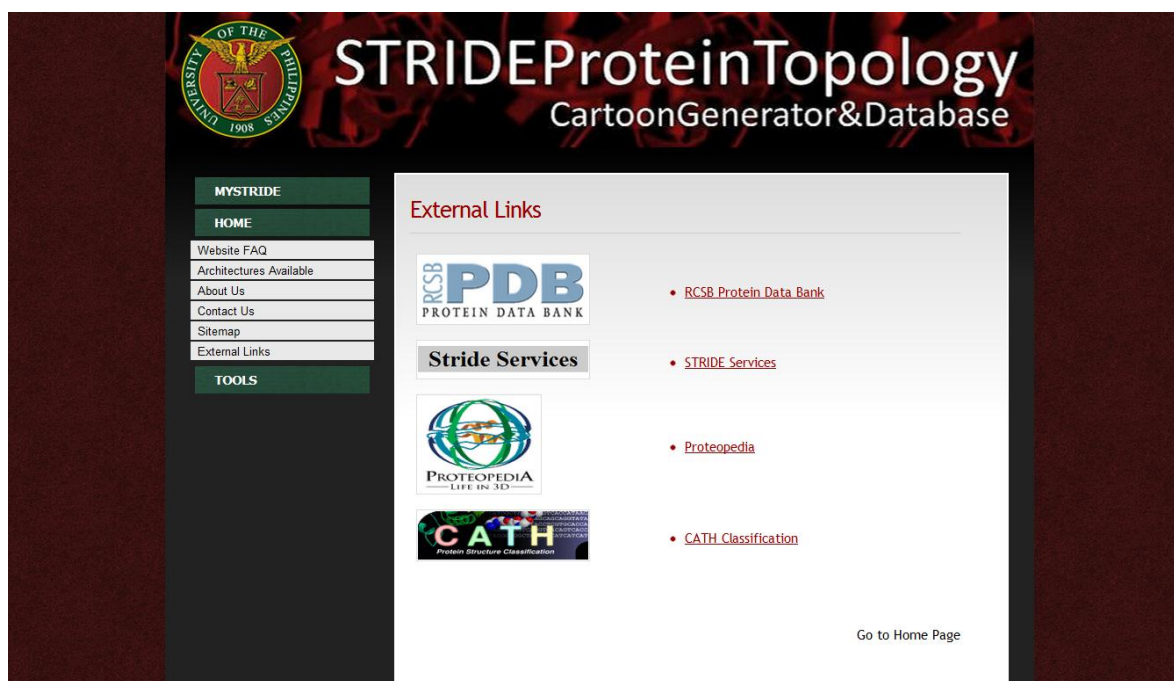


Figure 62. External Links Page

Search Protein Domain Page (see Figure 63) is the page where the user can search available and non-available (as long as its STRIDE output file is in the database) architectures in the website. This page only accepts complete PDB id as input. If the user does not want exact PDB id as an input, he/she must go to the Advanced Search page (see Figure 64 below).

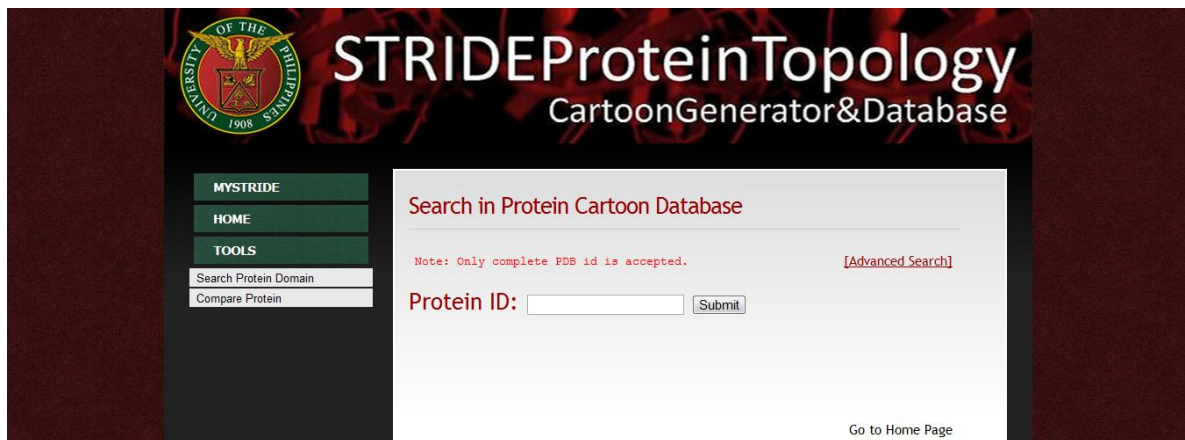


Figure 63. Search Protein Domain page

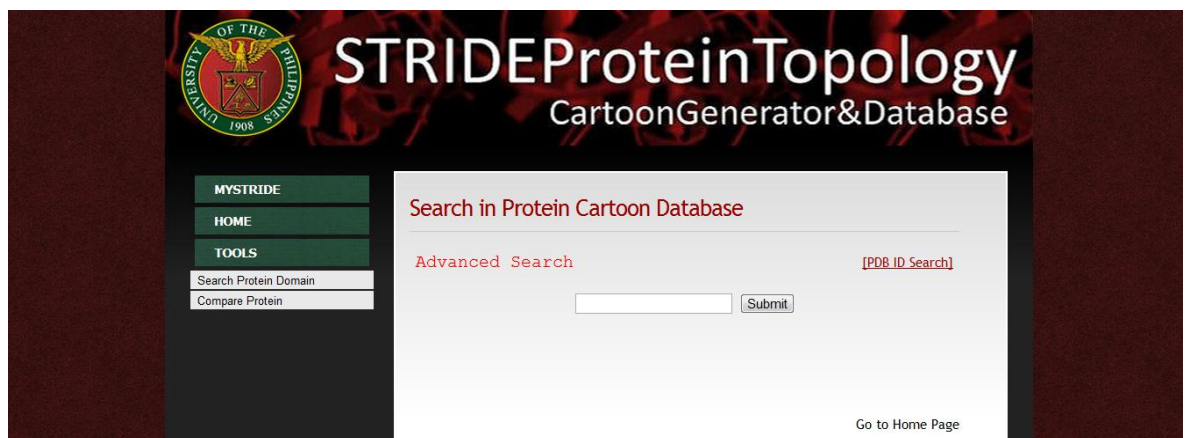


Figure 64. Search result for 1n7v

By clicking the link of the protein domain, a new browser tab will be opened showing its 2D representation. Figure 65 shows an example of the page which shows the 3-Propeller architecture of domain 1 of 1n7v.

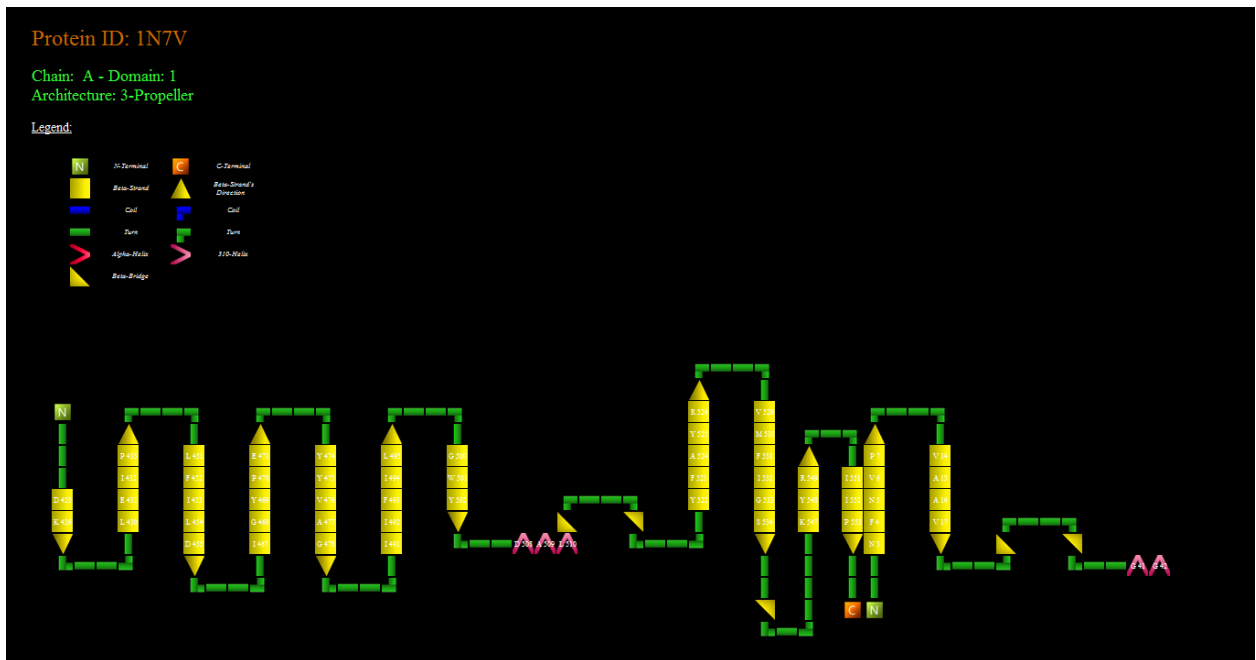


Figure 65. 3-Propeller architecture of the 1-domain of 1n7v shown upon searching protein domain

If the algorithm for an architecture is not yet implemented, it will display that the architecture is not available (see Figure 66).

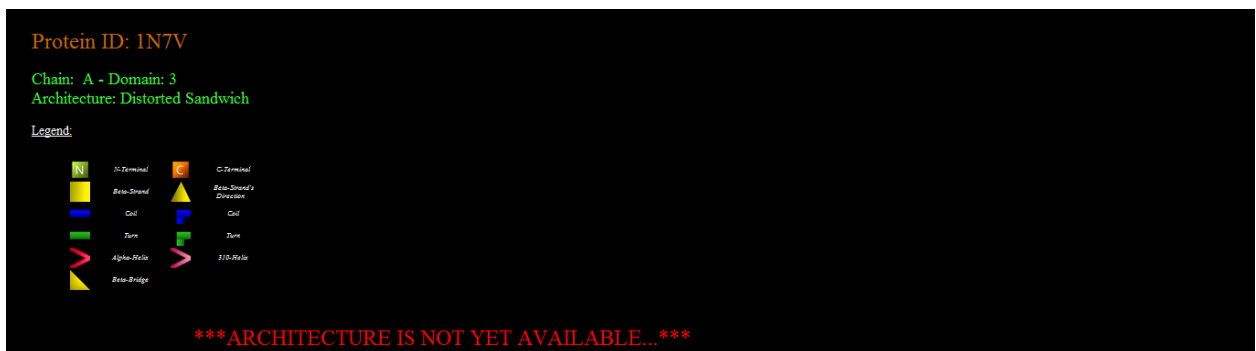


Figure 66. Distorted sandwich architecture of Domain 3 of 1n7v

The Compare Protein page (see Figure 67) enables a user to compare protein domains of the same architecture using methods such as Needleman-Wunsch, Smith-Waterman, CLUSTALW, FSA, POA-MSA and FASTA. Comparison can be pairwise or multiple depending on the user. The user may get the protein to be compared from the available proteins in the database or may upload a STRIDE output file which can be obtained by parsing a PDB file in the system. External programs of the said methods are used to perform comparison. Figure 68, 69 and 70 show an example of a result of comparison of 1auw and 1qrv using Needleman-Wunsch method.

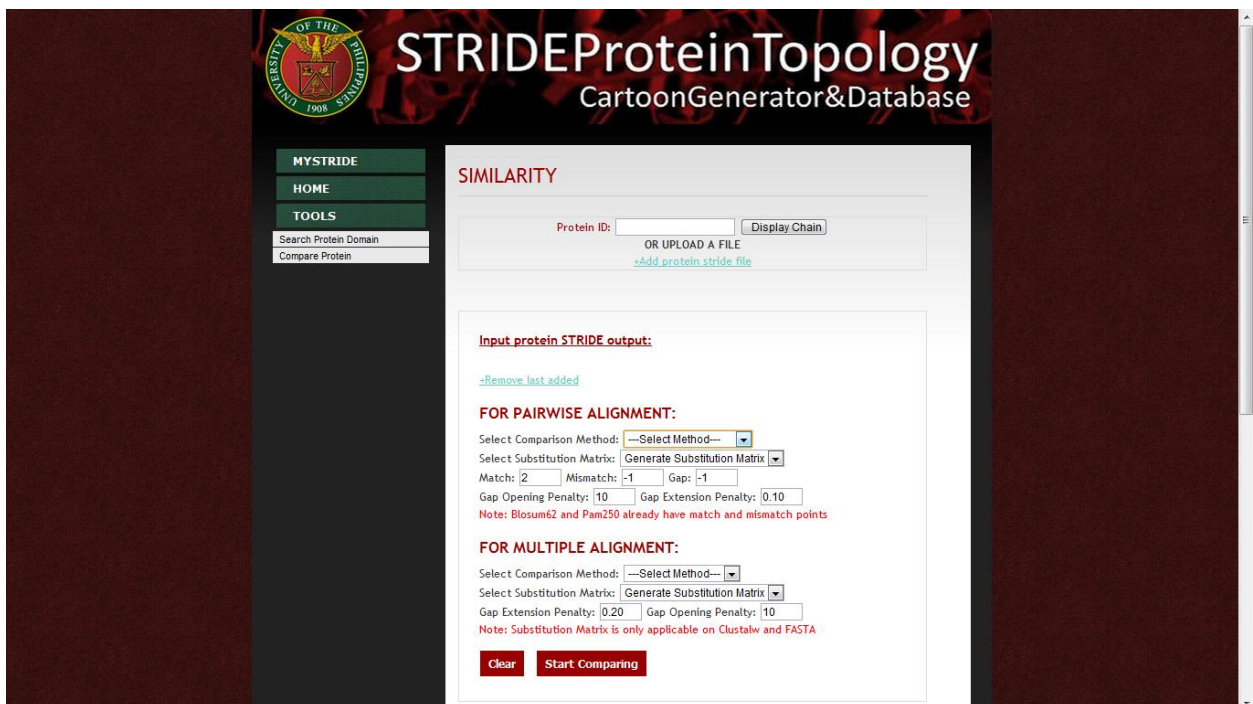


Figure 67. Compare Protein page





Figure 68. Example comparison of 1auw and 1qrv using the Needleman-Wunsch algorithm (First part)

