BCL::SAXS - SMALL ANGLE X-RAY SCATTERING PROFILES TO ASSIST PROTEIN STRUCTURE PREDICTION

By

Daniel Kent Putnam

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Biomedical Informatics

May, 2013

Nashville, Tennessee

Approved:

Jens Meiler Ph.D

Terry Lybrand Ph.D

David Tabb Ph.D

To my parents, L. Kent and Shauna Putnam and grandparents, Max and Louise Putnam

To my amazing children, Amelia, Max, Shauna, Jordan, and Spencer

To my treasured wife Marti - my eternal companion

To the Lord Jesus Christ

ACKNOWLEDGEMENTS

This work was funded by the National Library of Medicine Training Grant 5T15LM007450-09. Work in the Meiler laboratory is supported through NIH (R01 GM080403, R01 MH090192, R01 GM099842) and NSF (Career 0742762, OIA 0959454). I am especially indebted to Dr. Cindy Gadd, Director of Graduate Studies for Biomedical Informatics and Dr. Kevin Johnson, Chair of Biomedical Informatics at Vanderbilt and my committee members, Dr. Terry Lybrand and Dr. David Tabb. They have encouraged, advised and led me along this path.

I want to thank members of the Meiler Lab – Dr. Edward W. Lowe for his help with GPU acceleration, Brian Weiner for his help in programming and introduction to the lab, Mariusz Butkeivicz and Jeff Mendenhall for their insight and assistance throughout the development of BCL::SAXS. I want to thank the Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt for computer time on their cluster. I would especially like to thank Dr. Jens Meiler, the chairman of my committee. As my teacher and mentor he has taught me through word and action the type of scientist I would like to become.

My family has been critical to my success in this journey. My wife Marti has been my constant companion and my children Amelia, Max, Shauna, Jordan, and Spencer have been my cheerleaders and have made sacrifices of “Daddy” time enabling me to complete this work. I recognize the hand of the Lord in my life and give gratitude to my Savior Jesus Christ. “That which is of God is light; and he that receiveth light, and continueth in God, receiveth more light; and that light growth brighter and brighter until the perfect day.” (Doctrine and Covenants 50:24)

TABLE OF CONTENTS

Page

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF TABLES vi

LIST OF FIGURES vii

LIST OF ABBREVIATIONS viii

Chapter

I. INTRODUCTION 1

Introduction and Study Overview 1
Protein Structure 1
Limitations of Protein Structure Determination 3
Experimental Restraints Combined with Computational Methods 5
BCL::Fold Pushes Size and Complexity Limits in *de Novo* Protein Structure Prediction 6
Small Angle X-Ray Scattering (SAXS) 8
SAXS as an experimental restraint for protein structure prediction 10
BCL::SAXS is designed specifically to incorporate SAXS restraints into BCL::Fold 11
Overall Approach 12

II. RELATED WORK 13

III. ALGORITHM DEVELOPMENT 17

Generate SAXS profiles from rigid protein bodies 17
Initial Validation of BCL::SAXS 19
GPU Parallel processing to accelerate algorithm 20
GPU Acceleration Yields Orders of Magnitude Speed Improvements 22

IV. ALGORITHM APPROXIMATIONS 23

Approximate SAXS profiles for protein models with missing side chains and loops 22
Vector calculations to approximate path directions between two SSEs 26
Pathway calculations for loop approximations 27
Triangle Approximation for Normalization Factor Calculation 28
Regula Falsi Approximation for Normalization Factor Calculation 29
Arc Length Calculations 29
Apply Arc Length Calculations Generally 33

Numerically Implement Arc Length Calculations 35
One Dimensional Optimization 36
Regula Falsi Method 36

V. EVALUATION OF ALGORITHM AND APPROXIMATIONS 39

Generation of SAXS Profiles from Atomic Coordinates with CRYSOL 39
SAXS Profile Analysis 39
Model similarity assessed by dRMSD SAXS score 42
Alternative Methods to Compare SAXS Profiles 44
Accuracy of calculated SAXS profiles 45
Non-redundant Dataset for protein discrimination benchmark 46
SAXS dRMSD Score Analysis 48
Selecting the Optimal Loop Approximation Algorithm 49
Structural Similarity of Proteins with Similar SAXS Scores 50
SAXS Restraints Incorporated into BCL::Fold during Protein Folding 52

