2017

Plat: A Web Based Protein Local Alignment Tool

Stephen H. Jaegle
University of Rhode Island, sjaegle@uri.edu

PLAT: A WEB BASED PROTEIN LOCAL ALIGNMENT TOOL

BY

STEPHEN H. JAEGLE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER SCIENCE

UNIVERSITY OF RHODE ISLAND
2017

MASTER OF SCIENCE THESIS

OF

STEPHEN H. JAEGLE

APPROVED:

Thesis Committee:

Major Professor Lutz Hamel

Victor Fay-Wolfe

Ying Zhang

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2017

ABSTRACT

Protein structure largely determines functionality; three-dimensional structural alignment is thus important to analysis and prediction of protein function. Protein Local Alignment Tool (PLAT) is an implementation of a web-based tool with a graphic interface that performs local protein structure alignment based on user-selected amino acids. Global alignment compares entire structures; local alignment compares parts of structures. Given input from the user and the RCSB Protein Data Bank, PLAT determines an optimal translation and rotation that minimizes the distance between the structures defined by the selected inputs.

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my major professor Dr. Lutz Hamel, without whose support and patience this work would not have been completed.

I would also like to thank the members of my committee, Dr. Fay-Wolfe and Dr. Zhang, for their willingness to participate in my defense.

This work owes its existence in no small part to the efforts of faculty and staff of the Department of Computer Science and Statistics from the technical support of Dr. Bryan and the enabling wisdom of Jake Fonseca and Dr. Fay-Wolfe to the championing support of Dr. Peckham and Dr. Baudet.

Lastly, the events that led to this work would not have been started without the support of my family, especially my mother.

TABLE OF CONTENTS

CHAPTER

Page

APPENDIX

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Protein-protein interactions are fundamental to cellular activities. The three-dimensional structure of a protein largely determines how it functions with other molecules on the proteomic scale, and local structural similarities may be used to predict function [1, 2].

Families of related proteins that perform similar functions can have members with varying degrees of dissimilarity in overall conformation with similar functional centers or substructures preserved across members. The functionality of each member of such a family results from its unique grouping of similarities and dissimilarities; for example, each protein in a family such as the Ras superfamily [3] may perform a similar function using a similar functional center, but in response to different environmental conditions that interact with the dissimilar parts of the structures [4].

A protein is not necessarily constrained to having a single shape; it may fold into different structural conformations as a result of the presence of other proteins or other environmental factors. Some conformational changes may result in disease, as in Alzheimers disease, where misfolded amyloid beta peptide 1-42 proteins that normally penetrate the neuron cell membrane instead accumulate as plaques outside the cell [5].

The Research Collaboratory for Structural Bioinformatics Protein Data Bank (RCSB PDB) [6] stores files that describe three dimensional structure and additional attributes of proteins and other macromolecules. The PDB web site [7] offers search options for locating PDB entries and tools for visualizing proteins. PDB entries are frequently used to provide data for structural comparisons. The PDB contains over 128,000 entries and is accessed monthly by about 286,000 unique visitors [8], and calls itself “the single worldwide archive of structural data of biological macromolecules” [9].

Structural alignment techniques compare three-dimensional structure, and deal largely with tertiary and secondary structure. Structural alignment relies on information about three-dimensional conformations, and so can only be used with structural data. Though structures are usually determined through experimental methods and stored in the PDB [7, 6], theoretical structures may be constructed by structure prediction methods.

Protein structural alignments may be global or local. A global alignment of two proteins endeavors to align the spatial structure of one protein with the spatial structure of another protein. The distance between structures is generally measured by the root-mean-square deviation (RMSD) distance between the aligned input structures [10]. While global alignments take into account overall protein structure, local alignments perform similar operations on selected local sections of the global structure, minimizing the RMSD of the sections under consideration without regard to the remainder of global structure. Local alignment techniques can facilitate the analysis of local structural similarities that may predict function. Global alignment may align local substructures sub-optimally in favor of less similar but more numerous correspondences between the other parts of the proteins. If two proteins have identical sub-structures accompanied by significantly different overall structures, then a global alignment may not align the substructures optimally.

Existing tools that perform local alignments may restrict selection of substructure to predefined regions of a molecule, or may constrain the correspondence between query and reference structures in ways that make it difficult to compute alignments between local substructures such as functional centers or binding sites.

1.1 Goal and Objectives

The goal of this work is to develop a web-based application that performs local protein molecule structure alignments based on user-selected amino acid sequences in each molecule being aligned. Toward that goal, this work has the following objectives:

1. The application will be web-based, with a graphical user interface (GUI) for selecting regions of structure.
2. The user interface will provide cues that facilitate correct identification of the items being selected.
3. The application will obtain structure data from the PDB.
4. The application will perform local structural alignments.

1.2 Results

The PLAT application described herein meets the objectives set forth above. PLAT performs local protein molecule structure alignment between two molecules based on user-selected amino acids. Amino acids are selected through a web-based GUI that obtains structure data from the PDB.

1.3 Outline

The PLAT application is described in the following sections. The Background section describes how protein structure data is represented in the PDB and the mechanisms for programmatically retrieving and processing PDB data; related work is also described. The Methods section describes the computation of alignments, the application design and principal technologies employed, and communication between the user and between application components. The Results section compares the application and the objectives described, and describes the observed results produced by the application. Finally, the Conclusions section discusses conclusions and further work.

CHAPTER 2

Background

2.1 Protein Data Bank Representation of Structure

PDB entries may contain general structural descriptions, citations of papers that describe the molecule, the experimental methods used to determine its structure and its sequence of amino acid residues, a list of atoms and their coordinates, secondary structure annotations, and disulfide bonds and other linkages. Experimental methods used to determine structure include X-ray crystallography, NMR spectroscopy, and cryo-electron microscopy [11].

2.1.1 Organizational Levels of Protein Structure

From a structural perspective, protein molecules are comprised of one or more chains of amino acid residues held together by peptide bonds. Amino acids have a structure consisting of atoms, each of which has a location in three dimensional space. The structure of a protein molecule is characterized at four levels of organization: primary, secondary, tertiary, and quaternary; the three latter levels describe the three-dimensional spatial arrangement of the protein molecule. Each level is represented in the PDB.

2.1.2 Primary Structure in the PDB

Primary structure is the sequence of amino acids in a peptide bonded chain. Amino acids are formed from a carboxyl group (-COOH), an amine group (-NH2), and a carbon atom, termed the α-carbon, bound to a side chain, or R-group, that varies with the individual amino acid. Figure 1 illustrates the amino acid Lysine, with the α-carbon atom bound to the carboxyl acid group, amine group, and the side chain contains four additional groups with carbon atoms labeled β through ϵ. The amine groups of one amino acid forms peptide bonds with the carboxyl group of another amino acid [12] in succession to form a chain. The sequence of α-carbons associated with the peptide bonds linking amino acid residues forms the backbone of the amino acid chain. As primary structure is represented as the sequence of linked amino acids, the structural representation of the backbone is the sequence of spatial coordinates of its α-carbons.

The PDB represents primary structure in the form of SEQRES [13] records that contain the sequence of three-character amino acid codes. Table 2 shows an example of PDB SEQRES records for protein P21-H-Ras. Figure 2 shows a 3D rendering of P21-H-Ras structure with C, O, and N atoms represented as spheres, while Figure 3 illustrates the corresponding backbone formed by the sequence of bonds between the amino acids, using the spatial coordinates of the amino acid α-carbons. Appendix A contains a table of amino acid names and codes.

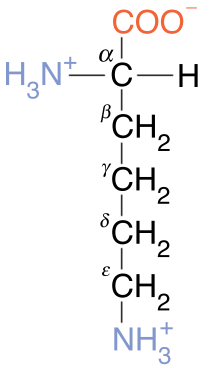


Figure 1: Lysine with side chain carbon atoms

2.1.3 Secondary Structure in the PDB

Secondary structure is a higher level of representation of three dimensional structure describing localized three dimensional features. Hydrogen-bonding interactions between amino acid residues along the polypeptide chain can cause the chain to fold into characteristic localized stable structures or pleated segments, the most common of which are α-helices and β-sheets. Secondary structure turns pro-

ATOM	37	N	LYS	A	5	-3.316	27.216	3.592	1.00	15.93	N
ATOM	38	CA	LYS	A	5	-3.586	27.580	4.978	1.00	16.80	C
ATOM	39	C	LYS	A	5	-2.569	26.825	5.797	1.00	14.24	C
ATOM	40	O	LYS	A	5	-2.689	25.602	5.979	1.00	14.36	O
ATOM	41	CB	LYS	A	5	-5.006	27.164	5.387	1.00	18.79	C
ATOM	42	CG	LYS	A	5	-5.762	28.262	6.168	1.00	24.69	C
ATOM	43	CD	LYS	A	5	-5.662	28.204	7.726	1.00	25.19	C
ATOM	44	CE	LYS	A	5	-4.314	28.470	8.439	1.00	23.75	C
ATOM	45	NZ	LYS	A	5	-3.792	29.796	8.192	1.00	22.37	N

Table 1: PDB ATOM Records Listing P21-H-Ras Chain A Lysine at position 5 (Source: PDB entry 121P [7, 6, 14])

SEQRES	1	A	166	MET	THR	GLU	TYR	LYS	LEU	VAL	VAL	VAL	GLY	ALA	GLY	GLY
SEQRES	2	A	166	VAL	GLY	LYS	SER	ALA	LEU	THR	ILE	GLN	LEU	ILE	GLN	ASN
SEQRES	3	A	166	HIS	PHE	VAL	ASP	GLU	TYR	ASP	PRO	THR	ILE	GLU	ASP	SER
SEQRES	4	A	166	TYR	ARG	LYS	GLN	VAL	VAL	ILE	ASP	GLY	GLU	THR	CYS	LEU
SEQRES	5	A	166	LEU	ASP	ILE	LEU	ASP	THR	ALA	GLY	GLN	GLU	GLU	TYR	SER
SEQRES	6	A	166	ALA	MET	ARG	ASP	GLN	TYR	MET	ARG	THR	GLY	GLU	GLY	PHE
SEQRES	7	A	166	LEU	CYS	VAL	PHE	ALA	ILE	ASN	ASN	THR	LYS	SER	PHE	GLU
SEQRES	8	A	166	ASP	ILE	HIS	GLN	TYR	ARG	GLU	GLN	ILE	LYS	ARG	VAL	LYS
SEQRES	9	A	166	ASP	SER	ASP	ASP	VAL	PRO	MET	VAL	LEU	VAL	GLY	ASN	LYS
SEQRES	10	A	166	CYS	ASP	LEU	ALA	ALA	ARG	THR	VAL	GLU	SER	ARG	GLN	ALA
SEQRES	11	A	166	GLN	ASP	LEU	ALA	ARG	SER	TYR	GLY	ILE	PRO	TYR	ILE	GLU
SEQRES	12	A	166	THR	SER	ALA	LYS	THR	ARG	GLN	GLY	VAL	GLU	ASP	ALA	PHE
SEQRES	13	A	166	TYR	THR	LEU	VAL	ARG	GLU	ILE	ARG	GLN	HIS			

Table 2: PDB SEQRES Records Listing P21-H-Ras Primary Structure (Source: PDB entry 121P [7, 6, 13])

vide links among helices and sheets. Secondary structure not classified as helices, sheets, or turns is classified as coil. Figure 4 illustrates the secondary structures of P21-H-Ras.