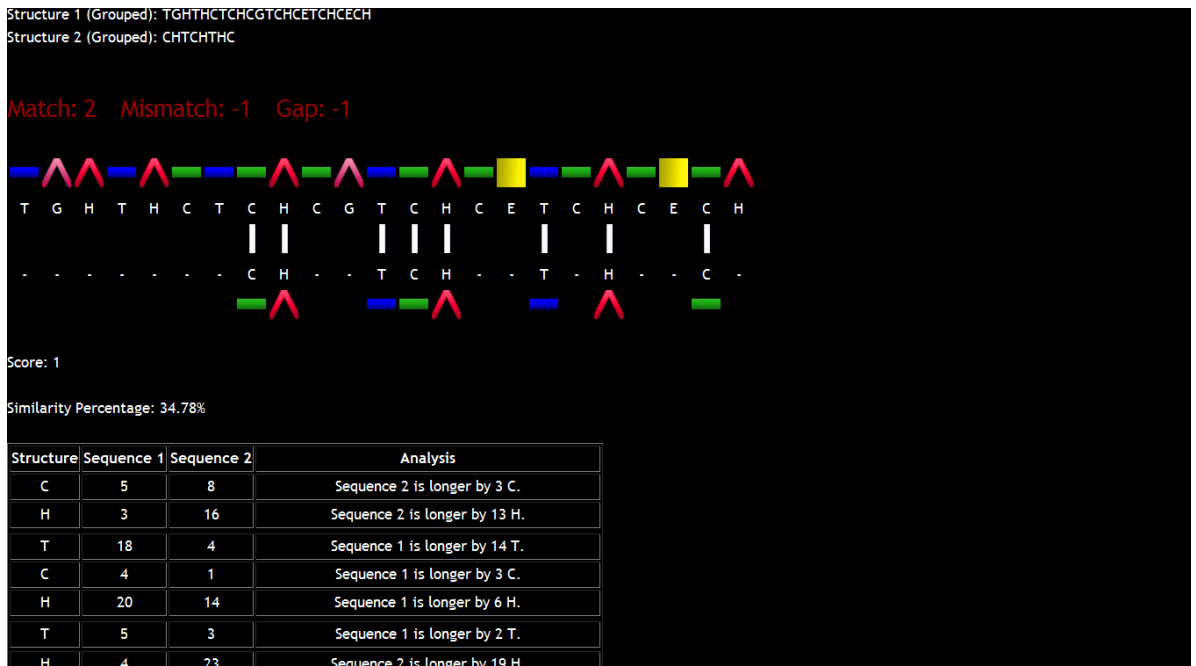


Figure 69. Example comparison of 1auw and 1qrv using the Needleman-Wunsch algorithm (Second part)

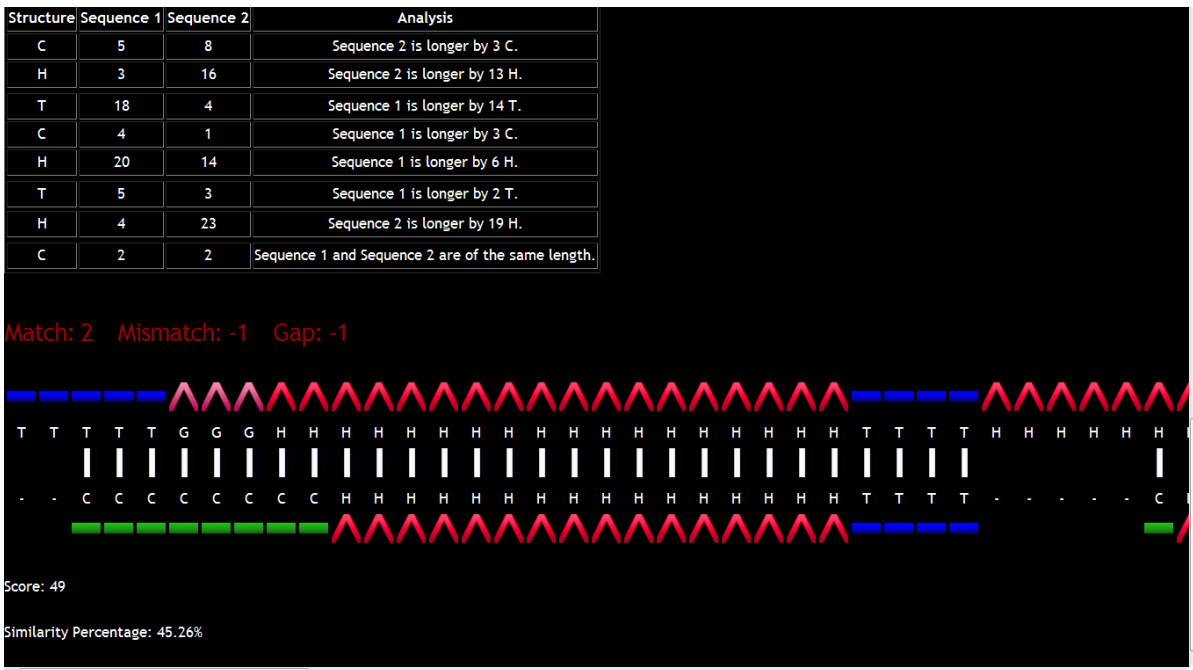


Figure 70. Example comparison of 1auw and 1qrv using the Needleman-Wunsch algorithm  
(Part 3)

The Registration page (see Figure 71) can be used to become a registered user of the system. Registered users can use tools such as STRIDE parser and Cartoon Generator. Application for registration is sent for system administrator approval. All of the fields, except the middle name, are required to be filled-up. An email of notification is sent to the user's email if he/she was approved to be a user of the site.

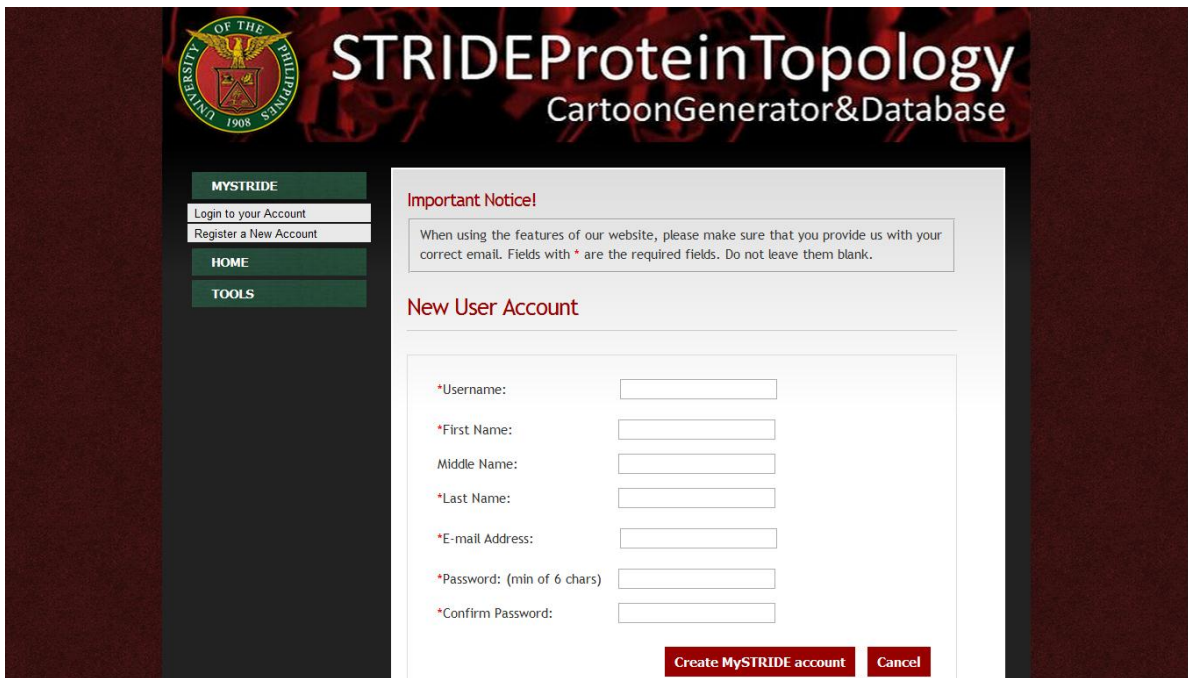


Figure 71. Registration Page

The Login page (see Figure 72) of the website can be used by registered users to access additional functionalities of the website. Username and password is required for the login procedure to proceed.

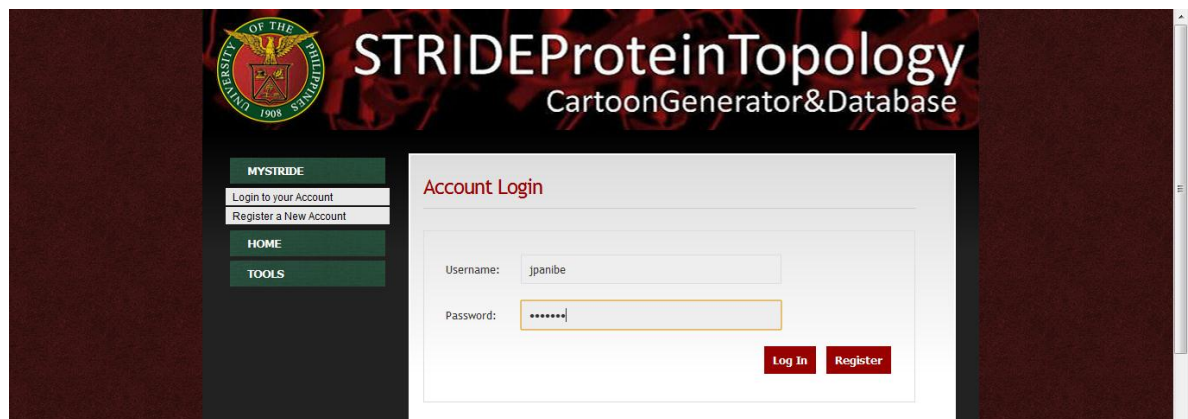
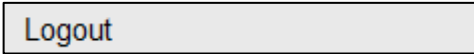


Figure 72. Login page

Pages that can be accessed by registered users and the system administrator only:

The Registered User Home page (see Figure 73) displays the tools that can be used in the website. The  menu in the upper-left corner of the websites logouts the user from the website.

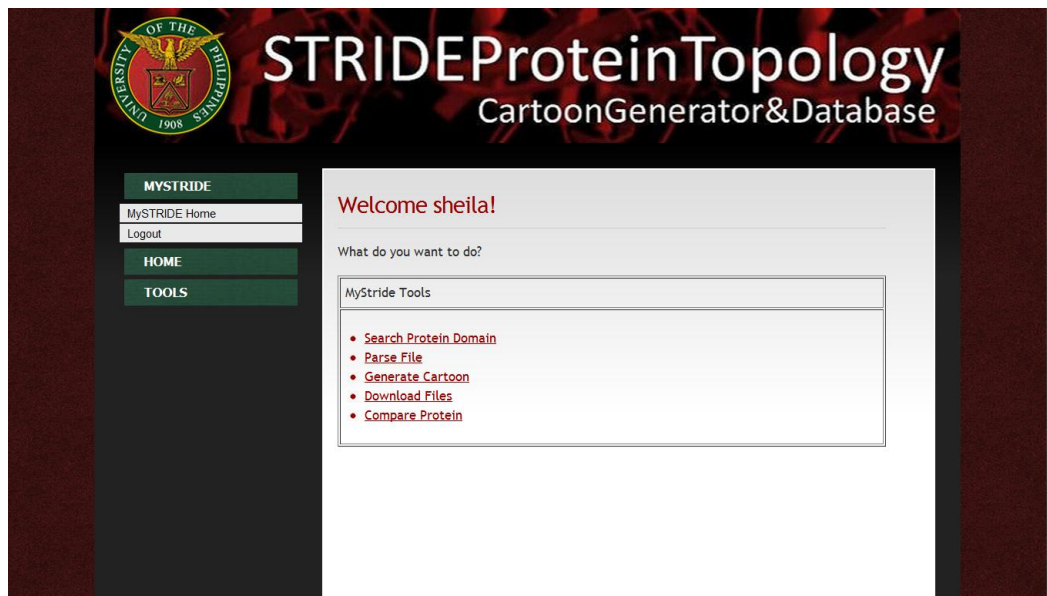


Figure 73. Registered User Home Page

By hovering the mouse in the tools menu (see Figure 74), a registered user can Search Protein Domain, Compare Protein, Parse PDB files and Generate Cartoon files from STRIDE output files. The MySTRIDE home page links (see Figure 75) also shows all of these available functionalities.

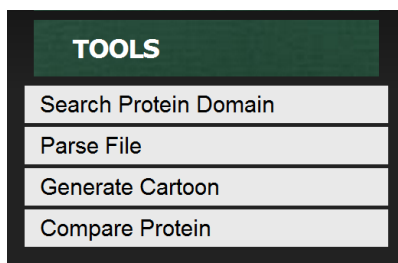


Figure 74. Tools Menu

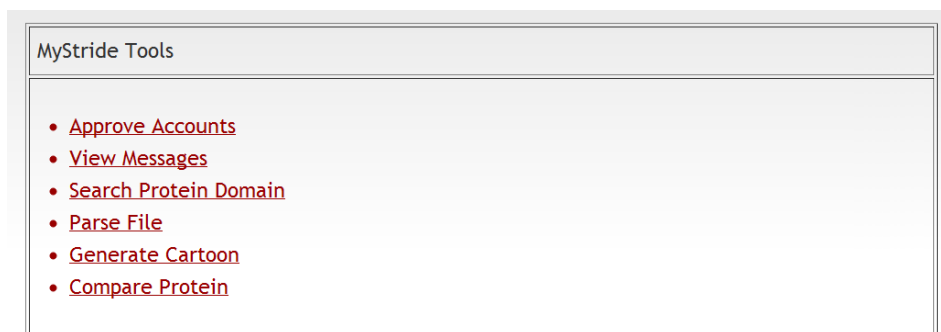


Figure 75. Tools in the MySTRIDE Home Page

If a registered user wants to parse a PDB file to obtain its STRIDE output, he/she must click the Parse PDB file link in the MySTRIDE Home Page or in the Tools Menu. Figure 76 shows the Parse PDB File page of the system.

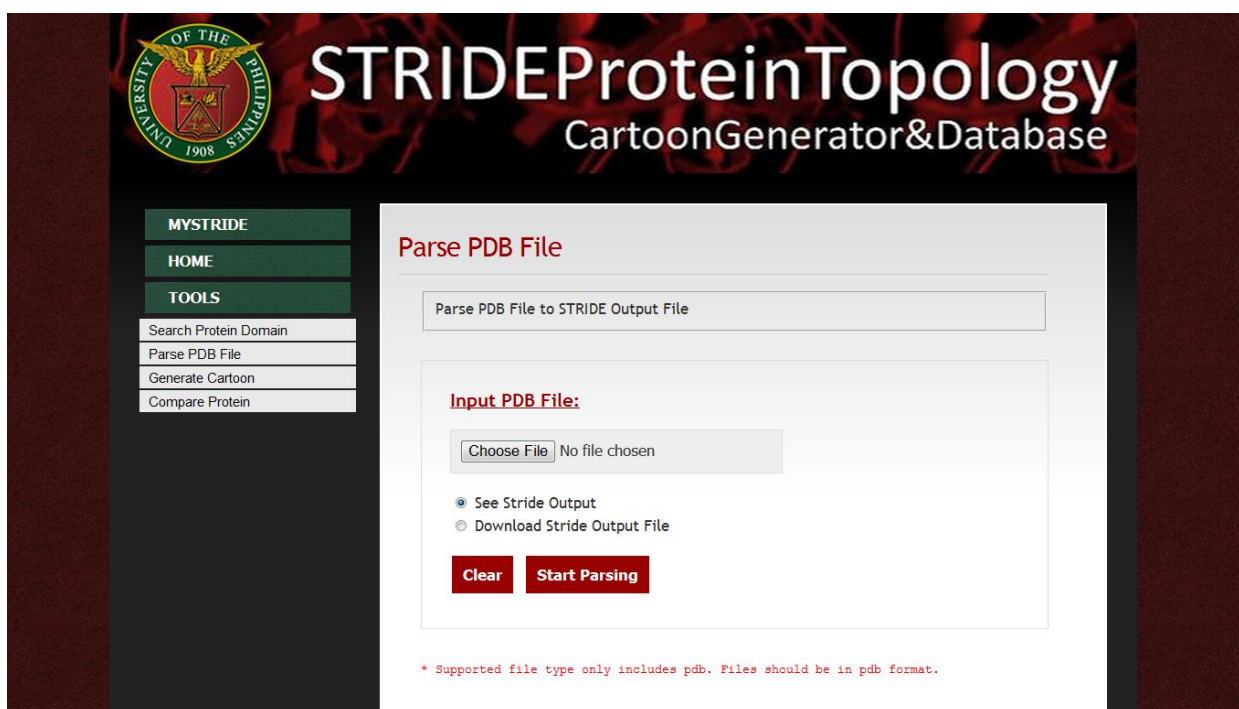


Figure 76. Parse PDB File Page

A PDB file can be obtained in the site of Research Collaboratory for Structural Bioinformatics (RCSB), <http://www.rcsb.org/pdb/home/home.do> (see Figure 77).