VI. DISCUSSION 55

VII. CONCLUSION 60

Summary 60
Study Limitations 60
Future Work 61

APPENDIX I: 1ENH C_α- C_α spacing 62

APPENDIX II: SAXS profiles before and after the Debye formula correction 64

APPENDIX III: SAXS profile with Midpoint Loop Approximation 65

APPENDIX IV: Logarithmic and Derivative SAXS Profiles 66

APPENDIX V: BCL::SAXS Benchmark set of 455 proteins 67

APPENDIX VI: Loop and Normalization Factor Optimization with Benchmark Set 68

APPENDIX VII: BCL::SAXS Command lines 71

BIBLIOGRAPHY 72

LIST OF TABLES

Table Page

1. The Cromer-Mann coefficients for scattering factors 18

2. Excluded volume, radius, and bound hydrogen count by atom type 19

3. Timing Results of GPU vs. CPU benchmarks 22

4. SAXS profile scores without derivative function 42

5. SAXS profile scores with derivative function 42

6. SAXS profile scores with Stovgaard function 43

7. SAXS profile scores with Cumulative Integral function 43

8. Replication of Stovgaard (S) score 44

9. Protein Folding Statistics 53

LIST OF FIGURES

Figure Page

Figure 1: Amino Acids and Peptide Bonds 2

Figure 2: Diagram of BCL::Fold 6

Figure 3: Experimental Solution Scattering from 3HZ7 9

Figure 4: Initial SAXS profiles with BCL::SAXS 19

Figure 5: Corrected SAXS profiles with BCL::SAXS 20

Figure 6: Schematic of GPU Acceleration 22

Figure 7: Diagram of Loop Approximation Methods 23

Figure 8: Scoring Matrix of Single atom vs. Linear Loop Approximation 24

Figure 9: Depiction of Parabolic Height Approximation 28

Figure 10: Linear and Curved path between SSEs 29

Figure 11: Regula Falsi Optimization 37

Figure 12: Original and Normalized SAXS Profiles 40

Figure 13: Logarithmic SAXS Profile 41

Figure 14: SAXS Profiles With and Without Derivative Scores 43

Figure 15: BCL::SAXS comparison with CRYSOL for different protein models 46

Figure 16: ROC analysis of benchmark set 47

Figure 17: Range of SAXS dRMSD Scores from Benchmark Set 48

Figure 18: Loop Approximation Results 49

Figure 19: MAMMOTH Z-score vs. SAXS similarity score 51

Figure 20: RMSD100 vs. dRMSD SAXS score 51

Figure 21: Folding Results for 1000 models with and without SAXS restraint 53

LIST OF ABBREVIATIONS

Cα ..... α-carbon on the backbone of the amino acid

CPU..... central processing unit

CRYOEM ..... cryo electron microspcopy

dCα..... distance between two α-carbons on the protein in angstroms(Å)

EPR .....electron paramagnetic resonance

GPU .....graphical processing unit

KBP ..... knowledge based potential

MC.....monte carlo

NMR .....nuclear magnetic resonance

PDB..... protein data bank

PSP ..... protein structure prediction

RMSD.....root mean square distance

SAXS ..... Small Angle X-Ray Scattering

SSE..... secondary structure element

CHAPTER I

INTRODUCTION

Overview

The objectives of this thesis are (1) to develop an algorithm to derive small angle X-ray (SAXS) scattering profiles from atomic coordinates; (2) to develop a scoring function comparing experimental SAXS profiles with profiles containing approximated loop and side chain regions for integration with BCL::Fold; (3) to benchmark the score against different protein models (4)to determine a suitable weight for the SAXS score implemented in the consensus knowledge-based scoring function; and (5) to evaluate the improvement of native-like sampling.

Protein Structure

The primary structure of a protein is formed by a long chain of amino acids. There are 20 natural types of amino acids that are distinguished by different properties. These properties include hydrophobicity, hydrophilicity, aromaticity, aliphaticity, size, charge, and the presence of hydroxyl groups. An amino acid is composed of backbone atoms and a side chain unit. The coordinate location of the backbone atoms, comprising a constant sequence of nitrogen, carbon, carbon, and oxygen, are influenced by the interactions of the differing side chain units. These side chain units, referred to as “R” groups, are responsible for the specific properties of a given amino acid. The linear sequence of amino acids is held together by covalent peptide bonds. Peptide bonds are formed at the c-terminus of amino acid one and the n-terminus of

amino acid two. (See figure 1) Peptide bond formation causes the release of water.