The PDB represents secondary structure through HELIX and SHEET records, where HELIX records are named, numbered, and typed, and SHEET records are named and numbered, and both include the sequence numbers of their initial and final residues [15]. For example, Table 3 contains the HELIX records for P21-H-Ras, and Table 4 contains the SHEET records.

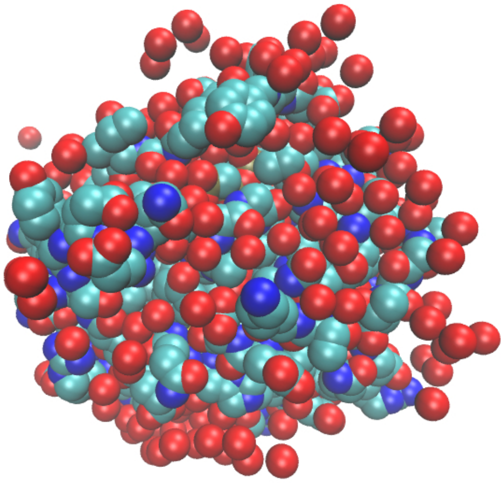


Figure 2: 3D Representation of P21-H-Ras Atoms as Spheres

Color Legend: Oxygen, Nitrogen, Carbon, Hydrogen not shown

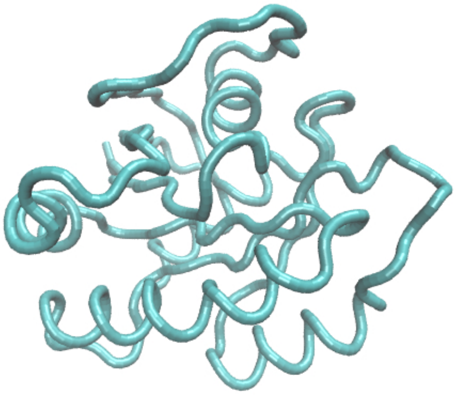


Figure 3: 3D Representation of P21-H-Ras Backbone

2.1.4 Tertiary and Quaternary Structure

As an arrangement of secondary structure elements [16], tertiary structure may arise from bonds between amino acids not adjacent along the backbone and from interactions with the surrounding environment, such as hydrophobic effects with a solvent that often result in a roughly globular shape. Figures 2, 3, and 4 are all representative of the tertiary structure of P21-H-Ras. Similarly, quaternary structure is the structure resulting from amino acid residue interactions between two or more polypeptide chains each of which has tertiary structure. PDB entries

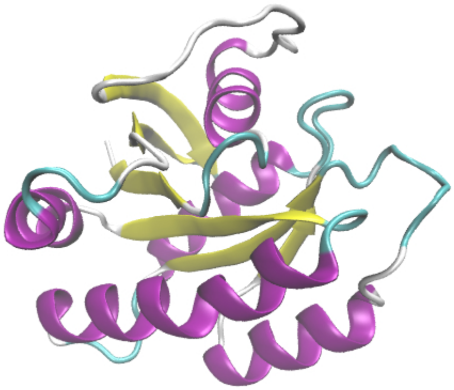


Figure 4: 3D Representation of P21-H-Ras Secondary Structure

Color Legend: α-helices, β-sheets, turns, coil

HELIX	1	A1	LYS	A	16	GLN	A	25	1	10
HELIX	2	A2	SER	A	65	THR	A	74	1	10
HELIX	3	A3	THR	A	87	VAL	A	103	1	17
HELIX	4	A4	SER	A	127	TYR	A	137	1	11
HELIX	5	A5	VAL	A	152	ARG	A	164	1	13

Table 3: HELIX Records for P21-H-Ras

Note: Empty spaces represent empty fields (Source: PDB entry 121P [7, 6, 15])

do not have explicit tertiary or quaternary structure records; instead, tertiary and quaternary structures are implicitly expressed as the structure of the chains and the structure of the entire molecule.

2.2 PDB Application Programming Interfaces

In addition to web browser based tools for exploring, visualizing, and comparing protein data, the PDB offers an extensive application programming interface (API). APIs specify the expected behavior of and interaction between software components, particularly with respect to syntax, inputs, and outputs, facilitating the sharing of data among components.

The method of exposing an API, i.e., making it available for communication with other software components, varies with the implementation. Web APIs are generally exposed via Hypertext Transfer Protocol (HTTP) protocol, and common

SHEET	1	S	6	GLU	A	37	ILE	A	46	0								
SHEET	2	S	6	GLU	A	49	THR	A	58	-1	O	LEU	A	53	N	LYS	A	42
SHEET	3	S	6	THR	A	2	VAL	A	9	1	N	LEU	A	6	O	ASP	A	54
SHEET	4	S	6	GLY	A	77	ALA	A	83	1	N	LEU	A	79	O	VAL	A	7
SHEET	5	S	6	MET	A	111	ASN	A	116	1	N	VAL	A	114	O	CYS	A	80
SHEET	6	S	6	TYR	A	141	GLU	A	143	1	N	ILE	A	142	O	LEU	A	113

Table 4: SHEET Records for P21-H-Ras

Note: Empty spaces represent empty fields (Source: PDB entry 121P [7, 6, 15])

practice is to make them available via URL as a set of *remote procedure call (RPC) endpoints* [17]. Exposed API components are often generally referred to as web services, and earlier versions of Internet web services were commonly implemented using Simple Object Access Protocol (SOAP) though they have been supplanted by implementations using representational state transfer (REST) [18]. REST does not specify a particular implementation, [19], and REST implementations are referred to as RESTful.

The PDB exposes some 30 RESTful web services classified into search or fetch categories [20]. Many of the current RESTful web services were available as SOAP services until the PDB retired its SOAP services in 2013 [21]. The PDB API makes available third party annotations that are supplied using the Distributed Annotation System (DAS) [22, 23], which distributes data across multiple sites and makes it available to clients as a single view. In 2011, there were an estimated 1,200 DAS servers [23].

The PDB third party annotation web service `pdbchainfeatures` [20] makes available third party features that include Dictionary of Protein Secondary Structure (DSSP) annotations for secondary structure [24]. DSSP assigns secondary structure according to hydrogen bond patterns, and has been accepted as a "gold standard" [24]. Not all PDB files contain structure fields, and when they are present, they may not be complete [24]. The PDB uses DSSP secondary structure as a default in its *Sequence Chain View*, as shown in Figure 5.

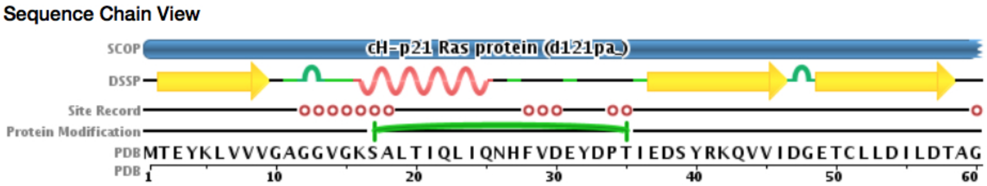

Sequence Chain View
SCOP
cH-p21 Ras protein (d121pa_)
DSSP
Site Record
Protein Modification
PDB
MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSYRKQVVIDGETCLLDILDTAG
PDB
1
10
20
30
40
50
60

Figure 5: PDB sequence chain view example

Source: PDB [25]

2.3 BioJava Library

The open-source BioJava project provides a Java framework for processing biological data [26]. It provides a toolkit of modules and APIs that load and parse pdb files, perform standard sequence and structure alignments, and allow the manipulation of sequences and 3D structures [27]. BioJava models data from a PDB file as a `Structure` object with methods for accessing header information and data. Unlike a PDB file, `Structure` maintains data as a hierarchy of sub-objects [28] facilitating the use of object oriented programming to access the data. The BioJava library provides functionality similar to both BioPerl [29] and Biopython [30] and uses Java, the same language as Google Web Toolkit discussed in §3.2.1.

2.4 Local vs Global Structural Alignment

Global structural alignment seeks an alignment optimization over the overall three dimensional structure of two proteins or chains. Local structural alignment seeks an alignment optimization over local parts of proteins or chains. Whole protein structures may contain information that interferes with optimizing the structural alignment of functional sites. Some alignment methodologies refine alignment by discarding atom pairs that diverge by more than some threshold.

Use cases for local alignment involve examination of relationships between structural arrangements where information from specific parts of molecules are to be used to the exclusion of others. For example, superpositions between neomycin-bound and paromomycin-bound ribosomes were performed while excluding "disordered or flexible" regions of 23S rRNA in [31]; in another example, an alignment between BRCA1-BARD1 and Ring1B-Bmi1 was performed using BRCA1 residues 2255 and 6076 and Ring1B residues 4979 and 86102 [32]. In a related use case, local alignments may be used to align and measure RMSD between existing and modeled structures [33], or in nanostructure modeling, as in [34], where local alignments were used to dock L-shaped monomers and KL complexes using four researcher-specific atoms.

2.5 Related Work

Existing structural alignment tools perform both global and local structural alignments. Structural alignment tools such as VAST [35] and DALI [36] perform alignment based primarily on secondary structure [4]. SSAP [37], based on DSSP [38, 39] employs a standard dictionary of secondary structure. Rosetta@home [40], the protein structure prediction distributed computing project, performs comparisons between sub-sequences of protein structure as part of computing predicted structure. Flexible structure AlignmenT by Chaining Aligned fragment pairs with Twists (FATCAT) [41] is available as a web-based tool that provides flexible pairwise 3D structure alignments. These tools and others use global alignment techniques.

While tools such as VAST [35, 42] employ identification of similar secondary structure elements as a step in producing global alignment, they do not offer a researcher the ability to select a local substructure and align proteins in favor of minimizing RMSD between the selected query and reference substructures. A commercial application named DS ViewerPro, from Accelrys, Inc., allows a researcher to select individual atoms to be aligned, but not to select amino acids for

3D structural alignment other than by selecting individual atoms of the residue.

Another application, Visual Molecular Dynamics (VMD) [43], allows alignments based on user selected residues restricted to the same positions in each primary sequence, i.e., the residues in positions i through j of one sequence must be aligned with the residues in the same positions i through j of another sequence when using only residue numbers. To align residues with different numbers in each sequence, VMD users must specify the residue number together with another attribute such as the name for the residues in each sequence, e.g. `(resid 10 and resname "GLY") or (resid 9 and resname "GLY")` [44]. The Basic Local Alignment Search Tool (BLAST) [45] searches for regions of similarity between amino acid sequences, i.e., primary structure, not 3D shapes and coordinates.

The PyMOL molecular visualization system [46, 47, 48] application allows both local and global structural alignment in subscriber and open source versions. PyMOL's GUI pair fitting wizard performs local structural alignments, and allows manual selection of atom pairs by zooming in on and clicking atom pairs in a graphic three dimensional model, or by command line syntax followed by a call to the `pair_fit` function. Selection of individual atom pairs through the GUI, analogous to the atom selection method of DS ViewerPro, has been termed "tedious" in SBGrid Consortium PyMOL documentation, which recommends scripting instead [49]. The `pair_fit` function performs the structural alignment and returns the RMSD of the aligned residues. Another PyMOL alignment function, `align` [50], combines sequence alignment and structural alignment algorithms that iteratively discard atoms from the fitting process and re-fit after each iteration.