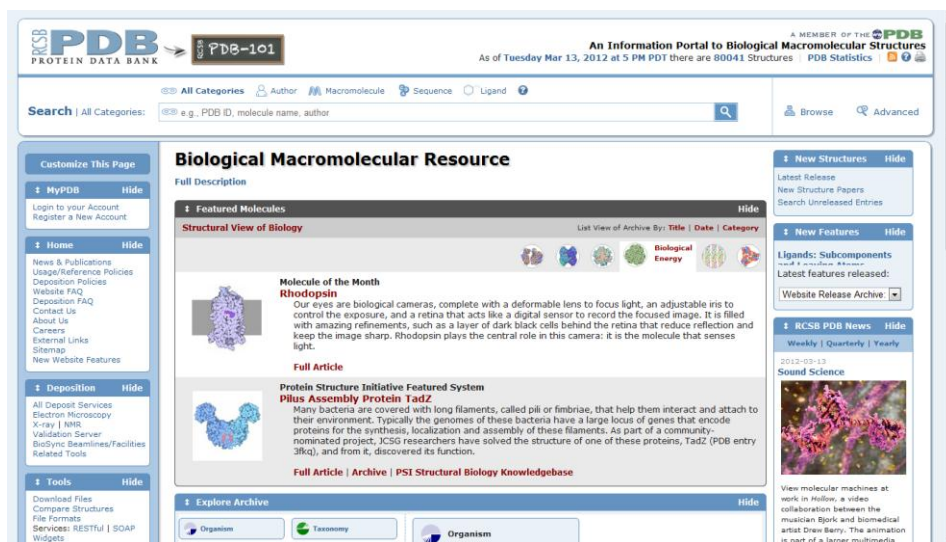


Figure 77. RCSB Website

The user will input the PDB id of the protein he/she wants to parse in the search bar

(see Figure 78) of the RCSB website. Click the  button to continue the search.

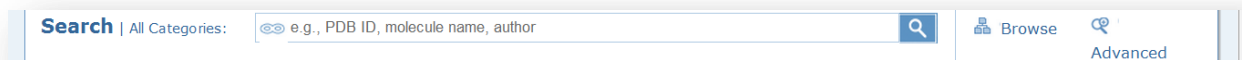
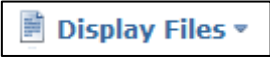



Figure 78. RCSB Search bar

Click the  drop-down box in the upper right side of the page (see Figure 79) and then click  link to download the PDB File of the protein the user wants to parse in the SPTCGaD system.

The screenshot shows the RCSB PDB website interface. At the top, there's the PDB logo and navigation links. A search bar is visible with the text 'e.g., PDB ID, molecule name, author'. The main content area is titled 'Solution Structure of the Reduced Form of the First Heavy Metal Binding Motif of the Menkes Protein' with ID 1KVI. A 'Display Files' dropdown menu is open, showing several file format options. The 'PDB File' option is highlighted with a red box. Below the menu, there's a 3D ribbon diagram of the protein structure. The left sidebar contains navigation links like 'MyPDB', 'Home', 'Deposition', and 'Tools'. The bottom section of the main content area shows 'Molecular Description' with classification as 'Hydrolase' and structure weight of 8691.91.

Figure 79. Display Files Menu in the RCSB Website

After downloading the PDB file, the user must click the **Choose File** button to search for the PDB file to be parsed. If the user wants to just see the STRIDE output in a new window, he/she must click the See Stride output radio button. On the other hand, if the user wants it to be download the STRIDE output, he/she must click the Download STRIDE Output file radio button (see Figure 76).

If the user wants to clear his input, he/she must click the **Clear** button. If the user wants to proceed in the parsing, click the **Start Parsing** button.

Take for example parsing the file 1RG8, if the user wants only to see the STRIDE Output file, it will direct the page to a window with the STRIDE Output (see Figure 77).

```

REM ----- 1RG8
REM STRIDE: Knowledge-based secondary structure assignment 1RG8
REM Please cite: D.Frushman & P.Argos, Proteins XX, XXX-XXX, 1995 1RG8
REM Residue accessible surface area calculation 1RG8
REM Please cite: F.Eisenhaber & P.Argos, J.Comp.Chem. 14, 1272-1280, 1993 1RG8
REM F.Eisenhaber et al., J.Comp.Chem., 1994, submitted 1RG8
REM ----- 1RG8
REM ----- General information ----- 1RG8
REM ----- 1RG8
HDR HORMONE/GROWTH FACTOR 11-NOV-03 1RG8
CMP MOL_ID: 1; 1RG8
CMP MOLECULE: HEPARIN-BINDING GROWTH FACTOR 1; 1RG8
CMP CHAIN: A, B; 1RG8
CMP SYNONYM: HBGF-1, ACIDIC FIBROBLAST GROWTH FACTOR, AFGF, 1RG8
CMP BETA-ENDOTHELIAL CELL GROWTH FACTOR, ECGF-BETA; 1RG8
CMP ENGINEERED: YES 1RG8
SRC MOL_ID: 1; 1RG8
SRC ORGANISM_SCIENTIFIC: HOMO SAPIENS; 1RG8
SRC ORGANISM_COMMON: HUMAN; 1RG8
SRC ORGANISM_TAXID: 9606; 1RG8
SRC GENE: FGf1, FGFA; 1RG8
SRC EXPRESSION_SYSTEM: ESCHERICHIA COLI; 1RG8
SRC EXPRESSION_SYSTEM_TAXID: 562 1RG8
AUT M.J.BERNETT,T.SOMASUNDARAM,M.BLABER 1RG8
REM 1RG8
REM ----- Secondary structure summary ----- 1RG8

```

Figure 77. View STRIDE Output file page of 1RG8

If a registered user wants to parse a STRIDE output file and view its 2D cartoon equivalent, he/she must click the Generate Cartoon link in the MySTRIDE Home Page or in the Tools Menu (see Figure 78).



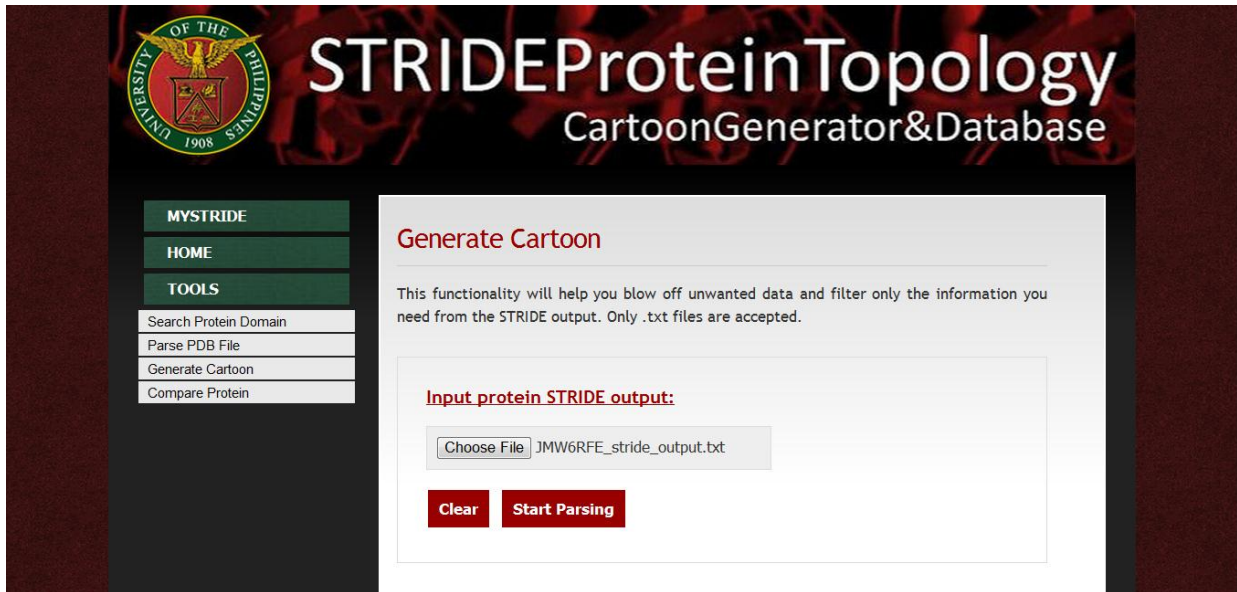


Figure 78. Generate Cartoon Page

The user must click the **Choose File** button to search for the STRIDE output file to be parsed. If the user wants to clear his input, he/she must click the **Clear** button. If the user wants to proceed in the parsing, click the **Start Parsing** button.

Take for example parsing the STRIDE output file of 1APS, after clicking the Start Parsing button, it would go to the result page (see Figure 79) displaying the chain and the type of architecture it belongs.

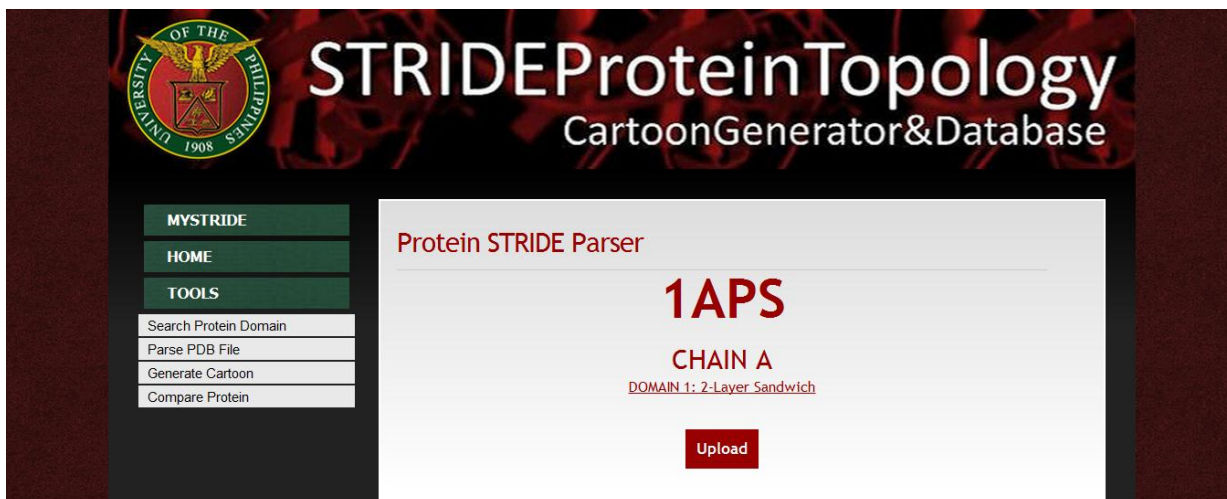


Figure 79. Generate Cartoon Result Page

Click the link of the domain and architecture type to see the 2D cartoon (see Figure 80). If the algorithm of the architecture domain is not yet implemented, it will display architecture not available (see example in Figure 66).

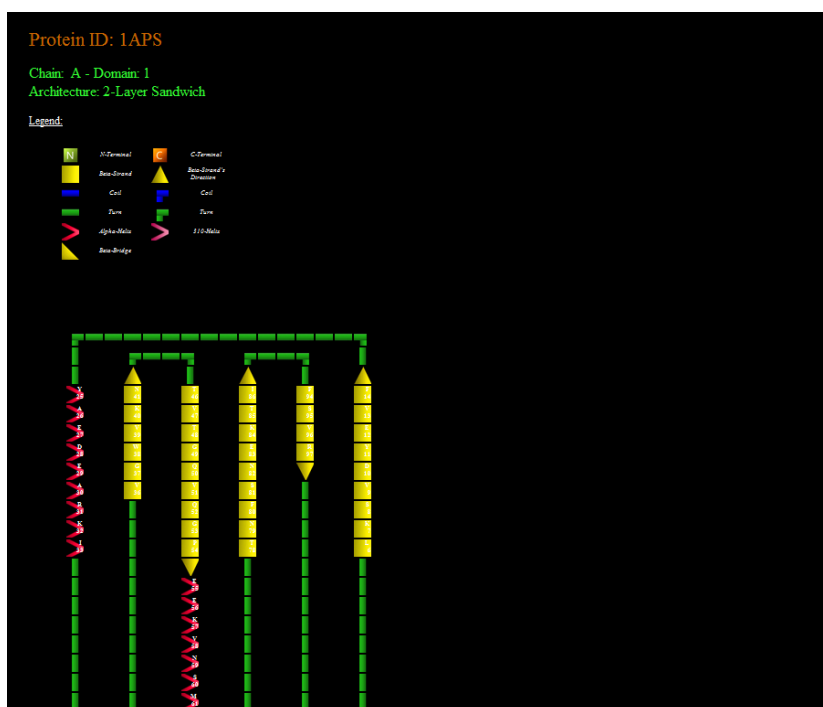



Figure 80. Cartoon of 1APS

If the registered user wants to upload the cartoon generated, he/she must click the  button. A prompt will be then displayed if the upload is successful.

Pages that can be accessed by the system administrator only:

If the user is a system administrator, his/her MyStride home page will look like (see Figure 81).

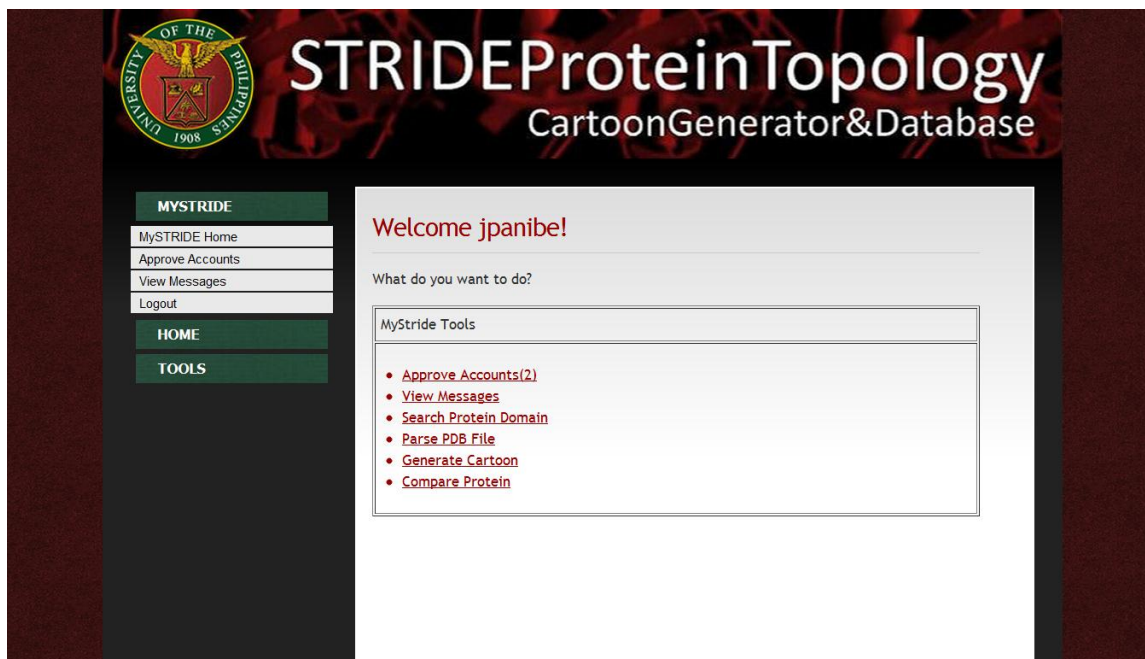


Figure 81. MySTRIDE Home Page for a System Administrator

If the system administrator wants to approve or disapprove accounts, he/she may click the Approve Accounts link in the MySTRIDE Home Page or in the Tools Menu. The list of pending accounts will be displayed to the system administrator (see Figure 82).

