CALCULATING THE STRUCTURE-BASED PHYLOGENETIC RELATIONSHIP OF DISTANTLY RELATED HOMOLOGOUS PROTEINS UTILIZING MAXIMUM LIKELIHOOD STRUCTURAL ALIGNMENT COMBINATORICS AND A NOVEL STRUCTURAL MOLECULAR CLOCK HYPOTHESIS

A DISSERTATION IN
Molecular Biology and Biochemistry
and
Cell Biology and Biophysics

Presented to the Faculty of the University
of Missouri-Kansas City in partial fulfillment of
the requirements for the degree

Doctor of Philosophy

by
SCOTT GARRETT FOY

B.S., Southwest Baptist University, 2005
B.A., Truman State University, 2007
M.S., University of Missouri-Kansas City, 2009

Kansas City, Missouri
2013

CALCULATING THE STRUCTURE-BASED PHYLOGENETIC RELATIONSHIP OF DISTANTLY RELATED HOMOLOGOUS PROTEINS UTILIZING MAXIMUM LIKELIHOOD STRUCTURAL ALIGNMENT COMBINATORICS AND A NOVEL STRUCTURAL MOLECULAR CLOCK HYPOTHESIS

Scott Garrett Foy, Candidate for the Doctor of Philosophy Degree

University of Missouri-Kansas City, 2013

ABSTRACT

Dendrograms establish the evolutionary relationships and homology of species, proteins, or genes. Homology modeling, ligand binding, and pharmaceutical testing all depend upon the homology ascertained by dendrograms. Regardless of the specific algorithm, all dendrograms that ascertain protein evolutionary homology are generated utilizing polypeptide sequences. However, because protein structures superiorly conserve homology and contain more biochemical information than their associated protein sequences, I hypothesize that utilizing the structure of a protein instead of its sequence will generate a superior dendrogram.

Generating a dendrogram utilizing protein structure requires a unique methodology and novel bioinformatic programs to implement this methodology. Contained within this dissertation is an original methodology that permits the aforementioned structure-based

dendrogram generation hypothesis. Additionally, I have scripted three novel bioinformatics programs required by this proposed methodology: a protein structure alignment program that proficiently superimposes distant homologs, an accurate structure-dependent sequence alignment program, and a dendrogram generation program that employs a novel structural molecular clock hypothesis. The results from this methodology support the proposed hypothesis by demonstrating that generating dendrograms utilizing protein structures is superior to those generated utilizing exclusively protein sequences.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled "Calculating the Structure-based Phylogenetic Relationship of Distantly Related Homologous Proteins Utilizing Maximum Likelihood Structural Alignment Combinatorics and a Novel Structural Molecular Clock Hypothesis", presented by Scott G. Foy, candidate for the Doctor of Philosophy degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Gerald Wyckoff, Ph.D., Committee Chair
Department of Molecular Biology and Biochemistry

Samuel Bouyain, Ph.D.
Department of Molecular Biology and Biochemistry

John Laity, Ph.D.
Department of Cell Biology and Biophysics

Thomas Menees, Ph.D.
Department of Cell Biology and Biophysics

Jakob Waterborg, Ph.D.
Department of Cell Biology and Biophysics

CONTENTS

ILLUSTRATIONS

TABLES

ACKNOWLEDGEMENTS

I would like to thank the School of Biological Sciences at the University of Missouri-Kansas City for funding my tuition, offering services and programs that improve my professional development, and for offering superior classroom instruction. Furthermore, I would like to extend a special acknowledgement to Karen Bame, Ph.D. for her immeasurable assistance.

I would like to express my gratitude to the members of my Supervisory Committee:

Samuel Bouyain, Ph.D.

John Laity, Ph.D.

Thomas Menees, Ph.D.

Jakob Waterborg, Ph.D.

for their beneficial suggestions, guidance, and time through this process.

Furthermore, I would like to thank Marilyn Yoder, Ph.D. for providing the pectate lyase and polygalacturonase protein structures and for helpful suggestions on the proposed programs.

I would also like to thank Lee Likins, M.S. for his educational instruction and insight.

Principally, I would like to express my sincerest gratitude to my Supervisory Committee Chair and dissertation advisor, Gerald Wyckoff, Ph.D., for his continuous supervision, guidance, time, and patience over the course of my tenure at the School of Biological Sciences.

This work is dedicated to my mother, father, and brother;

they helped when I required it most.

CHAPTER I

INTRODUCTION

Dendrograms establish the evolutionary relationships and homology of species, proteins, or genes. The importance of dendrograms is demonstrated in numerous biological disciplines: They are utilized in homology modeling to ascertain which sequences are the most evolutionarily homologous to the query sequence. This enables the calculated structure of the query protein to be based upon the most evolutionarily homologous structures. Additionally, the cost of ligand binding experiments for pharmaceutical testing is reduced by establishing accurate dendrograms of enzymes. Specifically, deriving evolutionary relationships of proteins with known ligands facilitates the calculation of the most probable possible ligands for a query protein. A final example demonstrating the importance of dendrograms is the homology of bacteria or viruses in medicine. The classification of pathogens by evolutionary homology enables the systematic selection of a medicine(s) for each pathogenic class. Furthermore, ascertaining the classification of a novel pathogen enables the manageable innovation of a treatment medicine.

Regardless of the specific algorithm, all dendrograms are generated utilizing nucleotide or polypeptide sequences. Unfortunately, when generating a protein dendrogram, utilizing sequences prevents the derivation of evolutionary relationships for distant homologs. Additionally, polypeptide sequences contain sparse information about a protein relative to that of other biochemical properties such as structure or function. Because protein

structures superiorly conserve homology and contain more biochemical information than their associated protein sequences, I hypothesize that utilizing the structure of a protein instead of its sequence will generate a superior dendrogram.

Although no established or substantiated methodology exists to generate a structure-based dendrogram, a possible example methodology utilizing current bioinformatics programs would involve: 1) structurally aligning the input proteins, 2) deriving a sequence alignment based upon the superimposed protein structures, and 3) inputting this derived sequence alignment into a conventional dendrogram generator. However, this example methodology possesses numerous limitations: First, current structural alignment programs are inaccurate when aligning complex, distant homologs. Although these alignments can possibly be simplified for a more accurate alignment, simplification requires considerable manual curation of the input data. The second limitation of the presented example methodology is the inability of any current bioinformatics program to accurately derive a sequence alignment utilizing superimposed protein structures. Finally, although the sequence alignment was derived utilizing protein structural information, the final step in the example methodology continues to utilize protein sequences to generate the dendrogram. Consequently, this final step disregards and contradicts any advantages initially obtained from implementing structural information. Although methodological variants exist for generating a dendrogram utilizing protein structure, they all possess the aforementioned technological and methodological limitations.

Generating a dendrogram utilizing protein structure requires a unique methodology and novel bioinformatic programs to implement this methodology. Contained within this dissertation is an original methodology that permits the aforementioned structure-based

dendrogram generation hypothesis. Additionally, I have scripted three novel bioinformatics programs required by this proposed methodology: a protein structure alignment program that proficiently superimposes distant homologs, an accurate structure-dependent sequence alignment program, and a dendrogram generation program that employs a novel structural molecular clock hypothesis. The results of this methodology support the proposed hypothesis by demonstrating that generating dendrograms utilizing protein structures is superior to those generated utilizing exclusively protein sequences.

The Molecular Clock and Dendrogram Generation

The molecular clock hypothesis relates nucleotide or polypeptide mutational accumulation to the amount of time required for these mutations[1] to occur (Krane and Raymer, 2003). This relationship is derived from both the number of mutations and the probability of each specific mutation occurring. Because of selection pressure or isolation mechanisms, inevitably, one common nucleotide or polypeptide sequence will evolve into two divergent sequences. As the time since mutational divergence increases, the two resultant sequences increase in dissimilarity both to the common ancestral sequence and to each other. Furthermore, as the time interval increases, the probability of the accumulation of a rare mutation (i.e., a mutation with a relatively low probability of occurrence) also increases.

The time required for the mutational divergence between two sequences to occur is called evolutionary distance. Graphically, the static states of sequences and the evolutionary distances between them are represented in a dendrogram or phylogenetic tree as vertices (nodes) and edges (lines) respectively (Orwant et al., 1999). Dendrograms illustrate the evolutionary relationship between homologous species, chromosomes, genes, or proteins. The final representations of species, chromosomes, genes, or proteins (whether extinct or extant) on the terminal nodes are called operational taxonomic units (OTUs). Establishing the evolutionary relationship of OTUs permits the inference of an unknown quality (e.g., structure, ligand binding, introns/exons, etc.) of one OTU utilizing the known quality of a homologous OTU. Practical biological implications of this inference include protein structural homology modeling, pharmaceutical drug engineering, and ligand binding.

[1] Throughout this dissertation, the term "mutation" encompasses amino acid substitutions, insertions, and deletions.

Implementation of Sequence and Structure in Dendrogram Generation

Calculating evolutionary distances and establishing the phylogenetic relationship of homologous OTUs is not trivial. Therefore, utilizing the molecular clock hypothesis, numerous dendrogram generation methods have been developed. Early methods included the implementation of substitution or distance matrices such as neighbor joining or UPGMA (Unweighted Pair Group Method with Arithmetic mean) (Ewens and Grant, 2005; Isaev, 2006; Krane and Raymer, 2003). These were followed by the maximum parsimony generation method which combinatorically calculates the evolutionary distance by determining which OTUs display the most sequence similarity (i.e., the maximal amount of parsimony) (Krane and Raymer, 2003). Modern dendrogram generation methods include the maximum likelihood (ML) and the Bayesian inference algorithms. The ML dendrogram generation algorithm combinatorically maximizes the total probability of alignment and clustering utilizing a probability matrix (containing the probabilities of nucleotide or amino acid substitutions) (Ewens and Grant, 2005; Isaev, 2006). Additionally, the Bayesian inference algorithm is similar to that of the ML algorithm but it further permits the inclusion of prior phylogenetic knowledge (Huelsenbeck and Ronquist, 2001).

Unfortunately, although the mathematical and statistical methodologies deriving dendrograms have improved, the quality of the aforementioned dendrogram generation calculations is limited by the quality of the input data. Specifically, to establish the evolutionary relationship of input proteins, current dendrogram generation methods are dependent upon the input protein sequences. As the homologies of proteins decrease, the difficulty of establishing the evolutionary relationship of these proteins increases as a result of their increasing sequence divergence. Additionally, if two proteins possess less than a

thirty percent sequence identity, then any corresponding sequence alignment is likely incorrect (Rost, 1999). Therefore, calculation of the evolutionary relationship between proteins is difficult unless the homology of the input sequences supersedes this threshold.

Amino acid sequences input into dendrogram generators are limited in the amount of sequence divergence they can possess. Consequently, dendrogram generators are unable to derive the evolutionary relationship of distantly related homologs (e.g., proteins belonging to different protein families). However, because selection pressure influences the structure of a protein more directly than it influences its amino acid sequence, the structure of a protein is more evolutionarily conserved (Marti-Renom et al., 2000). That is, as the evolutionary distance between proteins increases, the sequence disparity of the proteins increases at a greater rate than that of structural divergence. Therefore, when deriving the evolutionary relationships of distant homologs possessing inadequate sequence similarity, the corresponding protein structures will theoretically generate a superior dendrogram because of structural conservation.

Methodological and Algorithmic Overview

Methodological Overview

Generating a dendrogram based upon protein structure is a complex endeavor that requires a multistep methodology to complete. The following is an overview of the proposed methodology required for structure-based dendrogram generation:

1) Superimposing the protein structures. Regardless of algorithmic details, calculation of the relative evolutionary relationships between the input protein structures requires a method of protein structural comparison. The three-dimensional spatial coordinates of protein structures are stored in Protein Data Bank (PDB) files (Berman et al., 2000). Because the spatial location and orientation of each protein in each PDB file varies (despite homology), comparative structural calculations require that all input proteins be structurally superimposed.

2) Derivation of a structure-dependent sequence alignment. Calculating a comparative structural measurement not only requires superimposing the proteins, it also requires establishing atomic coordinate homology (i.e., determining which atoms will match and thus share calculations). Biologically, a sequence alignment that is derived utilizing the superimposed proteins permits the determination of atomic homology (and thus the ability to calculate a comparative measurement) because each amino acid comprises consistent backbone atoms. Therefore, atomic homology is established by determining amino acid homology. Importantly, the sequence alignment must be derived utilizing the spatial positions of the amino acids

comprising the structurally superimposed proteins; it cannot be conventionally derived by amino acid identity.

3) Calculating the evolutionary relationship of the proteins. Upon completion of superimposing the proteins and structurally-deriving the atomic homology, the protein structures must be comparatively measured. Furthermore, the calculated comparative quantities must be accurately and consistently translated into evolutionary distances. Finally, these evolutionary distances collectively comprise a matrix and a hierarchical clustering algorithm generates the dendrogram.

Novel Algorithms

Unfortunately, the three aforementioned procedures necessitate nonexistent programs and algorithms. Therefore, I created three novel programs and one module (the BioInfo module), each capable of accomplishing one of the aforementioned methodological challenges. Below are concise descriptions, methodological solutions, and outlines of the programming scripts for each program:

1) The Structural Alignment By Maximum Likelihood Estimation (SABLE) program utilizes only protein structural information to accurately superimpose both evolutionarily similar and distant homologous proteins. Furthermore, it possesses the versatility to align an unlimited number of input proteins, each composed of indiscriminant and variable numbers of subunits. Although other protein structure superimposing algorithms exist, none accurately and comprehensively superimpose distantly related, complex homologous proteins.

- Primary Program
    - print_time – prints the date and time of execution
    - recovery_input – permits the input of recovery data if program fails
    - recovery_output – outputs an emergency recovery file containing data
    - spatial_parameters – calculates the spatial perimeter and spatial midpoint of each input protein
    - largest_protein – calculates the protein possessing the largest spatial perimeter
    - protein_standard_deviation_calculation – calculates the protein and subunit standard deviations
    - template_protein_rotation – rotates the template protein
        - single_atom_translate_rotate – translates and rotates a single atom around a spatial center utilizing quaternion calculations
    - sa_probability_calculation – calculates the probability of each pseudostate in a list
        - chain_probability_filter – retains pseudostates with greatest chain probabilities (Phase 1 only)
        - chain_matching_with_trans_rot – determines chain homology for a pseudostate
            - single_atom_translate_rotate
        - protein_translate_rotate – translates and rotates all atoms in a protein around a spatial center utilizing quaternion calculations
            - single_atom_translate_rotate

- pseudostate_cutter – removes least probable pseudostates from a list
- first_translation_rotation – executes Phase 1 of SABLE
- second_translation_rotation – executes Phase 2 of SABLE
    - sa_probability_calculation
    - pseudostate_cutter
- third_translation_rotation – executes Phase 3 of SABLE
    - protein_translate_rotate
    - maxtrix_to_list_conversion – alternates data structures
    - optimal_template_protein – calculates optimal template protein based upon mean center
    - template_protein_rotation
    - sa_probability_calculation
    - pseudostate_cutter

- Required Modules
    - Bioinfo::Struct
        - pdb_input – inputs information from PDB files
        - pdb_output – generates output PDB files for aligned proteins
    - IO::Handle
        - autoflush – implements the automatic transference of buffer data during the interprocess communication (IPC) of parallel processes
    - Math::Quaternion
        - normalize – normalizes quaternion to unit length

        - rotate_vector – rotates a point around a spatial center utilizing the quaternion geometry
    - Math::Combinatorics
        - permute – generates all possible combinations from a list of values

2) Unfortunately, SABLE only superimposes proteins and does not derive a resultant structure-dependent sequence alignment. Therefore, the Universal True SDSA (Structure-dependent Sequence Alignment), or UniTS, program calculates the most probable sequence alignment derived from multiple superimposed protein structures. Although other algorithms have been developed to derive a sequence alignment from aligned structures utilizing atomic proximity, none of these appropriately manages multiple residue matches, prevents the incorrect ordering of residues, and sequentially aligns structurally nonconserved regions.

- Primary Program
    - spatial_parameters – calculates the spatial perimeter and spatial midpoint of each input protein
    - protein_standard_deviation_calculation – calculates the protein and subunit standard deviations
    - optimal_template_protein – calculates optimal template protein based upon mean center
    - multiple_chain_alignment – aligns multiple chains to a template chain
        - chain_matching - determines the chain homology of two proteins

- sdsa_protein_removal – removes proteins from the Grid
- find_amino_acid_name – derives the consensus amino acid sequence for each chain
- pairwise_sdsa – performs a pairwise SDSA
  - amino_acid_order_refinement – resolves multiple homologous amino acid matching
    - inconsistent_ordering_algorithm – generic algorithm that sorts amino acid matches
  - multiple_muscle_amino_acids – derives amino acid for a Grid position in which a residue is not numerically superior
- fasta_seq_align_output – outputs the SDSA in FASTA format
- rmsd_calculation – calculates quality assessment scores for pairwise superimposed proteins
- positional_sd_calculation – calculates quality assessment scores for multiple superimposed proteins
  - standard_deviation_algorithm – calculates the standard deviation of a list of numbers

- Required Modules
  - Bioinfo::Struct
    - pdb_input – inputs information from PDB files
  - Storable
    - dclone – duplicates a complex data structure
  - Math::Combinatorics

        - permute – generates all possible combinations from a list of values

3) The Phylogenetic Tree Using Structural Homology (PUSH) program generates a dendrogram utilizing a novel structural molecular clock hypothesis to derive a probability matrix. The dendrogram is then graphically displayed utilizing a hierarchical clustering algorithm. The calculation results of the proposed structural molecular clock hypothesis are unique and, therefore, require a unique program to implement. Furthermore, despite the existence of numerous hierarchical clustering and dendrogram generation modules, none permit the implementation of a custom matrix; instead, each derived a substitution matrix utilizing conventional sequence-based methods.

- Primary Program
    - print_time – prints the date and time of execution
    - evolutionary_distance_calculator – calculates the evolutionary distance between two input protein structures
- Required Modules
    - BioInfo::Struct
        - pdb_input – inputs information from PDB files
    - BioInfo::Phylo
        - hierarchical_clustering – utilizes a matrix to calculate a dendrogram

- _node_object_completion_ - completes the data in a node object then dichotomously divides the object into two node objects
    - _max_distance_using_matrix_ - calculates the maximum evolutionary distance in a distance matrix
- print_dendrogram – prints the dendrogram as a .gif file
    - _max_distance_using_object_list_ - normalizes evolutionary distances
    - _branch_y_coordinate_calculator_ - calculates the length of each branch in pixels
- node_object_to_newick – converts a dendrogram represented as node objects into the Newick format

o BioInfo::Seq

- fasta_input – inputs information from FASTA files

o GD::Image

- colorAllocate – assigns colors to a variable
- line – draws a line on a graphic
- string – permits a string of characters to be placed on a graphic

The included methodology detailing the derivation of a dendrogram utilizing protein structures requires the implementation of all the aforementioned programs. Importantly, however, I designed each program to be utilized independently and associate with a distinct category of bioinformatic algorithms. Additionally, I have empirically demonstrated that the

results of each of these programs are superior to those of other programs and algorithms in their respective bioinformatic categories. The following three sections discuss each category in detail, the algorithmic specifications of each program, and the comparative results generated utilizing each program.

CHAPTER II

SABLE: STRUCTURAL ALIGNMENT BY MAXIMUM LIKELIHOOD ESTIMATION

Introduction to Structural Alignment and Superpositioning

Superimposing proteins has become fundamental to molecular biology and is required for research in everything from homology modeling to comparing protein conformational states to sequence alignments of evolutionarily divergent proteins. Unfortunately, superpositioning programs require a preliminary sequence alignment and thus cannot be employed for superimposing divergent homologous protein structures. Alternatively, structural alignment programs are dependent on the influence of secondary structures and thus generate an alignment from incomplete information.

SABLE (Structural Alignment By Maximum Likelihood Estimation) is a protein superimposing program that combines the versatility of a structural alignment program with the accuracy and comprehensiveness of a superpositioning program. SABLE implements a maximum likelihood algorithm to thoroughly compare possible protein structural translocations. It then calculates the optimally superimposed position for each input protein utilizing a novel distance-based probability scoring algorithm that accurately manages extreme distances. Importantly, SABLE does not require a preliminary sequence alignment. Furthermore, it will theoretically accept an unlimited number of input proteins, each composed of indiscriminant and variable numbers of subunits.

Structural Alignment and Superpositioning Algorithms

Proteins are flexible, dynamic structures that most bioinformatics algorithms restrict to a static state. In PDB files, protein structure is represented by utilizing static atomic coordinates, while the flexibility of the protein is expressed utilizing the B factor (Berman et al., 2000). Many algorithms that superimpose protein structures (including SABLE) postulate each structure to be in both a static state and a single conformational state; therefore, they do not transmute the structures to improve the quality of the superimposition. Unfortunately this limits the biological implications of superimposing protein structures. Primarily, algorithms may be incapable of superimposing even highly homologous protein regions if these regions are flexible and in different positions. Furthermore, conformational changes in protein structure can translocate entire subunits, thus making it impossible to superimpose their static structures. Like many bioinformatics algorithms that utilize static protein structures, SABLE possesses the aforementioned biological limitations because it does not transmute input protein structures.

The method of measurement used to superimpose protein structures is a philosophical consideration without a single correct answer. That is, what exactly does it mean to superimpose proteins as "closely" as possible? Although many of these methods exist (e.g., energy minimization [Micheletti and Orland, 2009]), the distance-based method is the most intuitive. The distance-based method of superimposing proteins attempts to minimize the

spatial distance between matching or **homologous atoms**[2]. Importantly, this minimizing distance is not required to explicitly be direct distance; instead, it can be a measurement based on the distance. For example, the probability measurement of SABLE is derived from the distance between homologous atoms.

Protein structures can be superimposed utilizing a distance-based approach with or without *a priori* knowledge of the sequence alignment. Specifically, superpositioning software requires a preliminary sequence alignment, while structural alignment software does not. Because the preliminary sequence alignment required for superpositioning software determines atomic homology, the distance (or distance-based measurement) calculations between homologous atoms depend on this sequence alignment. Therefore, despite the specific superpositioning algorithm, the quality of the superpositioning solution is dependent on the quality of the sequence alignment. This dependency is unimportant for proteins demonstrating a high sequence identity because the sequence alignment is assumably correct. However, if the sequence identity is less than thirty percent, the sequence alignment is assumably incorrect, leading to incorrect structural superpositioning (Rost, 1999).

Because structural alignment software is utilized to infer protein homology and evolutionary relationships, the preliminary sequence alignment is unknown and must be derived by superimposing the protein structures. This enables structural alignment programs to theoretically align considerably divergent proteins since they are not dependent on an input sequence alignment. Distance-based structural alignment algorithms utilize a generic

[2] Homologous atoms are atoms located at the same position in homologous amino acids. Homologous amino acids are two amino acids that are considered to be a match or aligned. Note that "matching" is a broad term and does not exclusively occur through homology by sharing a common amino acid in an ancestral protein. For example, "homologous" amino acids are featured when aligning two of the same proteins in different conformational states. Importantly, homologous atoms are only present in the backbones of amino acids unless the amino acids are identical residues.

algorithm despite distinguishing details. The algorithm divides the input proteins into oligopeptide substructures based on the protein folds and secondary structures in a contact map[3]. Most structural alignment algorithms then implement a combinatorics algorithm to minimize the distance (or a distance-based measurement) between the oligopeptides (Holm and Sander, 1993; Konagurthu et al., 2006; Ortiz et al., 2002). Furthermore, the matched oligopeptides are used to generate the optimal sequence alignment.

Unfortunately, structural alignment programs also have limitations. Numerous programs alter protein structures by translocating relative atomic positions to generate an improved alignment (for a list of programs, see Micheletti and Orland [2009]) (Menke et al., 2008). Because protein structures are flexible *in vivo*, the researcher may prefer this feature. However, flexible proteins may generate negative consequences if the researcher requires the proteins retain their unaltered structures. Another limitation of structural alignment programs is the derivation of a sequence alignment based upon matching oligopeptide pairs (Konagurthu et al., 2006). This not only fails to generate a sequence alignment based upon the entire protein, it also fails to generate a complete sequence alignment (i.e., the alignment consists of only a fraction of the total number of amino acids). Additionally, many structural alignment programs use arbitrary values such as gap penalties to generate the sequence alignment (Konagurthu et al., 2006; Ortiz et al., 2002).

In addition to the aforementioned limitations, structural alignment algorithms that divide proteins into secondary structure and fold oligopeptides possess a primary disadvantage: The alignment overweighs the secondary structural influence in each protein. That is, consistent secondary structures or folds in the proteins will align at the expense of

[3] A contact map is generated utilizing a distance matrix containing the distances between all the alpha carbons in a protein. A distance threshold is arbitrarily established (usually four or five angstroms); any distance less than the threshold is black, while any distance greater than the threshold remains white. See Holm and Sander (1993) for more detailed information.

the remainder of the protein structures. Although the secondary structures are theoretically more conserved than other regions of the proteins (Marti-Renom et al., 2000), highly divergent proteins may possess inconsistent secondary structures. Furthermore, conserved regions of the protein such as active sites or a conserved "core" are not guaranteed to be composed of secondary structures. Additionally, current structural alignment algorithms align input proteins on a fold-level because they match consistent secondary structures and "superimpose" the proteins by minimizing the distance between these secondary structures (Ortiz et al., 2002). Superpositioning algorithms, however, superimpose the proteins by atomic distance minimization; therefore, the optimal superposition is an atomic-level calculation. Superimposing proteins at the atomic-level instead of the fold-level ensures the minimum possible distance between the proteins instead of simply ascertaining fold homology.