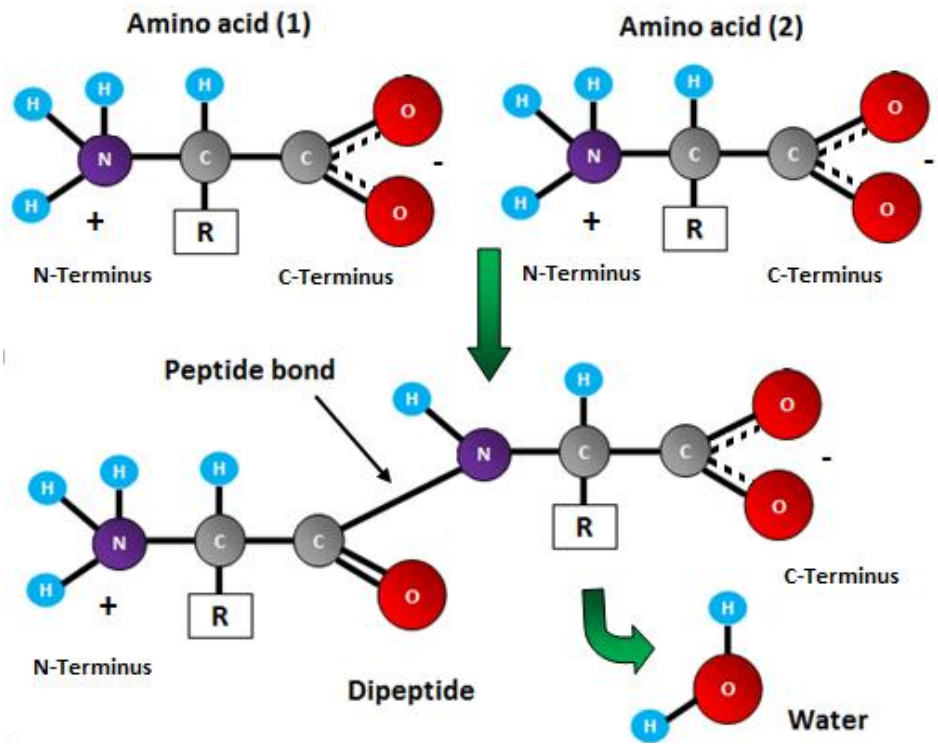


Figure 1: Amino acids and peptide bonds. The backbone structure of amino acids is depicted. (Top) The structure of the R group distinguishes the amino acids from each other. The formation of a dipeptide with a peptide bond is shown with the release of water (Bottom)

As the peptide chain grows in length, the side chain and backbone residues interact (exert forces) on one another thus influencing their position in Euclidean space. The main force for folding water soluble globular proteins is to pack hydrophobic side chains into the core of the protein forming a hydrophobic interior and a hydrophilic exterior surface. For this packing to be possible the backbone polar groups (NH, C'=O) must be neutralized by the formation of hydrogen bonds. This is accomplished by the formation of two types of secondary structure elements (SSEs): α helices or β sheets.

A standard α-helix is built from a continuous region of the peptide chain and has 3.6 residues per turn with hydrogen bonds between C'=O of residue n and NH of residue n+4. Although rare, other configurations of the α-helix exist. If it is more tightly coiled the hydrogen bonding occurs at n+3 (3_{10} helix), and if the α-helix is more loosely coiled, the hydrogen bonding

occurs at n+5 (π helix). In each case all the NH and C'=O groups are joined with hydrogen bonds except at the n-terminus and c-terminus. The termini of the α-helix are polar and are found at the surface of proteins. The α-helix can range in length from four residues to over forty residues, while the average length is ten residues. With each turn of the helix there is a 1.5Å rise along the helical axis.

The other type of SSE is the β-sheet. Different from the α-helix, this structure is built from several regions of the polypeptide chain. These regions are called β-strands and vary in an extended configuration from 5 – 10 residues in length. They are aligned adjacent to each other such that the NH group of one strand forms a hydrogen bond with the C=O group of the adjacent strand. In this configuration, the strands can align in the same direction (parallel) or alternating directions (anti-parallel).

After the secondary structure of helices and sheets are formed, they fold together packing the hydrophobic residues towards the core of the protein forming tertiary structure [3]. At this stage, disulfide bonds and salt bridges form providing stability to the protein. From a linear strand of amino acids, secondary structure elements form and then fold in 3-dimensional space to build a distinct shape - the tertiary structure of a protein. Finally, multiple subunits bind together to form quaternary structure of the protein.