Unlike PyMOL and VMD, MolLoc (Molecular Local surface comparison) [51] is a web based application that performs recognition and comparison of similar regions of Connolly molecular surfaces [52] including binding sites, cavities or arbitrary residue selections for two structures in PDB format, but does not allow direct comparison of structure. The MolLoc web server no longer appears to be available [53].

Some published comparisons of structural alignment techniques, e.g., [54, 55, 56], focus on global structure alignment tools. We are unaware of existing structural and sequence alignment web applications that offer the web based local protein structure alignment capabilities this work seeks to develop.

CHAPTER 3

Methodology

3.1 Minimum RMSD Computation

Protein structural alignment between a query structure and a reference structure is an optimization that endeavors to align spatial structures so as to minimize the RMSD distance between the aligned input structures. Results may be expressed as the rotation and translation of the three-dimensional atomic coordinate sets of each input structure, such that the molecules may be superpositioned with the minimal RMSD. For two sets of n coordinates a and b in $\mathbb{R}^3$, a transformation from a to b may be written as

$$Ma_i + t = b_i \tag{1}$$

where $M \in \mathbb{R}^{3\times3}$ is an orthogonal matrix with determinant 1, and so serves as a rotation matrix, and $t \in \mathbb{R}^3$ serves as a translation. Given $a_1 \ldots a_n$ and $b_1 \ldots b_n$, M is the solution to an orthogonal Procrustes problem, which may be computed using singular value decomposition [57]. Given two sets of 3D atom coordinates, the BioJava class `SVDSuperimposer` [58] uses singular value decomposition to compute a translation from the center of the second structure to the first, and a rotation from the coordinates of the second structure to the first. After receiving user input selecting amino acids to align, PLAT uses the α-carbon atom coordinates of each amino acid as input coordinates for `SVDSuperimposer`.

3.2 Application Design Pattern

The PLAT application design is based on the Model-View-Controller (MVC) design pattern, which separates components according to whether they represent the data, display the data, or provide data to view components [59]. PLAT uses the data modeled by the PDB, computes additional data with its server components, and presents data through its web components. PLAT uses executables that run in the browser environment to provide a user interface, and uses executables running in a server environment to access the PDB and perform structure alignments and other computations.

Separating the display of content in the browser from the acquisition and computation of data on the server increases the extensibility of PLAT by decoupling components so that they may be modified independently, contingent only on communicating through interfaces in ways independent of underlying implementation. Changes and additions can be made to the presentation components of PLAT without affecting computation components, and vice versa. For example, the local alignment algorithm can be changed without modification to the components that display data and collect user input. If an entirely separate alignment algorithm were added to PLAT, such as a global alignment, the changes required to the display components could be restricted to adding an interface element that enables the user to choose which algorithm to use.

3.2.1 Google Web Toolkit

The Google Web Toolkit (GWT) is used to provide the view components, i.e., the user interface, of PLAT. GWT provides tools to facilitate developing Internet applications with cross-browser compatibility [60, 61]. Web browsers generally retrieve and render information from the World Wide Web. While there are standards established by standards bodies such as the World Wide Web Consortium (W3C) [62], browsers may implement a standard in different ways, or may implement different standards. Code optimized for use in one browser may not be rendered the same way by another browser, necessitating browser-specific versions to achieve uniform rendering or functionality across browsers. GWT features a compiler that converts Java source code into multiple browser- and locale-specific versions of JavaScript, thereby adding a layer of abstraction between developer generated code and code supplied to browsers and increasing reusability of code [63]. In developing a GWT application, the user creates a single Java application that will be compiled into an equivalent JavaScript application with browser-specific versions [63]. The PLAT application user interface is an application in itself developed using GWT.

GWT is well-suited for use with a Model-View-Presenter (MVP) pattern, an extension of the MVC pattern where the Presenter provides control function for the view, and a separate AppController provides control function for logic not specific to a presenter [64]. The AppController component of the MVP pattern controls history events that represent various states within the application; PLAT in its current form does not have a requirement to maintain history.

GWT facilitates user interface performance because it improves browser page loading time by facilitating the loading of required resources only. In PLAT pages, resources other than those on the minimal initial screen are loaded when needed in response to user actions. GWT also places the burden of client rendering on the client browser using browser-specific javascript rather than on the server.

3.2.2 Server Components

The components of PLAT that access and cache data from the PDB and perform computations such as structural alignments are executed as Java Servlets on an Apache Tomcat server [65]. The servlets provide RESTful web services, receiving requests, performing computations, and sending responses via http, allowing the browser user interface (UI) components to interact with them. The BioJava library is used by PLAT servlets.

REST does not specify a particular format for data requests and responses [19], and both Extensible Markup Language (XML) and JavaScript Object Notation (JSON) are commonly used. Most PDB web services provide XML formatted responses. PLAT servlets communicate with PDB web services with components from the BioJava library, while PLAT servlets provide JSON encoded responses to the UI. JSON uses key-value pairs, and is generally though not necessarily more compact and more human readable than XML.

3.2.3 Jmol Interactive Viewer

PLAT uses the Jmol [66] applet to provide a 3D interactive view of aligned structures. Jmol can read many file types, including PDB. When an alignment is performed, PLAT constructs a pdb file to be rendered by Jmol.

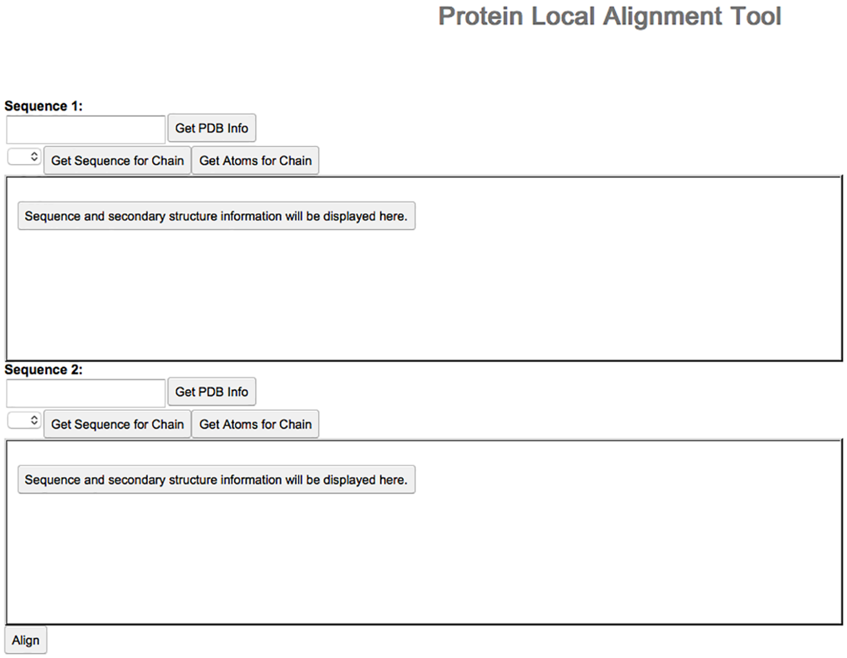
3.3 User Interface Design and Object Interaction

The user interface design provides users with the ability to:

1. Retrieve and display chains contained in a PDB file
2. Retrieve and display the amino acid sequence of a chain, together with its secondary structure annotations
3. Retrieve and display chains contained in an additional PDB file
4. Retrieve and display the amino acid sequence of a additional chain, together with its secondary structure annotations
5. Select amino acids from the displayed chains for local alignment
6. Display the translation and rotation that provides the minimum RMSD between the selected amino acids of each chain
7. Display the visual superposition of the two chains resulting from applying the calculated rotation and translation to the second chain

3.3.1 Main User Interface

The user interface (UI) consists largely of nested objects developed with GWT. Figure 6 shows the initial state of the UI when no data has been selected for analysis. Aside from the title at the top and the *Align* button at the bottom, the UI presents two instances of the sequence user interface panel `SequenceUIPanel`, which in turn contains several other panels, as shown in Figure 7, each developed with GWT, and each providing a user interface for viewing data or requesting an action.



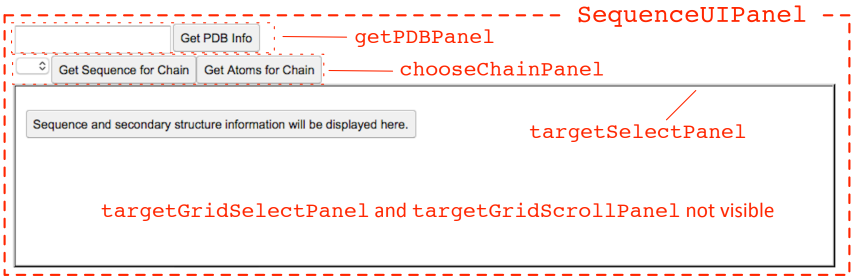
Sequence and secondary structure information will be displayed here.

Sequence and secondary structure information will be displayed here.

Figure 6: The initial UI state showing two `SequenceUIPanel` instances with no data presented

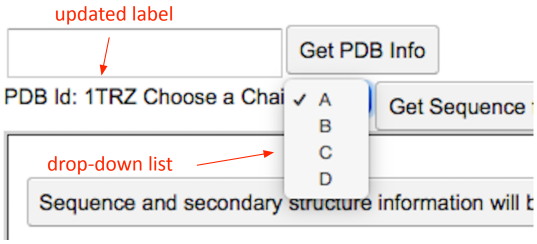
3.3.2 PDB Chains

From a UI perspective, to start a local alignment with PLAT, the user specifies a PDB entry id in the text field in the `getPDBPanel` shown in Figure 7 and clicks the *Get PDB Info* button, signaling PLAT to update and label the drop-down list containing the available chains, as shown in Figure 8.



Sequence and secondary structure information will be displayed here.

Figure 7: Major parts of the empty `SequenceUIPanel` labeled in red



Sequence and secondary structure information will b

Figure 8: UI updates after requesting a PDB id

In terms of objects, clicking the *Get PDB Info* button signals the UI to send an HTTP request for the amino acid chains of the PDB id from the web browser to the PLAT `PdbChainFetcher` servlet; the servlet then uses the BioJava library to request a PDB entry file from the PDB API, which it then incorporates into a BioJava `Structure` object. The servlet obtains the list of available chains from the `Structure` object, JSON-encodes the list, and returns the JSON encoded list as an HTTP response to the web browser, where the UI updates a label for the drop down menu to show the PDB entry id, parses the JSON encoded list of chains and uses the data to populate the drop down menu in the `chooseChainPanel`. The sequence of interactions is shown in Figure 9.