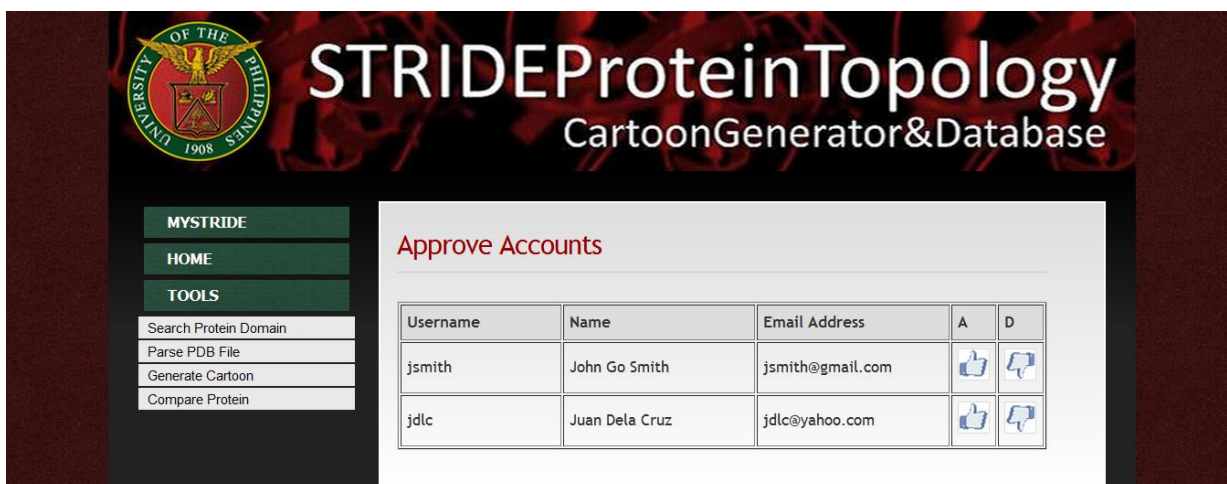


Figure 82. Approve Accounts page

To approve an account, the system administrator must click the button and an email will be sent to the approved user email. Otherwise, click button to disapprove the account. An email will also be sent to the user email about the disapproval of the account.

If the system administrator wants to view the messages that were sent to the system, he/she may click the View Messages link in the MySTRIDE Home Page or in the Tools Menu. The list of messages is then displayed to the user (see Figure 83). messages are read while messages are unread messages.

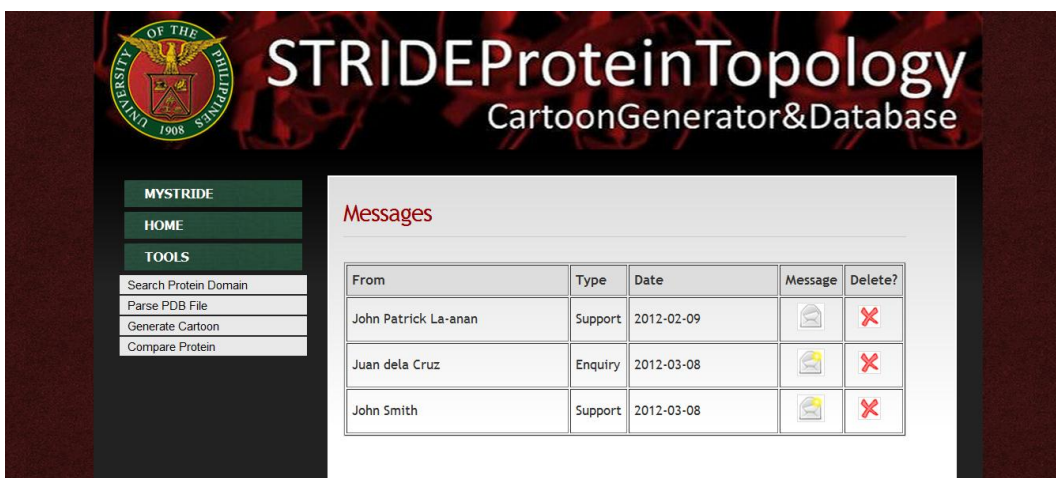



Figure 83. View Messages Page

If the system administrator wants to delete a message, he/she must click the  button beside the message to be deleted (see Figure 83)

By clicking the message icons, the user will be directed to a page (see Figure 84) displaying the message and its headers.

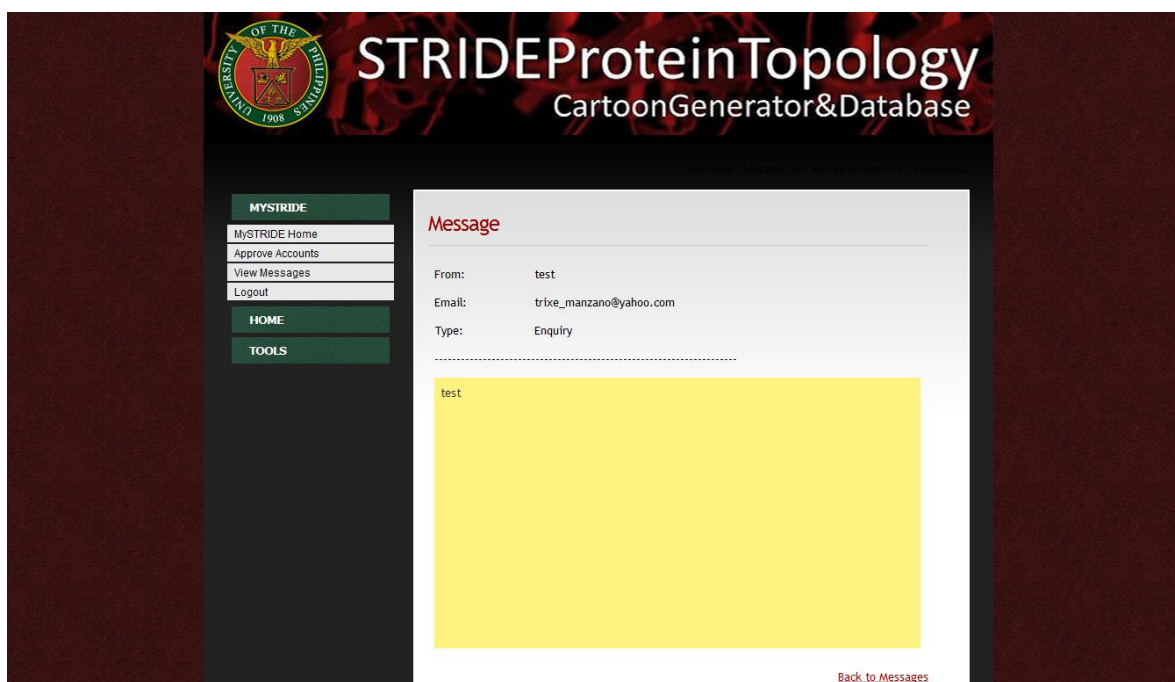


Figure 84. Message Page

If the system administrator wants to reply to the message sent by the website users, he/she must fill-up the message form under the message content in the Message page (see Figure 85).

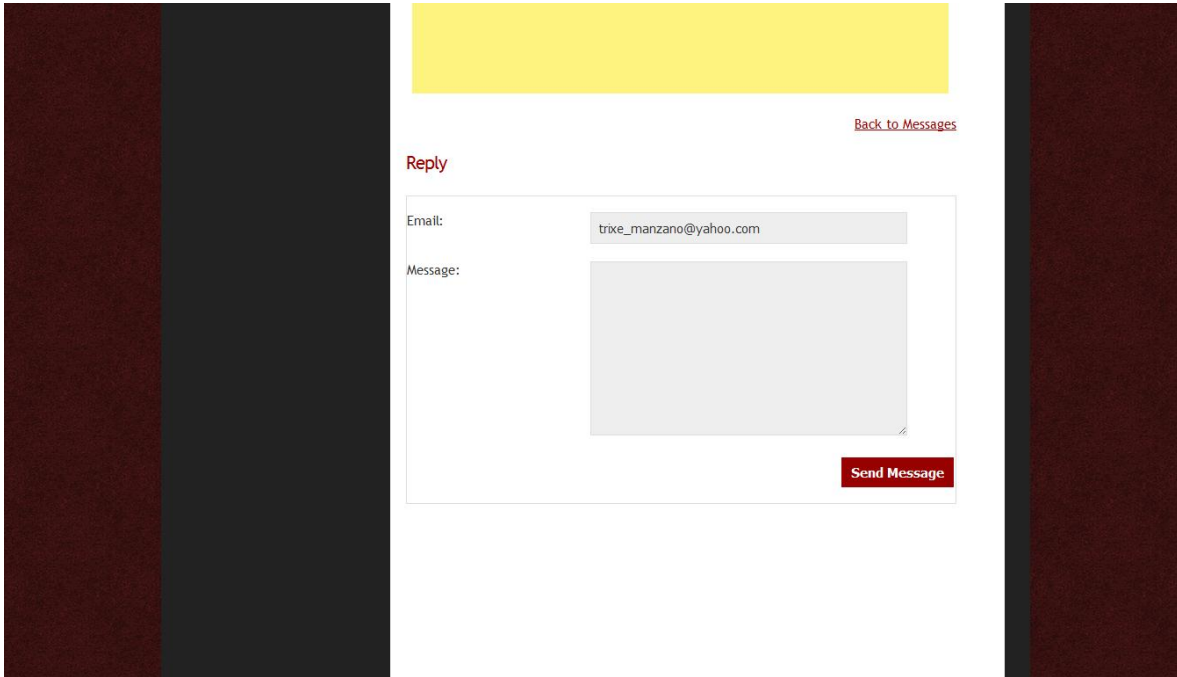
The image shows a screenshot of a web application interface for replying to a message. At the top, there is a yellow rectangular header. Below it, on the right side, is a link labeled "Back to Messages". The main section is titled "Reply" in red text. Below the title is a form with two input fields: "Email:" and "Message:". The "Email:" field contains the text "trixe\_manzano@yahoo.com". The "Message:" field is a large, empty text area. At the bottom right of the form is a red button labeled "Send Message". The entire form is set against a white background, which is centered on a dark red background.

Figure 85. Reply Message portion in the Message page

The system administrator must click the **Send Message** button to send the message to the website user's email.

## CHAPTER 6: DISCUSSION

The STRIDE Protein Topology Cartoon Generator and Database (SPCTGaD) is a system that mainly displays the 2D representation of protein domains. It is based from an external application, the STRIDE algorithm, which predicts the secondary structure sequence of the protein domain.

There are three levels of user in SPCTGaD: the non-registered user or the guest user, the registered user and the system administrator.

Limited functionalities or tools are available to non-registered users such as traversing the site, registering for an account, sending a message to the system administrator, searching for available protein domain architectures and comparing protein domains.

Registered users can use the parsing functionality of the system using an external application and using knowledge algorithms by an expert. The former is used when parsing a PDB file and outputs a text file (STRIDE output file) and the latter is for generating the two-dimensional cartoon of protein domains.

The system administrator, being the one responsible in maintaining the system, is able to approve and disapprove applications for an account in the system; and is able to view and delete messages sent by registered and non-registered users.

Added protein domain architecture algorithms in the system are trefoils, rolls and alpha barrel. Documentations for all algorithms are made.

## PROTEIN DOMAIN ARCHITECTURES:

### Trefoil

Trefoils are composed of the outline of overlapping rings. Its basic unit is beta hairpins (see Figure 86). Trefoils belong to class 2 and architecture number 80.

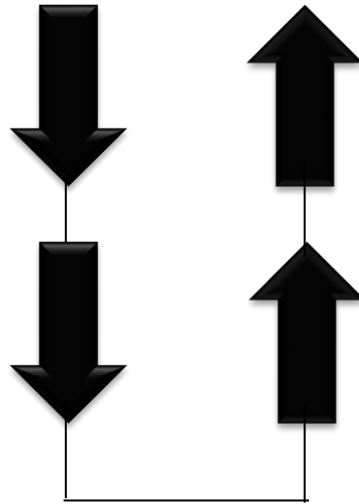


Figure 86. Topology cartoon of a Beta Hairpin

#### Algorithm:

Get all the structures, beta strands and helices, and store their necessary information in an array from the output of the parsing algorithm. The information includes the structure type, first residue number, the structure group and its corresponding residue name and numbers.

Initialize an array having 5 columns and twice the number of beta strands plus one rows. This will be the main array for the protein domain.

Group the betas into two then insert it in the second and fourth column of the main array while getting the maximum length of the beta strands in each column.



Insert the arrowhead depending on the orientation. The orientation will be alternating left or right. Rows whose corresponding number is divisible by two will have its contents, the residue number and name, reversed. If the beta strand that will be placed is less than the maximum length of the beta strand in that column, append coils.

Get all the helices and put them in the middle of the beta strands. Take note of the orientation to where the helices are placed and also the maximum length of helices in the column or row.

Put the connectors in the main array depending on the orientation to where it is placed. Connectors are coils and will be placed on the first and last column of the main array.

Put the starting point, the N terminal, and the ending point, the C terminal, in their corresponding places in the main array.

An example of a generated 2D cartoon of a trefoil is shown below (see Figure 87). Its 3D representation is shown in Figure 88.