Superpositioning by Structural Alignment

Because SABLE is a structural alignment program and does not require a preliminary sequence alignment, it is capable of aligning distant homologs without a minimum sequence identity threshold. However, unlike current structural alignment programs, SABLE does not divide input proteins into oligopeptides and match them at a fold-level. Instead, SABLE implements a maximum likelihood algorithm that superimposes proteins on an atomic-level. Superimposing proteins on an atomic-level provides SABLE with numerous advantages over traditional structural alignment programs. Because SABLE considers all amino acids in the structural alignment, the generated sequence alignment is derived from the entire protein structure, not merely oligopeptide structural matches. Furthermore, SABLE is able to

implement a more advanced measurement algorithm that compares to those utilized in superpositioning algorithms (Theobald and Wuttke, 2006a, 2006b, 2008).

The final advantage SABLE possesses over traditional structural alignment programs is the elimination of secondary structural bias without completely disregarding the conservative homology of the secondary structure. Although recently developed structural alignment programs distinguish conserved structures from secondary structures (e.g., the unit-vector RMS calculation in the MAMMOTH program [Kedem et al., 1999; Ortiz et al., 2002]), early alignment programs emphasize the contribution of secondary structure to the alignment because they are theoretically the most conserved regions in homologous proteins (although, as stated in the previous section, this is not guaranteed to be true). The SABLE algorithm, however, superimposes proteins by utilizing all input amino acids, thus eliminating the overweighed alignment contribution of secondary structures. Furthermore, by nature of the maximum likelihood algorithm, SABLE automatically considers the influence of the conservation of secondary structures. This influence is generated naturally because aligning the conserved regions of homologous proteins will produce the greatest structural superimposing likelihood, while attempting to align nonconserved regions will produce a lower likelihood. Otherwise stated, aligning the nonconserved regions at the expense of the conserved regions will result in a lower likelihood than aligning the conserved regions at the expense of the nonconserved regions. Therefore, SABLE considers the influence of all conserved regions (even those not composed of secondary structures) in a biologically accurate approach, while still utilizing the entire protein structure to superimpose the input proteins.

SABLE Algorithm

Maximum Likelihood Structural Alignment

Numerous bioinformatic programs use the maximum likelihood (ML) algorithm to ascertain the most probable solution for a problem. In its most generic format, the ML algorithm calculates every possible solution to a problem. The probability or likelihood (£) of each possible solution (i.e., a **pseudostate** [Krane and Raymer, 2003]) is then measured relative to the ideal solution. Consequently, the solution most similar to that of the ideal (i.e., the one with the greatest £; denoted *max*(£)) is selected as the most likely solution to the problem.

Given a pseudostate, £ of the pseudostate will change as the value of one of its parameters changes. Geometrically, the x-axis represents the range of the changing parameter and the y-axis represents £. The parameter value that generates *max*(£) can be calculated by setting the derivative of the curve equal to zero (the derivative-based approach) (Ewens and Grant, 2005). Alternatively, calculating £ at specific intervals of the parameter (the brute-force approach) will also generate the likelihood curve. Note that increasing or decreasing the parameter value by the interval generates a new pseudostate. Furthermore, decreasing the interval size increases the number of possible pseudostates to be generated.

A new axis is added to the geometric system for each changing parameter. Therefore, the brute-force approach dictates that increasing the number of changing parameters increases the number of possible pseudostates. SABLE changes seven parameters to generate a brute-force likelihood curve: three translation parameters, three rotation parameters, and the input sequence alignment (note that the three translation and three rotation parameters

delineate the location of the pseudostate). This combinatorics-like algorithm generates millions of pseudostates; therefore, the derivative-based approach would seem to be the superior option because it is computationally faster. Unfortunately, altering the sequence alignment will not generate a consistent likelihood curve, thus preventing the calculation of $max(£)$ using the derivative of the curve. Therefore, SABLE uses the brute-force approach to superimpose proteins with the accuracy of the ML method, while still not requiring a preliminary sequence alignment.

Protein Geometry Terminology

When represented as a Protein Databank (PDB) file, a protein is a static structure whose atoms are points in a three-dimensional space (Berman et al., 2000). This space possesses x-, y-, and z-axes to enable the PDB file to display atoms in terms of x-, y-, and z-coordinates. Using the coordinates of every atom in a protein, the greatest x-coordinate (x_{max}) and the least x-coordinate (x_{min}) indicate the x-axis' **spatial perimeter**. The distance (in angstroms) between x_{min} and x_{max} is the x-axis **diameter** (δ_x). These same definitions of spatial perimeter and diameter also apply to the y- and z-axes (i.e., δ_y and δ_z). Together, the spatial perimeters of all three axes compose the protein's total spatial perimeter.

The **spatial center** (c_Δ) of a protein, which is the center of the protein based upon the spatial perimeter, is defined as follows:

$$c_\Delta = \left(\frac{\delta_x}{2} + x_{min}\,, \frac{\delta_y}{2} + y_{min}\,, \frac{\delta_z}{2} + z_{min}\right)$$

The **mean center** (c_μ) of a protein is the average of all the atomic coordinates that compose the protein:

$$c_\mu = \left(\frac{\sum x_n}{n_{atoms}} , \frac{\sum y_n}{n_{atoms}} , \frac{\sum z_n}{n_{atoms}} \right)$$

where n_{atoms} is the total number of atoms in the protein.

Superimposing Measurement of SABLE

Traditionally, the root mean squared deviation (RMSD) is used to determine the quality of protein structural alignments. Unfortunately, long atomic distances (outliers) are overweighed, causing a disproportionate increase in the RMSD (Mechelke and Habeck, 2010). Although many structural alignment programs utilize RMSD-based measurements (e.g., the unit-vector RMS and the normalized weighted RMSD [Kedem et al., 1999; Wang and Dong, 2012]) to quantitatively calculate the alignment quality of protein structures (Konagurthu et al., 2006), recent structural superpositioning programs quantitatively superposition proteins utilizing variance or covariance matrices (Theobald and Wuttke, 2006a, 2006b, 2008). The novel scoring algorithm of the SABLE program further develops this concept by implementing the standard deviation (similar to variance) to calculate a probability curve. Therefore, to determine the probability of superimposing two atoms, SABLE utilizes a Gaussian probability curve (Figure 1).

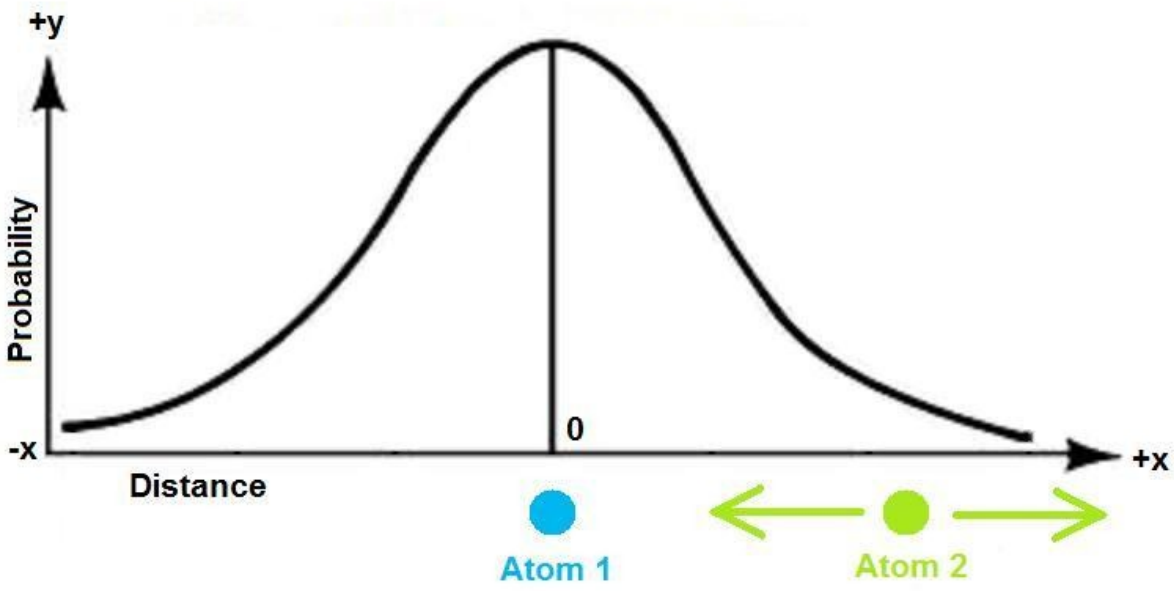


Figure 1. The relationship between Gaussian probability and atomic spatial distance.

As the distance between two atoms increases, the probability decreases by approaching zero. Alternatively, as the distance between two atoms decreases, the probability increases by approaching the zenith of the curve. This probability scoring measurement eliminates overweighing large atomic distances because the change in the probability decreases as the probability curve approaches the asymptote at the x-axis. Furthermore, when two atoms are close together, the algorithm will not emphasize moving them closer together at the expense of other atoms because of the plateau at the height of the probability curve.

Based upon a normal probability distribution curve, the general formula for the probability (p) of a random point being at x location along the x-axis is as follows:

$$p = \left(\frac{1}{\sigma\sqrt{2\pi}}\right) e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

where σ is the standard deviation and μ is the mean location of the points (Stewart, 2003, p. 616). However, when calculating the probability of two atoms superimposing, the mean (μ) is equal to zero and x becomes the distance between the two atoms (d), thus simplifying the equation to:

$$p = \left(\frac{1}{\sigma\sqrt{2\pi}}\right) e^{\frac{-d^2}{2\sigma^2}}$$

The σ is a measure of the deviation between the diameters (δ) of all the proteins (to determine subunit homology) or subunits (to determine atomic probability). Importantly, σ is averaged across all three dimensions, thus allowing the protein to rotate without having to recalculate σ for each dimension. The **protein standard deviation** (σ_p) is calculated using the mean of all diameters (δ_μ) of the program's input proteins. That is, δ_μ is the mean of the x-, y-, and z-axis diameters for every protein in the alignment.

$$\sigma_p = \sqrt{\frac{\sum(\delta - \delta_\mu)^2}{n_p - 1}}$$

where n_p is the total number of input proteins.

The **subunit standard deviation** (σ_s) must be calculated using the mean diameters from all the subunits of all input proteins. The mean diameter of a single subunit (δ_s) in an input protein is calculated using the following equation:

$$\delta_s = \left(\frac{\delta_x \cdot \delta_y \cdot \delta_z}{n_s}\right)^{\frac{1}{3}}$$

where n_s is the number of subunits in the protein and $\delta_{x/y/z}$ are the diameters for the subunit's respective protein. The mean diameter for all the subunits is then calculated by averaging δ_s for all the input proteins. The σ_s calculation is finished using a population standard deviation equation.

$$\sigma_s = \sqrt{\frac{\sum(\delta_s - \delta_\mu)^2}{n_t}}$$

where n_t is the total number of subunits in all the input proteins and δ_μ is the mean diameter of all the subunits of all input proteins. Note that δ_s represents each subunit, not each protein; therefore, the same δ_s will be used multiple times if the protein possesses multiple subunits. Further note that when calculating σ_p, δ_μ is the average of all the *protein* diameters, while δ_μ is the average of all the *subunit* diameters when calculating σ_s.

Importantly, no degrees of freedom are subtracted when calculating σ_s because every atom in the subunit is used in the σ_s application (to determine atom probability). Conversely, σ_p is a sample standard deviation equation that requires the subtraction of a degree of freedom. The sample standard deviation equation is necessary because its application (to determine subunit homology) uses only five amino acids per subunit (discussed subsequently), thus σ_p represents only a sample of the total population of atoms.

Although the aforementioned equations calculate the probability of superimposing two homologous atoms, SABLE must be able to calculate £ of superimposing entire protein structures for each pseudostate. First, the probability of each homologous pair of backbone atoms (alpha carbon, carboxyl carbon, and amine nitrogen)[4] is calculated. To derive £ of superimposing proteins in a pseudostate, SABLE multiplies the individual atomic probabilities (p_{atom}):

[4] Although the carboxyl oxygen is a consistent backbone atom, contrary to many structural measurements (e.g., backbone RMSD [Guex and Peitsch, 1997]), the probability measurement of SABLE does not incorporate this atom. Including the carboxyl oxygen in the probability calculation doubles the influence of the psi dihedral angle relative to the omega and phi angles. For example, assuming the amine nitrogen remains stationary, the spatial location of the carboxyl carbon represents the phi angle rotation. However, immobilizing the alpha carbon and rotating the psi angle relocates the spatial position of both the carboxyl oxygen and the amine nitrogen of the adjacent amino acid. Therefore, two atomic probability scores represent this dihedral rotation.

$$\mathcal{L}(I|S) = \prod p_{atom}$$

where the function $\mathcal{L}(I|S)$ is the likelihood of the input parameters (I) given a pseudostate (S) (Isaev, 2006). This multiplication is a logical "and" statement, representing the probability of superimposing all homologous atoms. The only input parameter (I) is the sequence arrangement that dictates homologous atoms (described below). Additionally, SABLE delineates the pseudostate location (S) using three translation and three rotation parameters.

Regardless of the specific scoring algorithm, any distance-based structural alignment program must determine homologous atom matches before calculating the distance (or distance-based measurement) between them. Generically, SABLE uses a combinatorics (i.e., an all-possible-combinations) algorithm to determine sequence homology. Duplicate amino acid matches are permitted and the sequence homology that generates $max(£)$ is the correct matching arrangement. Theoretically however, the number of possible sequence arrangements is too numerous to practically calculate £ for each arrangement. Therefore, SABLE performs heuristic steps to minimize the number of sequence arrangements to be calculated. Without these heuristic steps, calculating $max(£)$ would require calculating the probability between each amino acid from one protein to every amino acid in the second protein. However, SABLE predicts a limited range of amino acids from the second protein to match each amino acid from the first protein. This range is determined by first calculating the difference between the two sequence lengths ($\Delta\ell$; will be at least ten percent of the longer sequence). Then, each amino acid in both sequences is converted from a position in the sequence to a percentage of the sequence (the n-terminus is zero percent and the c-terminus is one hundred percent). For each amino acid in the first sequence, an amino acid from the second sequence is selected that possesses approximately the same sequence percentage. The

range for the second sequence is $\pm\Delta\ell$ from the selected amino acid. Generating this range operates on the assumption that a large deletion from one terminus and a correspondingly large insertion on the other terminus did not occur as the two sequences evolved independently.

Subunit Homology

Unfortunately, the PDB does not require consistent chain designations amongst homologous proteins (Berman et al., 2000). Therefore, to structurally align proteins composed of multiple subunits, SABLE must first determine subunit homology of the input proteins. Subunit homology is determined by calculating which amino acids are 0, 25, 50, 75, and 100 percent of the sequence (initiated at the n-terminus) for each subunit for all the input proteins. SABLE pretends each subunit is composed of only these five amino acids; therefore, when calculating subunit probabilities, it will match only amino acids with the same sequence percentage. These individual atomic probabilities are then multiplied to obtain the probability of one subunit superimposing the other. Having only five amino acids per subunit makes performing an all-possible-combinations (of subunits) algorithm practical. The likelihood of each subunit combination is calculated by multiplying the individual subunit probabilities to obtain the total likelihood of superimposing the proteins. Importantly, when determining subunit homology, because SABLE calculates these probabilities utilizing only five amino acids per subunit, they are not the true probabilities of superimposing two subunits or proteins. Furthermore, because subunits can statistically deviate on a protein-scale, the probability calculations require that σ_p be implemented as σ.

Reducing Infinite Pseudostates

Theoretically, once a method of measuring structural superimposing is established, a program can use an ML algorithm to determine optimal structural alignment. Generically, the ML algorithm will keep one protein stationary (the **template protein**; P_t) while translating and rotating another protein (the **mobile protein**; P_m) around the three-dimensional coordinate space. As P_m moves, its position at any point in time is a **pseudostate** (S_n) of the protein (i.e., a "snapshot" of the mobile protein). Because the coordinate space is infinite in all directions, boundless translation of P_m would produce an infinite number of possible pseudostates (S_∞). Even if P_m is contained within bounds, translating it by an infinitesimal fraction of an angstrom or rotating it by an infinitesimal fraction of a radian will produce a new pseudostate, thus S_∞ continues to be the number of possible pseudostates.

SABLE reduces S_∞ to S_t (where t is a finite number of total pseudostates) by endorsing the bounded translation and rotation of P_m by a specified distance or angle. The translation length between S_n and S_{n+1} is L_0, while the rotation angle between S_n and S_{n+1} is θ_0. P_m perpendicularly translates by L_0 in the direction of each three-dimensional axis (in both positive and negative directions). That is, P_m does not translate diagonally relative to the three axes. Additionally, P_m rotates unidirectionally by θ_0 around each of the three axes. Within the translation boundary, for each pseudostate that is translated by L_0, P_m will be rotated by θ_0 multiple times along each axis until P_m has been rotated by one radian around each axis. A new pseudostate is generated for each θ_0 around any axis. Despite a rotation of only one radian per axis, P_m can still invert completely due to cumulative rotation along all three axes.

SABLE: Phase 1

The SABLE program is divided into three structural alignment phases. For Phases 1 and 2, P_t is selected and each P_m aligns to P_t separately in a series of pairwise structural alignments. Each pairwise alignment features P_t remaining stationary, while P_m generates pseudostates by translating and rotating by L_0 and θ_0 respectively. After t pseudostates have been generated, the probability of each S_n in S_t is calculated using the aforementioned probability formulas. One quarter of S_t with the greatest probabilities are retained for Phase 2.

As P_m translates by L_0, it must remain within a boundary around P_t. The translation boundary for Phase 1 is the spatial perimeter of P_t. The c_Δ of P_m is prohibited from exiting the cube generated by this spatial perimeter. During Phases 1 and 2, P_t must be the largest input protein (based on cubic angstroms using the spatial perimeter dimensions) to establish the correct boundary. If P_m was the larger protein, it is possible that P_t would never superimpose the periphery of P_m.

Unfortunately, the generous boundary and numerous translation and rotations make Phase 1 the most prolonged phase of SABLE. Many pseudostates in S_t, however, produce P_m positioning that is either too distant or too transposed from P_t to result in a meaningful $\mathcal{L}(I|S_n)$. Therefore, SABLE utilizes a filtering algorithm to prevent the unnecessary lengthy probability calculation of significantly incorrect pseudostates. Calculating the complete £ requires the probability computation of every amino acid; therefore, the filtering algorithm uses the probability calculated when determining subunit homology because this calculation requires probability computation of only five amino acids per subunit. Once this probability

is calculated for each S_n, the filtering algorithm will discard a certain percentage of the least probable pseudostates in S_t.

When calculating the percentage of S_t to be removed by the filtering algorithm, SABLE assumes P_t is a globular protein with an ellipsoid shape. Furthermore, SABLE assumes that any S_n featuring the c_Δ of P_m outside the general ellipsoid shape of P_t but remaining within the spatial perimeter is likely to possess an unsatisfactory homologous subunit probability. Therefore, the percentage of S_t to be filtered ($S_\%$) is expressed by the following equation:

$$S_\% = 1 - \frac{V_e}{V_c}$$

where V_c is the cubic volume of the spatial perimeter and V_e is the volume of the ellipsoid (Weisstein):

$$V_e = \frac{4}{3} \cdot \pi \cdot \frac{\delta_x}{2} \cdot \frac{\delta_y}{2} \cdot \frac{\delta_z}{2} = \frac{\pi \delta_x \delta_y \delta_z}{6}$$

Note that δ is the diameter and must be halved to equal the radius. Using the above V_e equation, $S_\%$ equals the following value:

$$S_\% = 1 - \frac{\left(\frac{\pi \delta_x \delta_y \delta_z}{6}\right)}{\delta_x \delta_y \delta_z} = 1 - \frac{\pi}{6} \cong 0.48$$

Additionally, considering computational efficiency, SABLE assumes that half of the S_n featuring the c_Δ of P_m inside the general ellipsoid shape of P_t will also possess unsatisfactory probabilities. Therefore, SABLE increases the calculated percentage of S_t to be removed by the filtering algorithm to 75 percent.

SABLE: Phase 2

Because P_m translates within the voluminous spatial perimeter of P_t, the initial parameters of L_0 and θ_0 must be generously large to minimize the number of pseudostates in S_t. Unfortunately, although large initial parameters reduce the size of S_t, they also decreases the accuracy of the structural alignment. Therefore, Phase 2 of SABLE gradually decreases the translation length of L_0 and the rotation angle of θ_0 until they are less than or equal to a final pair of parameters containing lesser quantities (L_f and θ_f). Decreasing the sizes of the two initial parameters to those of the final parameters will increase the accuracy of the structural alignment without exponentially increasing S_t.

Phase 2 of SABLE receives S_t from Phase 1 and retains twenty percent of the pseudostates with the greatest £. The quantities of L_0 and θ_0 are then halved ($L_{0.5}$ and $\theta_{0.5}$) and used to generate new pseudostates. A portion of these new pseudostates is generated by translating the retained pseudostates in S_t by $L_{0.5}$ in the positive and negative direction of each axis. That is, each S_n is translated by $L_{0.5}$ in each of the six directions (positive x-direction, negative x-direction, etc.) to generate six new pseudostates. The final portion of the new pseudostates is generated by rotating each S_n (including those recently generated by a translation of $L_{0.5}$) along each axis in both the positive and negative directions by $\theta_{0.5}$. Once these new pseudostates are assembled into S_t, SABLE calculates £ for each S_n. SABLE then retains a specific number of those pseudostates in S_t possessing the greatest £. Importantly, to prevent the exponential growth of S_t, the number (not the percentage) of retained pseudostates equals the number of pseudostates retained immediately following Phase 1 and remains constant for all iterations. The parameters of $L_{0.5}$ and $\theta_{0.5}$ are then halved and the

process repeats for multiple iterations until $L \leq L_f$ and $\theta \leq \theta_f$. For each P_m, the S_n with the greatest £ is retained for Phase 3.

SABLE: Phase 3

In both Phases 1 and 2, SABLE selects a P_t based on the size of the spatial perimeter and structurally aligns it to each of the other input proteins (P_m) in a series of pairwise alignments. In Phase 3, however, SABLE selects a new P_t and aligns it to the remaining proteins using another series of pairwise alignments. The protein that possesses the least **mean center error distance** (MCED), which is the distance between the c_μ of each input protein and the c_μ of all the proteins combined, is selected as the new P_t. Designating P_t as the protein with the least MCED minimizes the total translational and rotational movement required for all the P_ms to structurally align to P_t. Therefore, each P_m is converging on a P_t that possesses the most "average" spatial position.

For each pairwise alignment between P_m and the newly designated P_t, SABLE generates new pseudostates similarly to those generated in Phase 2. However, only a single iteration using L_f and θ_f is performed instead of several iterations that half L and θ. Additionally, σ_s is hardcoded to equal four angstroms to increase the probability curve sensitivity. Furthermore, instead of P_m translating and rotating each iteration by only a single quantity of L and θ, Phase 3 permits P_m to translate and rotate by multiple quantities of L_f and θ_f. The number of quantities P_m can translate and rotate is equal to L_f / MCED (rounded to the nearest integer). That is, the MCED determines the translational and rotational deviation from the current position of P_m relative to the position P_m must achieve for adequately superimposing P_t. Finally, SABLE calculates $\mathcal{L}(I|S_n)$ of the pseudostates generated by Phase

3. For each P_m, SABLE then selects the S_n achieving $max(£)$ to be the final position of P_m in the structural alignment.

SABLE Accuracy Program

If more accuracy is required upon execution of SABLE, the aligned protein structures may be input into the SABLE Accuracy program. SABLE Accuracy is an independent program that increases the accuracy of the SABLE results by decreasing L_f and θ_f. The program executes multiple iterations similar to Phase 2 of the original SABLE until L and θ are reduced to the new L_f and θ_f. However, why not simply utilize these reduced quantities in the primary SABLE program? The spatial divergence of the proteins before the structural alignment necessitates that SABLE incorporates numerous calculations to prevent a heuristic problem. However, following the structural alignment of SABLE, the proteins will be relatively superimposed and the prevention of heuristic problems is unnecessary; therefore, many of these extraneous calculations are nonessential (e.g., the number of pseudostates retained per iteration decreases, σ_s is hardcoded to equal four angstroms, etc.). Therefore, compared to the primary SABLE program, SABLE Accuracy increases the speed at which L and θ are reduced.

SABLE Results

Because SABLE combines the versatility of a structural alignment program with the accuracy and comprehensiveness of a superpositioning program, optimal methodology requires that SABLE be compared to both these algorithms. To demonstrate the accuracy with which SABLE superimposes proteins, I comparatively superimposed several homologous proteins utilizing SABLE, the MUSTANG structural alignment program (Konagurthu et al., 2006), and the Theseus structural superpositioning program (Theobald and Wuttke, 2006b). The results indicate that, although all three programs accurately superimpose monomeric protein pairs, only SABLE accurately and consistently superimposes three or more multimeric proteins. Specifically, MUSTANG is incapable of aligning multimeric proteins (Konagurthu et al., 2006); furthermore, Theseus inconsistently superimposes multimeric proteins and is unable to competently superimpose more than four proteins.