Limitations of Protein Structure Determination

The understanding of protein structure is important because protein structure determines protein function. In humans, proteins are tiny biological machines that act as antibodies, contractile proteins, enzymes, hormonal proteins, structural proteins, storage proteins and transport proteins. They are classified into different types depending on their tertiary structure. For example, collagen (a support protein) has a super-coiled helical shape resembling a rope, while hemoglobin is a spherical compact globular protein. This spherical

shape is ideal for travel through the bloodstream. Two of the main types of proteins are globular proteins and membrane proteins. Globular proteins are soluble and act as enzymes, while membrane proteins act as receptors that provide channels for polar molecules to pass through the cell membrane. In the human genome 75% of the proteins are soluble proteins while 25% are membrane proteins. Despite the discrepancy in the number of membrane proteins vs. soluble proteins, 50% of all pharmaceuticals target membrane proteins[4] while the other 50% target soluble proteins.

The protein databank (PDB) is a repository of the atomic coordinates of each atom in a protein relative to each other [5]. The protein structures were determined through methods such as X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy and deposited into the database by experimentalists. In the PDB, 98% of the proteins are soluble proteins, while 2% of the proteins are membrane proteins. This discrepancy is not due to chance. Protein structure determination remains a major challenge in the field of structural biology [6-9], particularly for large proteins with highly flexible loop regions such as membrane proteins. While X-ray crystallography and NMR spectroscopy can provide high resolution rigid body structures, these techniques are limited by size [10], high flexibility[11], and membrane environment[11].

Membrane proteins, for example, are too large for NMR spectroscopy and do not crystallize for X-ray crystallography. Other low resolution experimental techniques such as Cryo electron microscopy, electron paramagnetic resonance (EPR), and small angle X-ray scattering (SAXS) are used to gain insights about the structure of proteins. Isolated, these low resolution techniques are not sufficient to determine the atomic coordinates of each atom of a protein in Euclidean space. Because of these limitations, the field of computational structural biology emerged as a discipline. The fundamental challenge in computational structural biology is to

write a computer algorithm to accurately compute the 3-dimensional coordinates of the tertiary protein structure given the linear sequence of amino acids.

Experimental Restraints Combined with Computational Methods

Protein structure prediction methods are classified into *de novo* structure prediction techniques (without a template) and comparative modeling techniques (models built from similar protein structures) [12]. Template-based modeling identifies templates based on sequence similarity. The template structure is then used as a basis for building target protein structures. If no suitable templates can be found, then *de novo* structure prediction techniques are used. Template based modeling techniques have traditionally provided the best results in the **C**ritical **A**ssessment of protein **S**tructure **P**rediction (CASP) assessment held every two years. In 2012 at CASP10, this was no different. From this conference held in Gaeta Italy, one of the major concerns was that we still do not understand how to separate a good prediction (when the method works) from an erroneous protein structure prediction (when the method fails). This is difficult because the number of physical conformations available for a given protein sequence is vast. The task of determining the optimal conformation of a protein given a potential-energy function requires computational time that is exponential in the number of degrees of freedom in the protein [7]. In fact, Levinthal's paradox states that it would take more time than the age of the universe to systematically test each conformation[9]. The native structure is hypothesized to be the conformation with the lowest free energy[13]. The problem of protein structure prediction is transformed into a search for all possible conformations of an amino acid sequence on the free energy landscape for the conformation of lowest energy [14, 15]. Critical to the success of this task is to 1) use a realistic energy function that can accurately determine the free energy of a given confirmation[16] and 2) use an efficient method to search the energy landscape. Finding the global minimum of the energy function on the energy

landscape is challenging because the potential-energy surface of a protein contains many local minima. Currently, no practical method for optimizing the potential energy of a protein exists. Because of this limitation, we seek to combine computational algorithms with experimental constraints to restrict the overall search space.

BCL::Fold Pushes Size and Complexity Limits in *de Novo* Protein Structure Prediction

Using a Monte Carlo algorithm with Metropolis criteria in a simulated annealing environment, BCL::Fold was designed in the Meiler Lab to predict the structure of large proteins by assembling secondary structure elements (SSEs) [17, 18].