3.3.3 Sequences, Secondary Structure, and Amino Acid Selection

With a chain selected, the user can click the *Get Sequence for Chain* button, and PLAT will populate the `targetSelectPanel` with rows of primary and

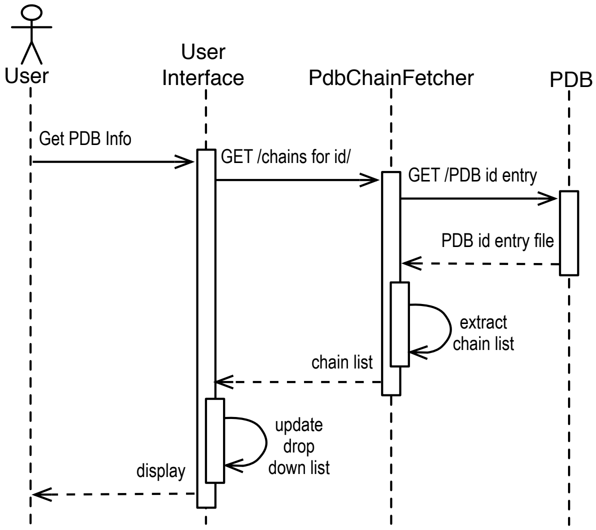
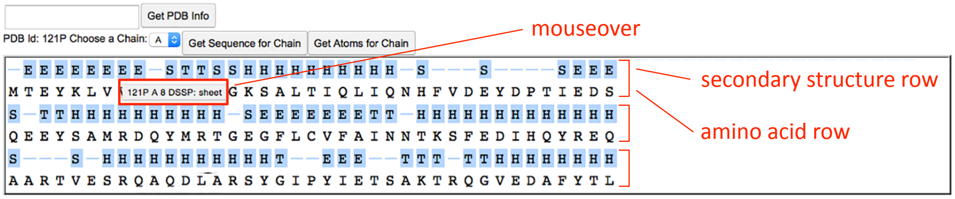


Figure 9: Interactions in response to a request for a list of chains

secondary structure information. Rows are displayed in sets of two, where the lower part of each set shows the single character amino acid abbreviation, and the upper blue-highlighted part shows the secondary structure code. Figure 10 shows an example with three sets of rows. Each amino acid or secondary structure code has a mouseover that displays relevant details of the item: pdb id, chain, sequence number, secondary structure annotation in the case of structure, and amino acid symbol in the case of amino acids. A secondary structure mouseover is displayed in Figure 10. The rows, the secondary structure labels, and the mouseovers provide hints to facilitate user navigation of the structure similar to those provided in the PDB *Sequence Chain View*, in Figure 5. Similar functionality is available when the user clicks the *Get Atoms for Chain* button.

The PLAT amino acid rows are comprised of objects that enable user selection of amino acids with visual feedback. The objects are based on the GWT `TextBox` class [67], with GWT dependent styling [68] that is displayed based on whether or not the box has been selected by the user; the UI displays selected amino acids



```
— E E E E E E E E — S T T S S H H H H H H H H H H — S — — — S — — — — S E E E
M T E Y K L V \ 121P A 8 DSSP: sheet G K S A L T I Q L I Q N H F V D E Y D P T I E D S
S — T T H H H H H H H H H H — S E E E E E E E T T — H H H H H H H H H H H H H
Q E E Y S A M R D Q Y M R T G E G F L C V F A I N N T K S F E D I H Q Y R E Q
S — — — S — H H H H H H H H H H H T — — E E E — — T T T — T T H H H H H H H H
A A R T V E S R Q A Q D L A R S Y G I P Y I E T S A K T R Q G V E D A F Y T L
```

Figure 10: Display of amino acids and secondary structure

with a yellow background. The `TextBox` objects are assigned uniform sizes, and each `TextBox` stores information about what amino acid it represents, as seen on the mouseover, in its object id. The UI does not need to compute cursor position or similar information about what was clicked; instead, the object sends a message containing the object id to the click handler. When the user clicks a `TextBox`, the click handler toggles selection status for the box with the correct id and any other affected boxes according to common practice for key-modified UI clicks; the click handler can address the objects by id without knowing where they are in the document or on the page. For example, a shift-click causes all the amino acids from the last one selected to the newly selected one to be added to the selection, while an unmodified click selects the current amino acid and deselects all others. To implement the acid and structure rows, PLAT departs from a pure MVC design pattern by employing a local model of data within the UI, largely in the form of arrays, to maintain the state of its display and selection functionality, only sending the state information to a servlet when the user initiates an action that requires the selection data.

SAX Parsing

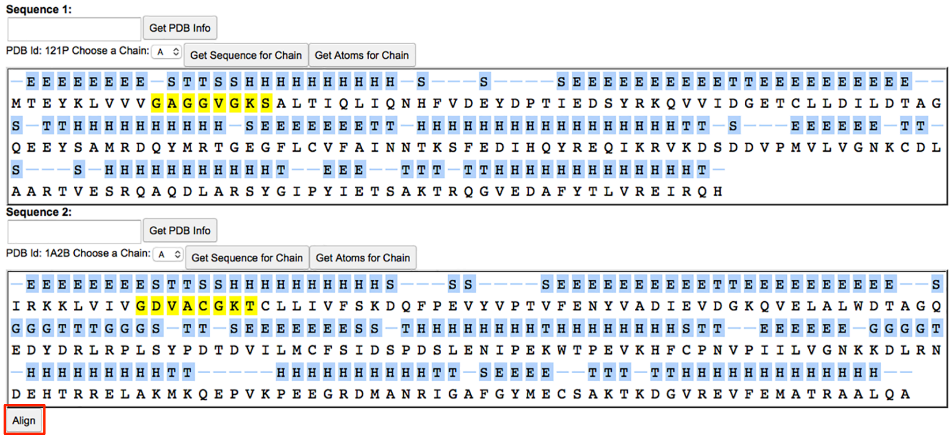
The sequence of interactions between the UI and other objects follows a similar model to that shown in Figure 9. For sequence information and atom coordinates, PLAT servlets use the BioJava library to obtain amino acid data from PDB web services, but, for secondary structure, PLAT servlets obtain data from the PDB DAS third party annotation web service `pdbchainfeatures` [20] and parse it using a Simple API for XML (SAX) [69] parser. SAX parsing uses an event-based model to parse an input stream as XML elements are started and completed, without parsing and maintaining a model of an entire xml document [70], in contrast to Document Object Model (DOM) [71] parsing, which creates a model of all XML elements that in some approaches must be traversed when accessing a single element. The `pdbchainfeatures` web service returns annotation types other than DSSP; a single amino acid may have several annotations. PLAT defines a `DSSP_START` event that allows its SAX parser to ignore non-DSSP annotations without attempting to parse them.

3.3.4 Local Alignment

Solution of the orthogonal Procrustes problem requires that each structure have the same number n coordinates, so the user must select the same number of amino acids from each chain. PLAT does not enforce a constraint that amino acids selected in each panel need be consecutive or adjacent. Adjacency is not necessary for the computation to succeed, though the results may be uninterpretable. The feature is left in place, though, to allow exploration. With appropriate amino acids selected, as in Figure 11, the user may click *Align*.

When the computation is complete and the result returned, the UI displays an $\mathbb{R}^{3\times3}$ rotation matrix and an $\mathbb{R}^3$ translation, which, when applied to the coordinates of the chain in the second `SequenceUIPanel`, will minimize the RMSD between the selected sequences of coordinates; the minimized RMSD after rotation and translation is displayed as well. The UI also displays a *Render alignment in JMOL* button. Figure 12 displays example output.

As in the case of obtaining sequence and structure data, UI behavior and



Sequence 1:

PDB Id: 121P Choose a Chain: A

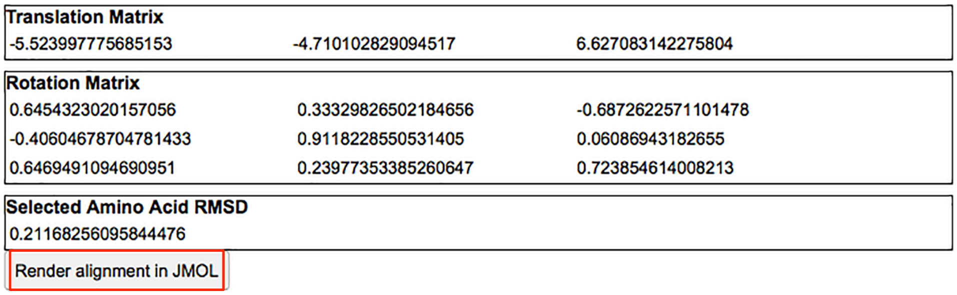
```
- E E E E E E E E - S T T S S H H H H H H H H H H - S - - - S - - - - S E E E E E E E E E E T T E E E E E E E E E E - -
M T E Y K L V V V G A G G V G K S A L T I Q L I Q N H F V D E Y D P T I E D S Y R K Q V V I D G E T C L L D I L D T A G
S - T T H H H H H H H H H H - S E E E E E E E T T - H H H H H H H H H H H H H H H H H H T T - S - - - E E E E E E - T T -
Q E E Y S A M R D Q Y M R T G E G F L C V F A I N N T K S F E D I H Q Y R E Q I K R V K D S D D V P M V L V G N K C D L
S - - - S - H H H H H H H H H H H T - - E E E - - T T T - T T H H H H H H H H H H H H H T -
A A R T V E S R Q A Q D L A R S Y G I P Y I E T S A K T R Q G V E D A F Y T L V R E I R Q H
```

Sequence 2:

PDB Id: 1A2B Choose a Chain: A

```
- E E E E E E E E S T T S S H H H H H H H H H H S - - - S S - - - - S E E E E E E E E E E T T E E E E E E E E E E - - S
I R K K L V I V G D V A C G K T C L L I V F S K D Q F P E V Y V P T V F E N Y V A D I E V D G K Q V E L A L W D T A G Q
G G G T T T G G G S - T T - S E E E E E E E S S - T H H H H H H H H T H H H H H H H H S T T - - E E E E E E - G G G G T
E D Y D R L R P L S Y P D T D V I L M C F S I D S P D S L E N I P E K W T P E V K H F C P N V P I I L V G N K K D L R N
- H H H H H H H H H T T - - - - - - H H H H H H H H H H T T - S E E E E - - T T T - T T H H H H H H H H H H H H H - -
D E H T R R E L A K M K Q E P V K P E E G R D M A N R I G A F G Y M E C S A K T K D G V R E V F E M A T R A A L Q A
```

Figure 11: Example of amino acid residue selections prepared for alignment



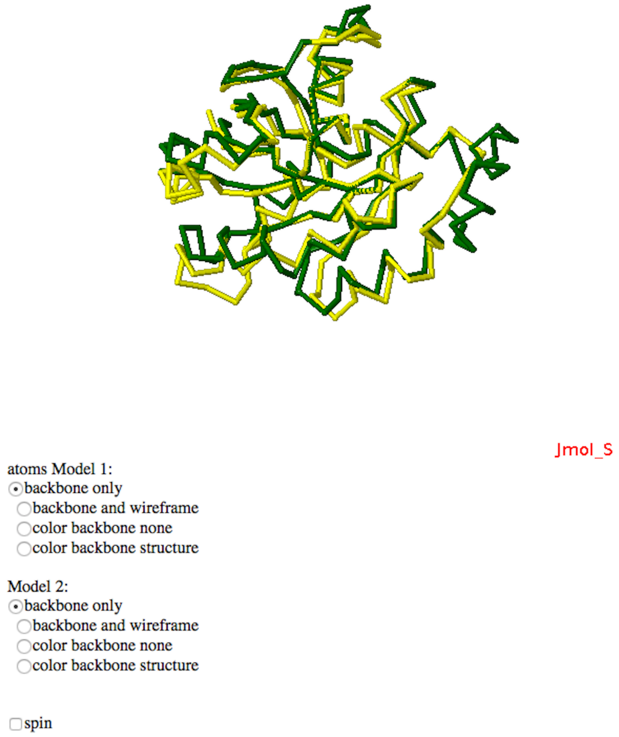
Translation Matrix		
-5.523997775685153	-4.710102829094517	6.627083142275804