Similar to other conventional structural alignment programs, MUSTANG identifies secondary structures utilizing a derived contact matrix (Holm and Sander, 1993; Konagurthu et al., 2006; Ortiz et al., 2002). It then aligns the complete structure of the proteins by superimposing these secondary structures. Importantly, MUSTANG only employs secondary structures to calculate the alignment, while nonconserved regions are consequently insignificant (Konagurthu et al., 2006). The Theseus program calculates structural translations and rotations utilizing a derivative-based ML algorithm (Theobald and Wuttke, 2006b). To establish the necessary homology of the alpha carbons, Theseus requires the input of a preliminary sequence alignment. For the Theseus results contained herein, I generated

this preliminary sequence alignment utilizing the MUSCLE program (Edgar, 2004a, 2004b; Edgar and Sjolander, 2004).

Measuring Structural Superimposition

Numerous methodologies quantitatively assess the quality of superimposing protein structures (e.g., the RMSD). However, all quality assessment methodologies fundamentally measure the distance between the spatial coordinates of atoms (although some methodologies require the mathematical modification of this distance). Therefore, calculation of the spatial distance measurements requires determining the homology (matching) of amino acid residues. To compare the SABLE, MUSTANG, and Theseus quality assessment scores of superimposed proteins, correct methodology requires the derivation of homologous amino acids utilizing the superimposed protein structures. Unfortunately, although Theseus calculates numerous quality assessment scores (including the classical RMSD) for pairwise superpositions (Theobald and Wuttke, 2006b), the amino acid homology utilized to generate these calculations is derived from the preliminary sequence alignment. Therefore, calculating the most probable sequence alignment derived from superimposed protein structures necessitates an unpublished structure-dependent sequence alignment (SDSA) program called UniTS (Universal True SDSA). The UniTS program calculates the most probable SDSA utilizing both the spatial distances between homologous atoms and sequence information in structurally nonconserved regions of the superimposed proteins. Utilizing this newly generated SDSA, UniTS calculates improved, structure-based, quality assessment scores for the superimposed proteins (more information regarding the UniTS program is available in Chapter III).

Importantly, SABLE does not derive an SDSA and MUSTANG only derives a partial SDSA (based upon the secondary structures), thus preventing the calculation of quantitative assessment scores for their generated structural alignments. Incidentally, I utilized UniTS to supplement these limitations; therefore, the combination of SABLE or MUSTANG and UniTS is capable of calculating a structural alignment, a corresponding sequence alignment, and quality assessment scores for the superimposed proteins. Because UniTS calculates the SDSA and structural quality assessment scores for the superimposed protein derivations of SABLE, MUSTANG, and Theseus, the results contained herein are measured utilizing equal (structure-based) methodologies. This eliminates the influence introduced into the results by inconsistent and possibly contrasting measurement methodologies.

Conventionally, the RMSD spatial distance measurement is utilized to quantitatively assess the quality of a pairwise protein structural superposition. However, the RMSD quantity is dependent upon the number of amino acids utilized in the calculation (i.e., the length of the sequence alignment), as is evident by the RMSD equation. The dependence of the RMSD on sequence length induces incorrect methodology because UniTS calculates marginally dissimilar, and thus incomparable, numbers of amino acid matches for each trial of superimposed proteins. Therefore, to quantitatively assess the quality of protein superimpositions, I formulated two novel measurements: the mean standard deviation and the mean probability. Calculation of the mean standard deviation (σ_m) consists of averaging the individual standard deviations of each homologous atom position (see Chapter III, section "UniTS Algorithm", subsection "Multiple SDSA Quality Assessment: Mean Standard Deviation" for additional details on this calculation). I utilize individual standard deviations instead of basic spatial distances to accommodate more than two superimposed proteins.

Although σ_m is superior to the RMSD because it is length independent and accommodates more than two superimposed proteins, σ_m also continues to overweigh structurally nonconserved regions (see Chapter II, section "SABLE Algorithm", subsection "Superimposing Measurement of SABLE" for more details). To compensate for this limitation, I derived a novel mean probability (p_m) measurement:

$$p_m = \left(\prod p_{atom}\right)^{(1/n)}$$

where n is the number of homologous amino acid positions and p_{atom} is the probability of two homologous atoms utilizing a σ of 4 angstroms (as detailed in Chapter II, section "SABLE Algorithm", subsection "SABLE: Phase 3"). Because the preceding equation frequently calculates a quantity too small for standard computational precision, I augmented the equation with logarithms:

$$p_m = \left(\sum \log p_{atom}\right)^{\left(1/\log n\right)}$$

Although the p_m measurement is comparable to that employed by SABLE to calculate comparative pseudostate likelihoods, as a mean quantity, p_m is independent of utilized amino acid (or homologous atom) quantities. The quality assessments of superimposed proteins contained herein are measured utilizing both σ_m and p_m.

Monomeric Pairwise Comparison

To compare the pairwise superimposing capabilities of SABLE, MUSTANG, and Theseus, I conducted four superimposition trials; each trial represents a protein family consisting of two homologous monomers. Table 1a displays the two PDB designations, the

mean sequence length, and the sequence identity characterizing each protein family, while Table 1b lists the species of each PDB designation (Berman et al., 2000).

Table 1a. Protein family characteristics for pairwise monomeric comparison

Protein Family	First Protein PDB	Second Protein PDB	Mean Length	Sequence Identity[b]
Protein Kinase C[c]	1BDY[a]	2ENJ	127	47.1
Isocitrate Dehydrogenase	1T09[a]	1XGV[a]	422	20.8
Pectate Lyase	1PLU	2BSP	375.5	20.3
Polygalacturonase	1CZF[a]	1HG8	342	40.1

[a]Chain A of the protein.
[b]The sequence identity was calculated utilizing the MEGA5 sequence analysis software (Tamura et al., 2011).
[c]This is the C2 domain of the protein kinase C protein.

Table 1b. PDB designation species for pairwise monomeric comparison

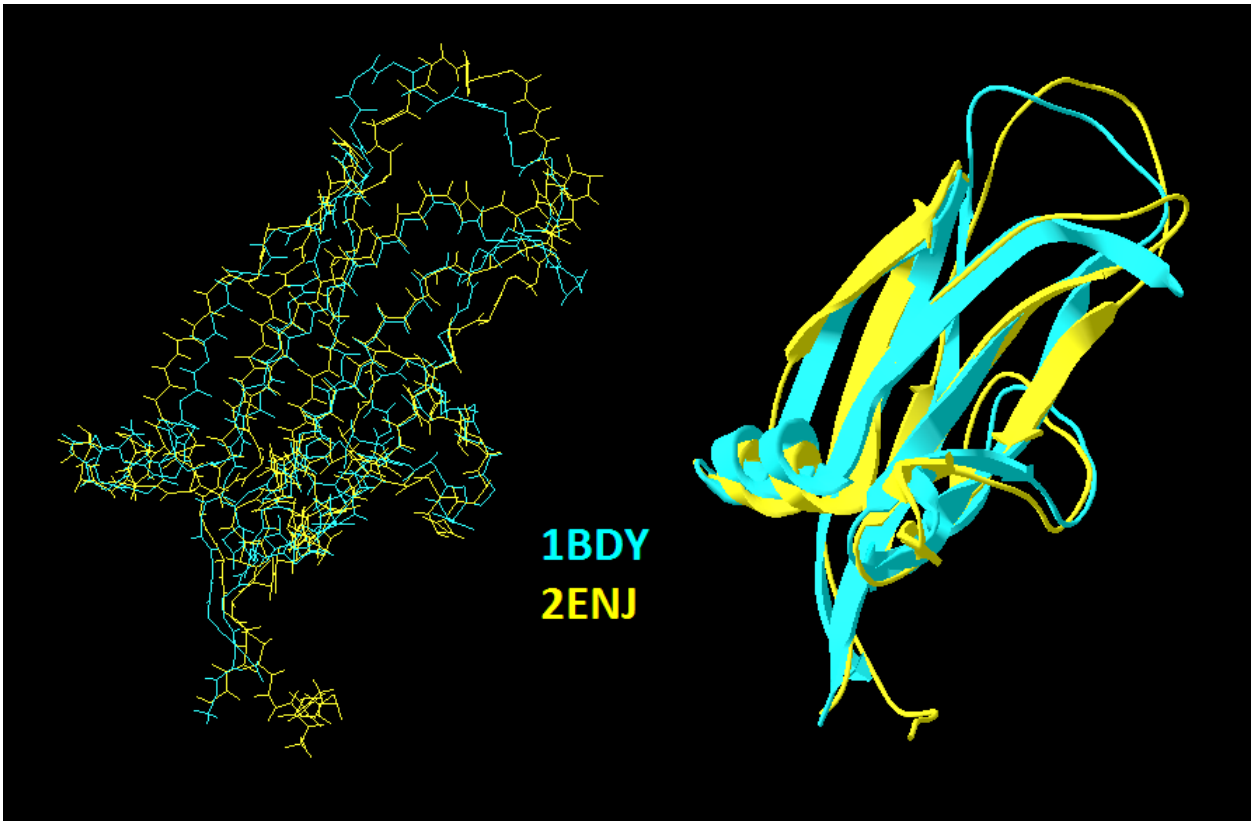
PDB Designations	First Protein Species	Common Name/Type
1BDY[a]	*Rattus norvegicus*	Brown Rat
2ENJ	*Homo sapiens*	Human
1T09[a]	*Homo sapiens*	Human
1XGV[a]	*Aeropyrum pernix*	Archea
1PLU	*Erwinia chysanthemi*	Bacteria
2BSP	*Bacillus subtilis*	Bacteria
1CZF[a]	*Aspergillus niger*	Fungi
1HG8	*Fusarium verticillioides*	Fungi

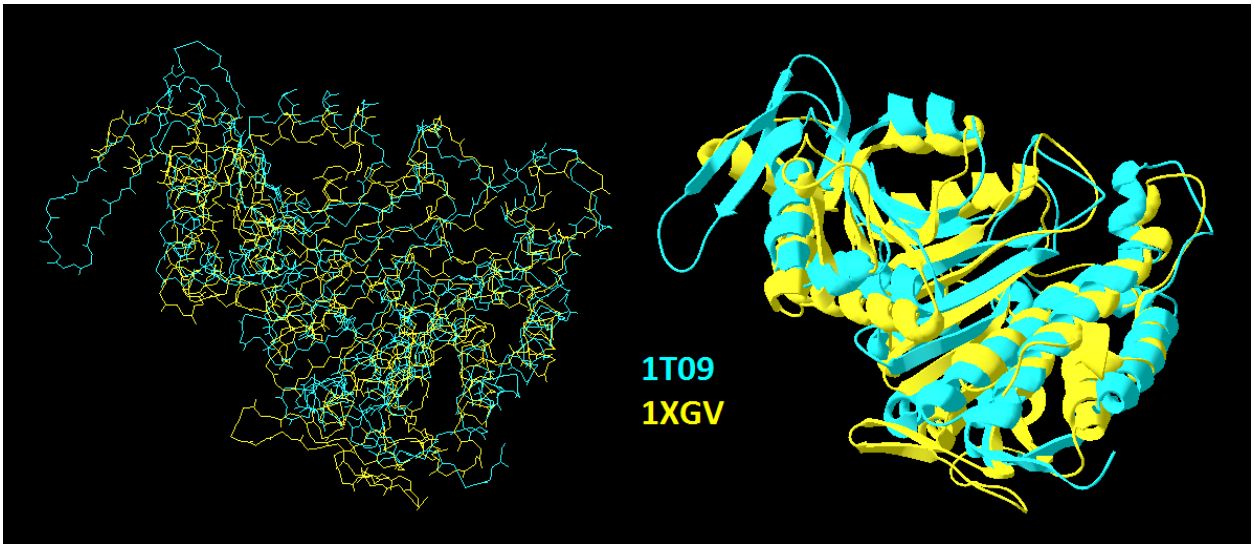
[a]Chain A of the protein.

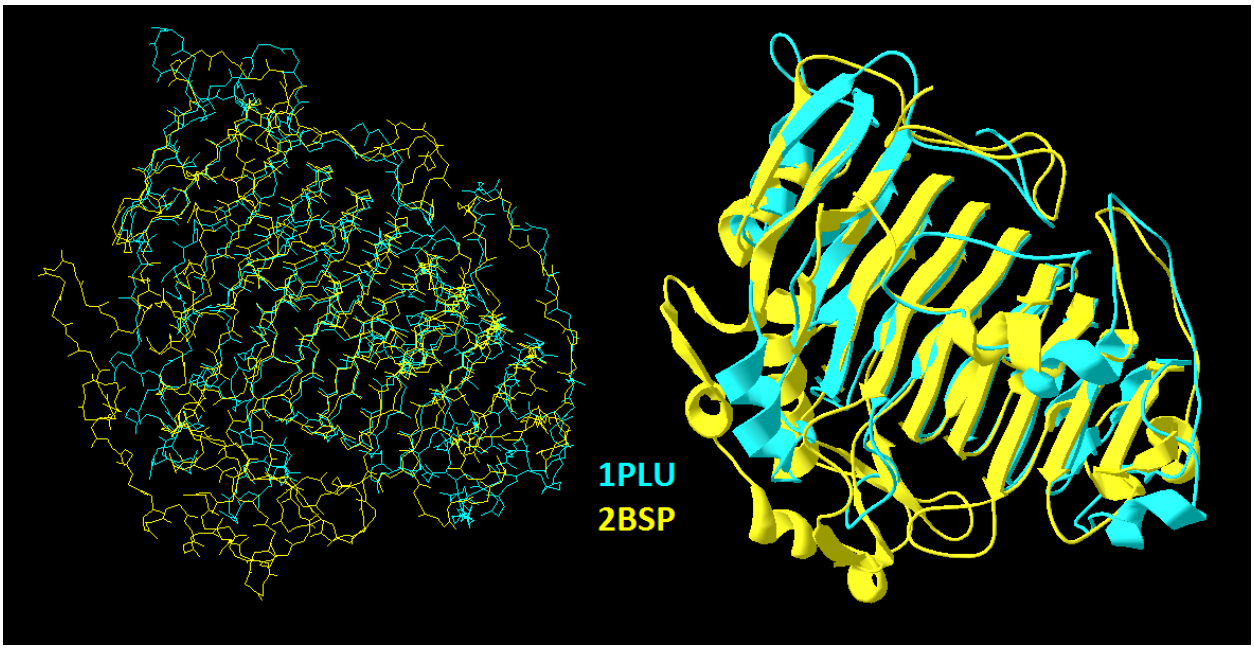
I chose each protein family either arbitrarily or for collaborative purposes. The C2 domain of protein kinase C (PKC) is present in the conventional and the novel subfamilies (but not present in the atypical subfamily). Although nonfunctional in the novel subfamily, the conventional subfamily C2 domain permits the binding of Ca^{2+}, one of the activation ligands of PKC (Voet and Voet, 2004). The PKC structures included herein (1BDY and 2ENJ) are both from the novel subfamily (Berman et al., 2000). Isocitrate dehydrogenase is a citric acid cycle protein that utilizes a decarboxylation reaction to convert isocitrate and NAD^+ (nicotinamide adenine dinucleotide) to α-ketoglutarate and NADH (reduced form of NAD) (Voet and Voet, 2004). Both pectate lyase and polygalacturonase catalyze the degradation of plant cell walls. Pectate lyase catalyzes the eliminative cleavage of pectin in cell walls, while polygalacturonase depolymerizes the pectin component polygalacturonic acid (Marín-

Rodríquez et al., 2002; Voet and Voet, 2004). Both protein families permit the softening of fruits during the ripening process and are utilized by microbes to cause diseases in plants (Marín-Rodríquez et al., 2002; Voet and Voet, 2004)

Figure 2 contains images of the SABLE structural alignment for each protein family. Images of the Theseus and MUSTANG superimpositions are not shown because they possess a similar spatial resemblance to those generated by SABLE.




1T09
1XGV


1PLU
2BSP

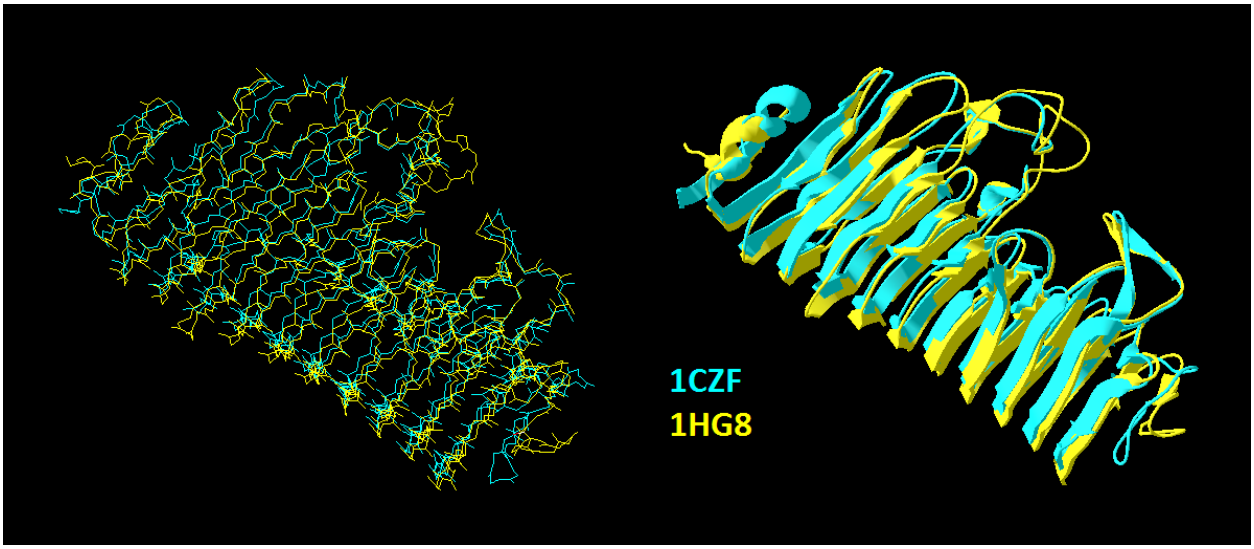


Figure 2. Atomic (left) and ribbon (right) image formats of each pairwise structure alignment generated by SABLE. The protein families represented include: (a) the C2 domain of protein kinase C, (b) isocitrate dehydrogenase, (c) pectate lyase, and (d) polygalacturonase. The PDB designations of each input protein are colored appropriately. Graphical imaging was performed utilizing Swiss-PdbViewer (Guex and Peitsch, 1997).

Tables 2 and 3 present the σ_m and p_m quantities of each superimposed protein family as calculated utilizing SABLE, MUSTANG, and Theseus. As previously stated, each time one of the aforementioned programs superimposed a protein family, I utilized UniTS to derive a consequent SDSA. I then utilized this resultant SDSA to determine the amino acid matches required to calculate σ_m and p_m. In Table 2, because the quantities are measured in angstroms, a lesser quantity indicates superiorly superimposed proteins; in Table 3, a greater probability equals a superior superimposition.

Table 2. Mean standard deviation (σ_m) of protein families

Protein Name	SABLE σ_m	MUSTANG σ_m	Theseus σ_m
Protein Kinase C	0.95 Å[a]	0.87 Å	0.92 Å
Isocitrate Dehydrogenase	2.47 Å	1.89 Å	2.17 Å
Pectate Lyase	1.98 Å	2.78 Å	1.70 Å
Polygalacturonase	0.54 Å	0.56 Å	0.54 Å

[a]The colors indicate the relative quality of the values. Green indicates the most superior quantity of the three; yellow the intermediate; and red the most inferior.

Table 3. Mean probability (p_m) of protein families

Protein Name	SABLE p_m	MUSTANG p_m	Theseus p_m
Protein Kinase C	3.989[a]	4.010	3.983
Isocitrate Dehydrogenase	3.106	3.444	3.160
Pectate Lyase	3.185	2.824	3.361
Polygalacturonase	3.771	3.768	3.770

[a]The colors indicate the relative quality of the values. Green indicates the most superior quantity of the three; yellow the intermediate; and red the most inferior.

The results of Tables 2 and 3 illustrate sporadic capability by each program. Therefore, no program consistently produces significantly superior structural results.

To compare the quality of the SDSAs generated utilizing the SABLE alignments to those generated utilizing the MUSTANG alignments and Theseus superpositions, I calculated conventional sequence alignment log-odds scores for each SDSA generated (Table 4). A greater log-odds score indicates a more probable sequence alignment as calculated

utilizing amino acid identity and similarity (information regarding the log-odds score calculation is described in Chapter III, section "UniTS Results", subsection "UniTS Compared to Chimera").

Table 4. Log-odds scores for each protein family

Protein Name	SABLE Log-odds	MUSTANG Log-odds	Theseus Log-odds
Protein Kinase C	314.4[a]	284.5	314.4
Isocitrate Dehydrogenase	-179.8	-180.4	-126.7
Pectate Lyase	68.2	-136.2	58.0
Polygalacturonase	677.2	673.9	677.2

[a]The colors indicate the relative quality of the values. Green indicates the most superior quantity of the three; yellow the intermediate; and red the most inferior.

The results illustrated in Table 4 indicate that UniTS generated the most probable SDSAs for the superimposed proteins derived utilizing SABLE and Theseus. Specifically, both SABLE and Theseus produced identical PKC (C2 domain) and polygalacturonase SDSAs. Furthermore, of the two remaining protein families, each program derived a single superior log-odds score. Importantly, the log-odds measurement is the same measurement employed by conventional sequence alignment algorithms. Therefore, because Theseus superpositions proteins utilizing a preliminary sequence alignment (which by default exhibts the maximum log-odds score), the proteins are superimposed indirectly utilizing a maximum log-odds score. However, the log-odds scores generated by the SABLE alignments contain no indirect sequence bias.

Despite the insignificant variation exhibited by the SABLE and Theseus SDSAs, Table 4 indicates that the SDSAs generated utilizing the MUSTANG alignments are significantly inferior. Consequently, although the MUSTANG quantities in Tables 2 and 3 indicate equally superimposed proteins to those of SABLE and Theseus, these quantities were calculated utilizing inferior SDSAs. Therefore, I recalculated each σ_m and p_m utilizing amino acid matches derived from the most probable SDSA for each protein family (Tables 5 and 6).

Table 5. Mean standard deviations (σ_m) of protein families calculated utilizing the SABLE and/or Theseus SDSAs

Protein Name	SABLE σ_m	MUSTANG σ_m	Theseus σ_m	SDSA From…
Protein Kinase C	0.95 Å[a]	0.95 Å	0.92 Å	SABLE/Theseus
Isocitrate Dehydrogenase	2.40 Å	2.18 Å	2.17 Å	Theseus
Pectate Lyase	1.98 Å	2.65 Å	1.95 Å	SABLE
Polygalacturonase	0.54 Å	0.55 Å	0.54 Å	SABLE/Theseus

[a]The colors indicate the relative quality of the values. Green indicates the most superior quantity of the three; yellow the intermediate; and red the most inferior.

Table 6. Mean probability (p_m) of protein families calculated utilizing the SABLE and/or Theseus SDSAs

Protein Name	SABLE p_m	MUSTANG p_m	Theseus p_m	SDSA From…
Protein Kinase C	3.989[a]	3.988	3.983	SABLE/Theseus
Isocitrate Dehydrogenase	3.142	3.191	3.160	Theseus
Pectate Lyase	3.185	3.072	3.164	SABLE
Polygalacturonase	3.771	3.770	3.770	SABLE/Theseus

[a]The colors indicate the relative quality of the values. Green indicates the most superior quantity of the three; yellow the intermediate; and red the most inferior.

Utilizing the most probable SDSAs, the comparative capabilities of SABLE remain approximately unchanged in Table 5; however, Table 6 illustrates that SABLE produces a superior p_m in nearly all protein families. Because p_m is a superior indicator of superimposition quality compared to σ_m, the results elucidated in Table 5 are justifiably more legitimate. Therefore, although the differences between the p_m quantities are likely inconsequential, SABLE predominantly produces superior superimposition and SDSA results.

I attribute the inferior alignment of isocitrate dehydrogenase to the sporadic positions of the composing secondary structures. The alignment accuracy of the regions containing secondary structures was likely compromised to compensate for the intermittent regions; however, algorithms utilizing a contact matrix ignore these intermittent regions. The significant discrepancy revealed in Table 3 between the p_m generated utilizing MUSTANG and those generated utilizing SABLE and Theseus further substantiates this explanation (although this discrepancy is reduced in Table 6). Note however, that the sporadic positions

of the secondary structures benefitted MUSTANG for isocitrate dehydrogenase because these structures superimposed optimally; however, if the secondary structures are unable to optimally superimpose, the inability of MUSTANG to compensate utilizing other protein regions will result in inferiorly superimposed structures.

Importantly, SABLE, MUSTANG, and Theseus all satisfactorily superimposed each protein family. Consequently, although Tables 2, 3, 5, and 6 compare the pairwise performances of the three programs, the p_m differential disparities equate to divergent RMSDs of only a fraction of an angstrom. However, superimposing more complex proteins will elucidate the comparative limitations of conventional structural alignment and superpositioning programs such as MUSTANG and Theseus.