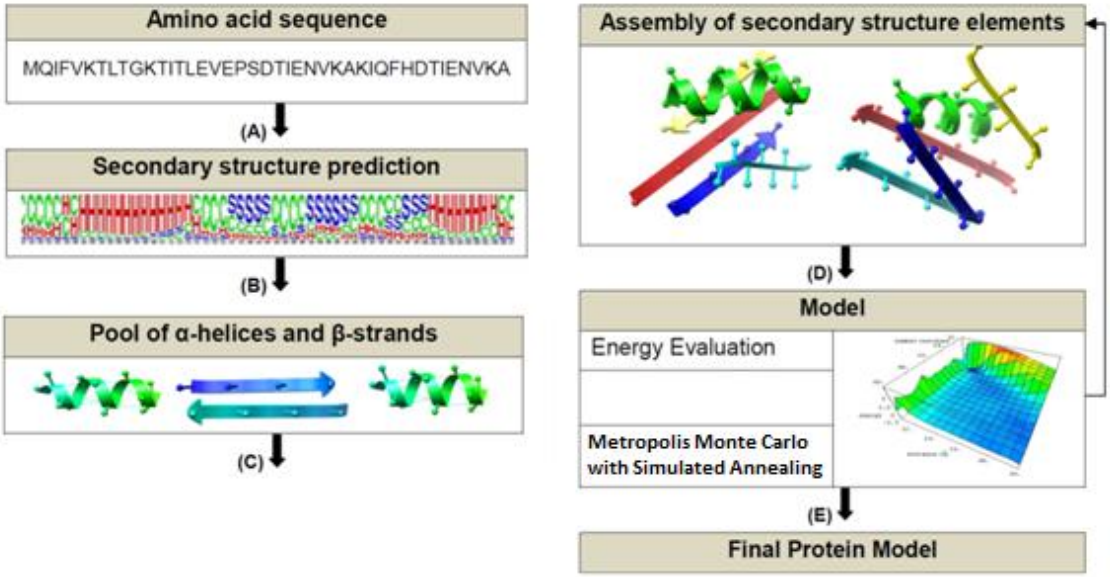


Figure 2: Diagram of the BCL::Fold Protocol

(A) BCL::Fold uses a primary amino acid sequence as input to identify a consensus secondary structure prediction.
(B) From secondary structure elements (SSE's) BCL::Fold generates a pool of candidate α-helices and β-sheets.
(C) The algorithm generates a proposed idealized protein model from the pool of secondary structure elements
(D) The proposed protein model is scored based on an energy potential function incorporating the BCL::SAXS score. The Metropolis Monte Carlo simulated annealing algorithm is used to sample the potential energy surface.
(E) Loop Regions and Side Chains added to the final SSE model

The advantages of this approach are 1) Low resolution / sparse experimental restraints are more readily available for SSEs and for backbone atoms; 2) The SSEs have well defined geometries and define the topology of a protein; 3) The Assembly of SSEs without the flexible loop regions facilitates the sampling of non-local residue contacts. This protein structure prediction algorithm is based on the placement of SSEs prior to the determination of side chain coordinates and loop region coordinates. The SSEs used in the BCL::Fold algorithm are built from a pool of predicted SSEs during five stages of assembly and one refinement stage. The algorithm begins with a random placement of candidate SSEs. The SSEs in the starting conformation are randomly moved to generate a new conformation.

Using a knowledge-based potential energy scoring function[19], the energy of each conformation generated is evaluated. If the new conformation or model is lower in energy than the current model, then it is accepted as the current model. If the new model is higher in energy then it is accepted with the Metropolis criteria – a probability dependent on a scaling factor given by the Boltzmann distribution. The Metropolis criterion provides a means to move uphill in the energy landscape and move out of local minima, thus more effectively sampling the search space of protein SSE conformations. After the SSE core of the protein is formed, the loop and side chains regions are added with established protocols such as ROSETTA [20, 21] to yield complete protein models.

Although the Metropolis criterion in BCL::Fold provides a way to test the energy landscape of SSE conformations more effectively, it cannot definitively identify the conformation in the lowest energetic state. To overcome computational limitations, hybrid methods (the combination of multiple experimental techniques) can be utilized to elucidate the structure of otherwise unsolved proteins [22-24]. To reduce the search space, experimental

restraints have been incorporated into the scoring function of BCL::Fold, including NMR, Electron microscopy density maps, and EPR. Another method gaining popularity in the structural biology community is the small angle X-ray scattering (SAXS) experimental method.