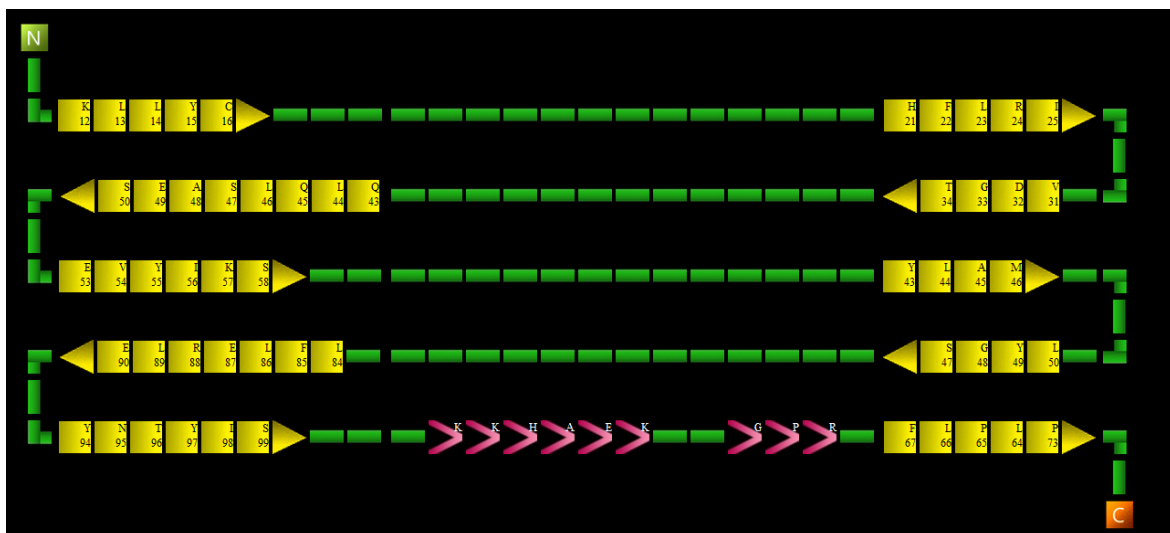


Figure 87. Generated trefoil cartoon diagram of 1rg8 Chain A Domain 1

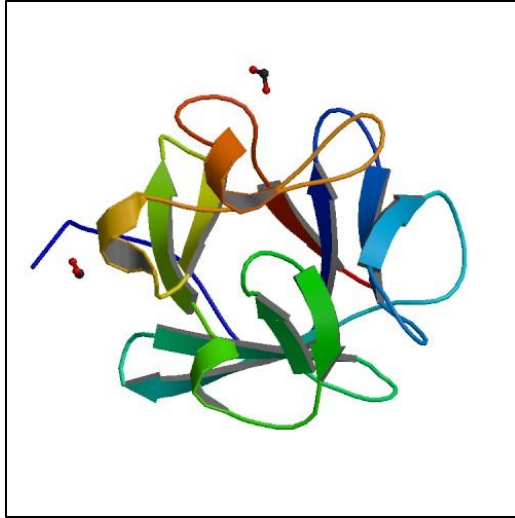


Figure 88. 3D representation of 1rg8

## **Rolls**

A roll is a sheet of secondary structures bent when viewed in 3D. In 2D, the secondary structures are straightened to form a single sheet. Rolls are a member of class 2 and architecture number 30.

Algorithm:

Get all the structures, beta strands and helices, and store their necessary information in an array from the output of the parsing algorithm. The information includes the structure type, first residue number, the structure group and its corresponding residue name and numbers.

Initialize an array with 3 rows and has the number of structures, beta strands and helices, plus one columns. This will be the main array. The structures will be placed in the second row. The connectors, coils, will be placed in the first and third row.

Place the structures in the second row in alternating, upward and downward, direction.

An example of a generated 2D cartoon of a roll is shown below (see Figure 89). Its 3D representation is shown in Figure 90.

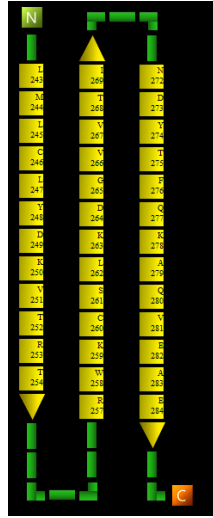


Figure 89. Generated roll cartoon diagram of 1nh2 Chain D Domain 2

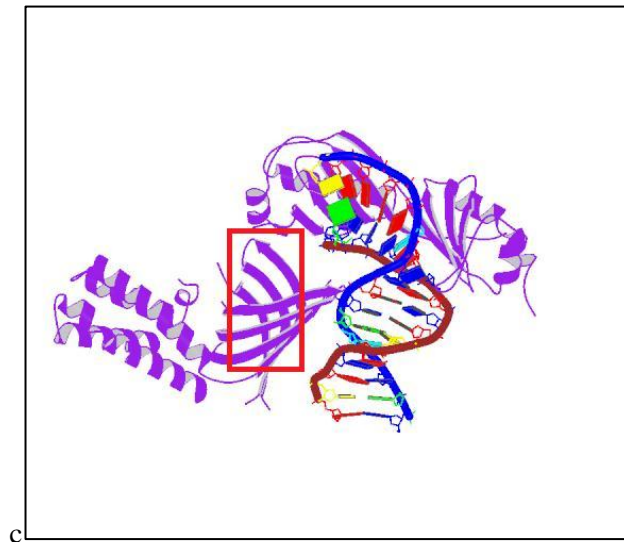


Figure 90. 3D representation of 1nh2

## Alpha Barrel

Alpha barrels are structures having alpha helices on top and beta strands on the bottom. Alpha barrels are members of class 1 and architecture number 50.

Algorithm:

Get all the structures, beta strands and helices, and store their necessary information in an array from the output of the parsing algorithm. The information includes the structure type, first residue number, the structure group and its corresponding residue name and numbers.

Initialize an array of 5 rows and twice the number of helices columns. This would be the main array placing the alpha helices in the second row and the beta strands, coils in the third row and beta strands in the fourth row.

Get all the helices in the array. Place the helices in the second row of the main array.

Get all the beta strands. Check if the beta strand/s obtained are between the alpha helices.

An example of a generated 2D cartoon of an alpha barrel is shown below (see Figure 91). Its 3D representation is shown in Figure 92.

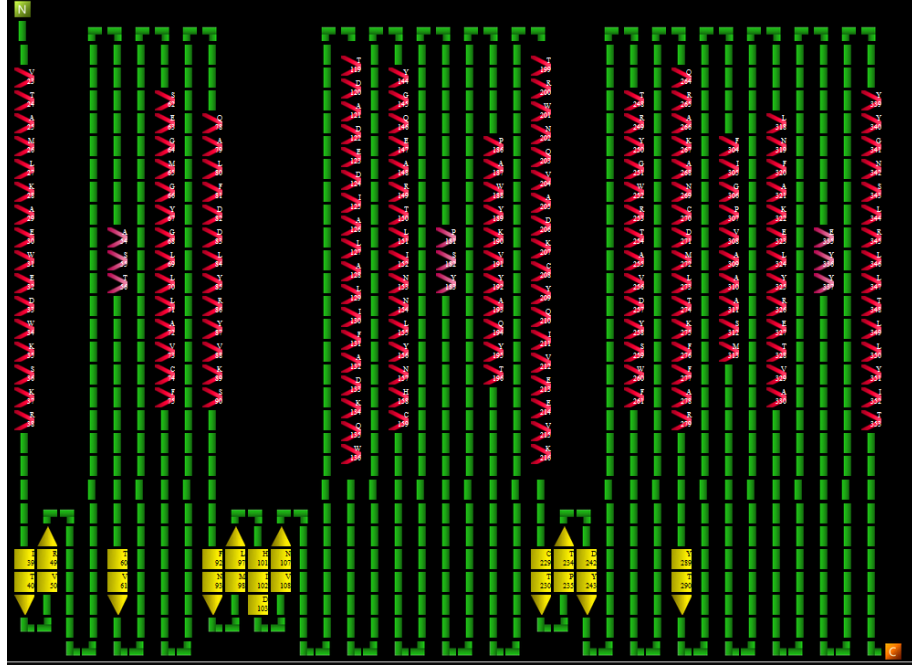


Figure 91. Generated alpha barrel cartoon diagram for lis9

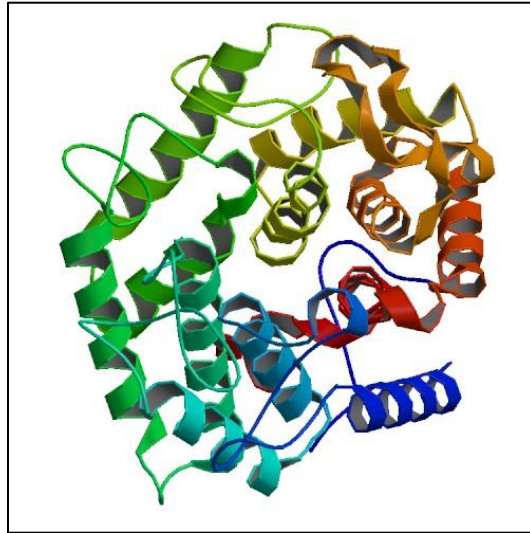


Figure 92. 3D representation of lis9

## **Other Architectures Existing in SPTCGaD**

### **Two-Layer Sandwich**

Two layer sandwiches are members of class 3 and architecture number 30.

Algorithm:

Get all the structures, beta strands and helices, and store their necessary information in an array from the output of the parsing algorithm. The information includes the structure type, first residue number, the structure group and its corresponding residue name and numbers.

Group tokens depending on type, if it is an alpha helix strand or a beta sheet strand.

Since the beta sheet strand is the focus of this architecture, assign the orientation for each beta strand group. Put the arrowheads depending on the orientation. Take note of the maximum length for all groups. After grouping all the secondary structures, traverse the group again to check if the secondary structure group is equal to the maximum length. If the length of the secondary structure group is less than the maximum length, append coils to the group until its length is equal to the maximum length.

The structures will then be arranged in a specific manner forming five groups of secondary structure (see Figure 93).

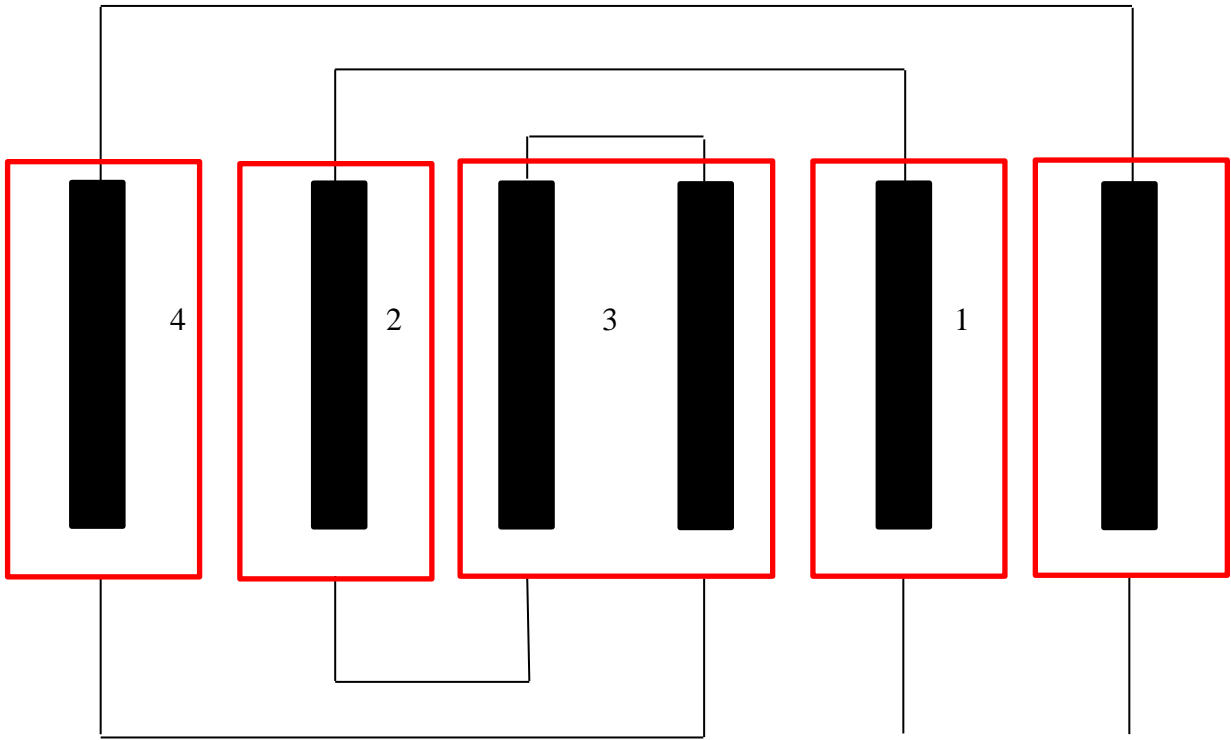


Figure 93. 2-Layer Sandwich Grouping for 1kvi based on the algorithm existing in the system

An example of a generated 2D cartoon of an 2-layer sandwich is shown below (see Figure 94). Its 3D representation is shown in Figure 95.

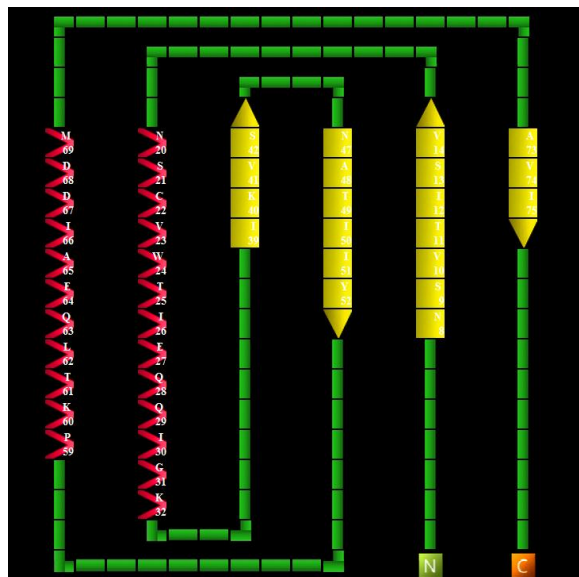


Figure 94. Generated 2-layer sandwich cartoon diagram for 1kvi

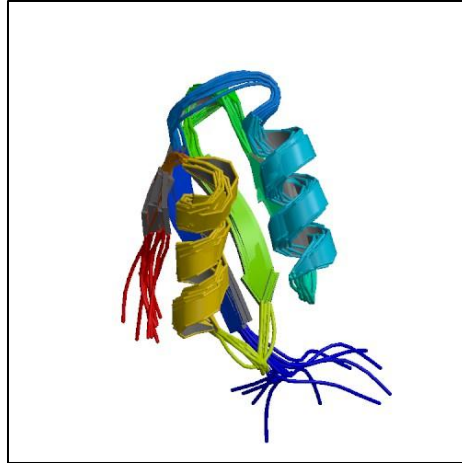


Figure 95. 3D representation of 1kvi

### **Three-Layer Sandwich (aba)**

Three layer sandwiches (aba) are basically three layer sandwiches but have its outer structure, when presented in 2D, consist of alpha helix strands. Three layer sandwiches (aba) are members of class 3 and architecture number 40.

Algorithm:

Get all the structures, beta strands and helices, and store their necessary information in an array from the output of the parsing algorithm. The information includes the structure type, first residue number, the structure group and its corresponding residue name and numbers.

Group tokens depending on type, if it is an alpha helix strand or a beta sheet strand.

Since the beta sheet strand is the focus of this architecture, assign the orientation for each beta strand group. Put the arrowheads depending on the orientation. Take note of the maximum length for all groups. After grouping all the



secondary structures, traverse the group again to check if the secondary structure group is equal to the maximum length. If the length of the secondary structure group is less than the maximum length, append coils to the group until its length is equal to the maximum length.

The structures will then be arranged in a specific manner forming seven groups of secondary structure having the outermost structures be alpha helix strands. These groups are 5 and 7 (see Figure 96).