Multisubunit Pairwise Comparison

To elucidate the multisubunit capabilities of SABLE, I superimposed the polygalacturonase proteins a second time utilizing a homodimeric 1CZF[5] (Chains A and B) and the original monomeric 1HG8. Because MUSTANG is unable to align multisubunit proteins, I compared the multisubunit capabilities of only SABLE and Theseus. For the Theseus superposition of the polygalacturonase proteins, UniTS calculated a $\sigma_m = 0.63$ angstroms and $p_m = 3.764$; furthermore, UniTS calculated $\sigma_m = 0.50$ angstroms and $p_m = 3.772$ for the SABLE alignment. Although the quality assessment scores for each program are comparable, the addition of a single subunit to a simple pairwise superimposition caused the quality of the Theseus superposition to decline. However, the scores of SABLE remained relatively consistent despite the addition of the subunit (as indicated by comparing the

[5] 1CZF is naturally a monomer. However, the 1CZF protein structure was derived utilizing x-ray crystallography and the asymmetric unit contains two proteins (Chains A and B). Therefore, the "homodimeric" 1CZF is actually two polygalacturonase proteins in the same asymmetric unit.

aforementioned quantities to those in Tables 5 and 6). Consequently, although adding a single subunit minimally decreased the quality of the Theseus superposition, the accumulation of numerous complexities (i.e., introducing additional subunits, heterogeneous numbers of subunits, and/or increasing the number of input proteins) would likely cumulatively result in a significant discrepancy. This significant difference in quality demonstrates both the inevitable inability of Theseus to accommodate more complex multisubunit superpositions and the proficiency of SABLE at segregating and differentiating the calculations of independent polypeptide chains.

Two reasons contribute to the inability of Theseus to superposition multisubunit proteins: First, MUSCLE is incapable of differentiating polypeptide chains while generating a sequence alignment. Second, the Theseus algorithm is dependent upon input proteins possessing similar polypeptide sequence lengths[6]; SABLE, however, is not limited by chain length differentiation. Furthermore, because the SABLE algorithm properly differentiates and aligns polypeptide chains, it is limited by neither multiple subunits nor heterogeneous numbers of subunits.

Multiple Alignment Comparison

To demonstrate the multiple structural alignment capability of SABLE, I superimposed three protein assemblages utilizing SABLE, MUSTANG, and Theseus. Each

[6] Because Theseus utilizes a preliminary sequence alignment to derive homologous atoms, similar monomeric sequence lengths prevent the occurrence of extreme atomic mismatches. Therefore, despite an inadequate input sequence alignment (i.e., a sequence alignment exhibiting a sequence identity of less than the conventional 25% threshold (Rost, 1999), homologous atom determination will remain approximately correct. However, the introduction of a single dimeric protein prevents the two input proteins from possessing comparable sequence lengths. These divergent sequence lengths preclude Theseus from compensating for an inadequate preliminary sequence alignment.

protein assemblage consists of multiple (i.e., greater than two) homologous proteins. As with the previous alignments, I assessed the quality of each superimposed protein assemblage utilizing both the p_m and σ_m measurements.

The initial two protein assemblages both consist of four homologous proteins: two pectate lyase and two polygalacturonase proteins (see Table 1 for specific PDB designations). One assemblage consists of all monomers (including the monomeric version of the 1CZF protein), while the other assemblage includes all monomers except the homodimeric version of 1CZF. Figures 3 and 4 graphically illustrate the SABLE alignment of each superimposed assembly of proteins.

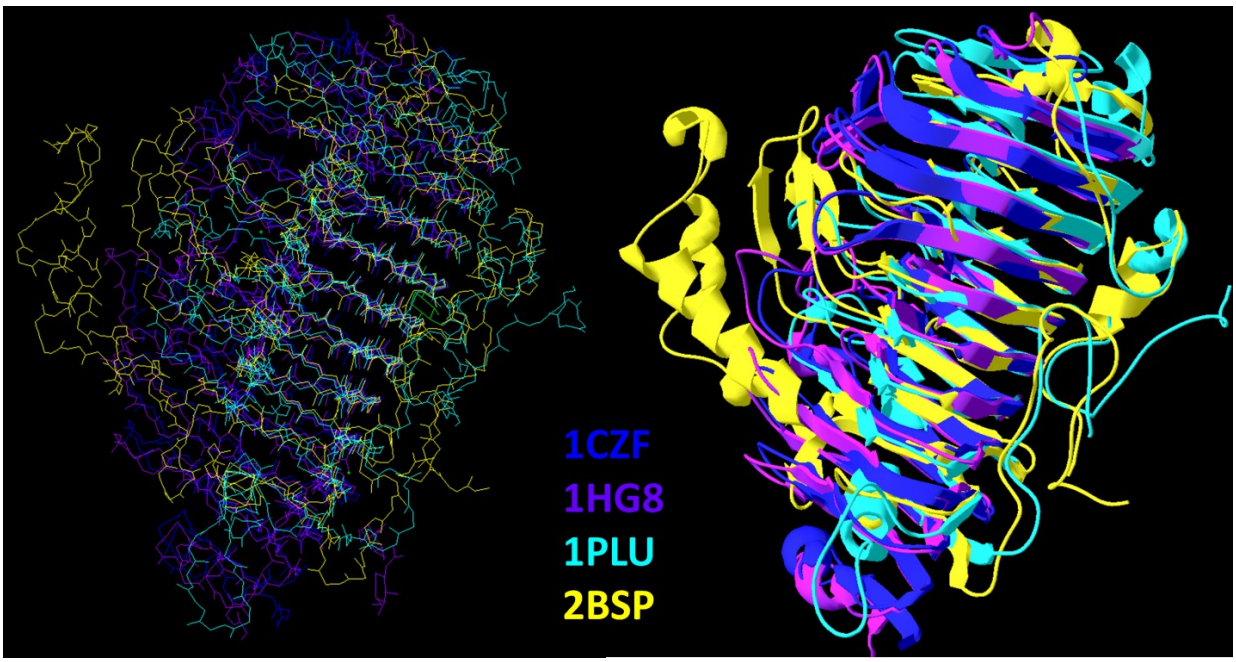


Figure 3. Atomic (left) and ribbon (right) image formats of a multiple structure alignment generated by SABLE. This structural alignment incorporates the monomeric version of the polygalacturonase protein possessing the PDB designation of 1CZF The PDB designations of each input protein are colored appropriately. Graphical imaging was performed utilizing Swiss-PdbViewer (Guex and Peitsch, 1997).

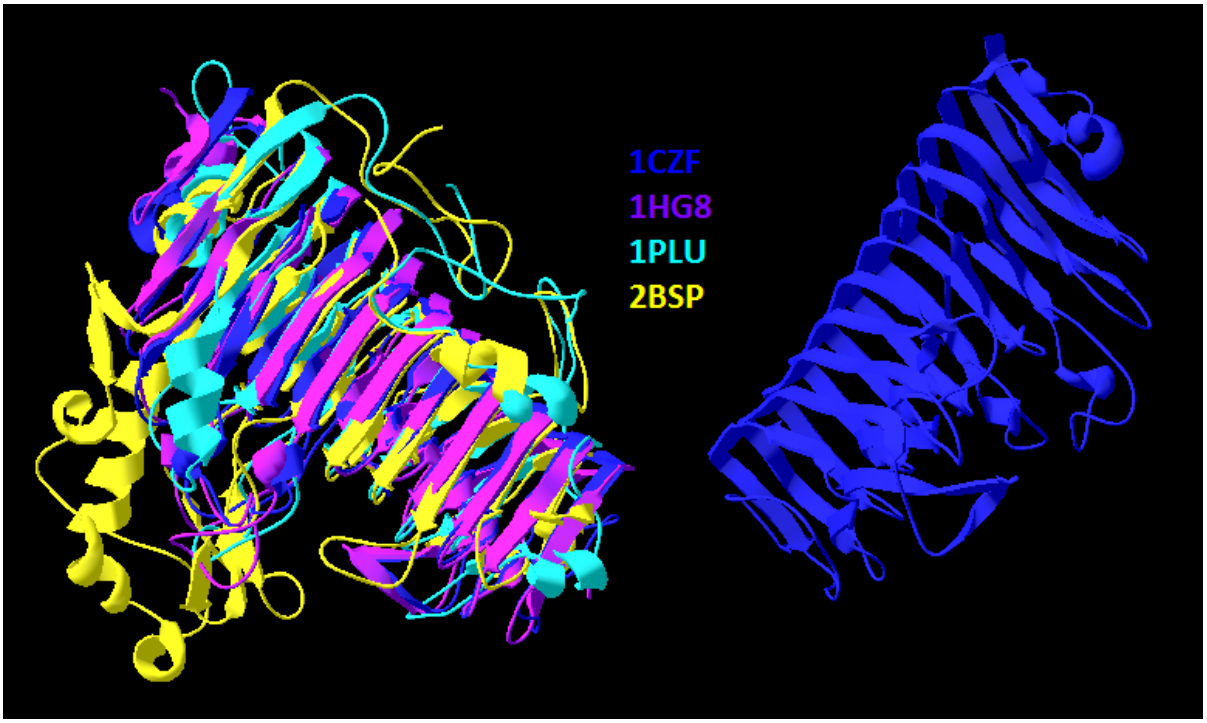


Figure 4. A ribbon image format of a multiple structure alignment generated by SABLE. This structural alignment incorporates the homodimeric version of the polygalacturonase protein possessing the PDB designation of 1CZF. The PDB designations of each input protein are colored appropriately. Note that because the other proteins are monomers, they only align to one of the 1CZF subunits. Graphical imaging was performed utilizing Swiss-PdbViewer (Guex and Peitsch, 1997).

Utilizing UniTS, I generated p_m and σ_m quantities for each assemblage of superimposed proteins as calculated by SABLE, MUSTANG, and Theseus (Table 7). The results indicate that SABLE superiorly superimposes both assemblages of proteins. Although the disparity between the monomeric proteins superimposed by SABLE and those superimposed by Theseus is likely insignificant, the additional of the homodimeric 1CZF protein increases this disparity to significant proportions. As initially demonstrated in the "Multisubunit Pairwise Comparison" subsection, the inability of Theseus to superimpose proteins composed of multiple or inconsistent subunits demonstrates its lack of versatility. Comparatively, MUSTANG unsuccessfully superimposed the quad polygalacturonase/pectate lyase

assemblages. In addition to its inability to superimpose input monomers to the homodimeric 1CZF, MUSTANG derived significantly inferior results to those of SABLE and Theseus when aligning the four polygalacturonase/pectate lyase monomers.

Table 7. Mean probabilities (p_m) and mean standard deviations (σ_m) for each assemblage of proteins

Assemblage	SABLE	MUSTANG	Theseus
Quad Alignment (1CZF monomer) p_m	3.608[a]	2.878	3.589
Quad Alignment (1CZF monomer) σ_m	3.31 Å	4.53 Å	3.36 Å
Quad Alignment (1CZF homodimer) p_m	3.492	NA[b]	3.294
Quad Alignment (1CZF homodimer) σ_m	3.41 Å	NA[b]	4.83 Å

[a]The colors indicate the relative quality of the values. Green indicates the most superior quantity of the three; yellow the intermediate; and red the most inferior.
[b]MUSTANG is unable to superimpose protein possessing multiple subunits.

The final assemblage consists of five monomeric polygalacturonase proteins. Two of these are published in the PDB (monomeric 1CZF and 2IQ7 [Berman et al., 2000]); two are unpublished but have reserved designations in the PDB (1ZEU and 1ZFW); the fifth is an unpublished tomato polygalacturonase that I designated as TOMA until a formal designation is assigned.[7] Figure 5 illustrates these five proteins as superimposed utilizing SABLE.

[7] The three unpublished protein structures were provided courtesy of Marilyn Yoder, Ph.D., at the Division of Cell Biology and Biophysics, School of Biological Sciences, University of Missouri-Kansas City.

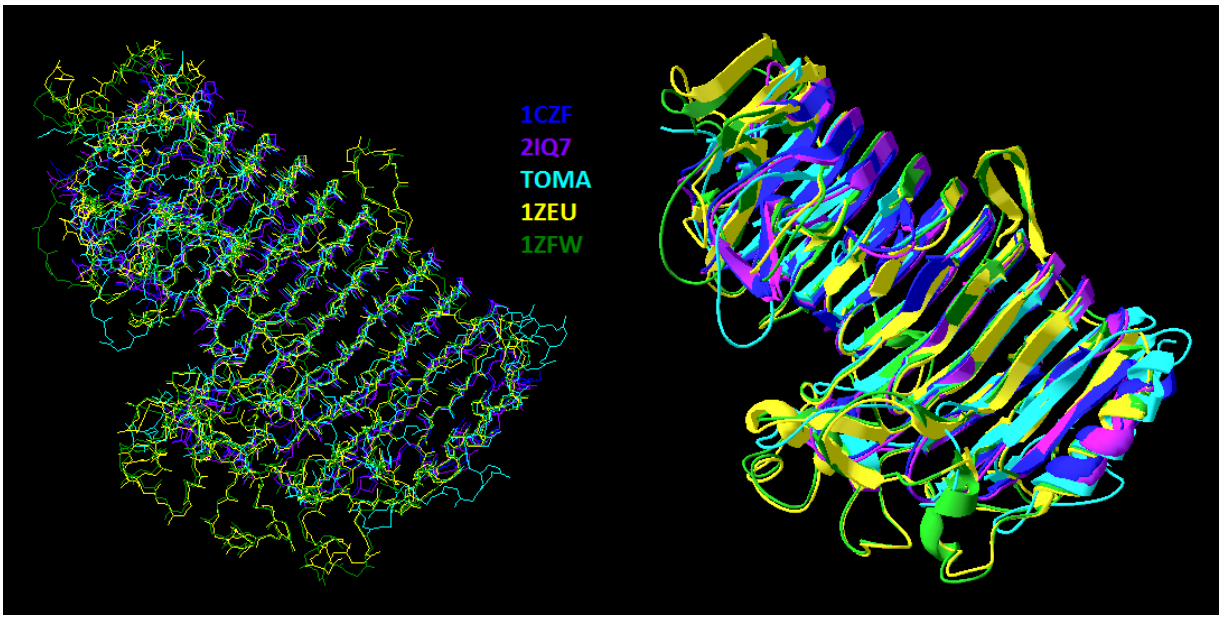


Figure 5. Atomic (left) and ribbon (right) image formats of a five-protein multiple structure alignment generated by SABLE. The PDB designations of each input protein are colored appropriately. Graphical imaging was performed utilizing Swiss-PdbViewer (Guex and Peitsch, 1997).

The quality assessment scores derived from UniTS for the proteins superimposed by SABLE are σ_m = 1.47 angstroms and p_m = 3.725. Furthermore, these scores derived utilizing the proteins superimposed by MUSTANG are σ_m = 1.38 angstroms and p_m = 3.730. Unfortunately, Theseus was unable to adequately superimpose the proteins. Throughout numerous superpositioning trials not included in this dissertation, Theseus repeatedly and inconsistently demonstrated an inability to adequately superimpose more than four proteins. This deficiency likely results from an error in the program rather than an algorithmic error. For this penta-polygalacturonase alignment, MUSTANG superiorly superimposes the proteins compared to SABLE (although the assessment score disparities are likely insignificant).

Although SABLE consistently aligned all three assemblages of proteins, the quality of the two MUSTANG alignments is inconsistent. The first assemblage superimposed by MUSTANG was significantly inferior to those of both SABLE and Theseus, while the quality of the MUSTANG alignment was slightly superior to that of SABLE for the third assemblage. To further complicate this demonstrated inconsistency, the proteins utilized in both assemblages are structurally similar. The inconsistency of the MUSTANG alignments possibly results from the additional and inconsistent secondary structures present in the monomeric quad polygalacturonase/pectate lyase superimposition (not including the beta-helix) as illustrated in Figure 3. The penta-polygalacturonase superimposition illustrated in Figure 5, however, contains only negligible secondary structures in addition to the prominent beta-helix. The two aforementioned examples demonstrate the dependence of MUSTANG on secondary structures, its inability to align heterogeneous secondary structures, and the inconsistent results generated from these deficiencies. Furthermore, the three multiple superimpositions demonstrate the capability of SABLE to consistently derive quality results despite the complexities of considerable numbers of input proteins, multiple or heterogeneous numbers of subunits, or inconsistent secondary structures.

Discussion of SABLE Results

SABLE combines the versatility of a structural alignment program with the accuracy and comprehensiveness of a superpositioning program by implementing a ML algorithm and a novel probability scoring algorithm. Comparable to conventional structural alignment algorithms, it implements structural information to superimpose protein structures rather than deriving information from amino acid sequences. The versatility of SABLE permits it to superimpose multiple input proteins, each comprising homo- or heterogeneous numbers of multiple subunits. However, SABLE is superior to conventional structural alignment algorithms because it implements the accuracy of a structural superpositioning algorithm, it is not limited by secondary structures in a contact matrix (Holm and Sander, 1993; Konagurthu et al., 2006; Ortiz et al., 2002), and it generates output files containing spatial coordinates.

Although SABLE does not require a preliminary sequence alignment, its functioning is comparable to that of a superpositioning program. The results presented herein indicate that the accuracy with which SABLE superimposes two monomeric proteins is equal to that of the Theseus superpositioning program. Furthermore, for increasingly complex alignments (i.e., increasing the number of input proteins, increasing the number of chains, or by incorporating heterogeneous numbers of chains), the flexibility of SABLE permits the generation of significantly more accurate superpositions than those generated by Theseus. This versatility and accuracy allows SABLE to calculate structural alignments with minimal human intervention, reducing the need to curate results and increasing output through automation. Consequently, SABLE can automate applications such as generating a mean structure from divergent homologous proteins or modeling ligand binding.

CHAPTER III

UNITS: UNIVERSAL TRUE SDSA (STRUCTURE-DEPENDENT SEQUENCE ALIGNMENT)

Introduction to Structure-dependent Sequence Alignments

Although protein structure is a superior indicator of protein homology because it is more evolutionarily conserved (Kim and Lee, 2007; Marti-Renom et al., 2000), utilization of protein sequences continues to be the primary method for determining protein homology due to sequence abundance, relatively inexpensive generation, and comparative algorithmic simplicity. However, as both computational power and the number of solved protein structures increase, the influence of structural information in protein biology is increasing. Evolutionary relationships and functional correlations between homologous proteins are increasingly determined utilizing protein structural alignment and superpositioning software instead of an exclusive reliance on sequence alignment software. While protein structural alignment and superpositioning software can calculate structural homology, the inability of this software to calculate an accurate corresponding sequence alignment limits its utilization. Although some algorithms either directly (e.g., the Chimera Match=>Align function [Meng et al., 2006; Pettersen et al., 2004]) or indirectly (e.g., inverse folding and structural alignment) attempt to calculate protein sequence homology utilizing structural information, no algorithmic solution permits the derivation of an accurate alignment while utilizing the complete protein (including nonhomologous regions).

Current SDSA Limitations

Inverse folding structure-dependent sequence alignment (SDSA) algorithms, precursors to protein structure prediction methods such as threading or homology modeling, attempt to align the amino acid sequence of one protein to the sequence of another with a known structure (Bowie et al., 1991; Hong et al., 2010; Yang, 2002). To align sequences possessing less than thirty percent sequence identity (Rost, 1999), these SDSA programs utilize profile-based sequencing techniques (Bowie et al., 1991; Edgar and Sjolander, 2004; Hong et al., 2010). That is, they generate amino acid profiles based upon structural information to assist with the sequence alignment. Importantly, these profile-based SDSA algorithms continue to utilize conventional sequence alignment algorithms (Edgar and Sjolander, 2004). The amino acid profiles generated by structural information only supplement the conventional sequence alignment; they are not truly dependent on this structural information (Bowie et al., 1991; Kuzlemko et al., 2011).

After superimposing protein structures, protein structural alignment algorithms[8] must calculate a sequence alignment utilizing this structural alignment. Unfortunately, the sequence alignments generated by structural alignment algorithms constitute only a fraction of the total protein. Consequent of utilizing a contact matrix, only those amino acids contained within matching submatrices are sequentially aligned (Holm and Sander, 1993; Konagurthu et al., 2006; Ortiz et al., 2002). Therefore, the homologous proteins are sequentially aligned intermittently rather than universally, resulting in an incomplete

[8] Protein superpositioning algorithms require a preliminary sequence alignment to determine amino acid matching and function to superimpose the structures. Protein structure alignment algorithms do not require a preliminary sequence alignment and function to identify and align evolutionarily homologous regions of protein structures (Gibas and Jambeck, 2001; Ortiz et al., 2002). Although both types of algorithms ultimately superimpose protein structures, they require different input and utilize different methodologies.

sequence alignment. In addition to the decreased number of amino acid matches preventing the sequence alignment of structurally nonconserved regions, it also prevents the calculation of accurate structural alignment quality assessment scores (e.g., RMSD).

Neither inverse folding SDSAs nor structural alignment algorithms are designed to specifically calculate a sequence alignment from superimposed protein structures. However, the Match=>Align function of the University of California-San Francisco's Chimera protein structure visualization and modeling program is designed for this purpose (Pettersen et al., 2004). Unfortunately, the Match=>Align function is unsophisticated and possesses numerous algorithmic deficiencies. First, while matching amino acids from homologous chains, it maintains the residue order of the polypeptide chains (i.e., it prevents the amino acids from becoming disordered) by serially matching them (Meng et al., 2006). This heuristic serial matching method prevents the algorithm from calculating the best SDSA for the entirety of the proteins. Second, similar to many structural alignment programs (Holm and Sander, 1993; Ortiz et al., 2002), the Match=>Align function of Chimera is unable to directly match amino acids whose spatial distance exceeds a predetermined distance threshold. Instead, it utilizes arbitrary scores such as gap penalties and negative scores to match amino acids with a spatial distance greater than the threshold distance (Meng et al., 2006). These arbitrary scores are inconsistent with the utilization of structure to calculate sequence matches and thus prevent the determination of an accurate sequence homology for structurally nonconserved regions.

The UniTS Solution

The Universal True SDSA, or UniTS, program calculates the most probable sequence alignment derived from multiple superimposed protein structures. Although designed to neither resolve the inverse protein folding problem (as are residue profile-based SDSA algorithms) nor superimpose protein structures, UniTS compensates for the aforementioned limitations and deficiencies inherent in residue profile-based SDSA programs, structural alignment programs, and other spatial SDSA programs such as Chimera. If superimposed protein structures are available, UniTS is *truly* structure-dependent because it derives the SDSA utilizing spatial coordinates instead of residue profiles. Additionally, compared to the incomplete or partial sequence alignment generated by a structural alignment algorithm, UniTS calculates a *universal* sequence alignment constituting information from the entire protein.

Although the Match=>Align function of the Chimera program also derives a SDSA from superimposed protein structures utilizing atomic proximity (Meng et al., 2006), UniTS calculates the sequence homology of structurally nonconserved regions utilizing sequence information. Predicated on the evolutionary model, this method is biologically superior to the utilization of arbitrary scores. Furthermore, UniTS calculates residue matches comprehensively based upon the totality of the proteins instead of the heuristic serial matching performed by Match=>Align.

The consequent SDSA derived by UniTS permits the calculation of improved quality assessment scores (e.g., RMSD) for the superimposed proteins relative to those calculated exclusively by structural alignment and superpositioning algorithms. Unfortunately, as aforementioned, protein structure alignment algorithms derive a partial sequence alignment;

additionally, superpositioning algorithms require an input sequence alignment that is derived utilizing a conventional sequence-based alignment program (Theobald and Wuttke, 2006b). Therefore, neither the structure alignment nor superpositioning algorithms derive an accurate sequence alignment utilizing structural information. Consequently, both algorithms utilize inadequate sequence alignments to calculate quality assessment scores for the superimposed protein structures. In contrast, after proteins have been superimposed, UniTS can modify and improve both the sequence alignment and the quality scores.

UniTS Algorithm

Pairwise SDSA

Given two structurally aligned proteins, the pairwise SDSA algorithm of the UniTS program will generate a sequence alignment based upon the structural alignment. The simplest and most intuitive SDSA algorithmic solution would calculate the distances between all **opposing** alpha carbons (i.e., alpha carbons located in different proteins). This algorithm would then consider two opposing amino acids to be a **structural match** if the distance between them is less than a predetermined distance threshold (four to five angstroms in many programs [Holm and Sander, 1993; Ortiz et al., 2002; Meng et al., 2006; Pettersen et al., 2004]). Unfortunately, this algorithmic solution is problematic despite being simple and intuitive.

The first problem with the aforementioned algorithm is the possibility of a single amino acid matching multiple opposing amino acids. If the multiple opposing amino acids are adjacent to each other, any one of them can structurally match to the single amino acid. Furthermore, if the multiple amino acids are remote or nonadjacent, it is possible for the amino acids to match in an incorrect sequence order (as detailed in Figure 6).

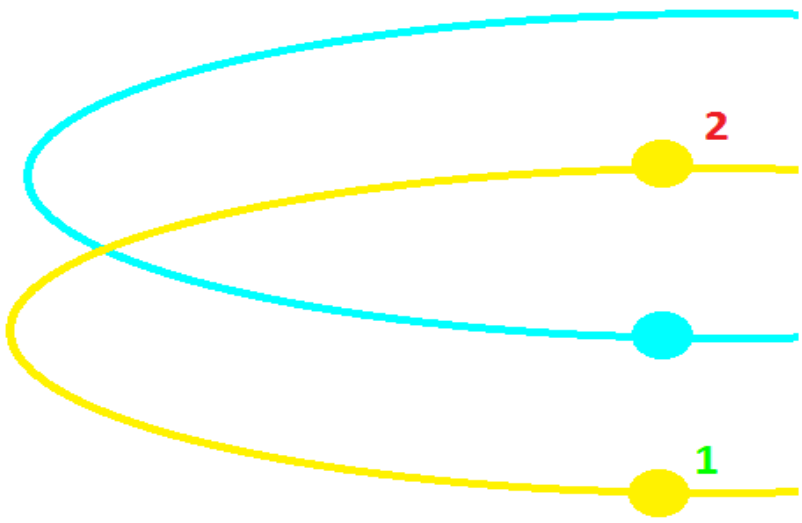


Figure 6. Disarranged amino acid matches illustrated utilizing the homologous loops of two protein chains (Yellow and Cyan) with dots representing the alpha carbons of noteworthy amino acids. The Cyan Amino Acid and the Yellow Amino Acid 1 are truly homologous and calculated to be a structural match. However, the Cyan Amino Acid also structurally matches to Yellow Amino Acid 2 due to their close proximity. If the remaining amino acids in the loop are matched correctly, the Yellow Amino Acid 2 amino acid will be disarranged in the amino acid sequence.