Small Angle X-ray Scattering (SAXS)

Small angle X-ray scattering (SAXS) is a sparse experimental structural characterization method for rapid analysis of biological macromolecules in solution [25-29]. SAXS is inherently a low resolution method because samples move freely in solution during data acquisition resulting in spherically averaged scattering intensity curves. To obtain a SAXS scattering profile, x-rays with a constant wavelength (λ) irradiate a purified protein sample in a ~1.0 mg/ml solution. As the X-rays collide with the sample, they scatter elastically. The scattered X-rays are captured on a detector as spots of varying intensity. The overall SAXS scattering profile is calculated by subtracting the scattering profile of the blank buffer solution from the profile of the sample dispersed in solution.

A SAXS scattering measurement represents a molecule's rotationally average intensity (I) as a function of scattering angle (q). In this representation, large pairwise atomic distances are represented by small scattering angles and small pair wise atomic distances are represented by large scattering angles. (See Figure 3) The information content of a SAXS profile is much less than other high resolution experimental techniques because the overall scattering curve represents the radially averaged contribution of all non hydrogen surface atoms in all orientations. Despite this limitation, several parameters can be extracted directly from the scattering curve which enables fast sample characterization. These parameters include the molecular mass (MM), radius of gyration (Rg), hydrated particle volume (Vp) and maximum

particle diameter (Dmax). Furthermore, the SAXS scattering curve contains information related to the overall shape of the molecule and is routinely used to validate structural models [30, 31].

The Guinier analysis is a rapid method to compute protein size, particle interactions (aggregation), oligomeric state, and overall data quality. First, the radius of gyration and forward scattering I(0) are easily obtained from a plot of ln[I(q)] vs. q^2. For monodisperse samples, this plot should be a linear line where the radius of gyration is the slope and the y intercept is I(0). If the Guinier plot is nonlinear, that may indicate inter-particle interactions, polydispersity, or improper background subtraction. The I(0) value normalized to solute concentration is proportional to the MM. The MM can be used to distinguish different oligomeric states.

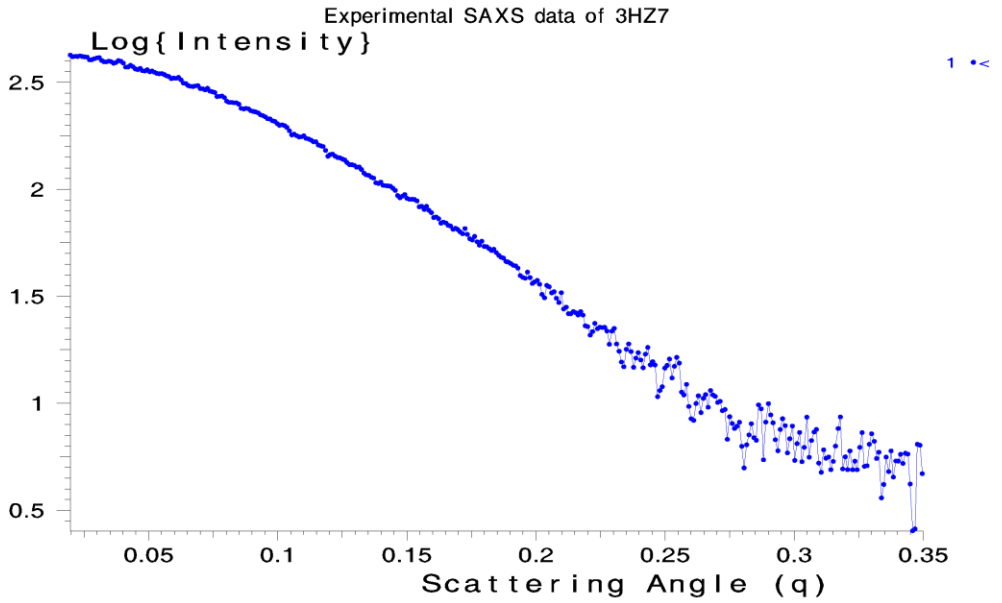


Figure 3: Experimental SAXS profile of a protein sample where the crystallographic structure is available. PDB ID: 3HZ7. This is a monomeric domain of a protein of unknown function. Each dot represents an intensity value for a given scattering angle.