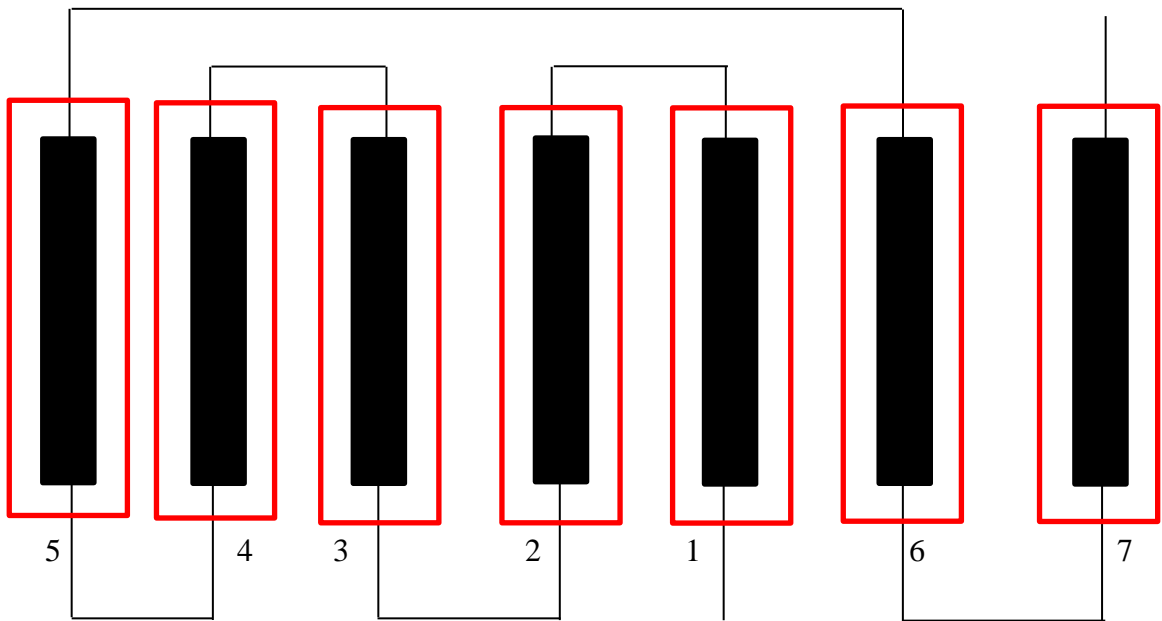


Figure 96. 3-Layer Sandwich (aba) grouping of 2fh1 Chain C Domain 3 based on the algorithm existing in the system

An example of a generated 2D cartoon of an 3-layer (aba) sandwich is shown below (see Figure 97). Its 3D representation is shown in Figure 98.

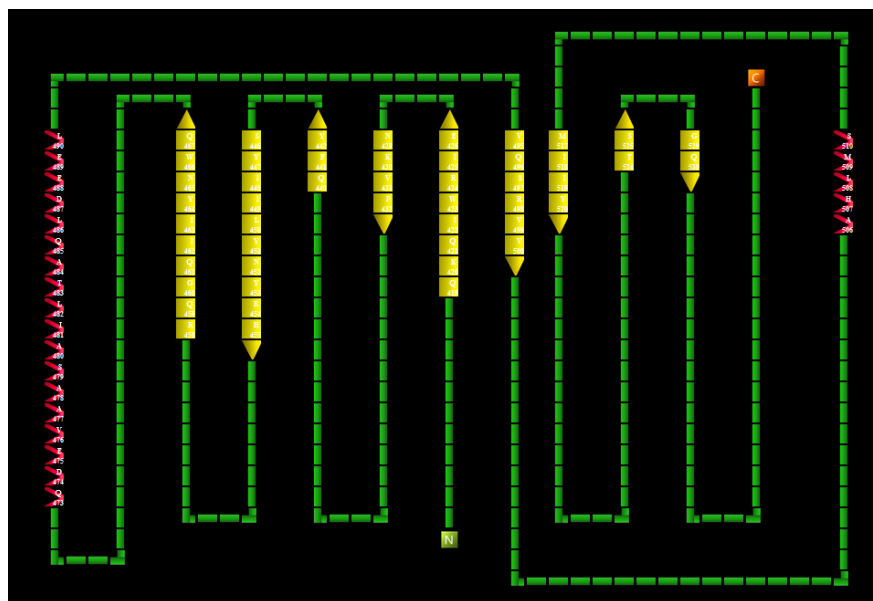


Figure 97. Generated 3-layer (aba) sandwich cartoon diagram of 2fh1 Chain A Domain 1

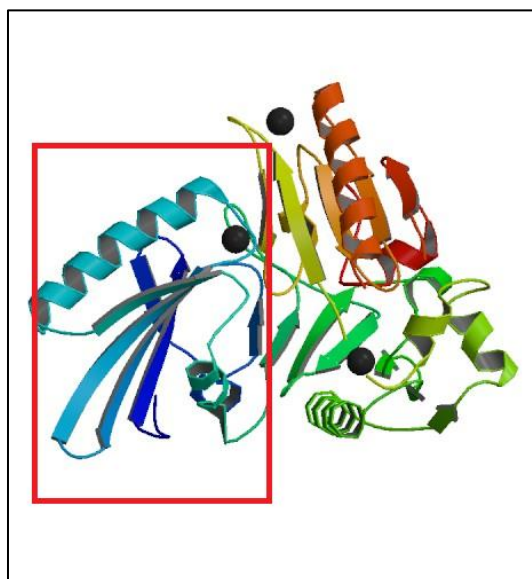


Figure 98. 3D representation of 2fh1

### Orthogonal Bundle

The algorithm for the generation of the 2D schematic diagram of orthogonal bundles is that the image would turn after a sequence of helices. The turns could either be upwards, downwards, or to the right.

Algorithm:

Get the residue names and their corresponding residue number. Save it in an array.

A variable is used in order to determine the orientation of the turns. If the value of the variable is right-down, the schematic diagram would turn downwards. On the other hand, if the value was right-up, the model would turn upwards. Otherwise, if the value was downright or upright, the turn would be towards the right.

The value of the orientation variable changes as it gets the secondary structures from the array.

An example of a generated 2D cartoon of an orthogonal bundle is shown below (see Figure 99). Its 3D representation is shown in Figure 100.

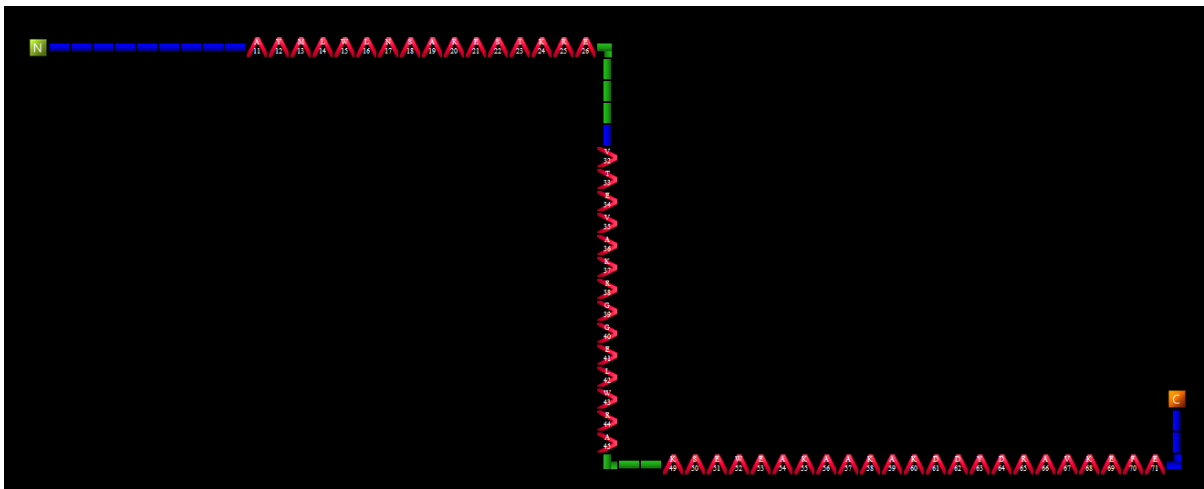


Figure 99. Generated orthogonal bundle cartoon diagram of 1qrv Chain A Domain 1



Figure 100. 3D representation of 1qrv

### Up-Down Bundle

Up-down bundle is basically an architecture consisting of secondary structures, mostly alpha helices, oriented up and down. Up-Down bundles are members of class 1 and architecture number 20.

Algorithm:

Get the residue names and their corresponding residue number. Save it in an array.

While traversing the residue array, take note if the residue constitutes a beta strand. If so, take note of its current orientation if its going to be up or down. If not, just increment the residue array traversal.

If the encountered residue name is not the same type as it was before, increment the column value to separate the new structure encountered from the old one.

An example of a generated 2D cartoon of an up-down bundle is shown below (see Figure 101). Its 3D representation is shown in Figure 102.

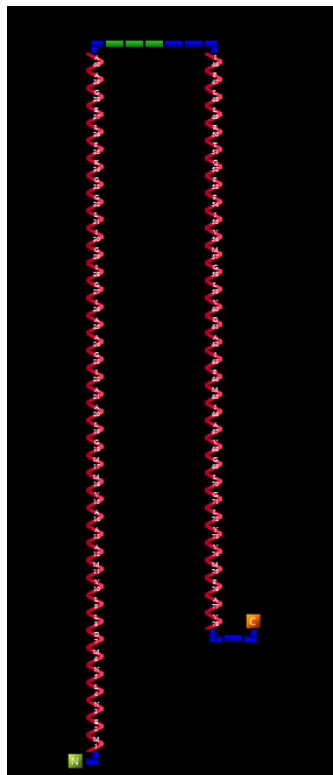


Figure 101. Generated up-down bundle cartoon diagram for 1a91

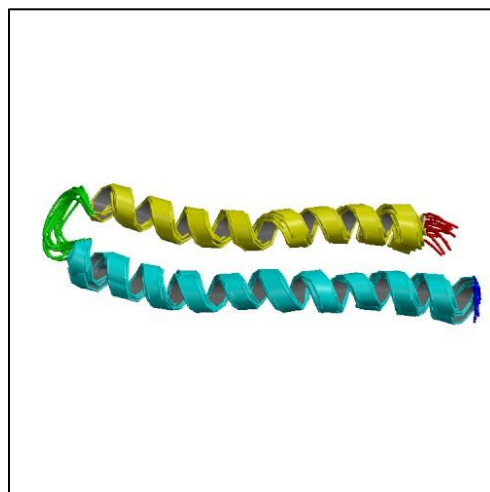


Figure 102. 3D representation of 1a91

## **Alpha Solenoid**

Alpha Solenoid is an architecture that is mostly made of alpha helices. It is drawn like a box. Alpha Solenoid bundles are members of class 1 and architecture number 40.

### **Algorithm:**

Get all the structures, beta strands and helices, and store their necessary information in an array from the output of the parsing algorithm. The information includes the structure type, first residue number, the structure group and its corresponding residue name and numbers.

Group tokens depending on type, if it is an alpha helix strand or a beta sheet strand.

Arrows indicating direction are not needed since alpha helices do not need their orientation to be noted. Take note of the length for all alpha helix groups as this will be the maximum length (vertical) of the array plus the connecting coils. After that, traverse the group again to check if the secondary structure group is equal to the maximum length. If the length of the secondary structure group is less than the maximum length, append coils to the group until its length is equal to the maximum length.

The structures will then be arranged in a specific manner forming eight groups of alpha helix secondary structure (see Figure 103).

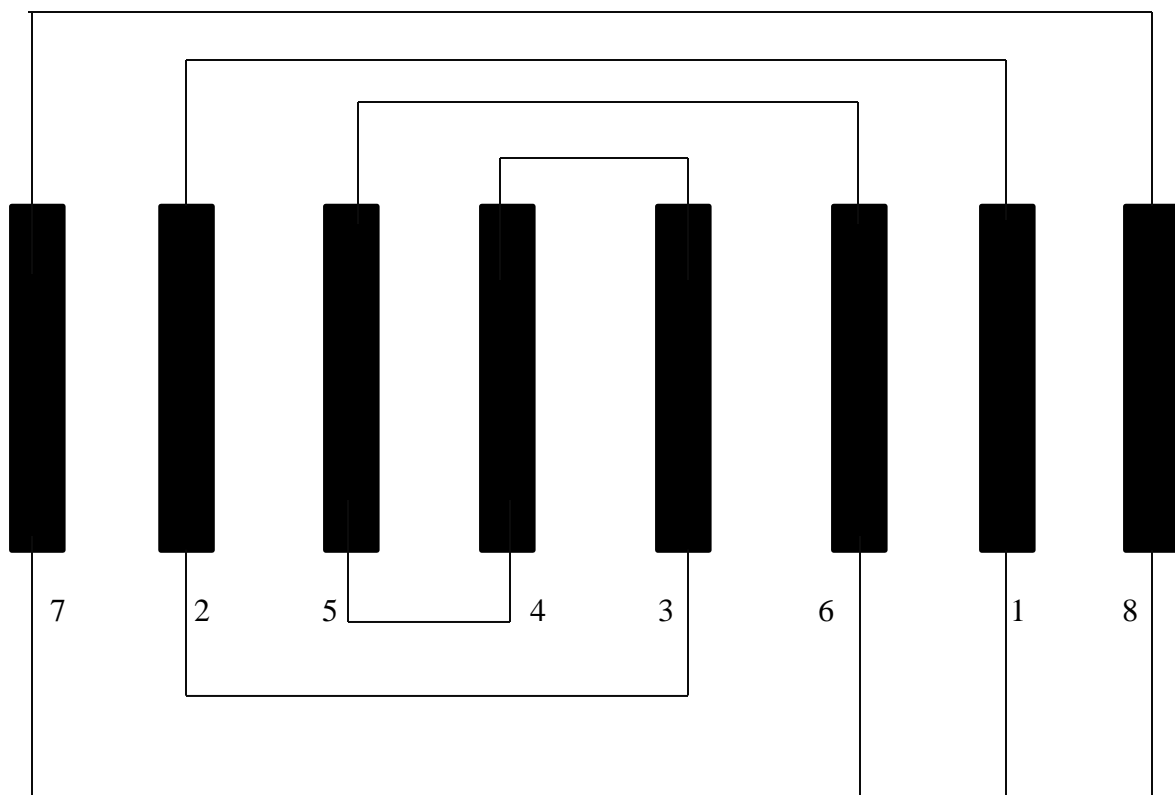


Figure 103. Alpha Solenoid grouping of 1ppr based on the algorithm existing in the system

An example of a generated 2D cartoon of an alpha solenoid is shown below  
 (see Figure 104). Its 3D representation is shown in Figure 105.

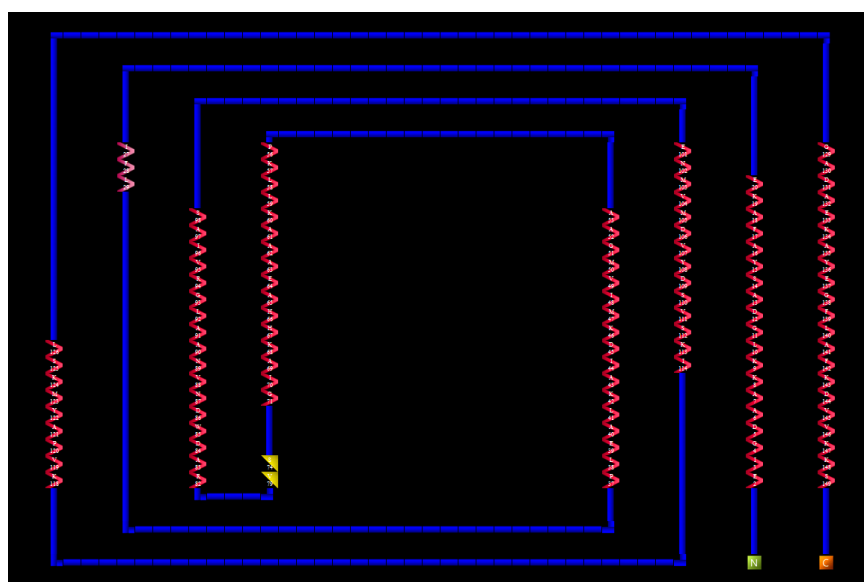


Figure 104. Generated cartoon diagram for 1ppr Chain A Domain 1

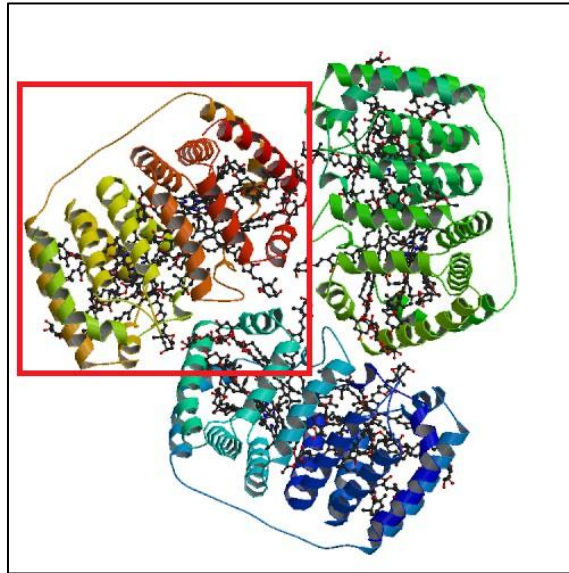


Figure 105. 3D representation of 1ppr

### Alpha Beta Barrel

Alpha Beta barrels are consists of pairs of alternating beta and alpha strand.