Rotation Matrix		
0.6454323020157056	0.33329826502184656	-0.6872622571101478
-0.40604678704781433	0.9118228550531405	0.06086943182655
0.6469491094690951	0.23977353385260647	0.723854614008213

Selected Amino Acid RMSD
0.21168256095844476

Figure 12: Example alignment $\mathbb{R}^{3\times 3}$ rotation and $\mathbb{R}^3$ translation output

interaction with other objects is similar to that shown in Figure 9. When the computation is complete, a servlet applies the alignment to the second chain to compute a superposition and produces a pdb file containing the resulting coordinates to act as input for Jmol. When a user invokes Jmol, the UI opens a new web page that includes the Jmol applet, which in turn renders the pdb file. Figure 13 illustrates the Jmol rendering page.



atoms Model 1:
● backbone only
○ backbone and wireframe
○ color backbone none
○ color backbone structure

Model 2:
● backbone only
○ backbone and wireframe
○ color backbone none
○ color backbone structure

□ spin

Figure 13: Example Jmol page

CHAPTER 4

Results

4.1 Development Objectives

PLAT meets the objective of being web-based, and its UI operates within the constraints of presenting rows of repeated objects. The use of GWT to create Javascript code for different browsers reduces the complexity of maintaining the UI and the use of arrays of `TextBox` objects reduces selection update complexity to that of traversing an array. PLAT meets the objective of facilitating correct identification of selected items through the use of graphical selection of amino acids displayed with secondary structure and position attributes. Finally, PLAT meets the objective of obtaining data from the PDB through the use of PDB APIs, BioJava, and web services, thus providing opportunities to extend functionality through further use of APIs or the addition of servlets that do not require modification to other parts of the application. Source code for PLAT is posted on the Github PLAT site [72].

4.2 Observed Differences Between Local and Global Alignments

Among structural features of proteins that determine protein function, specific functional sites may be of particular importance [2]. To examine the effects of the difference between local and global structural alignment, we compare alignments of proteins from the Ras superfamily, a group of proteins that act as switches that activate or inactivate cellular functions [73] through GDP/GTP bindings, where, for example, a GTP-binding protein binds to guanosine triphosphate (GTP) to activate a protein, or binds to guanosine diphosphate (GDP) to inactivate the protein [74].

Ras superfamily members conserve five G domains, regions related to

GDP/GTP bindings [3]. The G1 domain contains a phosphate binding loop (p-loop) with sequence GXXXXGK[S/T], where X may be any amino acid and the last amino acid may be S (Serine) or T (Threonine). PDB entries 121P and 1A2B are members of the HRas and RhoA subfamilies of the Ras superfamily. A global alignment and superposition of 121P and 1A2B performed with FATCAT [41] is shown in Figure 14.

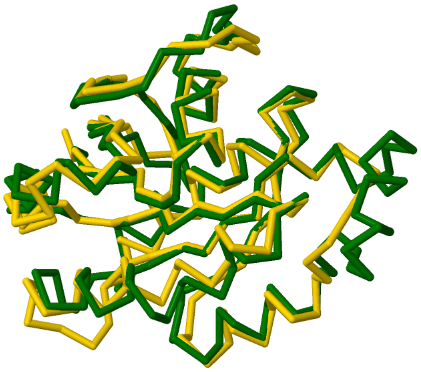


Figure 14: FATCAT global alignment and superposition of PDB structures 121P and 1A2B

The p-loop for 121P may be found as the sequence `GAGGVGKS`, staring at amino acid sequence position 10, and the p-loop for 1A2B as `GDVACGKT`, starting at amino acid position 9. Figure 11 shows the PLAT UI with the p-loop amino acids selected. The superposition of 121P and 1A2B resulting from a local alignment with PLAT is shown in Figure 15.

Using only the coordinates of the p-loop α-carbons, Figure 16 displays the effects of global versus local alignment on the superposition of the 121P and 1A2B p-loops using (a) FATCAT [41] global alignment and (b) PLAT local alignment. Numeric values indicate the distance in Å between the corresponding α-carbons, which are more closely aligned using local alignment.

As part of the protein-structure function analysis using self-organizing maps in [76], global alignments with FATCAT and p-loop based local alignments with

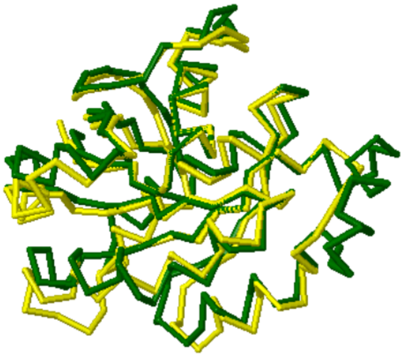


Figure 15: PLAT local alignment of p-loops and superposition of PDB structures 121P and 1A2B

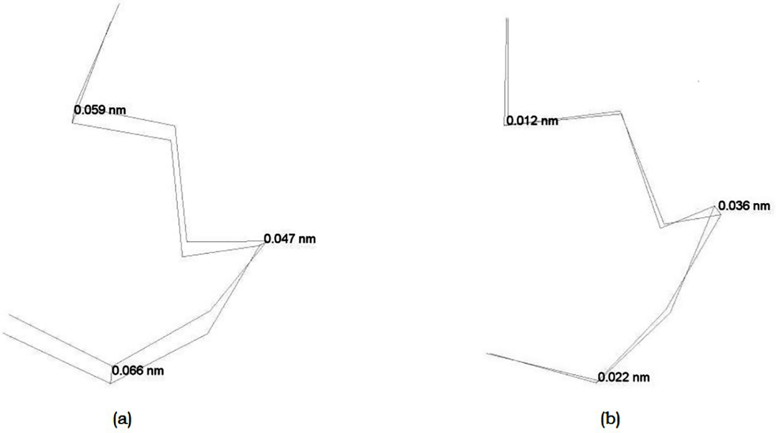


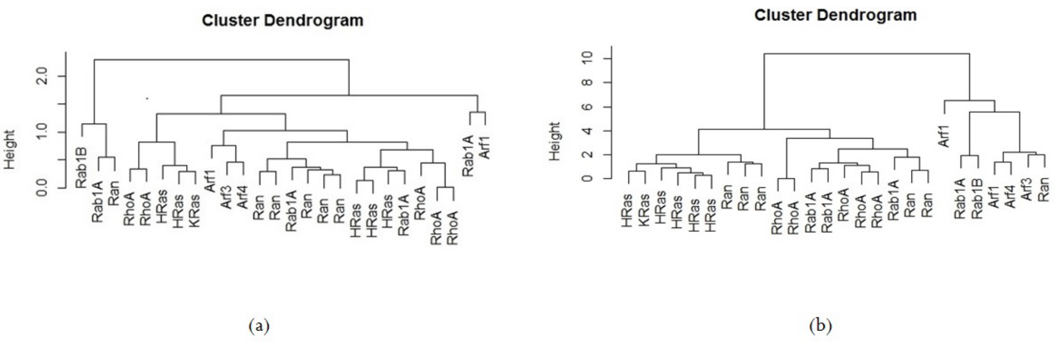
Figure 16: Comparison of p-loop alignments of PDB structures 121P and 1A2B with (a) FATCAT global alignment and (b) PLAT local alignment

Source: [75]

PLAT against 121P were performed using the PDB IDs listed in Table 5, which shows the hierarchical relationship of the Ras superfamily and the PDB IDs used for analysis. Cluster dendrogram models based on the alignment results were constructed using `hclust` in R [77, 78]. In these models, shown in Figure 17, global alignment clusters have significantly greater height than local alignment clusters, and thus significantly greater homogeneity, suggesting that small misalignments of the p-loops resulting from global alignment may be more predictive of protein functionality. This experiment contradicted our expectation that local alignments would be more predictive due to the elimination of alignment“noise” resulting from other parts of global structure.

Family	Subfamily	PDB ID
Ras	HRas	121P, 1QRA, 1CTQ, 1P2S, 1AGP
	KRas	4DSN
Rho	RhoA	1A2B, 1CC0, 1CXZ, 1DPF, 1FTN
Rab	Rab1A	2FOL, 2WWX, 3SFV, 3TKL
	Rab1B	3JZA
Arf	Arf1	1HUR
	Arf2	1U81
	Arf3	1RE0
	Arf4	1Z6X
Ran		1I2M, 1IBR, 1RRP, 3CH5, 3EA5, 3GJ3

Table 5: Hierarchy of the Ras superfamily and the list of proteins used for analysis.
(Source: [76])



(a)

(b)

Figure 17: Cluster dendrograms of results of (a) local alignment and (b) global alignment of the Ras Superfamily.
Source: [75]

The FATCAT global alignment between PDB entries 121P and 1A2B produced a 0.625 Å RMSD between the p-loop α-carbons, versus 0.212 Å generated by PLAT local alignment of the p-loops. A global alignment using the PyMOL `align` function produced a 0.466 Å RMSD alignment between the p-loop α-carbons. Further work will be required to compare the effects of lower RMSD between p-loops produced by global alignments using PyMOL `align` on clustering characteristics to those observed using FATCAT in protein-structure function analysis as in [76].

4.3 Comparison of PLAT and PyMOL Local Alignment Results

In the p-loop example, PLAT returns numeric results that compare to those of PyMOL `pair_fit`; PLAT and PyMOL represent values using different precision, resulting in slightly different output. To calculate rotation, translation, and RMSD, BioJava SVDSuperimposer uses Java `double` representations [58, 79] while PyMOL uses Python `float` representations [80]. Table 6 displays the rotated and translated coordinates of the 1A2B p-loop locally aligned with the 121P p-loop as reported by both PLAT and PyMOL. Both sets of coordinates return a rounded RMSD of 0.212 versus the 121P p-loop coordinates.

4.4 Comparison of Identical Structures

PLAT has been tested to align identical structures, i.e., to align an amino acid sequence to itself by using the same sequence and selecting the same residues in each sequence panel. As expected, PLAT returns a 3 × 3 identity matrix for rotation, and a zero matrix for translation aligning a structure with itself.