The second problem with this algorithm regards the handling of singular omega loops. As detailed in Figure 7, exclusively utilizing spatial coordinates in structurally divergent regions possessing random indels (amino acid insertions or deletions) prevents the determination of amino acid homology. Therefore, determining the SDSA is impossible in protein regions possessing divergent structural alignments.

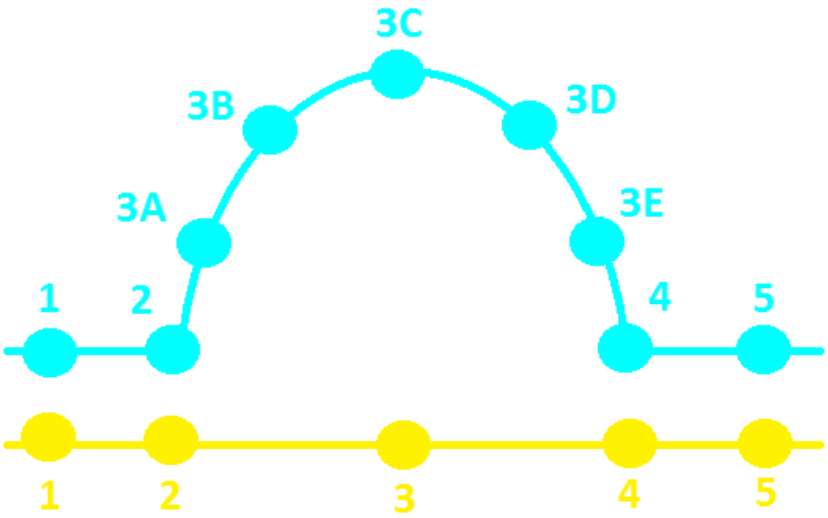


Figure 7. The omega loop problem illustrated utilizing two homologous protein chains with dots representing the alpha carbons of noteworthy amino acids. Amino Acids 1, 2, 4, and 5 for both proteins will be structurally matched. However, to which amino acid composing the omega loop (3A, 3B, 3C, 3D, or 3E) will Amino Acid 3 match? Assuming three of the four amino acids composing the omega loop were evolutionarily inserted (as opposed to the loop being deleted in the opposing protein), the original homologous amino acid can be any one of the loop amino acids. Conversely, utilizing only structural coordinates, the amino acid opposing the omega loop can be homologous to any of the amino acids composing the loop.

The pairwise SDSA algorithm of the UniTS program proposed herein determines pairwise structural matches similarly to the aforementioned algorithm. UniTS considers two opposing amino acids to be structurally matched if the distance between the alpha carbon of each amino acid is less than three angstroms. The algorithm uses the distance of three angstroms because it is approximately the maximum distance that prevents the frequent occurrence of an amino acid matching multiple opposing amino acids. Note, however, that multiple and disarranged matches can still emerge and their occurrence must be resolved. Therefore, following the calculation of structural matches, the pairwise SDSA algorithm utilizes a sorting algorithm to resolve multiple and disarranged matches.

The sorting algorithm orders a list of unordered amino acid positions by removing any position that obstructs the correct order. Positions are removed based upon the distance

from their ideal ordered index (Appendix A). Any position possessing multiple possible matches is input into the sorting algorithm as an element containing an "or" statement. However, the sorting algorithm continues to calculate an index distance for each possible match and removes them accordingly. Importantly, although the sorting algorithm can resolve multiple matches in which the opposing amino acids are not adjacent, it may be unable to resolve matches containing adjacent opposing amino acids. Therefore, if an amino acid position continues to match multiple opposing amino acids upon completion of the sorting algorithm, the algorithm will reject this position as a structural match. That is, UniTS is unable to structurally match this position based exclusively upon structural information. Instead, UniTS will resolve the match utilizing the same methodology it uses to align the remaining unmatched positions.

Although the sorting algorithm resolves matching multiple and disarranged residues, amino acids located in highly divergent regions of the structural alignment remain unmatched. A **divergent region** in a protein is an unmatched oligopeptide located between two structural matches and is composed of residues whose alpha carbons are greater than three angstroms from any opposing alpha carbon. As detailed in Figure 7, divergent regions of the structural alignment do not provide sufficient structural information to match homologous residues. Therefore, the UniTS program utilizes sequence information to align the amino acids of the divergent regions.

The pairwise SDSA algorithm utilizes the structurally matched amino acids to determine which divergent regions of each protein match. Divergent regions from opposing proteins that are located between the same structurally matched amino acids are **matching divergent regions**. The algorithm inputs the sequence from a divergent region of one protein

and the sequence from the matching divergent region of the other protein into the MUSCLE sequence alignment program and sequentially aligns (i.e., utilize sequence information) the regions (Edgar, 2004a, 2004b). This process is repeated for all divergent regions of both proteins. The algorithm inserts gaps as necessary to compliment any unmatched divergent regions. Upon completion, all amino acids will be matched (either to another amino acid or to a gap) by either the structural matching algorithm or the MUSCLE sequence alignment.

The Grid

Although deriving a pairwise alignment (conventional sequence, SDSA, or structure) is relatively straightforward, aligning multiple proteins introduces a fundamental difficulty in bioinformatics: How does one align multiple proteins at the same time? Many alignment programs (regardless of the specific type of information being aligned) solve this problem by subdividing the alignment into multiple pairwise-alignments. Inevitably, the program generates a multiple alignment by combining the results of these pairwise alignments (Gibas and Jambeck, 2001; Krane and Raymer, 2003).

Like many alignment programs, UniTS subdivides multiple alignments into numerous pairwise alignments. It subdivides and recombines pairwise alignments comparably to the Clustal algorithm (Gibas and Jambeck, 2001). Additionally, the way in which it dynamically modifies the results of each iteration in a data structure is analogous to the position specific scoring matrix (PSSM) in PSI-BLAST (Krane and Raymer, 2003). This PSSM equivalent data structure in the UniTS program, designated as the **Grid**, relates the amino acid positions of the overall multiple alignment (including gaps) to the amino acid positions of each input protein. Figure 8 is a visual representation of the Grid. UniTS designates the Grid to be an

abstract protein (i.e., the **Grid protein**) and utilizes it as the template protein to which the input proteins are aligned. The spatial coordinates of an alpha carbon at a certain position in the Grid protein are the mean of the alpha carbon spatial coordinates of any positions aligned to that Grid position. Note that because the aligned positions in the Grid are dynamic, UniTS continuously modifies the spatial coordinates of the Grid protein as it calculates new alignment iterations.

Grid Position	1	2	3	4	5
Protein A Position	1	2	-	3	4
Protein B Position	-	1	-	2	3
Protein C Position	1	2	3	4	-
Protein D Position	1	-	2	3	-

Figure 8. Visual representation of the Grid. A gap is represented by a dash ("-"). The first amino acid in each of Proteins A, C, and D are aligned, while the first amino acid in Protein B aligns to the second amino acid in each of Proteins A and C. The spatial coordinates of the alpha carbon of the first Grid position are the mean of the coordinates of the first alpha carbon in each of Proteins A, C, and D.

Residue Determination

In the aforementioned algorithm, the Grid protein is an abstract protein derived using the mean alpha carbon spatial coordinates of the input proteins. Deriving the mean of coordinates is possible because numbers are analog and capable of being averaged together. Conversely, divergent regions in the pairwise SDSA algorithm require the input of amino acid sequences into the MUSCLE program. Because the pairwise SDSA algorithm aligns the Grid protein to an input protein, the algorithm necessitates the sequence of the Grid protein. However, the digital or discrete nature of amino acid residues prevents the derivation of a mean sequence (e.g., how does one average a glycine and a phenylalanine?).

The pairwise SDSA algorithm designates the amino acid identity for a Grid position as the most frequently occurring residue for that respective Grid position. However, if the residues aligned to a Grid position occur with equal frequency, deriving the Grid residue requires a more complex solution. Before UniTS inputs a divergent region of the Grid into MUSCLE, any Grid position without an established residue identity (because no residue is the most frequently occurring) receives the designation of an unknown amino acid (i.e., assigned an IUPAC abbreviation of "X" [Dixon et al., 1984]). For each Grid position featuring an unknown amino acid, UniTS substitutes the unknown amino acid with each of the possible amino acids available in the Grid position. MUSCLE then performs a sequence alignment for each of these substitutions. Note that only one amino acid is substituted for each sequence alignment; the other positions retain the unknown designation. For each of these alignments, MUSCLE outputs a scorefile containing the average BLOSUM62 score for each position aligned (Edgar, 2010; Gibas and Jambeck, 2001). Therefore, each of the possible amino acids for each unknown Grid position receives a BLOSUM62 score. The amino acid receiving the greatest BLOSUM62 score for a given Grid position is selected to represent that Grid position. Importantly, the amino acid positional designations are utilized exclusively for the MUSCLE alignment; furthermore, because the Grid is dynamic, the designations change with each iteration of the multiple SDSA algorithm.

Multiple SDSA

Calculating the SDSA of multiple structures initiates by selecting one of the input proteins to be the initial template protein. Appendix B describes the methodology UniTS employs to select the template protein. UniTS inserts the selected template protein into the

Grid. Because this is initially the only protein in the Grid, the alpha carbon spatial coordinates assumed by the Grid protein will equal those of the template protein (i.e., the mean of a single number is that number). The UniTS program then performs a pairwise SDSA of the Grid protein (initially only the template protein) and another input protein. Thereafter, UniTS will insert the input protein into the Grid based upon this pairwise alignment. Because the Grid now contains two proteins, the alpha carbon spatial coordinates of the Grid protein are recalculated by averaging the coordinates of both proteins. UniTS performs another pairwise SDSA of the newly calculated Grid protein and another input protein. This process is repeated until all input proteins have been inserted into the Grid.

Because the spatial coordinates of the Grid protein are updated as each input protein is iteratively inserted, the alignment of the initial template protein (or any of the early subsequent proteins) to the final Grid protein may now be inaccurate. Therefore, after the insertion of all input proteins into the Grid, the UniTS program will individually remove each protein (beginning with the original template protein) from the Grid. Upon removal of a protein, UniTS recalculates the spatial coordinates of the Grid protein utilizing those proteins remaining in the Grid. The removed protein will then be realigned to the recalculated Grid protein (via the pairwise SDSA algorithm) and reinserted into the Grid. This removal and realignment calculation is repeated for each protein. A single iteration is delineated by the removal and realignment of all the proteins. The aforementioned iteration is repeated until the Grid stabilizes. That is, until all the amino acid positions of the input proteins remain in consistent Grid positions. Importantly, the UniTS program will not cease in the middle of an iteration. Once the first protein is removed and realigned, the iteration must be completed by removing and realigning the remaining subsequent proteins. Only after the removal and

realignment of the final protein will UniTS compare the current state of the Grid to its state at the conclusion of the previous iteration. If the positional state of the Grid in the current iteration equals that of the previous iteration, the iterations cease and UniTS achieves Grid positional stabilization. Upon stabilization, the final state of the Grid is the final SDSA.

Multiple SDSA Quality Assessment: Mean Standard Deviation

Traditionally, structural alignment and superpositioning algorithms utilize the RMSD score to quantitatively assess the quality of two superimposed proteins (i.e., a pairwise alignment). Unfortunately, superimposing more than two proteins (i.e., a multiple alignment) prevents the calculation of the RMSD for quantitative analysis. Therefore, UniTS performs the quantitative assessment of a multiple protein superposition or structural alignment utilizing the mean standard deviation. Calculation of the mean standard deviation consists of averaging the individual standard deviations of each Grid position. UniTS calculates each individual standard deviation utilizing the spatial coordinates of all the alpha carbons constituting each Grid position. Specifically, it calculates the mean and standard deviation for the coordinates of each axis separately. The three mean coordinates (one from each axis) combine to establish the three-dimensional spatial coordinates representing the mean. UniTS then derives the standard deviation coordinates by adding the calculated standard deviation distance for each axis to each respective mean coordinate. The positional standard deviation equals the spatial distance between the mean coordinates and the standard deviation coordinates.

UniTS Results

To determine the accuracy of the UniTS program, I utilized UniTS to calculate the SDSA results of four protein families. The two PDB files comprising each protein family are illustrated in Table 8 (Berman et al., 2000). Species information for each PDB file (except those of the hemopexin repeats) can be found in Table 1b, while protein functional information can be found in the text subsequent of Table 1b. The hemopexin repeats are from the rabbit species of *Oryctolagus cuniculus*. Each repeat structurally composes a propeller-shaped region of hemopexin. Hemopexin recovers unbound heme to prevent the oxidative damage it causes to tissues.

Table 8. PDB designations associated with each protein family

Protein family	First protein PDB	Second protein PDB
Isocitrate Dehydrogenase	1T09[a]	1XGV[a]
Pectate Lyase	1PLU	2BSP
Polygalacturonase	1CZF[a]	1HG8
Hemopexin Repeats	1QHU[b]	1QHU[c]

[a]Chain A of the protein.
[b]Residues 56-134.
[c]Residues 263-353.

Because UniTS requires superimposed input proteins, I utilized the Theseus structural superpositioning program to superimpose the two proteins for each family (Theobald and Wuttke, 2006a, 2006b, 2008). I then compared the SDSAs, quality assessment scores, or generation parameters derived by UniTS to those results generated by the Theseus, DALI,

and Chimera programs (Holm and Rosenstrom, 2010; Pettersen et al., 2004; Theobald and Wuttke, 2006b). The subsequent results demonstrate that UniTS is currently the most capable and accurate algorithm for producing a SDSA if superimposed protein structures are available.

Because UniTS requires no input parameters (other than PDB files), I executed all comparison programs utilizing their default parameters. Additionally, although UniTS is capable of calculating a SDSA for multiple input protein structures (i.e., those involving more than two proteins), conducted comparisons utilize only pairwise alignments to reduce the complexity of manual analysis. Furthermore, all RMSD distances calculated herein incorporate only the alpha carbon atoms. Importantly, UniTS, Theseus, and DALI all perform distinctive primary functions. Therefore, UniTS does not replace these or other superpositioning and structural alignment programs; instead, UniTS supplements them by modifying their results. Finally, I performed no comparison of UniTS to a residue profile-based SDSA because UniTS requires superimposed protein structures. This requirement prevents the utilization of UniTS to solve the protein folding problem, thus a comparison is unwarranted.

UniTS Compared to Theseus

I performed the first UniTS comparison utilizing data output from the Theseus structural superpositioning program against the output data as subsequently refined by UniTS (Theobald and Wuttke, 2006b). Importantly, Theseus does not generate a resultant sequence alignment; instead, Theseus requires the input of a preliminary sequence alignment. Therefore, the conventional sequence alignment program MUSCLE derived the input

preliminary sequence alignment for all utilizations of Theseus presented herein (Edgar, 2004a, 2004b). Upon input of the MUSCLE sequence alignment, Theseus superimposed the input proteins and calculated the classical RMSD utilizing the amino acid matches established by the MUSCLE alignment. I then input the superpositioned protein structures into UniTS to calculate a comparison RMSD and sequence alignment for each protein family.

The RMSD calculation is utilized to measure spatial similarity and requires the establishment of amino acid matching derived by various forms of homologous alignment. UniTS utilizes protein structure to derive a SDSA while Theseus utilizes a conventional sequence-based MUSCLE alignment; therefore, the amino acid matches established utilizing the SDSA of UniTS will more accurately represent the spatial homology of the two proteins. Table 9 illustrates a significantly decreased resultant RMSD calculated by the improved amino acid matches established by the UniTS SDSA.

Table 9. Original RMSD reported by Theseus compared to the UniTS RMSD calculated from the proteins superimposed by Theseus for each protein family

Protein family	Theseus RMSD	UniTS RMSD
Isocitrate Dehydrogenase	15.23 Å	7.00 Å
Pectate Lyase	11.85 Å	6.19 Å
Polygalacturonase	1.99 Å	1.57 Å
Hemopexin Repeats	3.22 Å	1.74 Å

UniTS Compared to DALI

I next compared UniTS to the DALI structural alignment program (Holm and Rosenstrom, 2010). In addition to calculating the SDSA and RMSD for each protein family as detailed in the previous section, I also calculated the number of structurally matched residues UniTS utilized for these calculations. As displayed in Table 10, for three of the four protein families, UniTS utilized more residue matches than DALI when calculating the sequence alignment and RMSD (even the shorter hemopexin chains utilized the same percentage of matched residues for each methodology). The increased number of amino acid matches permits UniTS to produce a more complete and comprehensive SDSA and thus a more accurate RMSD calculation.

Table 10. The number of residue matches of DALI compared to the UniTS

Protein family	Shortest chain length[a]	DALI res matches[b]	UniTS res matches[b]	DALI RMSD
IDH[c]	414	299 (72%)	362 (87%)	2.57 Å
Pectate Lyase	353	261 (74%)	307 (87%)	1.59 Å
Polygalacturonase	335	325 (97%)	331 (99%)	1.13 Å
HPX Rep[d]	79	70 (89%)	70 (89%)	1.53 Å

[a]Least number of amino acids comprising the two polypeptide chains representing each protein family.
[b]Number of amino acid residue matches. Parentheses contain the percentage of matched residues utilized out of the total number of amino acids.
[c]Isocitrate Dehydrogenase.
[d]Hemopexin Repeats.

Notably, the RMSD returned from DALI is less than that calculated by UniTS. Although appearing favorable to DALI, this discrepancy is a product of the fewer matched residues DALI utilizes to calculate the RMSD. Specifically, the RMSD calculated by DALI is derived utilizing exclusively structurally conserved residues (i.e., those residue matches containing spatially proximate amino acids), thus resulting in a lower RMSD value (Holm and Sander, 1993).

UniTS Compared to Chimera

I performed the final comparison against the Match=>Align function of the Chimera protein structure visualization and modeling program (Pettersen et al., 2004). Although relatively unsophisticated, the Match=>Align function is a SDSA algorithm similar to UniTS. After superimposing each protein family utilizing Theseus, I derived two SDSAs for each superimposed family utilizing UniTS and Chimera respectively (Appendix C).

To quantitatively determine which SDSA represents the more accurate evolutionary homology for each family, I calculated the log-odds score of each SDSA utilizing a similar methodology to that of a sequence alignment algorithm. Specifically, the log-odds score for each SDSA represents the significance of the similarity between the composing polypeptide sequences given the amino acids matching therein. That is, it represents the significance of a nonrandom homologous relationship existing, with a greater score indicating a more significant, nonrandom alignment (Gibas and Jambeck, 2001). I calculated the log-odds score utilizing the Gonnet substitution matrix to score individual amino acid matches (Gonnet et al., 1992). The summation of these individual scores was then calculated to represent the sequence accuracy of the alignment.

I quantitatively compared the SDSAs for each protein family twice: The first comparison employed a gap opening penalty of -10.0 and a gap extension penalty of -0.1 because these constitute the standard default penalties in many sequence alignment programs (Tamura et al., 2011). However, the second comparison featured gap penalties of -5.0 and -0.1 respectively. The decreased gap opening penalty compensates for the increased number of gaps that will inevitably form when generating a SDSA as compared to a standard sequence alignment.

Tables 11 and 12 contain the log-odds scores derived by both UniTS and the Match=>Align function of Chimera for each protein family. The polypeptide sequences contained within the SDSAs generated by UniTS demonstrate superior significance of evolutionary homology relative to those generated utilizing the Match=>Align function of Chimera. The single exception to the aforementioned results is the log-odds score of the hemopexin repeats derived utilizing the -10.0 gap opening penalty. This inconsistent result can likely be attributed to the relatively short length of the repeats (see Table 10 for a length comparison). The short length of the hemopexin repeats prevents an adequate number of amino acid matches that are required to counterweigh the large gap opening penalty. This explanation is reinforced by UniTS producing the superior SDSA when utilizing the lesser -5.0 gap opening penalty.

Table 11. Comparison of the UniTS and Chimera alignment scores utilizing a gap opening penalty of -10

Protein family	UniTS Alignment Score	Chimera Alignment Score
Isocitrate Dehydrogenase	-126.7	-437.9
Pectate Lyase	58.0	-108.8
Polygalacturonase	677.2	643.3
Hemopexin Repeats	14.6	16.0

Table 12. Comparison of the UniTS and Chimera alignment scores utilizing a gap opening penalty of -5

Protein family	UniTS Alignment Score	Chimera Alignment Score
Isocitrate Dehydrogenase	58.3	-192.9
Pectate Lyase	203.0	71.2
Polygalacturonase	742.2	713.3
Hemopexin Repeats	64.6	56.0

Multiple PL/PG SDSA

To demonstrate the complete capability of UniTS, I calculated a multiple SDSA (Figure 9) and quantitatively assessed the structural superpositioning of four homologous proteins (Figure 10). The quad structural superpositioning was performed utilizing Theseus and consisted of two pectate lyase (PL) and two polygalacturonase (PG) proteins (see Table 8 for specific PDB designations). The mean standard deviation for the quad superposition is 3.36 angstroms.

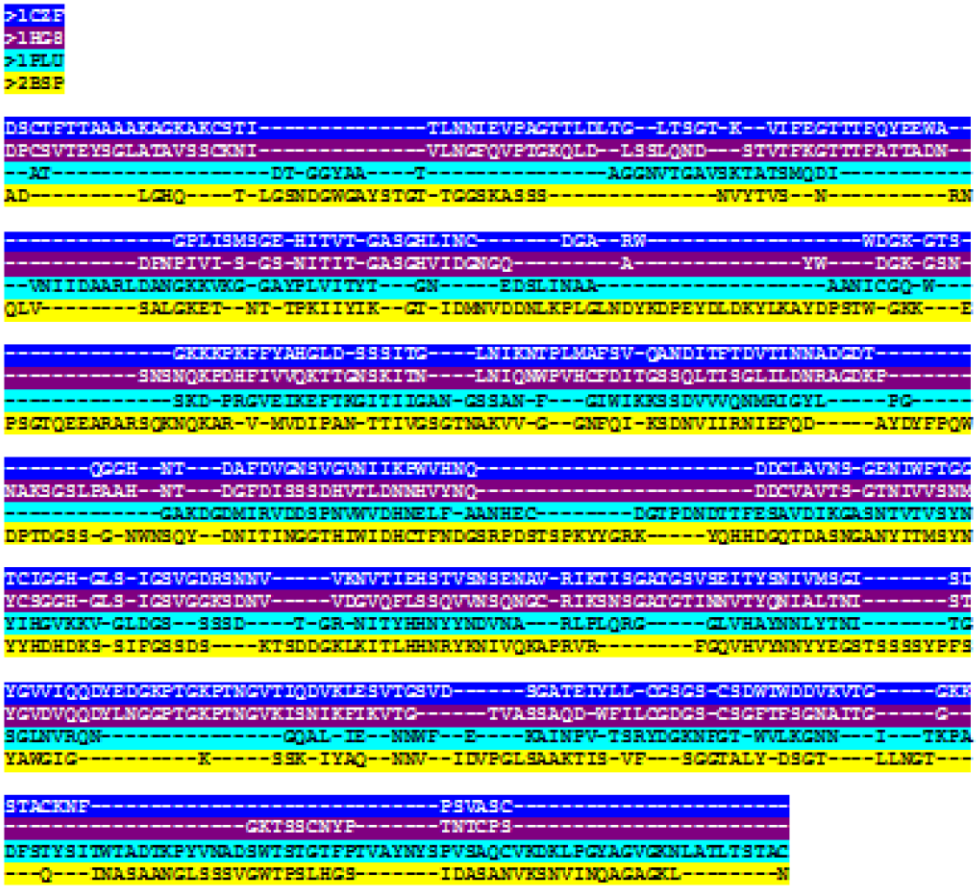


Figure 9. PL/PG multiple SDSA derived utilizing UniTS.