They belong to class 3 with architecture number 20.

Algorithm:

Get all the structures, beta strands and helices, and store their necessary information in an array from the output of the parsing algorithm. The information includes the structure type, first residue number, the structure group and its corresponding residue name and numbers.

Group the structures according to type: alpha and 310 helices, and beta strands; noting also the sequence of the structures in an array. It means that while the current structure group (a helix or a strand) is the same with the past structure group,



it will be grouped together or else the array counter will increment. The process will continue till all the structure groups are grouped.

The array contents will be printed using the image cartoons, noting that the first structure group in the pair will be a beta strand in an upward direction connected to the second structure group, the helices. After each pair, another pair of structure group will be printed. It will start to print upward based on the lowest y-coordinate the last alpha helix is printed. The process will be repeated again until all structure groups are printed.

An example of a generated 2D cartoon of an alpha beta barrel is shown below (see Figure 106). Its 3D representation is shown in Figure 107.

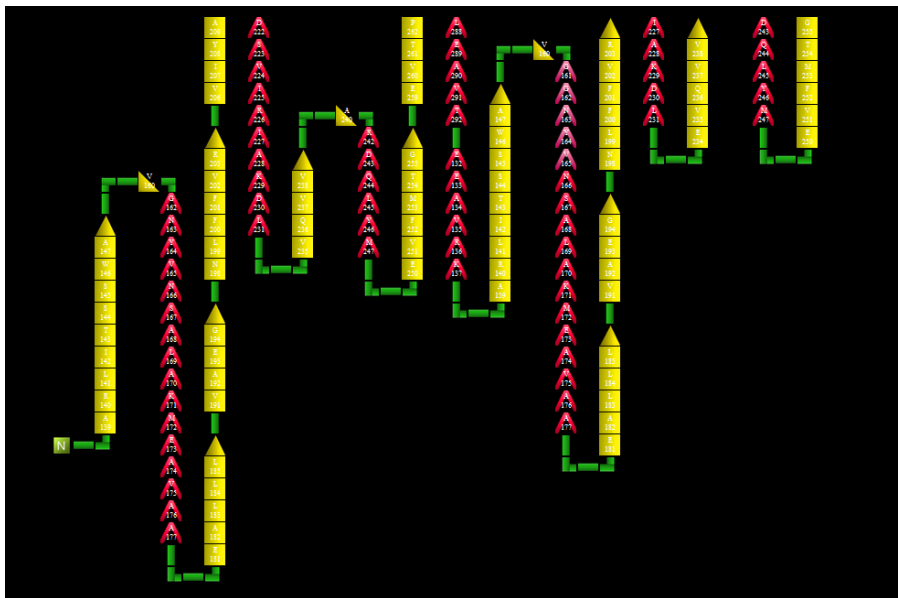


Figure 106. Generated Cartoon of 2eyj Chain A Domain 2

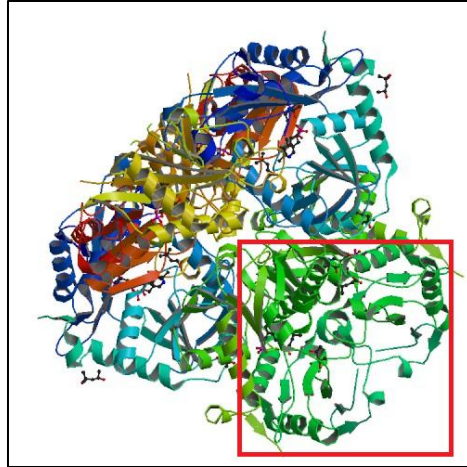


Figure 107. 3D representation of 2ei9

## Ribbon

Ribbon is an architecture wherein the structure groups (alpha helix or beta strand) are grouped into two and will form a chain of structures. It belongs to class number 2 with architecture number 10.

### Algorithm:

Initialize a main array. While traversing the sequence array obtained in the parsing algorithm, check the type of structure and if the turn flag is activated.

While a beta strand is not encountered, put the structure in the array. If a beta strand is encountered, activate turn flag. Beta strand encountered will be placed in an array and directed downward.

If turn flag is activated, put two horizontal coils in the array. After that, deactivate turn flag and start putting the next set of structure group until a beta strand is again encountered. If a beta strand is encountered, activate again the turn flag and put in the array four coils horizontally for separation. Deactivate again the turn flag.

This means that the algorithm has already encountered two beta strands and will proceed to the rest of the structures following the same pattern.

An example of a generated 2D cartoon of an alpha beta barrel is shown below (see Figure 108). Its 3D representation is shown in Figure 109.

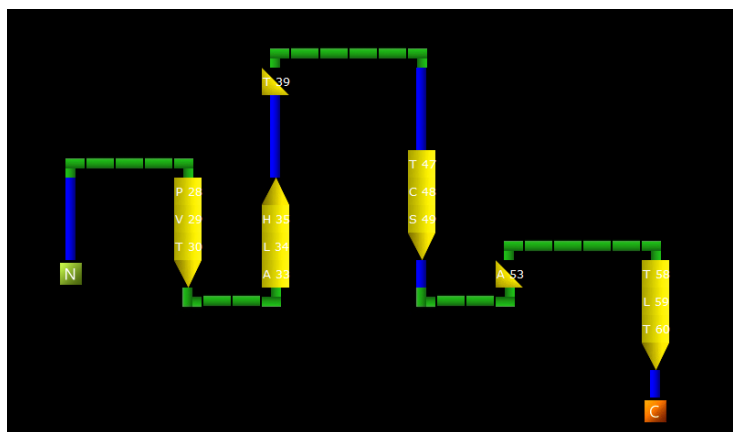


Figure 108. Generated Cartoon of 1h8p Chain A Domain 1

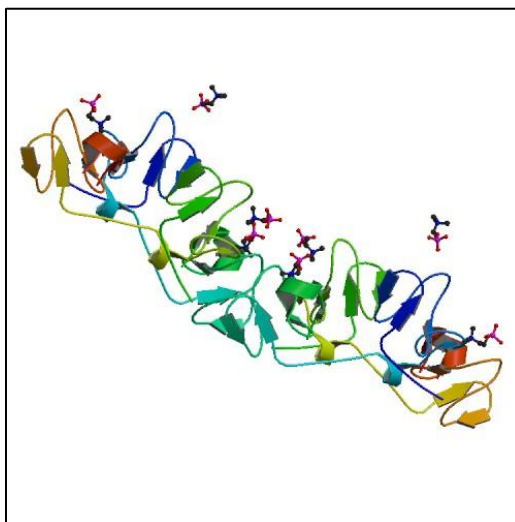


Figure 109. 3D representation of 1h8p

## Single Sheet

Single sheet is an architecture showing a sheet of structure groups (helix or beta strands). It belongs to class 2 with architecture number 20.

### Algorithm:

Traverse the array which is the output from the parsing algorithm. If any secondary structure is encountered except a coil, put it in the main array. If a turn is encountered, print coils. Put also a turn in the end to indicate that the next secondary structure is a helix or beta strand. Note that all of the beta strands are anti-parallel and will start printing vertically upward from the lowest y-coordinate of the latter strand. The algorithm will continue until the end of the sequence string outputted from the parsing algorithm.

An example of a generated 2D cartoon of an single sheet is shown below (see Figure 110). Its 3D representation is shown in Figure 111.

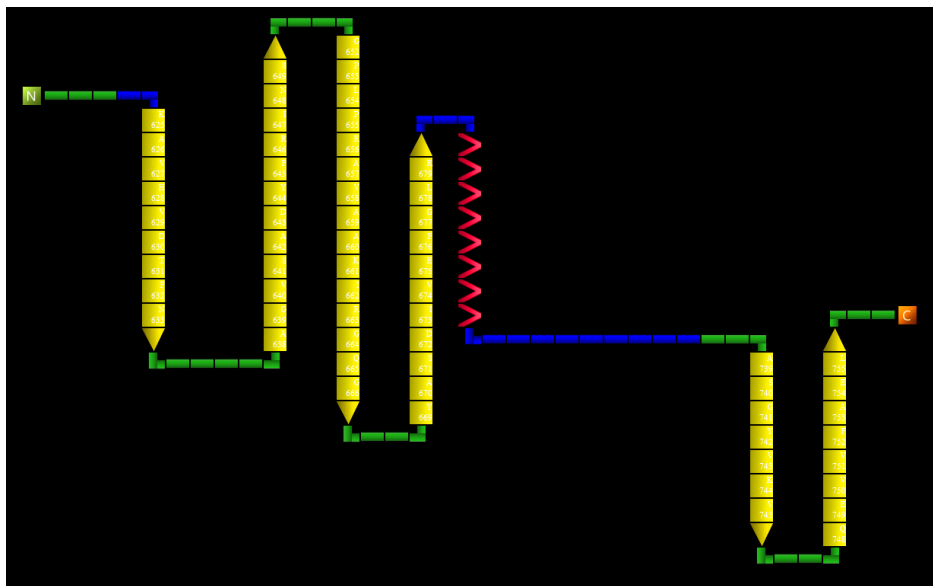


Figure 110. Generated cartoon for 1lsh Chain A Domain 3

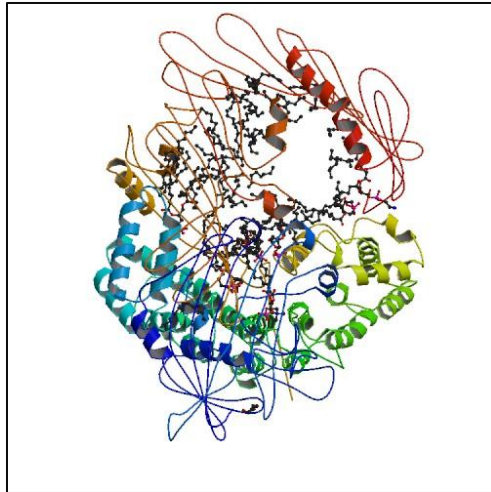


Figure 111. 3D representation of 1lsh

### Orthogonal Prism

Orthogonal prism is an architecture that only contains beta strands. The maximum number of beta strands it can contain is 12. It belongs to class 2 with architecture number 90.

Algorithm:

Traverse the array which is the output of the algorithm. Figure 112 shows the positioning of the beta strands in the final output.

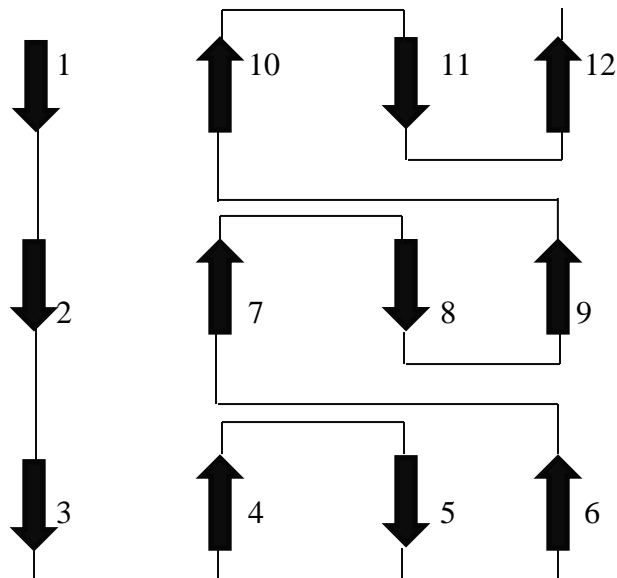


Figure 112. Topological Diagram of an Orthogonal Prism

An example of a generated 2D cartoon of an single sheet is shown below (see Figure 113). Its 3D representation is shown in Figure 114.

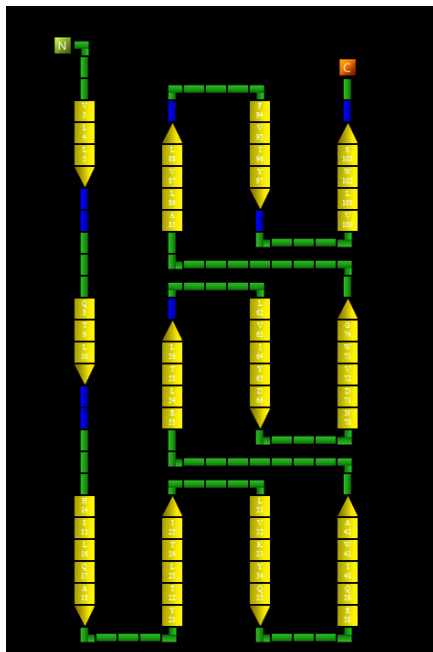


Figure 113. Generated cartoon diagram for 2dpf Chain A Domain 1

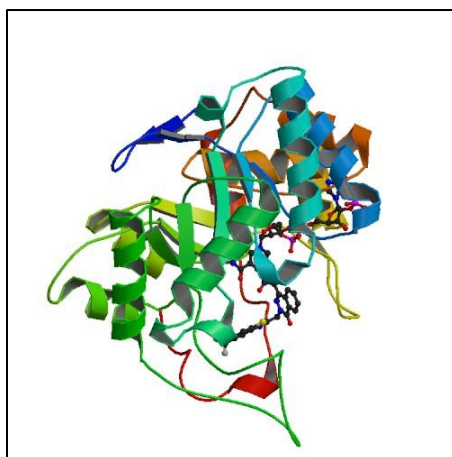


Figure 114. 3D representation of 2dpf

### 3-Propellor

3-propellor is an architecture composed of three groups of hydrogen bonding beta strands. They are usually separated by alpha helix strands. It belongs to class 2 with architecture number 105.

#### Algorithm:

Traverse the array which is the output of the parsing algorithm. If the encountered structure group is a beta strand, print the array vertically, having one strand in one column, noting its direction. The beta strands will follow anti-parallelism thus having an alternate up-down direction. The beta strands will start to print based on the y-coordinate of the last structure. Each beta strand pair will be separated by two horizontal coils. These processes will be repeated until a helix structure group is encountered or the next beta strand is not hydrogen bonded with another.

The beta strands will start to print based on the y-coordinate of the last structure. Each beta strand pair will be separated by two horizontal coils. These processes will be repeated until a helix structure group is encountered or the next beta strand is not hydrogen bonded with another.

If the encountered group is a helix (alpha or 310), it will be printed horizontally. After that, it will continue to print the remaining structures.

An example of a generated 2D cartoon of an single sheet is shown below (see Figure 115). Its 3D representation is shown in Figure 116.