Residue	x	y	z
1A2B p-loop PLAT Post Alignment Results			
Gly	5.148 572 277 413 82	24.282 221 037 583 923	18.877 883 494 551 73
Asp	6.373 214 846 227 8	24.932 178 322 636 698	22.453 015 814 656 197
Val	6.698 731 261 147 962	28.276 479 158 452 688	24.223 444 006 698 454
Ala	9.441 034 413 867 95	30.612 635 482 886 37	22.914 784 751 433 533
Cys	10.622 850 331 599 512	28.114 552 474 887 972	20.297 407 548 631 84
Gly	10.751 304 138 382 444	30.889 173 463 144 38	17.672 040 909 274 585
Lys	7.392 683 412 632 035	30.411 634 761 766 074	15.958 102 576 298 248
Thr	6.720 609 318 728 46	34.143 125 298 641 85	15.607 320 898 455 392
1A2B p-loop PyMOL Post Alignment Results			
Gly	5.148 572 444 92	24.282 222 747 8	18.877 883 911 1
Asp	6.373 214 721 68	24.932 178 497 3	22.453 016 281 1
Val	6.698 731 422 42	28.276 481 628 4	24.223 445 892 3
Ala	9.441 034 317 02	30.612 636 566 2	22.914 785 385 1
Cys	10.622 850 418 1	28.114 553 451 5	20.297 407 150 3
Gly	10.751 303 672 8	30.889 173 507 7	17.672 040 939 3
Lys	7.392 683 029 17	30.411 634 445 2	15.958 102 226 3
Thr	6.720 608 711 24	34.143 127 441 4	15.607 321 739 2

Table 6: α-carbon coordinates returned by PLAT and PyMOL `pair_fit` for the p-loops of PDB structures 1A2B and 121P.

CHAPTER 5

Conclusions and Further Work

5.1 Conclusions

We have presented a tool that performs structural protein alignments using local structures, and demonstrated that its local structural alignments have the ability to align local structures, including functional centers such as p-loops, more closely than global structural alignments. The tool is web-based, uses a graphical user interface, and obtains data from the PDB.

5.2 Further Work

Further work on PLAT will focus on the user interface and alignment functionality. The user interface input methods should be tested for usability. Error handling and other communication with the user may also be made more robust. For example, a requirement of the SVD technique used by PLAT and PyMOL is that each structure needs to have the same number of coordinate vectors. When the number of coordinates is unequal, PLAT fails gracefully but without warning to the user. Similarly, although PLAT returns the correct result, it does not warn the user when identical structures are being compared, so additional informative communications from PLAT to the user will improve usability. User testing by researchers with use cases requiring alignments between specified residues or atoms to the exclusion of others, such as in [31] and [32], would aid the improvement of usability as well. PLAT currently uses data from the PDB, and configuring PLAT to accept user-supplied structures will enable it to work with modeled structure use cases, as in [33] and [34].

Functionally, although PLAT has demonstrated the ability to return valid results, additional validity testing is required, particularly for edge cases. PLAT uses the Java applet version of Jmol, which currently requires users to modify security-related browser and system settings; an upgrade to an HTML5 version would simplify use, as would a change to another visualization system such as PyMOL.

PLAT can be extended to support multiple alignments against a single reference molecule; the current architecture will facilitate the addition of what will largely be more instances of existing components. PLAT results can be made persistent and downloadable by storing them in a user accessible file, requiring the addition of UI and servlet components.

PLAT rendering performance for larger molecules could be evaluated, as well as performance limits for servlets and limits for PDB data access. To improve multi-user access, per-user storage and retrieval of results could be added.

The PLAT superposition methodology may be applied to any atoms in a structure in the same way as it is applied to amino acid residue α-carbons. The PLAT atom user interface accessible via the `SequenceUIPanel` *Get Atoms for Chain* button uses single character symbols for atoms, so additional development of that interface will improve usability.

PLAT currently uses data from the PDB, and configuring PLAT to accept user-supplied structures will enable it to work with modeled structure use cases, as in [33, 34].

LIST OF REFERENCES

[1] R. J. Najmanovich, J. W. Torrance, and J. M. Thornton, “Prediction of protein function from structure: insights from methods for the detection of local structural similarities,” *BioTechniques*, vol. 38, no. 6, pp. 847, 849, 851, Jun 2005.

[2] H. A. Maghawry, M. G. Mostafa, M. H. Abdul-Aziz, and T. F. Gharib, “Structural protein function prediction-a comprehensive review.” *International Journal of Modern Education & Computer Science*, vol. 7, no. 10, 2015.

[3] K. Wennerberg, K. L. Rossman, and C. J. Der, “The ras superfamily at a glance,” *Journal of cell science*, vol. 118, no. 5, pp. 843–846, 2005.

[4] L. Hamel, G. Sun, and J. Zhang, “Toward protein structure analysis with self-organizing maps,” in *Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB’05. Proceedings of the 2005 IEEE Symposium on.* Embassy Suites Hotel La Jolla, La Jolla, CA,: IEEE, 2005, pp. 1–8.

[5] M. Hashimoto, E. Rockenstein, L. Crews, and E. Masliah, “Role of protein aggregation in mitochondrial dysfunction and neurodegeneration in alzheimer’s and parkinson’s diseases,” *NeuroMolecular Medicine*, vol. 4, no. 1, pp. 21–35, 2003. [Online]. Available: http://dx.doi.org/10.1385/NMM:4:1-2:21

[6] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, “The protein data bank,” *Nucleic Acids Research*, vol. 28, no. 1, p. 235, 2000. [Online]. Available: +http://dx.doi.org/10.1093/nar/28.1.235

[7] RCSB. “Rcsb pdb.” March 2017. [Online]. Available: http://rcsb.org

[8] RCSB. “About the pdb archive and the rcsb pdb.” March 2017. [Online]. Available: http://www.rcsb.org/pdb/static.do?p=general_information/about_pdb/index.html

[9] J. Westbrook, Z. Feng, S. Jain, T. N. Bhat, N. Thanki, V. Ravichandran, G. L. Gilliland, W. F. Bluhm, H. Weissig, D. S. Greer, P. E. Bourne, and H. M. Berman, “The protein data bank: unifying the archive,” *Nucleic Acids Research*, vol. 30, no. 1, pp. 245–248, 01 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC99110/

[10] V. N. Maiorov and G. M. Crippen, "Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins," *Journal of Molecular Biology*, vol. 235, no. 2, pp. 625 – 634, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022283684710175

[11] RCSB. "Methods for determining atomic structures." March 2017. [Online]. Available: https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/methods-for-determining-structure

[12] M. Nic, J. Jirat, and B. Kosata. International Union of Pure and Applied Chemistry (IUPAC). "Iupac compendium of chemical terminology - the gold book." March 2017. [Online]. Available: http://goldbook.iupac.org/

[13] Worldwide Protein Data Bank. "Primary structure section." March 2017. [Online]. Available: http://www.wwpdb.org/documentation/file-format-content/format33/sect3.html

[14] Worldwide Protein Data Bank. "Coordinate section." March 2017. [Online]. Available: http://www.wwpdb.org/documentation/file-format-content/format33/sect9.html

[15] Worldwide Protein Data Bank. "Secondary structure section." March 2017. [Online]. Available: http://www.wwpdb.org/documentation/file-format-content/format33/sect5.html

[16] RCSB. "Glossary of technical terms." March 2017. [Online]. Available: http://www.rcsb.org/pdb/static.do?p=help/glossary.html

[17] M. Amundsen, *RESTful Web Clients.* O'Reilly Media, Inc., 2017.

[18] M. Pagni, J. Hau, and H. Stockinger, "A multi-protocol bioinformatics web service: Use soap, take a rest or go with html," in *2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, May 2008, pp. 728–734.

[19] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, Irvine, CA, 7 2000.

[20] RCSB. "The rcsb pdb restful web service interface." March 2017. [Online]. Available: http://www.rcsb.org/pdb/software/rest.do

[21] RCSB. "Retirement of soap services." March 2017. [Online]. Available: http://www.rcsb.org/pdb/news.do?year=2013&article=5764422199cccf72e74ca356

[22] R. D. Dowell, R. M. Jokerst, A. Day, S. R. Eddy, and L. Stein, "The distributed annotation system," *BMC Bioinformatics*, vol. 2, 2001.

[23] BioDAS.org. “Biodas: Main page.” March 2017. [Online]. Available: http://www.biodas.org

[24] J. Martin, G. Letellier, A. Marin, J.-F. Taly, A. G. de Brevern, and J.-F. Gibrat, “Protein secondary structure assignment revisited: a detailed analysis of different assignment methods,” *BMC Structural Biology*, vol. 5, 2005.

[25] RCSB. “Sequence display for the entities in pdb 121p.” March 2017. [Online]. Available: http://www.rcsb.org/pdb/explore/remediatedSequence.do?structureId=121p

[26] A. Prli, A. Yates, S. E. Bliven, P. W. Rose, J. Jacobsen, P. V. Troshin, M. Chapman, J. Gao, C. H. Koh, S. Foisy, R. Holland, G. Rima, M. L. Heuer, H. BrandsttterMller, P. E. Bourne, and S. Willis, “Biojava: an open-source framework for bioinformatics in 2012,” *Bioinformatics*, vol. 28, no. 20, p. 2693, 2012. [Online]. Available: +http://dx.doi.org/10.1093/bioinformatics/bts494

[27] The BioJava Project. “Biojava.” March 2017. [Online]. Available: http://biojava.org

[28] The BioJava Project. “Interface structure.” March 2017. [Online]. Available: http://biojava.org/docs/api4.2.1/org/biojava/nbio/structure/Structure.html

[29] The Bioperl Project. “Bioperl.” July 2017. [Online]. Available: http://bioperl.org

[30] Biopython. “Biopython.” July 2017. [Online]. Available: http://biopython.org/wiki/Biopython

[31] M. R. Wasserman, A. Pulk, Z. Zhou, R. B. Altman, J. C. Zinder, K. D. Green, S. Garneau-Tsodikova, J. H. Cate, and S. C. Blanchard, “Chemically related 4,5-linked aminoglycoside antibiotics drive subunit rotation in opposite directions,” *Nat Commun*, vol. 6, p. 7896, Jul 2015. [Online]. Available: https://dx.doi.org/10.1038/ncomms8896

[32] R. K. McGinty, R. C. Henrici, and S. Tan, “Crystal structure of the prc1 ubiquitylation module bound to the nucleosome,” *Nature*, vol. 514, no. 7524, pp. 591–596, Oct 2014, article. [Online]. Available: http://dx.doi.org/10.1038/nature13890

[33] Z. Zhao, D. Worthylake, L. LeCour, G. A. Maresh, and S. H. Pincus, “Crystal structure and computational modeling of the fab fragment from a protective anti-ricin monoclonal antibody,” *PLoS One*, vol. 7, no. 12, p. e52613, Dec 2012, pONE-D-12-19605[PII]. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3526572/

[34] W. Kasprzak, E. Bindewald, T.-J. Kim, L. Jaeger, and B. A. Shapiro, "Use of rna structure flexibility data in nanostructure modeling," *Methods*, vol. 54, no. 2, pp. 239–250, 2011. [Online]. Available: https://doi.org/10.1016/j.ymeth.2010.12.010

[35] NCBI. "Vast vector alignment search tool." March 2017. [Online]. Available: http://www.ncbi.nlm.nih.gov/Structure/VAST/vast.shtml

[36] L. Holm and P. Rosenström. Institute of Biotechnology, University of Helsinki. "Dali server." March 2017. [Online]. Available: http://ekhidna.biocenter.helsinki.fi/dali_server/

[37] C. Orengo and W. Taylor, "Ssap: Sequential structure alignment program for protein structure comparison." *Methods in Enzymology*, vol. 266, pp. 617–635, 1996.