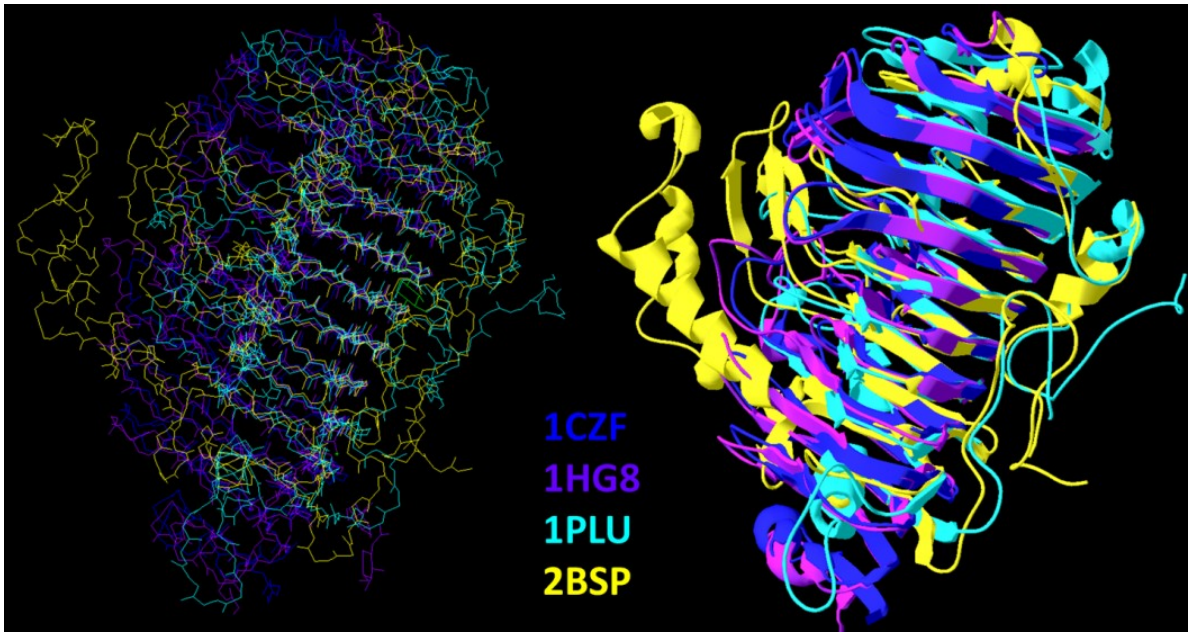


Figure 10. Graphical representation of the PL/PG protein superposition in both atomic and ribbon formats. Protein superpositioning was derived by Theseus and graphical imaging was performed utilizing Swiss-PdbViewer (Guex and Peitsch, 1997).

In addition to calculating the total mean standard deviation for the entirety of the four protein structures superpositioned, UniTS also exhibits the capability of outputting the standard deviation for each individual amino acid position (i.e., the Grid position as described in Chapter III, section “UniTS Algorithm”, subsection “The Grid”). Furthermore, one can generate a graph correlating these individual standard deviations to their respective amino acid positions (e.g., the graph in Figure 11 demonstrates this capability utilizing the aforementioned quad superposition). This graph permits intelligible differentiation of those regions of the protein superposition that are structurally conserved from those that are nonconserved.

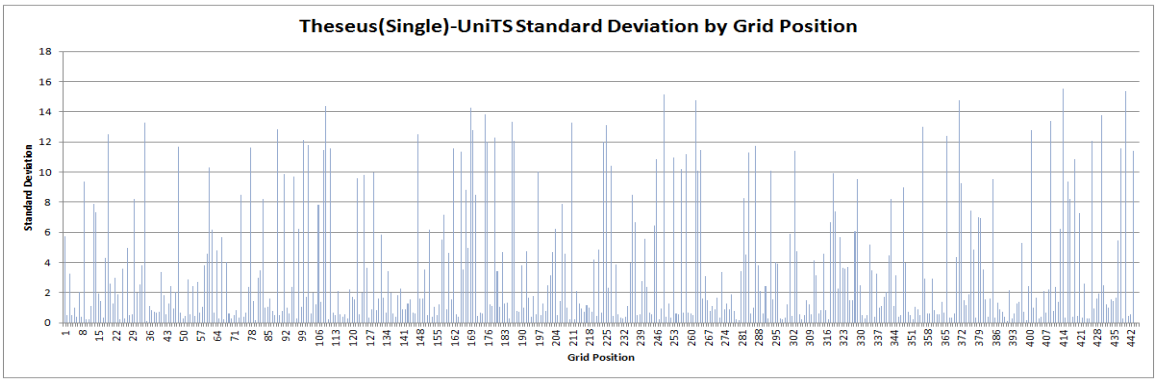


Figure 11. Standard deviation for each Grid position of the PL/PG protein alignment.

Discussion of UniTS Results

Although only the Match=>Align function of the Chimera program directly calculates a sequence alignment utilizing spatial information from superimposed protein structures, other algorithms (e.g., inverse folding sequence alignments and structural alignments) are capable of performing this function indirectly. However, the aforementioned results indicate that UniTS is the most capable SDSA program to date. Furthermore, these results demonstrate the capability of UniTS to refine the sequence alignment input into a superpositioning program and utilize this refined alignment to calculate improved structural quality assessment scores. Importantly, because these quality assessment scores are consistently derived, they also provide the capability to compare different superpositioning and structural alignment algorithms.

Most significantly, implementation of the UniTS program requires a more formalized analysis of sequence alignments derived utilizing sequence information versus those derived utilizing structural information. That is, does amino acid sequence or protein structure primarily influence the evolutionary homology of proteins? Although the solution to this question is extraordinarily complex, the problem is reconcilable in many situations. However, consider the following question: Provided the results of a conventional sequence alignment and dissimilar results of an SDSA derived utilizing the same input proteins, which alignment most accurately represents the homology of the proteins? Unfortunately, this complex but reconcilable solution must now be simplistically reduced to two incompatible options. This inevitable problem substantiated by the UniTS program necessitates further research into protein sequential/structural correlation and robustness.

CHAPTER IV

PUSH: PHYLOGENETIC TREE USING STRUCTURAL HOMOLOGY

Conventional dendrogram generation utilizes protein sequences and the established molecular clock hypothesis to generate results. However, because protein structure is more evolutionarily conserved than its corresponding amino acid sequence (Marti-Renom et al., 2000), a dendrogram generated utilizing the structure of the input proteins will derive superior and more complete evolutionary results. Unfortunately, the conventional molecular clock hypothesis fails to establish the required correlation between structural divergence and evolutionary distance. Therefore, I developed the proposed structural molecular clock hypothesis to establish this correlation with biological accuracy. To implement this novel hypothesis, I developed the unique Phylogenetic Tree Using Structural Homology (PUSH) program that is capable of generating a dendrogram utilizing protein structures instead of conventional sequences. It generates a dendrogram utilizing the proposed structural molecular clock hypothesis to derive a probability matrix. The dendrogram is then graphically displayed utilizing a hierarchical clustering algorithm.

Structural Molecular Clock Hypothesis

The Application of the Sequence-based Molecular Clock Hypothesis to Protein Structure

A molecular clock establishes a correlation between the magnitude of evolutionary modifications on a genome, gene, or protein and the quantity of time necessary for these modifications to evolve (Krane and Raymer, 2003). The time required for evolution is linearly expressed as the evolutionary distance (d_e). Apropos of protein sequences, the established sequence-based molecular clock hypothesis (MCH) assumes that random amino acid mutations (R_m), including insertions and deletions, accumulate at a consistent rate throughout time (Krane and Raymer, 2003). Specifically, although individual residues possess distinct mutation rates, the mean rate at which mutations manifest over time remains constant. This assumption allows the direct correlation relating d_e and sequential R_m. R_m is derived by the residue differentiation portrayed in the sequences of multiple homologous proteins. Therefore, the correlation relating d_e and R_m, and calculating R_m using sequence divergence, establishes a correlation between amino acid sequence divergence and d_e. That is, as the sequences of two proteins become more divergent, the d_e between the two proteins increases (Krane & Raymer, 2003).

Although the established MCH successfully calculates d_e from input amino acid sequence data, the theoretical foundation of this hypothesis fails upon inputting protein structure data. To calculate d_e utilizing protein structure, structural dendrogram generation algorithms (e.g., DALI [Holm & Sander, 1993]) extend the conventional sequence-based MCH to a structural equivalent based upon the spatial distance between the atomic coordinates of multiple protein structures (Deeds, 2007). This logical extension of the MCH

is presumably correct because sequence mutations theoretically invoke structural transformation, thus establishing a correlation between structural divergence and R_m. Therefore, as the total spatial divergence of two proteins increases, d_e between the two proteins hypothetically increases proportionally.

Because the protein structures of homologous proteins are more conserved than their respective sequences (Marti-Renom, et al., 2000), a frequent problem experienced upon utilizing protein structural information to calculate d_e is the negligible spatial divergence of sequentially conserved regions (Deeds, 2007). Therefore, the spatial dissimilarity in sequentially nonconserved regions of homologous proteins most significantly contributes to d_e calculations. Unfortunately, the effect amino acid mutations in sequentially nonconserved regions assert on protein structure is unpredictable. Nonconserved regions are capable of significant spatial movements resulting from a single amino acid mutation (Deeds, 2007; Glasner, 2007). These considerable spatial displacements prevent the correlation of structural change to sequential R_m because they generate too many variables and possibilities. For example, consider a single mutation substantially spatially relocating a nonconserved peripheral loop. Sequentially, the single mutation (correctly) will derive a relatively short d_e; however, structurally, the substantial spatial relocation of the loop will generate a (incorrect) lengthy d_e.

In addition to the inconsistent calculation of d_e in mobile protein loops, structural information prevents the correct calculation of d_e for oligopeptide indel regions. It is possible for an oligopeptide indel to occur in a single evolutionary event as a solitary lengthy indel, thus producing a short d_e; however, the possibility also exists that this oligopeptide indel develops by multiple indels, each consisting of a single amino acid. The latter possibility will

derive a lengthy d_e because it requires numerous evolutionary events. Although resolving this discrepancy is possible, the resolution requires the utilization of sequence information rather than structural information.

The absence of significant spatial divergence between the sequentially conserved regions of homologous proteins and the inability of nonconserved regions to consistently predict R_m prevent the structural implementation of the established MCH. Therefore, calculating d_e utilizing protein structural information requires an alternative hypothesis. The structure-based MCH proposed herein resolves the aforementioned difficulties by employing a protein structural and functional evolutionary model.

Proposal of a New Evolutionary Mechanism for Use as a Molecular Clock

The evolution of a protein can be monitored by observing the change in sequence, structure, or function of the protein. The rate of evolution is derived by establishing a molecular clock to calculate d_e; the established sequence-based MCH calculates d_e utilizing R_m as a molecular clock. Therefore, limitless types of quantitative biological information can be utilized to calculate d_e provided this information can be correlated to R_m. Unfortunately, an imperfect correlation decreases the accuracy of the molecular clock.

As illustrated in Figure 12, the central dogma of molecular biology ensures the linear transference of the correlation between protein sequence, structure, and function data types (Krane & Raymer, 2003). That is, establishing a correlation from protein structural information to sequence information is required before the calculation of d_e can proceed.[9]

[9] Conventionally, spatial coordinates or dihedral angles represent protein structural information. Therefore, the amino acid sequence extracted from structural information is likewise considered "sequence information". The aforementioned approach is simply an alternate methodology for obtaining the amino acid sequence.

Additionally, a correlation from protein functional information to structural information is required prior to correlating this structure information to sequence information. Therefore, the established MCH will calculate a more accurate d_e utilizing sequence information rather than structural information because it requires one fewer correlation.

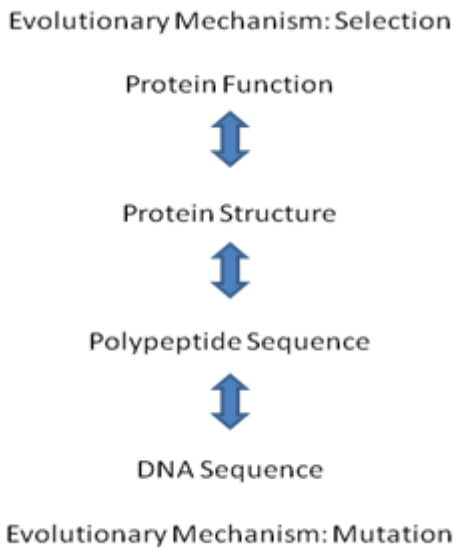


Figure 12. The central dogma of molecular biology

While the molecular clock requires amino acid R_m to calculate d_e, the results derived utilizing structural information will be inferior to those derived utilizing sequence information. Fortunately, a second evolutionary mechanism influences protein structural evolution: selection pressure. Both random amino acid mutations and evolutionary selection pressure (P_s) influence the evolution of protein structure. Random sequence mutations continuously attempt to transform the structure of a protein, while P_s determines which of these mutations remain and which mutations are removed from the population.

P_s directly influences the function of a protein. Because the promoted function (i.e., the function selected for by P_s) influences the structure of the protein, P_s indirectly influences the structure of the protein. Furthermore, P_s also indirectly influences protein sequence because structural constraints influence the sequence of a protein. However, the influence of P_s on the structure of a protein is more direct than the influence of P_s on the sequence (as illustrated by the linear trajectory of influence in Figure 12). Therefore, a protein structure-based molecular clock that utilizes the evolutionary mechanism of P_s to measure d_e will be superior to one that utilizes R_m.

Protein Structural and Function Evolution

The correlation of the structural divergence of homologous proteins and the magnitude of P_s exhibited on them requires knowledge of the evolution of protein structure and function. Research suggests that most novel families of proteins originate from pseudogenes or cryptic genes (i.e., genes that exclusively express under specific atypical conditions; sometimes lying dormant for generations). However, the origination of a novel class of proteins is rare because pseudogenes and cryptic genes accumulate mutations in the absence of P_s. Therefore, the respective proteins expressed from these genes will likely be nonfunctional (Glasner, 2007).

Although the creation of novel protein families is rare, creating a protein within a class is more common. The evolution of function within a class of proteins originates from promiscuous intermediate proteins. A promiscuous protein is an enzyme whose active site catalyzes one or more "promiscuous" reactions in addition to the primary reaction. The catalytic activity of these promiscuous reactions occurs on several orders of magnitude less

than the primary reaction. Additionally, if P_s promotes (i.e., selects in favor of) a promiscuous function in a promiscuous intermediate enzyme, the gene coding the promiscuous intermediate begins to evolve before gene duplication occurs. Once the magnitude of catalytic activity equalizes between the primary and promiscuous reactions, P_s manifests gene duplication because neither reaction can catalyze at an optimal rate unless two distinct proteins are generated (Glasner, 2007).

Additionally, research also suggests that an enzymatic active site evolves by incrementally changing a single elementary reaction in a multiple-step reaction mechanism. For example, if a catalytic reaction mechanism requires five elementary reactions, the active site structurally evolves to alter only a single elementary reaction. This minor structural alteration produces an evolved reaction mechanism that remains similar to the original reaction. Changing only elementary reactions is necessary to allow the protein to perform both the primary reaction and the promiscuous reaction. Furthermore, theoretically, this allows the construction of a dendrogram based upon protein functional evolution by traversing the changes in the elementary reactions over time (Glasner, 2007).

Correlating Protein Structural Divergence and Evolutionary Selection Pressure

The magnitude of P_s exhibited on a protein influences the structure and function of the protein. Realistically, the magnitude of P_s fluctuates at any instant in time $\left(\frac{dP_s}{dt}\right)$ based upon several variables including adaptability of the organism and the rate of environmental change. However, the first assumption of the proposed structure-based MCH is that the changes in P_s will average over a length of time $\left(\frac{\Delta P_s}{\Delta t}\right)$ into a consistent and gradual

magnitude of P_s (i.e., $\frac{\Delta P_s}{\Delta t} = 0$). Even the increased deviation in P_s exhibited under conditions of punctuated equilibrium is averaged over time (Eldredge and Gould, 1972). This assumption is similar to that made by the conventional MCH regarding R_m (i.e., while R_m at any instant in time may vary, it will average to produce a consistent R_m) (Krane and Raymer, 2003).

Assuming P_s is gradual and consistent over Δt, the functional change (ΔF) exhibited by the protein must also be consistent over Δt $\left(\frac{P_s}{\Delta t} = \frac{\Delta F}{\Delta t}\right)$. This direct correlation exists because all proteins input into the proposed structural MCH must be extant proteins (otherwise data on the structure would be nonexistent). Any protein whose function is unable to adapt to P_s $\left(\frac{\Delta F}{\Delta t} \neq \frac{P_s}{\Delta t}\right)$ will inevitably become extinct (i.e., removed from the population); therefore, an extant protein must have $\frac{\Delta F}{\Delta t} = \frac{P_s}{\Delta t}$ to prevent extinction.

The general function of a protein determines the structural component of the protein that correlates with the function. In a structural protein the entire structure correlates with function because the structure of the protein is its function. Unlike structural proteins, however, in enzymes the active site correlates with function because it is the location at which the function is directly determined and produced. Function evolves by changing single elementary reactions in an enzymatic reaction mechanism (Glasner, 2007). To alter an elementary reaction, a corresponding structural transformation (caused directly or indirectly) in the active site is required. Therefore, the number of elementary reactions that change is directly proportional to the magnitude of structural change in the active site.

A Novel Structural Molecular Clock Hypothesis

The aforementioned section illustrates a correlation between the change in protein structure and P_s that can ultimately be utilized to calculate the relative lengths of d_e for a system of homologous proteins. Specifically, the spatial divergence of the enzymatic active site is directly correlated with the length of d_e. The active site and other conserved regions of homologous proteins lack structural divergence when compared to nonconserved regions of a protein. Therefore, any divergence in the active site that illustrates functional change will be eclipsed by the spatial alterations of peripheral loops and other nonconserved regions. This suggests that including nonconserved regions in the spatial divergence calculations will actually decrease the accuracy of d_e.

Unfortunately, the structural conservation of the active site results in insufficient structural divergence if only the active site is used to measure d_e (Deeds, 2007). Considering the level of error involved in structural determination and computation, this minimal structural divergence is likely inadequate for calculating d_e. Importantly, active site conservation only occurs if P_s promotes the retention of the primary function of the enzyme. However, if P_s promotes the expression of a promiscuous function, the structural divergence of the active site increases. Fortunately, the second assumption of the proposed structure-based MCH is that input "homologous" enzymes possess folds exhibiting different or varying levels of catalytic functions. Therefore, because the input enzymes possess significant ΔF, the structure of the enzymatic active site will possess enough structural divergence to adequately calculate d_e.

Practical utilization of the proposed MCH dictates that it must possess a method of locating the enzymatic active site. Conventionally, the active site is located by finding the

sequentially conserved regions of the enzyme. This not only locates the active site, but also any region of the protein upon which P_s is acting. Unfortunately, the second assumption of the proposed MCH prevents the utilization of the aforementioned method because no region of the input enzymes can be sequentially conserved if the enzymes posses significant ΔF. To change the function of an enzyme, the active site must change in one of three ways: 1) A nonconserved peripheral region of the enzyme can mutate, thus indirectly changing the active site; 2) A mutation in the active site on an amino acid not directly involved in the catalytic reaction, thus causing a structural change in the active site; 3) A mutation in an amino acid directly involved in the reaction[10] (Glasner, 2007). Because mutations possess an equal probability of occurring in any region of the protein (before the influence of P_s), and because the active site can be structurally changed by a sequence mutation occurring in any region of the enzyme, no region of an enzyme is sequentially conserved given ΔF.

Although the active site of an enzyme is not sequentially conserved, it remains structurally conserved. Because ΔF occurs by changing single elementary reactions, the structure of the active site will only change in minute increments as the enzyme evolves. A lengthy d_e would be required for the active site to be structurally nonconserved. Importantly, although the structure of the active site is relatively conserved, it is not completely conserved (such as if P_s promoted the primary enzymatic function).

The active site is not the exclusive structurally conserved region in the enzyme. Fortunately, although all regions of an enzyme featuring ΔF are sequentially nonconserved, all structurally conserved regions of the enzyme are influenced by P_s. Therefore, all structurally conserved regions of an enzyme (not only the active site) can correlate with P_s.

[10] The latter mutation has a more profound effect on the enzymatic function compared to the former two. However, the probability of a mutation occurring on an amino acid directly involved in catalysis is low because these residues are few in number and this mutation may prevent the promiscuous reaction from occurring.

Structural Molecular Clock Hypothesis Discussion

The structural divergence of homologous proteins should not be used to measure the length of the evolutionary distance between proteins utilizing the conventional molecular clock hypothesis. Correlating changes in protein structure to the rate of amino acid mutations required by the established molecular clock is discouraged because nonconserved regions exhibit unpredictable spatial movements while conserved regions demonstrate minimal structural divergence if enzymatic function remains static. Therefore, the novel molecular clock hypothesis proposed herein is required to correlate protein structural divergence with the length of evolutionary distance.

When selective pressure promotes functional change, any enzymatic region upon which the selective pressure influences must be determined utilizing structural conservation rather than sequence conservation. The spatial divergence of these structurally conserved regions correlates with the change in function over time exhibited by the enzyme. If the selection pressure influencing enzymatic function is applied gradually and consistently, the change in function must also be consistent. Therefore, the spatial divergence of structurally conserved regions of homologous proteins directly correlates to the length of the evolutionary distance between these proteins.

PUSH Algorithm

Derivation of the Evolutionary Distance Matrix

As stated in the aforementioned structural molecular clock hypothesis, the amount of structural divergence exhibited between two proteins is directly proportional to the d_e between them. The PUSH algorithm utilizes superimposed input protein structures to compare the structural divergence of the proteins. The algorithm performs a pairwise comparison of all combinations of input proteins. The d_e results of these pairwise comparisons are placed in a $n \times n$ matrix, where n is the number of proteins input into the algorithm. The d_e between two proteins i and j is equal to the mean distance ($d_{i,j}$) between their homologous atoms. Importantly, the magnitude of d_e is equal to that of $d_{i,j}$; however, $d_{i,j}$ is measured in angstroms, while d_e is measured as an arbitrary unit of time. If a time calibration scale is known, each d_e can be converted into the appropriate unit of time. Although the distance lengths will never change relative to each other, converting the unit of time will modify the magnitude of each d_e.

Conducting an evolutionary comparison of proteins structures by calculating the mean standard deviation requires the determination of homologous atoms. The PUSH algorithm utilizes an input sequence alignment to determine homologous amino acid matching. Although this sequence alignment can originate from any program, the methodology proposed herein requires the alignment be structure-based such as one generated utilizing UniTS. As outlined by the proposed structural molecular clock hypothesis, only those homologous amino acids located within structurally conserved regions of the proteins are considered in the evolutionary distance calculation. Specifically, each

amino acid is spatially represented by its associated alpha carbon, amine nitrogen, and carboxyl carbon. The spatial distances between the three homologous atoms for each amino acid are averaged into a mean distance ($\bar{d}_{aa}$). Homologous amino acids are located within a structurally conserved region of the protein if $\bar{d}_{aa} \leq 3$ angstroms. Furthermore, any unmatched amino acids located within these conserved regions are ignored. To generate the d_e matrix, PUSH calculates $d_{i,j}$ for each pairwise structural comparison by averaging all the individual $\bar{d}_{aa}$ quantities located within structurally conserved regions.

Hierarchical Clustering

Hierarchical clustering is the process of graphing a dendrogram based upon the quantities of the calculated d_e matrix. To complete the hierarchical clustering process, two algorithms are required: one algorithm will derive the relative orientation of the nodes, where each internal node represents a speciation event and each terminal node represents the final representation of an OTU (whether extinct or extant); the other algorithm will calculate the relative d_e between each node (both internal and terminal). The node orientation algorithm begins with the assumption that each internal node comprises all OTUs to whose terminal nodes they connect (Figure 13).

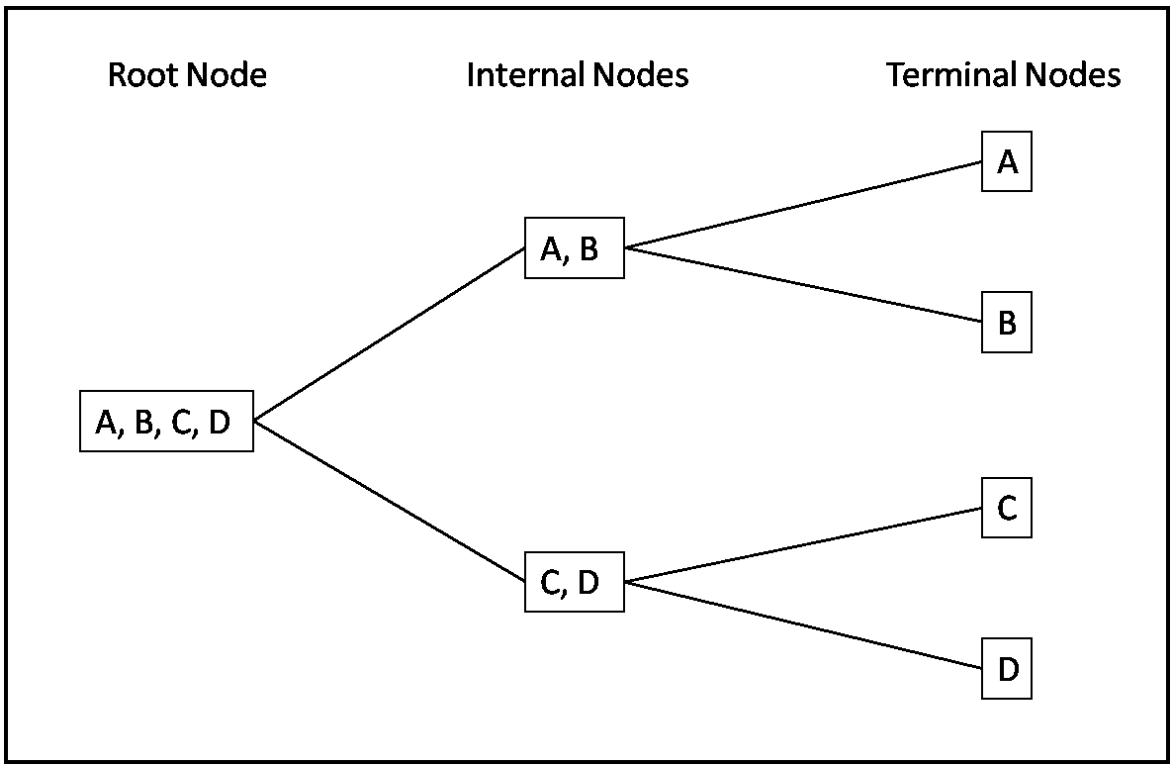


Figure 13. Node composition; each letter represents an OTU.