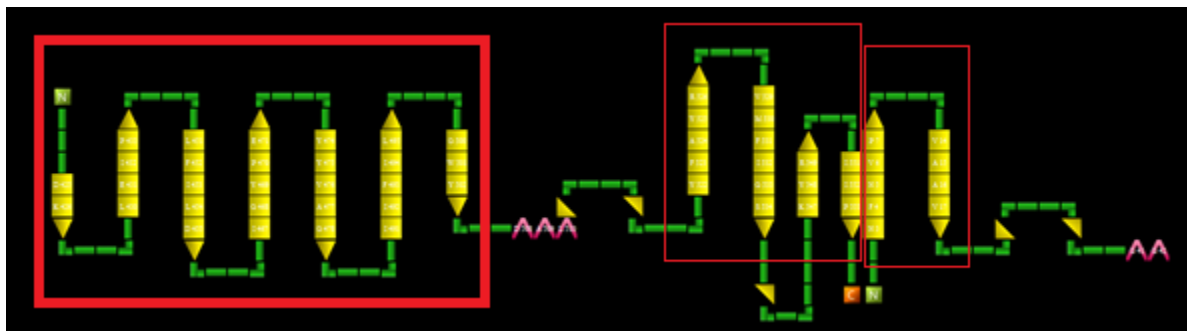


Figure 115. Generated Cartoon Diagram for 1n7v

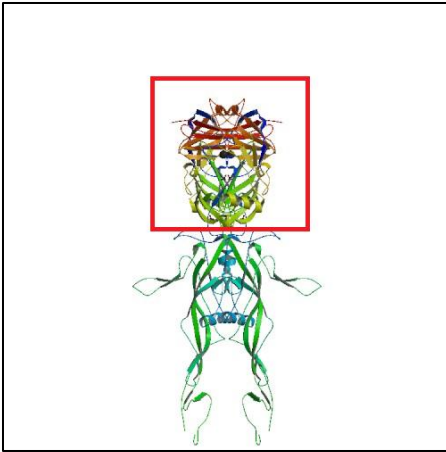


Figure 116. 3D representation of 1n7v

### 5-Propellor

5-propellor is an architecture composed of five groups of hydrogen bonding beta strands. It belongs to class 2 with architecture number 115.

#### Algorithm:

Traverse the array which is the output of the parsing algorithm. If the encountered structure group is a beta strand, print the array vertically, having one strand in one column, noting its direction. The beta strands will follow anti-parallelism thus having an alternate up-down direction.

The beta strands will start to print based on the y-coordinate of the last structure. Each beta strand pair will be separated by two horizontal coils. These processes will be repeated until a helix structure group is encountered or the next beta strand is not hydrogen bonded with another.

If the encountered group is a helix (alpha or 310), it will be printed horizontally. After that, it will continue to print the remaining structures.



An example of a generated 2D cartoon of a single sheet is shown below (see Figure 117). Its 3D representation is shown in Figure 118.

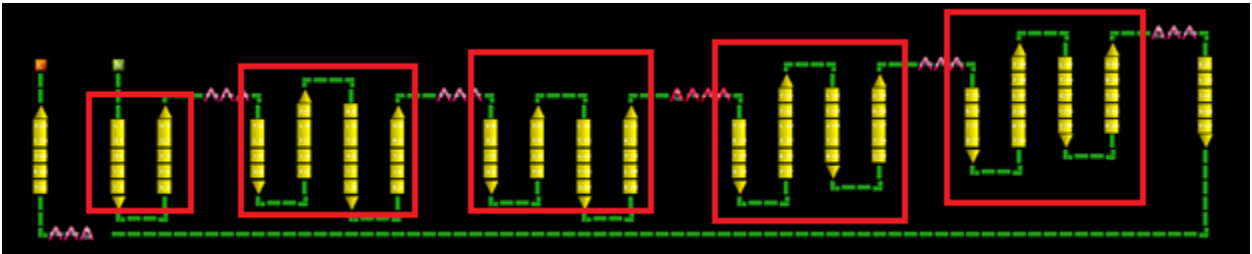


Figure 117. Generated cartoon of 1tl2

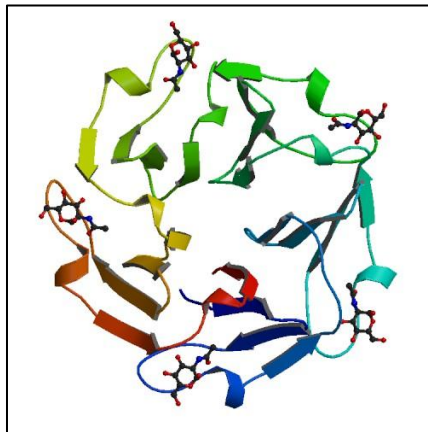


Figure 118. 3D representation of 1tl2

### 7-Propellor

7-propellor is an architecture composed of seven groups of hydrogen bonding beta strands. It belongs to class 2 with architecture number 130.

Algorithm:

Traverse the array which is the output of the parsing algorithm. If the encountered structure group is a beta strand, print the array vertically, having one

strand in one column, noting its direction. The beta strands will follow anti-parallelism thus having an alternate up-down direction.

The beta strands will start to print based on the y-coordinate of the last structure. Each beta strand pair will be separated by two horizontal coils. These processes will be repeated until a helix structure group or a non-hydrogen bonding beta strand is encountered.

If the encountered group is a helix (alpha or 310), it will be printed horizontally. After that, it will continue to print the remaining structures.

An example of a generated 2D cartoon of a single sheet is shown below (see Figure 119). Its 3D representation is shown in Figure 120.

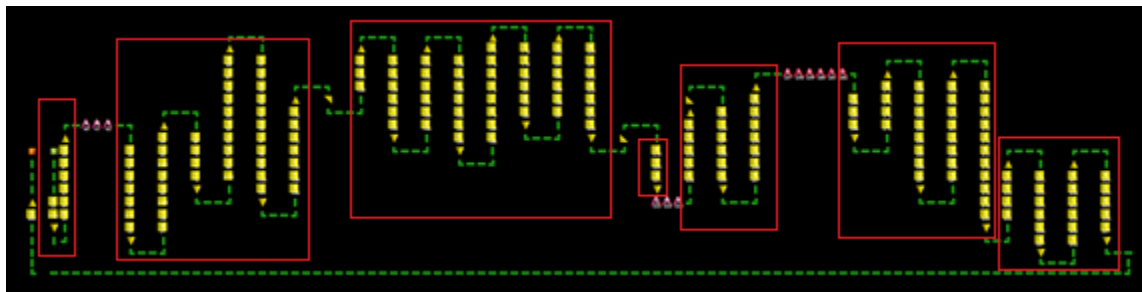


Figure 119. Generated cartoon diagram for 2bbk Chain A Domain 1

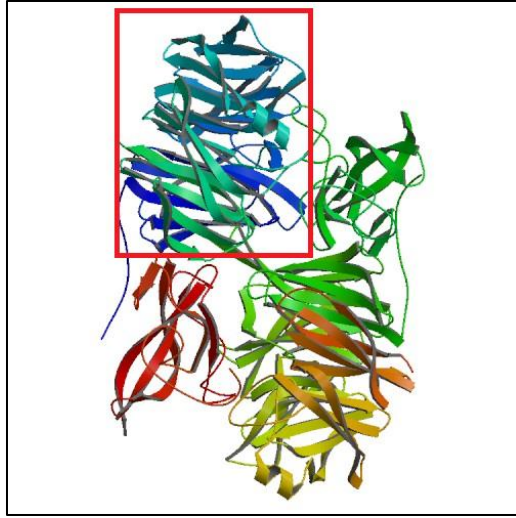


Figure 120. 3D representation of 2bbk

## CHAPTER 7: CONCLUSION

The STRIDE Protein Topology Cartoon Generator and Database (SPCTGaD) is a system that mainly displays the 2D representation of protein domains. It is based from an external application, the STRIDE algorithm, which predicts the secondary structure sequence of the protein domain.

A PDB file is parsed using the STRIDE algorithm. The output of the algorithm will be then the input of the architecture algorithms that generates the 2D representation. The 2D representations focus on the secondary structure helices and beta strands.

The parsing of files are tools that can be used by registered users. Non-registered users can search protein domains, which are only available in the system, and compare them. Registration for an account in the system is also available and can be approved or disapproved by the system administrator. Sending message to the system administrator for inquiry, support, report bug and other concerns has also been made available. The system administrator, being the one monitoring the site, is able to approve or disapprove accounts for application in the system. He or She can also view the messages sent by users and delete them.

## CHAPTER 8: RECOMMENDATION

The STRIDE Protein Topology Cartoon Generator and Database (SPTCGaD) system can be further improved by implementing the following:

- Completion of other protein architectures (see Table 11 for available and not available architectures)

Representative Protein Domain	Representative Protein Domain Subfamily	Architecture Availability
1. Bundles	1.1 Orthogonal Bundle	Available
	1.2 Up-Down Bundle	Available
2. Barrels	2.1 Alpha-Beta Barrel	Available
	2.2 Alpha Barrel	Available
	2.3 Beta Barrel	Not Available
3. Ribbon	Ribbon	Available
4. Single Sheet	Single Sheet	Available
5. Sandwiches	5.1 Sandwich	Not Available
	5.2 2-Layer Sandwich	Available
	5.3 3-Layer Sandwich	Not Available
	5.4 3-Layer (aba) Sandwich	Available
	5.5 3-Layer(aab) Sandwich	Not Available
	5.6 3-Layer(bba) Sandwich	Not Available
	5.7 3-Layer(bab) Sandwich	Not Available
	5.8 4-Layer Sandwich	Not Available
	5.9 Distorted Sandwich	Not Available
6. Prisms	6.1 Orthogonal Prism	Available
	6.2 Aligned Prism	Not Available
	6.3 Alpha-Beta Prism	Not Available
7. Propellers	7.1 3-Propellor	Available
	7.2 5-Propellor	Available
	7.3 7-Propellor	Available
	7.4 4-Propellor	Not Available
	7.5 6-Propellor	Not Available
	7.6 8-Propellor	Not Available
	7.7 5-stranded Propeller	Not Available
8. Solenoids	8.1 Alpha solenoid	Available
	8.2 2-Solenoid	Not Available
	8.3 3-Solenoid	Not Available
9. Horseshoes	9.1 Alpha Horseshoe	Not Available
	9.2 Alpha-Beta Horseshoe	Not Available
10. Rolls	10.1 Roll	Available
	10.2 Super Roll	Not Available
11. Clams	Clam	Not Available
12. Trefoils	Trefoil	Available
13. Complexes	13.1 Beta Complexes	Not Available
	13.2 Alpha-Beta Complexes	Not Available

14. Box	Box	Not Available
15. Ribosomal Protein	Ribosomal Protein L15	Not Available
16. Irregular	Irregular	Not Available

Table 11 shows the updated list of representative protein domain algorithms based on CATH that are now existing in SPTCGaD

- Exportation of the 2D cartoon generated in a PDF or image file
- Usage of different secondary structure prediction algorithms such as DSSP for comparison of the 2D cartoons
- Add other tools for protein analysis like generation of Ramachandran plots of proteins

## CHAPTER 9: BIBLIOGRAPHY

### References:

- [1] Crooks , Gavin E., and Brenner, Steven E. “Protein secondary structure: entropy, correlations and prediction” *Bioinformatics* 20/10 (2004): 1603–1611
- [2] Heinig, Matthias, and Frishman, Dmitrij. “STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins”. *Nucleic Acids Research* 32/Web Server issue (2004)
- [3] Michalopoulos, Ioannis. et. al. “TOPS: an enhanced database of protein structural topology” 32. *Nucleic Acids Research* (2004):D251-D254
- [4] Martin, Andrew C.R. <http://www.bioinf.org.uk/talks/topscan>
- [5] Bond, Charles S. “TopDraw: a sketchpad for protein structure topology cartoons” 19. *Bioinformatics Applications Note* (2003):311-312
- [6] Tyagi, Manoj, et al. “Analysis of loop boundaries using different local structure assignment methods” (2009)
- [7] Best, Beuth B., et al. “PDBe: Protein Data Bank in Europe” *Nucleic Acids Research* 38/Database issue (2009)
- [8] Orengo, C. A. et. al. “The CATH database provides insights into protein structure/function relationships” 27/1. *Nucleic Acids Research* (1999):275-279
- [9] Peng, Zou. Shang, Zhicai. “2D molecular graphics: a flattened world of chemistry and biology” 10/3. *Briefings in Bioinformatics* (2009):247-258
- [10] Kurgan, Lukasz A. et. al. “Secondary structure-based assignment of the protein structural Classes” *Amino Acids* (2008):551-564
- [11] <http://www.cathdb.info/>
- [12] Martin, Juliette, et al. “Protein secondary structure assignment revisited: a detailed analysis of different assignment methods” *BMC Structural Biology* (2005): 5-17
- [13] May, Patrick. et al. “PTGL: a database for secondary structure-based protein topologies” *Nucleic Acids Research* 38/Database issue (2004)
- [14] Rose, Peter W., et al. “The RCSB Protein Data Bank: redesigned web site and web services” *Nucleic Acids Research* 39/Database Server issue (2010)

- [15] Portugal, Elon et al. "Assessment of Protein Domain Classifications: SCOP, CATH, Dali and EVEREST" (2008)
- [16] Cuff, Allison L. et al. "The CATH classification revisited—architectures reviewed and new ways to characterize structural divergence in superfamilies" 37. *Nucleic Acid Research* (2009)
- [17] Klose, D. P., et al. "2Struc: the secondary structure server" *Structural Bioinformatics* 26/10 (2010):2624-2625
- [18] Veeramalai, Mallika. Gilbert, David. "A novel method for comparing topological models of protein structures enhanced with ligand information" 24/23 *Structural Bioinformatics* (2008):2698–2705
- [19] Mallika, Veermalai. et al. "TOPS++FATCAT: Fast flexible structural alignment using constraints derived from TOPS+ Strings Model" *BMC Bioinformatics* (2008)
- [20] Stivala, Alex. et al. "Automatic generation of protein structure cartoons with Pro-origami Bioinformatics" doi:10.1093/bioinformatics/btr575:(2011)
- [21] Martin, Andrew C.R. "The Ups and Downs of Protein Topology; Rapid Comparison of Protein Structure" 13/12. *Protein Engineering* (2000):829-837
- [22] Nagano, Nozomi et. al. "Barrel structures in proteins: Automatic identification and classification including a sequence analysis of TIM barrels" *Protein Science* (1999):2072-2084
- [23] ExSer: A standalone tool to mine protein data bank (PDB) for secondary structural elements
- [24] [http://www.pdb.org/pdb/file\\_formats/pdb/pdbguide2.2/PDB\\_format\\_1992.pdf](http://www.pdb.org/pdb/file_formats/pdb/pdbguide2.2/PDB_format_1992.pdf)
- [25] [http://deposit.rcsb.org/adit/docs/pdb\\_atom\\_format.html](http://deposit.rcsb.org/adit/docs/pdb_atom_format.html)
- [26] <http://www.biomedsearch.com/nih/Atomic-interaction-networks-in-core/20186337.html>
- [27] <http://structure.usc.edu/stride/#output>
- [28] <http://searchsqlserver.techtarget.com/definition/database-management-system>
- [29] <http://www.computingstudents.com/dictionary/index.php?word=database+management+system>
- [30] [http://whatis.techtarget.com/definition/0,,sid9\\_gci212396,00.html](http://whatis.techtarget.com/definition/0,,sid9_gci212396,00.html)
- [31] Stair, R. Reynolds, G. *Principles of Information Systems*. 2008.