[38] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–637, Dec 1983.

[39] R. P. Joosten, T. A. H. te Beek, E. Krieger, M. L. Hekkelman, R. W. W. Hooft, R. Schneider, C. Sander, and G. Vriend, "A series of pdb related databases for everyday needs," *Nucleic Acids Res*, vol. 39, no. Database issue, pp. D411–9, Jan 2011.

[40] Rosetta@home, Howard Hughes Medical Institute, University of Washington. "Rosetta@home." March 2017. [Online]. Available: http://boinc.bakerlab.org/rosetta/

[41] Y. Ye and A. Godzik, "Flexible structure alignment by chaining aligned fragment pairs allowing twists," *Bioinformatics*, vol. 19, no. suppl 2, pp. ii246–ii255, 2003.

[42] L. P. Chew, "Exact computation of protein structure similarity," in *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*. ACM Press, 2006.

[43] W. Humphrey, A. Dalke, and K. Schulten, "VMD – Visual Molecular Dynamics," *Journal of Molecular Graphics*, vol. 14, pp. 33–38, 1996.

[44] K. Callenberg. Calgorithms. "How to easily align two molecular structures in vmd." July 2017. [Online]. Available: http://www.calgorithms.com/blog/2014/06/12/how-to-easily-align-two-molecular-structures-in-vmd/

[45] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, no. 3, pp. 403–410, Oct 1990.

[46] Schrödinger, LLC, “The PyMOL molecular graphics system, version 1.8,” November 2015.

[47] Schrödinger, LLC. “Pymol — www.pymol.org.” July 2017. [Online]. Available: http://pymol.org

[48] pymol.org. “Pymol molecular graphics system.” July 2017. [Online]. Available: https://sourceforge.net/projects/pymol/

[49] SBGrid Consortium. “Pair fit - pymolwiki.” July 2017. [Online]. Available: https://pymolwiki.org/index.php/Pair_fit

[50] Schrödinger, LLC. “command:align [pymol documentation].” June 2017. [Online]. Available: https://pymol.org/dokuwiki/doku.php?id=command:align

[51] S. Angaran, M. E. Bock, C. Garutti, and C. Guerra, “Molloc: a web tool for the local structural alignment of molecular surfaces,” *Nucleic Acids Res*, vol. 37, Web Server Issue, pp. W565–W570, Jul 2009, 19465382[pmid]. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2703929/

[52] M. L. Connolly, “Analytical molecular surface calculation,” *Journal of Applied Crystallography*, vol. 16, pp. 548–558, 1983.

[53] N. Shivashankar, S. Patil, A. Bhosle, N. Chandra, and V. Natarajan, “Ms3align: an efficient molecular surface aligner using the topology of surface curvature,” *BMC Bioinformatics*, vol. 17, no. 1, p. 26, Jan 2016. [Online]. Available: http://dx.doi.org/10.1186/s12859-015-0874-8

[54] C. Berbalk, C. S. Schwaiger, and P. Lackner, “Accuracy analysis of multiple structure alignments,” *Protein Science*, vol. 18, no. 10, pp. 2027–2035, 2009.

[55] G. Mayr, F. S. Domingues, and P. Lackner, “Comparative analysis of protein structure alignments,” *BMC Structural Biology*, vol. 7, no. 1, p. 50, 2007. [Online]. Available: http://dx.doi.org/10.1186/1472-6807-7-50

[56] D. Petrova, “Protein structure comparison methods,” *Information Technologies and Control*, vol. VII, no. 2, 2009.

[57] T. Viklands, “Algorithms for the weighted orthogonal procrustes problem and other least squares problems,” Ph.D. dissertation, Umeå University, Umeå, Sweden, 2006.

[58] The BioJava Project. “Class svdsuperimposer.” March 2017. [Online]. Available: http://biojava.org/docs/api4.2.0/org/biojava/nbio/structure/SVDSuperimposer.html

[59] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, 1995.

[60] GWT Project. "Google web toolkit." March 2017. [Online]. Available: http://www.gwtproject.org

[61] GWT Project. "Browsers and servers." March 2017. [Online]. Available: http://www.gwtproject.org/doc/latest/FAQ_GettingStarted.html#Browsers_and_Servers

[62] World Wide Web Consortium. "W3c." March 2017. [Online]. Available: https://www.w3.org

[63] GWT Project. "Compiling and debugging." March 2017. [Online]. Available: http://www.gwtproject.org/doc/latest/DevGuideCompilingAndDebugging.html

[64] GWT Project. "Building mvp apps: Mvp part i." March 2017. [Online]. Available: http://www.gwtproject.org/articles/mvp-architecture.html

[65] Apache Software Foundation. "Apache tomcat." March 2017. [Online]. Available: https://tomcat.apache.org

[66] Jmol. "Jmol: an open-source java viewer for chemical structures in 3d." March 2017. [Online]. Available: http://www.jmol.org

[67] GWT Project. "Class textbox." March 2017. [Online]. Available: http://www.gwtproject.org/javadoc/latest/com/google/gwt/user/client/ui/TextBox.html

[68] GWT Project. "Class uiobject." March 2017. [Online]. Available: http://www.gwtproject.org/javadoc/latest/com/google/gwt/user/client/ui/UIObject.html#addStyleName-java.lang.String-

[69] SAX Project. "Sax." March 2017. [Online]. Available: http://www.saxproject.org

[70] D. Brownell, *SAX2: Processing XML Efficiently with Java.* O'Reilly Media, Inc., 2002.

[71] World Wide Web Consortium. "Document object model (dom)." March 2017. [Online]. Available: https://www.w3.org/DOM/

[72] PLAT. "plat." July 2017. [Online]. Available: https://github.com/sjaegle/plat

[73] M.] Kennedy, H. Beale, H. Carlisle, and L. Washburn, "Achieving signalling specificity: the ras superfamily." *Nature Reviews Neuroscience*, vol. 6, pp. 423 – 434, 2005.

[74] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*. Garland Science, 2002.

[75] S. Lim, “Functional site based protein structure analysis with self-organizing maps,” Master’s thesis, University of Rhode Island, Kingston, RI, 2015.

[76] S. Lim, S. Jaegle, and L. Hamel, “Protein structure-function analysis with self-organizing maps,” in *Proceedings of the 17th International Conference on Bioinformatics & Computational Biology (BIOCOMP’16)*. CSREA Press, 2016, pp. 10–16.

[77] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015. [Online]. Available: http://www.r-project.org/

[78] R Core Team. R Foundation for Statistical Computing. “hclust hierarchical clustering.” March 2017. [Online]. Available: https://www.rdocumentation.org/packages/stats/versions/3.3.2/topics/hclust

[79] The BioJava Project. “Svdsuperimposer.java.” March 2017. [Online]. Available: https://github.com/biojava/biojava/blob/b56672e7ec6d7aed3e43d3224de62bc21b41a071/biojava-structure/src/main/java/org/biojava/nbio/structure/SVDSuperimposer.java

[80] Schrödinger, LLC. “fitting.py.” July 2017. [Online]. Available: https://github.com/speleo3/pymol/blob/91ddc53199f40f12d186dee2a3745cd777a57877/modules/pymol/fitting.py

APPENDIX

Appendix A

Amino Acid	Single Character Code	3 Character Code
Alanine	A	Ala
Cysteine	C	Cys
Aspartic acid	D	Asp
Glutamic acid	E	Glu
Phenylalanine	F	Phe
Glycine	G	Gly
Histidine	H	His
Isoleucine	I	Ile
Lysine	K	Lys
Leucine	L	Leu
Methionine	M	Met
Asparagine	N	Asn
Proline	P	Pro
Glutamine	Q	Gln
Arginine	R	Arg
Serine	S	Ser
Threonine	T	Thr
Valine	V	Val
Tryptophan	W	Trp
Tyrosine	Y	Tyr

Table A.1: 20 Common Amino Acids and their Codes in the Protein Data Bank

BIBLIOGRAPHY

] Kennedy, M., Beale, H., Carlisle, H., and Washburn, L., "Achieving signalling specificity: the ras superfamily." *Nature Reviews Neuroscience*, vol. 6, pp. 423 – 434, 2005.

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P., *Molecular Biology of the Cell.* Garland Science, 2002.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J., "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, no. 3, pp. 403–410, Oct 1990.

Amundsen, M., *RESTful Web Clients.* O'Reilly Media, Inc., 2017.

Angaran, S., Bock, M. E., Garutti, C., and Guerra, C., "Molloc: a web tool for the local structural alignment of molecular surfaces," *Nucleic Acids Res*, vol. 37, Web Server Issue, pp. W565–W570, Jul 2009, 19465382[pmid]. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2703929/

Apache Software Foundation. "Apache tomcat." March 2017. [Online]. Available: https://tomcat.apache.org

Berbalk, C., Schwaiger, C. S., and Lackner, P., "Accuracy analysis of multiple structure alignments," *Protein Science*, vol. 18, no. 10, pp. 2027–2035, 2009.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E., "The protein data bank," *Nucleic Acids Research*, vol. 28, no. 1, p. 235, 2000. [Online]. Available: +http://dx.doi.org/10.1093/nar/28.1.235

BioDAS.org. "Biodas: Main page." March 2017. [Online]. Available: http://www.biodas.org

The BioJava Project. "Biojava." March 2017. [Online]. Available: http://biojava.org

The BioJava Project. "Interface structure." March 2017. [Online]. Available: http://biojava.org/docs/api4.2.1/org/biojava/nbio/structure/Structure.html

The BioJava Project. "Class svdsuperimposer." March 2017. [Online]. Available: http://biojava.org/docs/api4.2.0/org/biojava/nbio/structure/SVDSuperimposer.html

The BioJava Project. “Svdsuperimposer.java.” March 2017. [Online]. Available: https://github.com/biojava/biojava/blob/b56672e7ec6d7aed3e43d3224de62bc21b41a071/biojava-structure/src/main/java/org/biojava/nbio/structure/SVDSuperimposer.java

The Bioperl Project. “Bioperl.” July 2017. [Online]. Available: http://bioperl.org

Biopython. “Biopython.” July 2017. [Online]. Available: http://biopython.org/wiki/Biopython

Brownell, D., *SAX2: Processing XML Efficiently with Java.* O’Reilly Media, Inc., 2002.

Callenberg, K. Calgorithms. “How to easily align two molecular structures in vmd.” July 2017. [Online]. Available: http://www.calgorithms.com/blog/2014/06/12/how-to-easily-align-two-molecular-structures-in-vmd/

Chew, L. P., “Exact computation of protein structure similarity,” in *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry.* ACM Press, 2006.

Connolly, M. L., “Analytical molecular surface calculation,” *Journal of Applied Crystallography*, vol. 16, pp. 548–558, 1983.

Dowell, R. D., Jokerst, R. M., Day, A., Eddy, S. R., and Stein, L., “The distributed annotation system,” *BMC Bioinformatics*, vol. 2, 2001.

Fielding, R. T., “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, University of California, Irvine, Irvine, CA, 7 2000.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, 1995.