Note that although Figure 13 independently labels the root node, the root node is considered an internal node. The hierarchical clustering algorithm is initiated at the root node and works towards the future terminal nodes, generating dichotomous child nodes for each internal node present. For all the OTUs contained within an internal node, the d_e between each OTU pair is known and located in the d_e matrix. By default, the pair of OTUs possessing the maximum d_e will be moved to opposite dichotomous child nodes. That is, one OTU will move to the upper child node and the other will move to the lower child node[11]; these initially moved OTUs are labeled **upper child OTU** and **lower child OTU** respectively. Any additional OTUs contained within the original node (i.e., the "parent node") are placed in one of the child

[11] The terms "upper" and "lower" are derived assuming a horizontally oriented dendrogram (such as the one in Figure 13). For each internal node, dichotomous child nodes are generated. Conventionally, these are oriented vertically, with one being the relative "upper node" and the other being the relative "lower node".

nodes. Each additional OTU is moved to the child node possessing the lesser d_e. That is, if d_e between the additional child OTU and the upper child OTU is less than that between the additional child OTU and the lower child OTU, then the additional OTU is moved to the upper child node. Conversely, the additional OTU is moved to the lower child node if d_e between the additional child OTU and the lower child OTU is the lesser quantity. This process of node generation continues until only terminal nodes (i.e., those possessing only one OTU) remain and the complete orientation of the nodes composing the dendrogram is established.

The second hierarchical clustering algorithm calculates d_e between all connected nodes in the dendrogram. Ideally, the quantities within the d_e matrix should consistently calculate the d_e between all dendrogram nodes. Unfortunately, realistic protein structural divergence produces d_e quantities in the d_e matrix that do not derive a consistent d_e between two nodes. Therefore, it is impossible to generate a perfect dendrogram whose d_e lengths correspond exactly to those contained within the matrix. If matrix quantities inconsistently represent a d_e length on the dendrogram, the discrepancy is commonly solved by utilizing the mean d_e of the inconsistent lengths. Although the calculation direction is reversed, PUSH calculates the mean derivation of each d_e similarly to that of the UPGMA algorithm (Ewens and Grant, 2005; Isaev, 2006; Krane and Raymer, 2003). Specifically, PUSH resolves inconsistent d_e lengths by averaging the d_e of each combination of upper and lower child nodes respectively for all internal nodes. Note that any internal node possessing terminal child nodes utilizes the d_e from the distance matrix because it is the only d_e to average.

PUSH Results

I generated three dendrograms to examine the proficiency of the structure-based dendrograms generated by the PUSH program. These dendrograms are modeled utilizing the three protein assemblages superimposed by SABLE to examine its multiple alignment capabilities (see Chapter II, section "SABLE Results", subsection "Multiple Alignment Comparison" for more details on each protein assemblage). PUSH requires superimposed protein structures and a corresponding sequence alignment as input; therefore, each assemblage input into PUSH was structurally superimposed utilizing SABLE and the resultant SDSA was derived utilizing UniTS.

Upon completion of the aforementioned PUSH dendrograms, I input the same proteins into the conventional amino acid sequence-based ML dendrogram generator from the MEGA5 evolutionary analysis program to generate corresponding comparison dendrograms (Tamura et al., 2011). Each comparison ML dendrogram was generated by deleting any sequence gaps and utilizing the Jones-Taylor-Thornton (JTT) substitution model (Jones et al., 1992). The input sequence alignment required for each ML dendrogram was derived utilizing the MUSCLE sequence alignment program (Edgar, 2004a, 2004b).

Quad PG-PL Dendrograms

The initial two of the aforementioned dendrograms each consists of two polygalacturonase (PG) proteins (PDB designations: 1CZF and 1HG8) and two pectate lyase (PL) proteins (PDB designations: 1PLU and 2BSP). One dendrogram was generated utilizing the monomeric version of 1CZF and the other was generated utilizing the homodimeric

version. Because two proteins belong to the PG family and the other two belong to the PL family, the shape of the ideal dendrogram should reflect this familial divergence. That is, one internal node should evolutionarily relate the two PG proteins while another internal node should relate the two PL proteins. Figures 14 and 15 illustrate the dendrograms generated utilizing the conventional sequence-based ML algorithm, while Figures 16 and 17 illustrate the complimentary dendrograms generated utilizing PUSH.

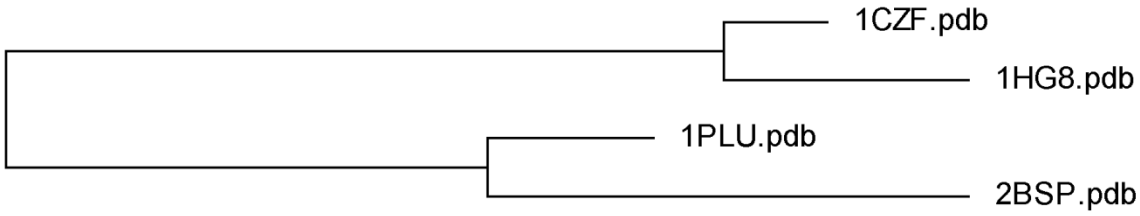


Figure 14. Dendrogram of two monomeric PG (1CZF and 1HG8) and two monomeric PL (1PLU and 2BSP) proteins generated utilizing a sequence-based ML algorithm. Dendrogram image was generated utilizing the MEGA5 evolutionary analysis program (Tamura et al., 2011).

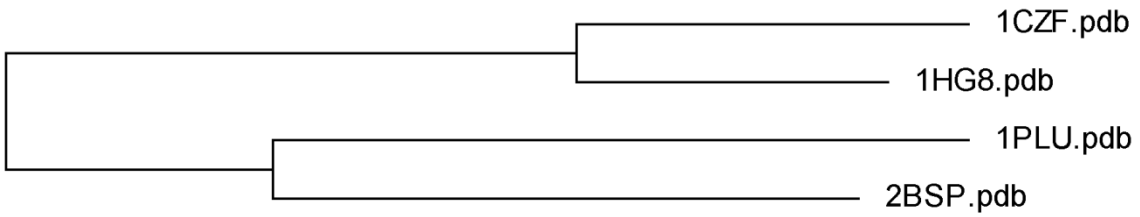


Figure 15. Dendrogram of one monomeric PG (1HG8), one homodimeric PG (1CZF), and two monomeric PL (1PLU and 2BSP) proteins generated utilizing a sequence-based ML algorithm. Dendrogram image was generated utilizing the MEGA5 evolutionary analysis program (Tamura et al., 2011).

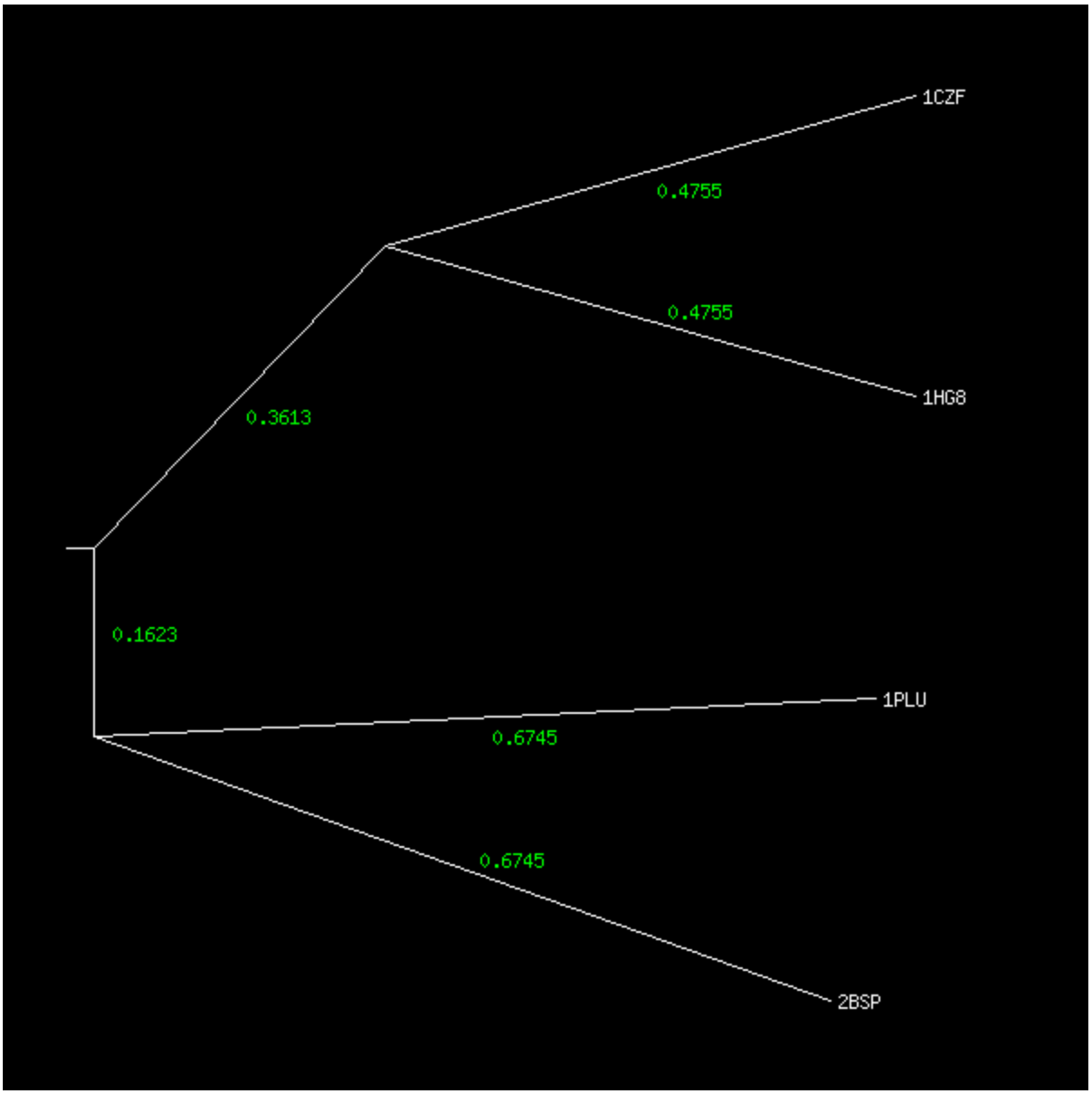


Figure 16. Dendrogram of two monomeric PG (1CZF and 1HG8) and two monomeric PL (1PLU and 2BSP) proteins generated utilizing the structure-based PUSH algorithm. Dendrogram image was generated utilizing the PUSH program.

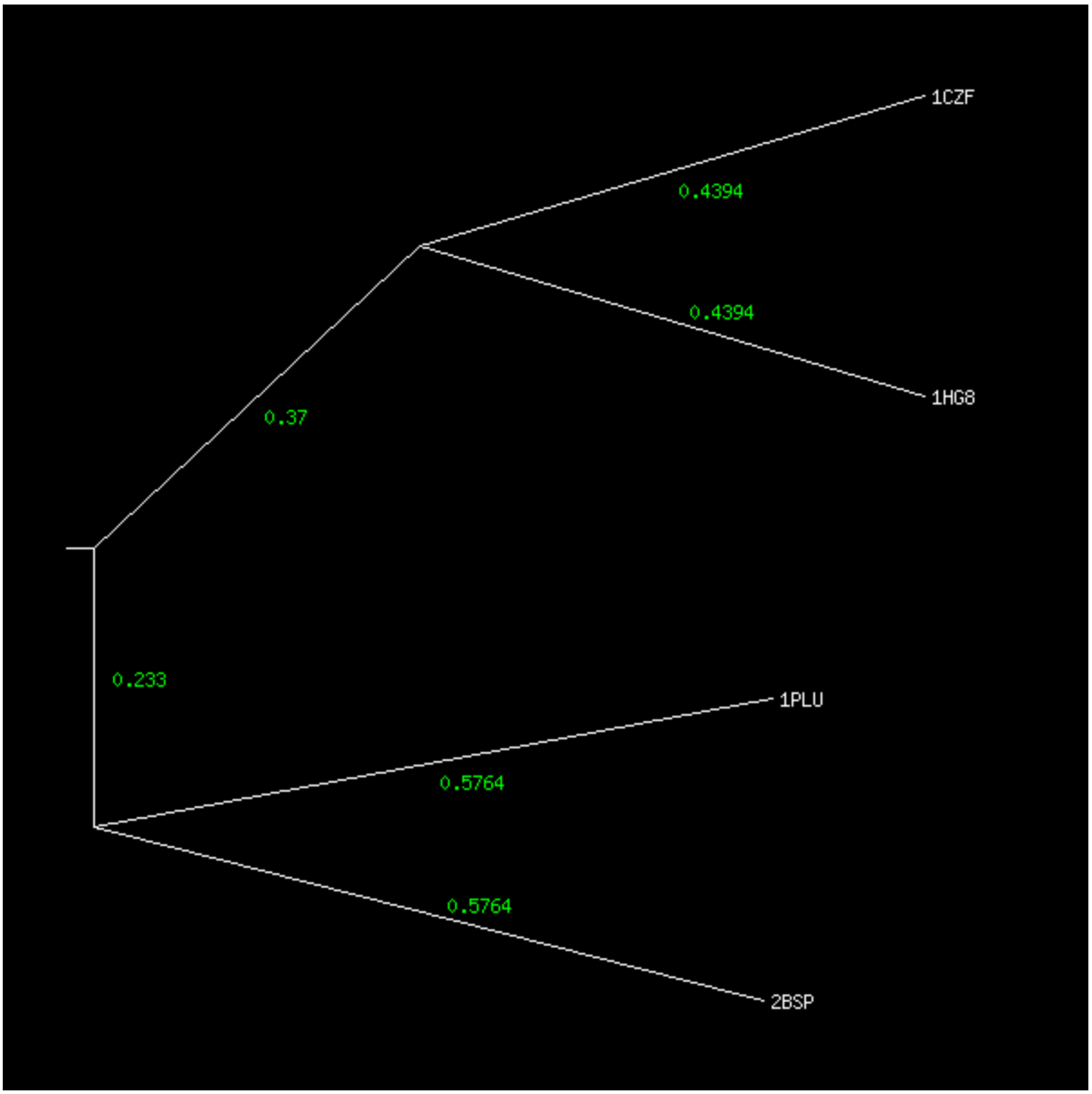


Figure 17. Dendrogram of one monomeric PG (1HG8), one homodimeric PG (1CZF), and two monomeric PL (1PLU and 2BSP) proteins generated utilizing the structure-based PUSH algorithm. Dendrogram image was generated utilizing the PUSH program.

Derivation of a dendrogram representing the "true" evolutionary relationship of the aforementioned proteins is simple because only four proteins are utilized to generate each dendrogram and the evolutionary relationship exhibited from the four proteins is straightforward. The simplicity of these dendrograms indicates that those generated utilizing

the conventional ML algorithm are likely a correct representation of the "true" dendrogram. Therefore, because the PUSH dendrograms in Figures 16 and 17 are identical to those generated utilizing the conventional ML algorithm in Figures 14 and 15, the PUSH dendrograms are also likely the correct representation of the "true" dendrogram. Importantly, when generating simple dendrograms, the correct PUSH dendrogram is likely identical to that produced by a conventional sequence-based ML algorithm.

Penta-PG Dendrograms

The final dendrogram comparison was generated utilizing five monomeric PG proteins. Although all proteins included in these comparative dendrograms correspond to a single protein family, structural evolutionary distinctions exist. Figure 5 illustrates that the protein structures of 1CZF and 2IQ7 differentiate from those of the unpublished 1ZEU and 1ZFW proteins. Furthermore, Figure 5 suggests that the unpublished (and unofficially designated) TOMA protein is a structural intermediate to the two PG groups. The dendrograms generated by both methodologies (Figures 18 and 19) confirm the close homology of the 1CZF protein to the 2IQ7 protein, as well as the close homology of the 1ZEU protein to the 1ZFW protein. The dendrograms also verify the structural differentiation illustrated in Figure 5 of the two homologous pairs (i.e., the structural divergence of the 1CZF/2IQ7 pair and the 1ZEU/1ZFW pair).

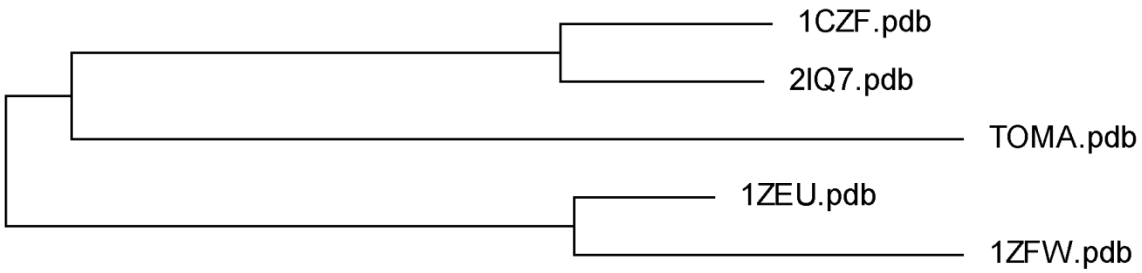


Figure 18. Dendrogram of five monomeric PG proteins (1CZF, 2IQ7, TOMA, 1ZEU, and 1ZFW) generated utilizing a sequence-based ML algorithm. Dendrogram image was generated utilizing the MEGA5 evolutionary analysis program (Tamura et al., 2011).

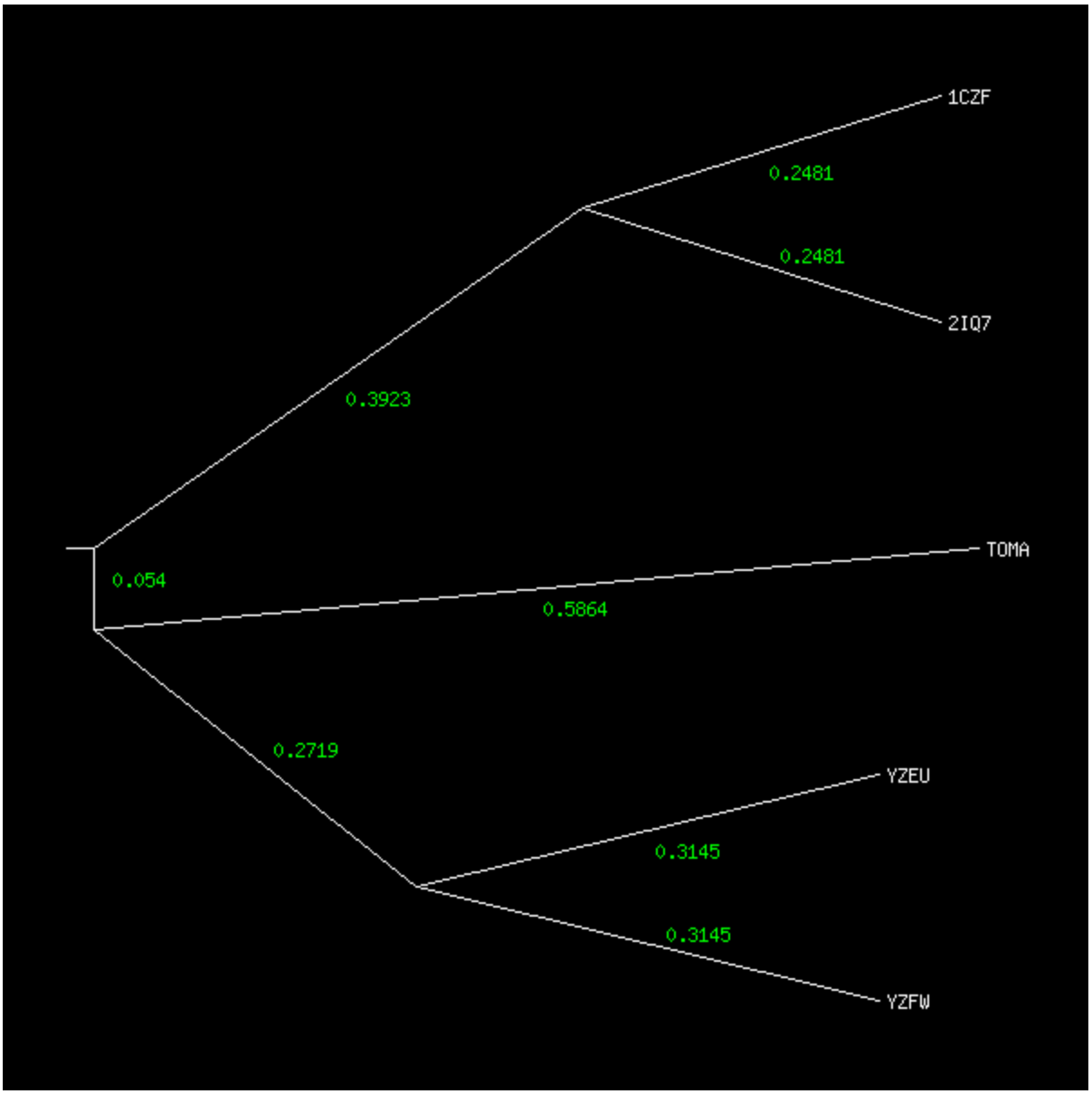


Figure 19. Dendrogram of five monomeric PG proteins (1CZF, 2IQ7, TOMA, 1ZEU, and 1ZFW) generated utilizing the structure-based PUSH algorithm. Dendrogram image was generated utilizing the PUSH program.

The results of the sequence-based ML algorithm indicate the TOMA protein is more closely related to 1CZF and 2IQ7, while the results of the PUSH algorithm indicate the TOMA protein is more closely related to the 1ZEU and 1ZFW proteins. However, both dendrogram generation algorithms demonstrate that the d_e between the TOMA protein and

the other PG proteins is approximately equal to the d_e between the 1CZF/2IQ7 pair and the 1ZEU/1ZFW pair. Therefore, in the absence of the "true" dendrogram, the insignificantly short differential d_e (0.054 angstroms in the PUSH dendrogram) between the internal node (i.e., the node that relates TOMA to its most homologous pair) and the root node in each dendrogram suggests that this difference is negligible for purposes of comparing dendrogram generation algorithms. Additionally, the short differential d_e in both dendrograms indicates that if 1CZF and 2IQ7 belong to a subfamily[12] and 1ZEU and 1ZFW belong to a separate subfamily, then TOMA likely belongs to a third distinct subfamily.

[12] The term "subfamily" in this context is utilized as an unofficial designation. In this context it is referring to a conglomerate of several closely-related homologous OTUs within a protein family.

CHAPTER V

PUSH DISCUSSION AND GENERAL CONCLUSION

The hypothesis that the dendrogram generation methodology and software presented herein is superior to any current methodology utilizing conventional technology is thus far inconclusive because additional trials featuring proteins of distant homology are required. The SABLE program is capable of consistently and accurately superimposing pairwise monomers, pairwise proteins composing inconsistent numbers of subunits, and more complex multiple alignments. The capabilities of the comparison superimposing programs, however, were inconsistent, demonstrated inferior accuracy, induced an inferior SDSA, or required additional preliminary curation. Additionally, Theseus and other superpositioning programs require input of a preliminary sequence alignment; this requirement prevents superpositioning programs from superimposing distant homologs. Although the comparison programs were generally capable of superimposing monomeric proteins featuring close homologies, they demonstrated an inferior capability to superimpose proteins from distant homologs. For example, SABLE demonstrated significant superior proficiency when superimposing either the quad PG/PL assemblage of proteins or any assemblage containing the homodimeric 1CZF protein. Further note that, these assemblages are still relatively homologous; therefore, MUSTANG and Theseus would likely be incapable of accurately and consistently superimposing proteins from more evolutionary distant protein families.

Analysis of the PUSH results in Chapter IV ("PUSH Results" section) also suggests the necessity of additional trials featuring proteins exhibiting distant homologies. The relatively few OTUs included in each dendrogram generation trial and the evident evolutionary relationships between the OTUs suggest the relative simplicity of the dendrograms generated herein. Because these dendrograms are relatively simplistic, a conventional sequence-based ML algorithm should correctly derive dendrograms that accurately represent the "true" evolutionary relationships of the inclusive OTUs. Therefore, ideally, the PUSH dendrograms would be identical to those generated utilizing a conventional ML algorithm. As hypothesized, the PUSH dendrograms are identical (the dendrogram relating the five PG proteins is insignificantly nonidentical) to those generated utilizing protein sequence.

Although the aforementioned results confirm the proficiency of PUSH discerning the evolutionary relationships between close homologs, more data is required to completely examine the competency of the PUSH algorithm utilizing proteins exhibiting distant homologies. Conventional sequence-based ML dendrogram generation algorithms are likely incapable of deriving the evolutionary relationships of multiple protein families all possessing a homogenous class of protein subunit. However, SABLE is proficient at superimposing any homologous proteins, UniTS calculates amino acid matches based upon these superimposed structures, and PUSH utilizes these protein structures and amino acid matches to derive evolutionary relationships. These programs permit the conservation of protein structural information throughout the entire proposed methodology. Therefore, additional data will likely demonstrate that the methodology and software proposed herein is likely capable of superiorly deriving evolutionary relationships of distant homologs.

APPENDIX A

GENERIC SORTING ALGORITHM

The following algorithm will delete numbers from an array that do not constitute the general ascending order of the numbers contained therein. The input for the algorithm is an array whose elements are numbers in a general ascending order. However, throughout this array are numerical elements that disrupt the general ascending order of the numbers. The algorithm will determine which numerical elements compose the general order and which ones need to be deleted. The final (output) array will contain only those elements retained. Furthermore, the final array will be in ascending order and will comprise fewer elements than the Input Array. Although removing unordered array elements sounds simple, the difficulty lies in establishing which elements comprise the general ascending order—a seemingly subjective task.

The algorithm functions by copying the Input Array into a second array called the Sorted Array. The Sorted Array is then sorted into ascending order, thus moving all those elements that are not in order already. The algorithm then measures how many indexes each element moves after being sorted using the following equation:

$$Distance = (Index\ of\ Input\ Array - Index\ of\ Sorted\ Array)^2$$

The number of indexes moved is squared to eliminate negative distances. The numerical element(s) that moves the most (i.e., possesses the greatest Distance) is deleted from the Input Array. Using the refined Input Array, a new Sorted Array is generated and the process is repeated until the Input Array matches the Sorted Array (i.e., the Input Array is in perfect ascending order). Importantly, for each element that is deleted, the remaining

elements will shift more towards their Sorted Array positions. Therefore, even though all of the elements may have a Distance greater than zero(0) for the first iteration, these Distances will approach zero(0) as more elements are deleted.

The following are additional notes; however, because this algorithm is a generic algorithm, not all of the following notes apply to the UniTS program:

1. Duplicate numbers in the Input Array are permitted and are not deleted from the Input Array because the previous duplicate number is not less than the subsequent duplicate number (although they may still be deleted for compromising the general order relative to the other numbers).
2. The elements may contain decimal digits as well as integers.
3. The numbers within the elements do not have to be contiguous.
4. Finally, having a Distance of one(1) means that an element is only one index away from its ideal location in the Sorted Array, thus indicating than it must be switched with an adjacent element (which will also have a Distance of 1). Because both elements are switched, the algorithm is unable to determine which element will remain in the Input Array and which will be deleted; therefore, the algorithm will delete both elements.

APPENDIX B

TEMPLATE PROTEIN SELECTION

The **mean center** (c_μ) of a protein is the average of all the atomic coordinates that compose the protein:

$$c_\mu = \left(\frac{\sum x_n}{n_{atoms}}, \frac{\sum y_n}{n_{atoms}}, \frac{\sum z_n}{n_{atoms}}\right)$$

where n_{atoms} is the total number of atoms in the protein. UniTS selects the template protein as the protein that possesses the least **mean center error distance**, which is the distance between the c_μ of each input protein and the c_μ of all the proteins combined (calculated by averaging all the c_μs).

APPENDIX C

UNITS AND CHIMERA SDSAS

Isocitrate Dehydrogenase SDSAs

```
>Chimera_1T09.pdb

MSKKI----------------------------SGGSVVEMQGDEMTRIIWELIKEKL-I
FP---YV-------ELDLHSYDLGIENRD--AT-N-DQ-VTKDAAEAIKKHNVGVKCATI
TPDEKRVEEFKLKQMW-----KSPNGTIRN-ILGGTVFREAIICKNIPRLVSGWV-KPII
IGRHAYGDQ----YRATDFVVPGPGK--------V--EITYTPSDGTQKV--TYLVHN-F
EEGGGVAM-GMYNQDKSIEDFAHSSFQMAL-SKGW-PLYLSTKNTILKKYDGRFKDIFQE
IYDKQYK-SQF-----------------EAQKIWYEHRLIDDMVAQAMKSEGG--FIWAC
K-NYDGDVQSDSVAQG----------Y--G-SLGMMTSVLVCPDGKTVEAEAAHGTVTRH
YRM-YQKGQETSTNPIASIFAWTRGLA-HRAKLDNNKELA-FFANALEEVSIETIEAGFM
TKDLAACIKGLPNVQRS--D-YLNTFEFMDKLGENLKIKLAQAKL---------

>Chimera_1XGV.pdb

-----SPPCTTEELSPPPGGSLVEYSGGSLRVPDNPVVAFIRGDGVGPEVVESAL-KVVD
--AAVK-KVYGGSRRIVWWELLAGHLA-REKC-GELLPKATLEGIRL---ARVALKGPLE
TPV-------------GTGYRSL-NVAIRQALDLYANIRPVRYYGQPA-PHKYADRVDMV
IFRENT---EDVYAGIEWPH------DSPEAARIRRFL------------AEEFGIS-IR
---EDAGIGVKPISRFATRRLMERALEW-ALRNGNTVVTIMHKGNIMKYTEGAFMRWAYE
VALEKFREH-VVTEQEVQEKYGGVRPEGK---ILVNDRIADNMLQQIITRPW-DYQVIVA
PNL-------------NGDYISDAASALVGGIG-MAAGMNMG-D-GIAVAEPVHGTAPKY
A--GK-----DLINPSAEILSASLLIGEFM-G-------WREVKSIVEYAIRKAVQSKKV
TQDLAR-HM--------PGVQPLRTSEYTETLIAYIDEA--DL—NEVLAGKRG

>UniTS_1T09.pdb

----MSKKIS---GGSVVEMQG-----------------DEMTRIIWELIKEKL-IFP-Y-
--------VELDLHSYDLGIENRDATNDQ-VTKDAAEAIKKHNVGVKCATITPDEKRVEE
FKLKQMWKSPNGTIRNILG-GTVFREAIICKNIPRLVSGWVKP-III-GRHAYGDQYRAT
-DFV-VPGPGK-V--EITYTPSDGTQKVTYLVHNFEEGGGVAMGMYNQD-KSIEDFAHSS
FQMALSKGW-PLYLSTKNTILKKYDGRFKDIFQEIYDKQYKSQFEAQK------------
--IWYEHRLIDDMVAQAMKSEGG-FIWACKNYDGDVQSDSVAQGYGSLGMMTSVLVCPDG
KTVEAEAAHGTVTRHYRMYQKGQETSTNPIASIFAWTRGLAHRAKLDNNKELA-FFANAL
EEVSIETIEAGFMTKDLAACIKGLPNVQRSDYLNTFEFMDKLGENLKIKLAQA-------
KL

>UniTS_1XGV.pdb

SPPCTTEELSPPPGGSLVEYSGGSLRVPDNPVVAFIRGDGVGPEVVE-SALKVVDAAVKK
VYGGSRRIVWWELLAGHLAREKCGELLPKAT-LE---GIRLARVALKGPLETPV------
---GTGYRSLNVAIRQALDLYANIRPVRY-YGQPAPHKYADRVDMVIFR-ENTEDVYAGI
EWPHDSPEAARIRRFL--------AEE--FGISIRE---DAGIGVKPISRFATRRLMERA
LEWALRNGNTVVTIMHKGNIMKYTEGAFMRWAYEVALEKFREHVVTEQEVQEKYGGVRPE
GKILVNDRIADNMLQQIITRPWDYQVIVAPNLNGDYISDAASALVGGIGMAAGMNMG-DG
-IAVAEPVHGTAPKYAG-KDL-----INPSAEILSASLLIGEFMG-------WREVKSIV
EYAIRKAVQSKKVTQDLAR---HMPGVQ---PLRTSEYTETLIAYIDEA--DLNEVLAGK
RG
```

Pectate Lyase SDSAs

>Chimera_1PLU.pdb

------ATDT--GGYAA----TAGGNVTGA---VSKTATSMQDIVNIIDAA---RLDANG
KKVKGGAYPLVITYTGNEDSLINAAAANICGQWS----------K----------
----------DP--RGVEIKEFTKGITIIGAN-GSSAN-FGI
WIKKSSDVVVQNMRIGYLPGG----------AKDGDMIRVDDSPNVWVDHNELF
A---ANHE--C-DG----TPDNDTTFESAVDIKGASNTVTVSYNYIHGVKKVGLDGSSSS
D--TG--RNITYHHNYYNDVNARLPLQRGGLVHAYNNLYTNI--------TGSGLNVRQN
GQALIENNWFEKA--I---NPVTSRYDGKNFGTWVLKGNN---ITKPADFSTYSITWTAD
TKPYVNADSW--TS----TGTF-PTVAYNYSPVSAQCVKDKLPGYAGVGKNLATLTSTAC
-

>Chimera_2BSP.pdb

ADLGHQTLG-SNDGWGAYSTGTTGGSK--ASSSNVYTVSNRNQLVSALG--KETN-----
-----T-TPKIIYIKGTI----------DMNVDDNLKPLGLNDYKDPEYDLDKY
LKAYDPSTWGKKEPSGTQEEARARSQKNQKARVMVDI--PA-NTTIVGSGTNAKVVGGNF
QIK-SDNVIIRNIEFQDAYD-YFPQWDPTDGSSGNWNS-QYDNITINGGTHIWIDHCTFN
DGSRP-D-STSPKYYGRKY----QHHDGQTDASNGANYITMSYNYYHDHDKSSIFGSSDS
KTSDDGKLKITLHHNRYKNIVQKAPRVRFGQVHVYNNYYEG-STSSSSYPFSYAWGIGKS
SKIYAQNNVIDV-PGLSAAKTISVF---SGGTALYDSGTLLNGTQI--------------
---------NASA-ANGLSSSVGWTPSLHGSIDASANVKSNVINQAGAGKL---------
N

>UniTS_1PLU.pdb

AT------DT-GGYAA----TAGGNVTGA---VSKTATSMQDIVNIIDAARLDANGKKVK
GGAYPLVITYTGN-----EDSLINAAAANI----------CGQWSK---------
--------DP--RGVEIKEFTKGITIIGAN-GSSANFGIWIKKSSDVVVQNMRIGYLPG-
-----------G---AKDGDMIRVDDSPNVWVDHNELFA---ANHEC-DG----TPDNDT
TFESAVDIKGASNTVTVSYNYIHGVKKVGLDGS---SSSDTGR-NITYHHNYYNDVNARL
PLQRGGLVHAYNNLYTNI-------TGSGLNVRQNGQALIENNWFEK---AINPVTSRYD
GKNFGTWVLKGNNITKPADFSTYSITWTADTKPYVNADSWTSTGTF-PTVAYNYSPVSAQ
CVKDKLPGYAGVGKNLATLTSTAC

>UniTS_2BSP.pdb

ADLGHQTLGSNDGWGAYSTGTTGGS--KASSSNVYTVSNRNQLVSAL--------GKETN
T--TPKIIYIKGTIDMNVDDNLKPLGLNDYKDPEYDLDKYLKAYDPSTWGKKEPSGTQEE
ARARSQKNQKARVMVDIPA---NTTIVGSGTNAKVVGGNFQIKSDNVIIRNIEFQDAYDY
FPQWDPTDGSSGNWNSQY-DNITINGGTHIWIDHCTFNDGSRPDSTSPKYYGRKYQH---
-HDGQTDASNGANYITMSYNYYHDHDKSSIFGSSDSKTSDDGKLKITLHHNRYKNIVQKA
PRVRFGQVHVYNNYYEGSTSSSSYPFSYAWGIGKSSKIYAQNNVIDVPGLSAAKTISVF-
-SG-GTALYDSGTLLNGTQINASAANGL--------------SSSVGWTPSLHGSIDASA
NVKSNVINQAGAGKL--------N

Polygalacturonase SDSAs

>Chimera_1CZF.pdb

DS--CTFTTAAAAKAG-KAKCSTITLNNIEVPAGTTLDLTGLTSGTKVIFEGTTTFQYEE
WA-GPLISMSGEHITVTGASGHLINCDGARWWDGKGTS--GKKKP-KFFYAHGLDS-SSI
TGLNIKNTPLMAFSVQ-ANDITFTDVTINNADGDTQ--------GGHNTDAFDVGNSVGV
NIIKPWVHNQDDCLAVNSGENIWFTGGTCIGGHGLSIGSVGDRSNNVVKNVTIEHSTVSN
SENAVRIKTISGATGSVSEITYSNIVMSGISDYGVVIQQDYEDGKPTGKPTNGVTIQDVK
LESVTGSVDSGATEIYLLCGSGSCSDWTWDDVKVTGG-KKSTACKNFPSVA--SC-

>Chimera_1HG8.pdb

--DPCSVTEYSGL-ATAVSSCKNIVLNGFQVPTGKQLDLSSLQNDSTVTFKGTTTFATTA
DNDFNPIVISGSNITITGASGHVIDGNGQAYWDGKGSNSNSNQKPDHFIVVQKTTGNSKI
TNLNIQNWPVHCFDITGSSQLTISGLILDNRAGDKPNAKSGSLPAAHNTDGFDISSSDHV
TLDNNHVYNQDDCVAVTSGTNIVVSNMYCSGGHGLSIGSVGGKSDNVVDGVQFLSSQVVN
SQNGCRIKSNSGATGTINNVTYQNIALTNISTYGVDVQQDYLNGGPTGKPTNGVKISNIK
FIKVTGTVASSAQDWFILCGDGSCSGFTFSGNAITGGGKTSS-CN-YPT—NTCPS

>UniTS_1CZF.pdb

DSCTFTTAAAAKAG-KAKCSTITLNNIEVPAGTTLDLTGLTSGTKVIFEGTTTFQYEEWA
GP-LISMSGEHITVTGASGHLINCDGARWWDGKGTS--GKKKP-KFFYAHGLDS-SSITG
LNIKNTPLMAFSVQ-ANDITFTDVTINNADGDTQG--------GHNTDAFDVGNSVGVNI
IKPWVHNQDDCLAVNSGENIWFTGGTCIGGHGLSIGSVGDRSNNVVKNVTIEHSTVSNSE
NAVRIKTISGATGSVSEITYSNIVMSGISDYGVVIQQDYEDGKPTGKPTNGVTIQDVKLE
SVTGSVDSGATEIYLLCGSGSCSDWTWDDVKVTGG-KKSTACKNFPSVASC—

>UniTS_1HG8.pdb

DPCSVTEYSGL-ATAVSSCKNIVLNGFQVPTGKQLDLSSLQNDSTVTFKGTTTFATTADN
DFNPIVISGSNITITGASGHVIDGNGQAYWDGKGSNSNSNQKPDHFIVVQKTTGNSKITN
LNIQNWPVHCFDITGSSQLTISGLILDNRAGDKPNAKSGSLPAAHNTDGFDISSSDHVTL
DNNHVYNQDDCVAVTSGTNIVVSNMYCSGGHGLSIGSVGGKSDNVVDGVQFLSSQVVNSQ
NGCRIKSNSGATGTINNVTYQNIALTNISTYGVDVQQDYLNGGPTGKPTNGVKISNIKFI
KVTGTVASSAQDWFILCGDGSCSGFTFSGNAITGGGK-TSSCN-YPT-NTCPS

Hemopexin Repeats SDSAs

>Chimera_1QHU_(residues_56-134).pdb

HRGIRELISERWKNFIGPVDAAFRHGHTSVYLIKGDKVWVYT-S---------PKSLQDE
FPGI----PFPLDAAVEC--HRGECQDEGILFFQG----

>Chimera_1QHU_(residues_263-353).pdb

-GWHSWPIAHQWPQGPSTVDAAFSWE-DKLYLIQDTKVYVFLTKGGYTLVNGYPKRLEKE
LGSPPVISLEAVDAAFVCPGS------SRLHIMAGRRLW

>UniTS_1QHU_(residues_56-134).pdb

HRGI-RELISERWKNFIGPVDAAFRHGHTSVYLIKGDKVWVYT-S---------PKSLQD
EFPG---I-PFPLDAAVEC--HRGECQDEGILFFQG----

>UniTS_1QHU_(residues_263-353).pdb

--GWHSWPIAHQWPQGPSTVDAAFSWE-DKLYLIQDTKVYVFLTKGGYTLVNGYPKRLEK
ELGSPPVISLEAVDAAFVCPGS------SRLHIMAGRRLW

REFERENCES

Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000,** *28* (1), 235-242. www.pdb.org.

Bowie, J. U.; Lüthy, R.; Eisenberg, D. A method to identify protein sequences that fold into a known three-dimensional structure. *Science* **1991,** *253* (5016), 164-170.

Deeds, E. J. A structure-centric view of protein evolution, design, and adaptation. In *Advances in Enzymology and Related Areas of Molecular Biology;* Toone, E. J., Ed.; Protein Evolution, Vol. 75; John Wiley & Sons: Hoboken, NJ, 2007; pp 133-191.

Dixon, H. B. F.; Cornish-Bowden, A.; Liébecq, C.; Loening, K. L.; Moss, G. P.; Reedijk, J.; Velick, S. F.; Vliegenthart, J. F. G. Nomenclature and symbolism for amion acids and peptides. *Pure & Appl. Chem.* **1984,** *56* (5), 595-624.

Edgar, R. C. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics* [Online] **2004a,** *5* (113). DOI: 10.1186/1471-2105-5-113.

Edgar, R. C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **2004b,** *32* (5), 1792-1797.

Edgar, R. C. *MUSCLE User Guide;* User Guide for the MUSCLE Sequence Alignment Program, 2010; Version 3.8. http://www.drive5.com/muscle/.

Edgar, R. C.; Sjolander, K. A comparison of scoring functions for protein sequence profile alignment. *Bioinformatics* **2004,** *20* (8), 1301-1308.

Eldredge, N.; Gould, S. J. Punctuated Equilibria: An alternative to phyletic gradualism. In *Models in Paleobiology*; Schopf, T. J. M., Ed.; Freeman Cooper: San Francisco, 1972; pp 82-115.

Ewens, W.; Grant, G. *Statistical Methods in Bioinformatics: An Introduction,* 2nd ed.; Springer: New York, 2005.

Gibas, C.; Jambeck, P. *Developing Bioinformatics Computer Skills,* 1st ed.; O'Reilly: Sebastopol, CA, 2001.

Glasner, M. E. Mechanisms of protein evolution and their application to protein engineering. In *Advances in Enzymology and Related Areas of Molecular Biology;* Toone, E. J., Ed.; Protein Evolution, Vol. 75; John Wiley & Sons: Hoboken, NJ, 2007; pp 193-239.

Gonnet, G. H.; Cohen, M. A.; Benner, S. A. Exhaustive matching of the entire protein sequence database. *Science* **1992,** *256* (5062), 1443-1445.

Guex, N.; Peitsch, M. C. SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling. *Electrophoresis* **1997,** *18* (15), 2714-2723.

Holm, L.; Rosenstrom,P. Dali server: conservation mapping in 3D. *Nucleic Acids Res.* **2010,** *38,* 545-549.

Holm, L; Sander, C. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol*. **1993,** *233* (1), 123-138.

Hong, Y.; Ko, K. D.; Bhardwaj, G.; Zhang, Z.; van Rossum, D. B.; Patterson, R. L. Towards solving the inverse protein folding problem. *Physics Archives* **2010,** Aug. http://arxiv.org/abs/1008.4938.

Huelsenbeck, J. P.; Ronquist, F. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics* **2001,** *17* (8), 754-755.

Isaev, A. *Introduction to Mathematical Methods in Bioinformatics;* Springer: Berlin, 2006.

Jones, D. T.; Taylor, W. R.; Thornton, J. M. The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci*. **1992,** *8* (3), 275-282.

Kedem, K.; Chew, P.; Elber, R. Unit-vector RMS (URMS) as a tool to analyze molecular dynamics trajectories. *Proteins: Struct., Funct., and Genet.* **1999,** *37* (4), 554-564.

Kim, C.; Lee, B. Accuracy of structure-based sequence alignment of automatic methods. *BMC Bioinformatics* [Online] **2007,** *8* (355). DOI:10.1186/1471-2105-8-355.

Konagurthu, A. S.; Whisstock, J. C.; Stuckey, P. J.; Lesk, A. M. MUSTANG: a multiple structural alignment algorithm. *Proteins* **2006,** *64* (3), 559-574.

Krane, D. E.; Raymer, M. L. *Fundamental Concepts of Bioinformatics;* Benjamin Cummings: San Francisco, 2003.

Kuzlemko, A.; Honig, B.; Petrey, D. Using structure to explore the sequence alignment space of remote homologs. *PLoS Comput. Biol.* [Online] **2011,** *7* (10), e1002175. DOI: 10.1371/journal.pcbi.1002175.

Marín-Rodríguez, M. C.; Orchard, J.; Seymour, G. B. Pectate lyases, cell wall degradation and fruit softening. *J. Exp. Bot.* **2002,** *53* (377), 2115-2119.

Marti-Renom, M. A.; Stuart, A. C.; Fiser, A.; Sánchez, R.; Sali, A. Comparative protein structure modeling of genes and genomes. *Annu. Rev. Biophys. Biomol. Struct.* **2000,** *29,* 291-325.

Mechelke, M.; Habeck, M. Robust probabilistic superposition and comparison of protein structures. *BMC Bioinformatics* [Online] **2010,** *11* (363). DOI: 10.1186/1471-2105-11-363.

Meng, E. C.; Pettersen, E. F.; Couch, G. S.; Huang, C. C.; Ferrin, T. E. Tools for integrated sequence-structure analysis with UCSF Chimera. *BMC Bioinformatics* [Online] **2006,** *7* (339). DOI: 10.1186/1471-2105-7-339.

Menke, M.; Berger, B.; Cowen, L. Matt: local flexibility aid protein multiple structure alignment. *PLOS Comput. Biol.* **2008,** *4* (1), 88-99.

Micheletti, C.; Orland, H. MISTRAL: a tool for energy-based multiple structural alignment of proteins. *Bioinformatics* **2009,** *25* (20), 2663-2669.

Ortiz, A. R.; Strauss, C. E. M.; Olmea, O. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci*. **2002,** *11* (11), 2606-2621.

Orwant, J.; Hietaniemi, J.; Macdonald, J. *Mastering Algorithms with Perl;* O'Reilly: Sebastopol, CA, 1999.

Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E. UCSF Chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem.* **2004,** *25* (13), 1605-1612.

Rost, B. Twilight zone of protein sequence alignments. *Protein Eng*. **1999,** *12* (2), 85-94.

Stewart, J. *Calculus,* 5th ed.; Thomson Learning: Belmont, CA, 2003.

Tamura, K.; Peterson, D.; Peterson, N.; Stecher, G.; Nei, M.; Kumar, S. MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods. *Mol. Biol. Evol.* **2011,** *28* (10), 2731-2739.

Theobald, D. L.; Wuttke, D. S. Empirical Bayes hierarchical models for regularizing maximum likelihood estimation in the matrix Gaussian Procrustes problem. *Proc. Natl. Acad. Sci.* **2006a,** *103* (49), 18521-18527.

Theobald, D. L.; Wuttke, D. S. THESEUS: maximum likelihood superpositioning and analysis of macromolecular structures. *Bioinformatics* **2006b,** *22* (17), 2171-2172.

Theobald, D. L.; Wuttke, D. S. Accurate structural correlations from maximum likelihood superpositions. *PLoS Comput. Biol.* [Online] **2008,** *4* (2), e43. DOI: 10.1371/journal.pcbi.0040043.

Voet, D.; Voet, J. G. *Biochemistry,* 3rd ed.; John Wiley and Sons: Hoboken, NJ, 2004.

Wang, X.; Dong, J. A Normalized weighted RMSD for measuring protein structure superposition. *IPCBEE* **2012,** *34,* 68-72.

Weisstein, E. W. "Ellipsoid" From *MathWorld* – A Wolfram Web Resource. http://mathworld.wolfram.com/Ellipsoid.html (accessed June 7, 2013).

Yang, A. S. Structure-dependent sequence alignment for remotely related proteins. *Bioinformatics* **2002,** *18* (12), 1658-1665.

VITA

Scott G. Foy was born in Voorhees, New Jersey on January 24, 1982. When he was young, Mr. Foy's family moved Mountain View, Missouri, where he spent his childhood. In 2000, Mr. Foy graduated with honors from Liberty High School in Mountain View. He later attended college at Southwest Baptist University in Bolivar, Missouri. In 2005, Mr. Foy received a Bachelor of Science in Psychology with emphases in both Counseling and Premedicine. However, he decided to pursue neither a Doctor of Psychology nor a Doctor of Medicine (Psychiatry). Instead, Mr. Foy decided to pursue a degree in biology and was accepted into the selective Truman State University in Kirksville, Missouri, where he received a Bachelor of Arts in Biology in 2007.

Following his graduation in 2007, Mr. Foy initially attended the University of Missouri-Kansas City in pursuit of a Master of Science in Cell and Molecular Biology with an emphasis in Bioinformatics. His thesis was to be completed in the evolutionary genetics and bioinformatics laboratory of Dr. Gerald Wyckoff. However, Mr. Foy enjoyed his thesis experience so much that he decided to remain in Dr. Wyckoff's laboratory to complete his dissertation in the pursuit of a Doctor of Philosophy. Following the reception of a Doctor of Philosophy in Molecular Biology and Biochemistry degree in 2013, Mr. Foy has accepted a postdoctoral position in the laboratory of Dr. Joanna Masel at the University of Arizona.

Mr. Foy is an Eagle Scout and was a member of both Psi Chi (psychology) and Beta Beta Beta (biology) undergraduate honor societies. Mr. Foy received the Graduate Teaching Assistant Superior Teaching Award in 2013.