GWT Project. “Google web toolkit.” March 2017. [Online]. Available: http://www.gwtproject.org

GWT Project. “Browsers and servers.” March 2017. [Online]. Available: http://www.gwtproject.org/doc/latest/FAQ_GettingStarted.html#Browsers_and_Servers

GWT Project. “Compiling and debugging.” March 2017. [Online]. Available: http://www.gwtproject.org/doc/latest/DevGuideCompilingAndDebugging.html

GWT Project. “Class uiobject.” March 2017. [Online]. Available: http://www.gwtproject.org/javadoc/latest/com/google/gwt/user/client/ui/UIObject.html#addStyleName-java.lang.String-

GWT Project. “Building mvp apps: Mvp part i.” March 2017. [Online]. Available: http://www.gwtproject.org/articles/mvp-architecture.html

GWT Project. “Class textbox.” March 2017. [Online]. Available: http://www.gwtproject.org/javadoc/latest/com/google/gwt/user/client/ui/TextBox.html

Hamel, L., Sun, G., and Zhang, J., “Toward protein structure analysis with self-organizing maps,” in *Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB’05. Proceedings of the 2005 IEEE Symposium on.* Embassy Suites Hotel La Jolla, La Jolla, CA,: IEEE, 2005, pp. 1–8.

Hashimoto, M., Rockenstein, E., Crews, L., and Masliah, E., “Role of protein aggregation in mitochondrial dysfunction and neurodegeneration in alzheimer’s and parkinson’s diseases,” *NeuroMolecular Medicine*, vol. 4, no. 1, pp. 21–35, 2003. [Online]. Available: http://dx.doi.org/10.1385/NMM:4:1-2:21

Holm, L. and Rosenström, P. Institute of Biotechnology, University of Helsinki. “Dali server.” March 2017. [Online]. Available: http://ekhidna.biocenter.helsinki.fi/dali_server/

Humphrey, W., Dalke, A., and Schulten, K., “VMD – Visual Molecular Dynamics,” *Journal of Molecular Graphics*, vol. 14, pp. 33–38, 1996.

Jmol. “Jmol: an open-source java viewer for chemical structures in 3d.” March 2017. [Online]. Available: http://www.jmol.org

Joosten, R. P., te Beek, T. A. H., Krieger, E., Hekkelman, M. L., Hooft, R. W. W., Schneider, R., Sander, C., and Vriend, G., “A series of pdb related databases for everyday needs,” *Nucleic Acids Res*, vol. 39, no. Database issue, pp. D411–9, Jan 2011.

Kabsch, W. and Sander, C., “Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features,” *Biopolymers*, vol. 22, no. 12, pp. 2577–637, Dec 1983.

Kasprzak, W., Bindewald, E., Kim, T.-J., Jaeger, L., and Shapiro, B. A., “Use of rna structure flexibility data in nanostructure modeling,” *Methods*, vol. 54, no. 2, pp. 239–250, 2011. [Online]. Available: https://doi.org/10.1016/j.ymeth.2010.12.010

Lim, S., “Functional site based protein structure analysis with self-organizing maps,” Master’s thesis, University of Rhode Island, Kingston, RI, 2015.

Lim, S., Jaegle, S., and Hamel, L., “Protein structure-function analysis with self-organizing maps,” in *Proceedings of the 17th International Conference on*

Bioinformatics & Computational Biology (BIOCOMP'16). CSREA Press, 2016, pp. 10–16.

Maghawry, H. A., Mostafa, M. G., Abdul-Aziz, M. H., and Gharib, T. F., "Structural protein function prediction-a comprehensive review." *International Journal of Modern Education & Computer Science*, vol. 7, no. 10, 2015.

Maiorov, V. N. and Crippen, G. M., "Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins," *Journal of Molecular Biology*, vol. 235, no. 2, pp. 625 – 634, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022283684710175

Martin, J., Letellier, G., Marin, A., Taly, J.-F., de Brevern, A. G., and Gibrat, J.-F., "Protein secondary structure assignment revisited: a detailed analysis of different assignment methods," *BMC Structural Biology*, vol. 5, 2005.

Mayr, G., Domingues, F. S., and Lackner, P., "Comparative analysis of protein structure alignments," *BMC Structural Biology*, vol. 7, no. 1, p. 50, 2007. [Online]. Available: http://dx.doi.org/10.1186/1472-6807-7-50

McGinty, R. K., Henrici, R. C., and Tan, S., "Crystal structure of the prc1 ubiquitylation module bound to the nucleosome," *Nature*, vol. 514, no. 7524, pp. 591–596, Oct 2014, article. [Online]. Available: http://dx.doi.org/10.1038/nature13890

Najmanovich, R. J., Torrance, J. W., and Thornton, J. M., "Prediction of protein function from structure: insights from methods for the detection of local structural similarities," *BioTechniques*, vol. 38, no. 6, pp. 847, 849, 851, Jun 2005.

Nic, M., Jirat, J., and Kosata, B. International Union of Pure and Applied Chemistry (IUPAC). "Iupac compendium of chemical terminology - the gold book." March 2017. [Online]. Available: http://goldbook.iupac.org/

Orengo, C. and Taylor, W., "Ssap: Sequential structure alignment program for protein structure comparison." *Methods in Enzymology*, vol. 266, pp. 617–635, 1996.

Pagni, M., Hau, J., and Stockinger, H., "A multi-protocol bioinformatics web service: Use soap, take a rest or go with html," in *2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, May 2008, pp. 728–734.

Petrova, D., "Protein structure comparison methods," *Information Technologies and Control*, vol. VII, no. 2, 2009.

PLAT. "plat." July 2017. [Online]. Available: https://github.com/sjaegle/plat

Prli, A., Yates, A., Bliven, S. E., Rose, P. W., Jacobsen, J., Troshin, P. V., Chapman, M., Gao, J., Koh, C. H., Foisy, S., Holland, R., Rima, G., Heuer, M. L., BrandsttterMller, H., Bourne, P. E., and Willis, S., "Biojava: an open-source framework for bioinformatics in 2012," *Bioinformatics*, vol. 28, no. 20, p. 2693, 2012. [Online]. Available: +http://dx.doi.org/10.1093/bioinformatics/bts494

Schrödinger, LLC. "command:align [pymol documentation]." June 2017. [Online]. Available: https://pymol.org/dokuwiki/doku.php?id=command:align

Schrödinger, LLC. "Pymol — www.pymol.org." July 2017. [Online]. Available: http://pymol.org

pymol.org. "Pymol molecular graphics system." July 2017. [Online]. Available: https://sourceforge.net/projects/pymol/

SBGrid Consortium. "Pair fit - pymolwiki." July 2017. [Online]. Available: https://pymolwiki.org/index.php/Pair_fit

Schrödinger, LLC. "fitting.py." July 2017. [Online]. Available: https://github.com/speleo3/pymol/blob/91ddc53199f40f12d186dee2a3745cd777a57877/modules/pymol/fitting.py

R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015. [Online]. Available: http://www.r-project.org/

R Core Team. R Foundation for Statistical Computing. "hclust hierarchical clustering." March 2017. [Online]. Available: https://www.rdocumentation.org/packages/stats/versions/3.3.2/topics/hclust

RCSB. "Rcsb pdb." March 2017. [Online]. Available: http://rcsb.org

RCSB. "Sequence display for the entities in pdb 121p." March 2017. [Online]. Available: http://www.rcsb.org/pdb/explore/remediatedSequence.do?structureId=121p

RCSB. "Glossary of technical terms." March 2017. [Online]. Available: http://www.rcsb.org/pdb/static.do?p=help/glossary.html

RCSB. "Methods for determining atomic structures." March 2017. [Online]. Available: https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/methods-for-determining-structure

RCSB. "The rcsb pdb restful web service interface." March 2017. [Online]. Available: http://www.rcsb.org/pdb/software/rest.do

RCSB. "Retirement of soap services." March 2017. [Online]. Available: http://www.rcsb.org/pdb/news.do?year=2013&article=5764422199cccf72e74ca356

RCSB. "About the pdb archive and the rcsb pdb." March 2017. [Online]. Available: http://www.rcsb.org/pdb/static.do?p=general_information/about_pdb/index.html

Rosetta@home, Howard Hughes Medical Institute, University of Washington. "Rosetta@home." March 2017. [Online]. Available: http://boinc.bakerlab.org/rosetta/

SAX Project. "Sax." March 2017. [Online]. Available: http://www.saxproject.org

Schrödinger, LLC, "The PyMOL molecular graphics system, version 1.8," November 2015.

Shivashankar, N., Patil, S., Bhosle, A., Chandra, N., and Natarajan, V., "Ms3align: an efficient molecular surface aligner using the topology of surface curvature," *BMC Bioinformatics*, vol. 17, no. 1, p. 26, Jan 2016. [Online]. Available: http://dx.doi.org/10.1186/s12859-015-0874-8

NCBI. "Vast vector alignment search tool." March 2017. [Online]. Available: http://www.ncbi.nlm.nih.gov/Structure/VAST/vast.shtml

Viklands, T., "Algorithms for the weighted orthogonal procrustes problem and other least squares problems," Ph.D. dissertation, Umeå University, Umeå, Sweden, 2006.

World Wide Web Consortium. "Document object model (dom)." March 2017. [Online]. Available: https://www.w3.org/DOM/

World Wide Web Consortium. "W3c." March 2017. [Online]. Available: https://www.w3.org

Wasserman, M. R., Pulk, A., Zhou, Z., Altman, R. B., Zinder, J. C., Green, K. D., Garneau-Tsodikova, S., Cate, J. H., and Blanchard, S. C., "Chemically related 4,5-linked aminoglycoside antibiotics drive subunit rotation in opposite directions," *Nat Commun*, vol. 6, p. 7896, Jul 2015. [Online]. Available: https://dx.doi.org/10.1038/ncomms8896

Wennerberg, K., Rossman, K. L., and Der, C. J., "The ras superfamily at a glance," *Journal of cell science*, vol. 118, no. 5, pp. 843–846, 2005.

Westbrook, J., Feng, Z., Jain, S., Bhat, T. N., Thanki, N., Ravichandran, V., Gilliland, G. L., Bluhm, W. F., Weissig, H., Greer, D. S., Bourne, P. E., and Berman, H. M., "The protein data bank: unifying the archive," *Nucleic Acids Research*, vol. 30, no. 1, pp. 245–248, 01 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC99110/

Worldwide Protein Data Bank. “Coordinate section.” March 2017. [Online]. Available: http://www.wwpdb.org/documentation/file-format-content/format33/sect9.html

Worldwide Protein Data Bank. “Primary structure section.” March 2017. [Online]. Available: http://www.wwpdb.org/documentation/file-format-content/format33/sect3.html

Worldwide Protein Data Bank. “Secondary structure section.” March 2017. [Online]. Available: http://www.wwpdb.org/documentation/file-format-content/format33/sect5.html

Ye, Y. and Godzik, A., “Flexible structure alignment by chaining aligned fragment pairs allowing twists,” *Bioinformatics*, vol. 19, no. suppl 2, pp. ii246–ii255, 2003.

Zhao, Z., Worthylake, D., LeCour, L., Maresh, G. A., and Pincus, S. H., “Crystal structure and computational modeling of the fab fragment from a protective anti-ricin monoclonal antibody,” *PLoS One*, vol. 7, no. 12, p. e52613, Dec 2012, pONE-D-12-19605[PII]. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3526572/