After Guinier analysis, the distance distribution function, p(r), is computed by a Fourier transformation of the SAXS pattern. This function represents the SAXS data in real space and gives information about the overall shape of the macromolecule. For example, a spherical particle has a bell shaped p(r) curve. Direct Fourier transformation of the SAXS profile is not possible and indirect Fourier methods must be used. Because the original SAXS profile contains all of the information from which the p(r) curve is derived, we used SAXS profiles directly for analysis without computing the distance distribution function.

SAXS as an Experimental Restraint for Protein Structure Prediction

The combination of SAXS experimental data with computational protein structure prediction algorithms provides a potential opportunity to predict structures closer to the native topology [32-34]. SAXS profiles have been used to identify native-like protein models from a large set of alternative protein models [35, 36]. Furthermore, SAXS profiles have been used to reconstruct proteins in protein structure prediction algorithms. [27, 28, 30, 31, 37] Because the SAXS experimental technique represents proteins with spherically averaged election densities, multiple structures can be reconstructed from the same SAXS profile. To address this challenge, experimental data collected from SAXS is integrated with additional structural information to reduce the number of proteins models consistent with a SAXS profile. For example, Boura et. al characterized the structure of ESCRT-I in solution by simultaneous structural refinement against SAXS and double electron-electron resonance spectroscopy of spin-labeled complexes[38]. Mishraki et. al used SAXS experiments to monitor the hexagonal state of the HII mesophase lattice structure. They also used electron paramagnetic resonance (EPR) to measure insulin entrapment within the lattice structure [39]. Wang et. al combined residual dipolar coupling (RDCs) from nuclear magnetic resonance spectroscopy (NMR) with SAXS restraints to orient

subunits and define the global shape of multi-component proteins and protein complexes [40]. Grishaev et al. used NMR and SAXS restraints to refine the solution structure of the 82-kDA enzyme malate synthase G [41].

BCL::SAXS Designed Specifically to Incorporate SAXS Restraints into BCL::Fold

Here we describe our newly developed algorithm BCL::SAXS – a module inside of our *de novo* protein structure prediction algorithm BCL::Fold. The integration of BCL::SAXS with BCL::Fold provides a means to utilize SAXS scattering profiles as an additional term in the potential energy function[19]. Our algorithm computes complete SAXS scattering profiles for complete protein models and an approximate scattering profile for these idealized protein models that consist of secondary structure elements only. We then compare the calculated scattering profile with the 'experimental' profile to identify likely protein structures. For benchmark purposes, the 'experimental' profile was simulated from many proteins using CRYSOL[42] - a freely available software package to simulate SAXS profiles from atomic coordinates. It is the current state of the art program for scattering vectors with lengths up to $0.75Å^{-1}$.

Computational approaches for SAXS fitting are classified as either ab initio or rigid body modeling. The ab initio methods search for three dimensional shapes represented by beads that fit the experimental SAXS profile [28, 43]. This method does not yield a (unique) structural model, but provides a variety of configurations of spheres that correspond to a given scattering pattern. The rigid body modeling approach uses the three dimensional atomic coordinates from a solved structure as input to compute the scattering profile [37]. The main methods to calculate a SAXS scattering profile from atomic coordinates are multipole expansion, Monte Carlo methods and coarse grain sampling with the Debye formula [42, 44-46]. Multipole

expansion methods have been shown to be highly accurate, but difficult to modify for approximations. The Debye formula is easy to modify, but comes with a high computational cost[30]. We need to compare BCL::Fold models – i.e. protein structure that lack loops and side chains – with SAXS profiles. To facilitate this, we chose to use the Debye formula, implement approximations for missing loops and side chain atoms, and address the computational cost with GPU acceleration.

Overall approach

In BCL::SAXS inter-atomic pairwise distances are computed explicitly for each heavy atom using the Debye formula for atomic scatterers [47]. We accelerated the algorithm performance by using graphical processing unit (GPU) parallel threads. We demonstrate the discriminatory power of SAXS at three different abstraction levels: 1) Complete protein models, 2) protein models with approximated side chain coordinates, 3) protein models with approximated side chain coordinates and approximated loop regions. We quantify the performance of the protocol from a benchmark set of 455 proteins. Further, we evaluate the effect of using the SAXS score as a weighted term in the knowledge-based energy function of BCL::Fold for four soluble protein examples. Finally we introduce a new approximation for crude protein models missing side chain and loop regions. Following the benchmarking presented in this paper, BCL::SAXS will be made available to the scientific community.

CHAPTER II

RELATED WORK

Different methods are available to compute scattering profiles from atomic models. In one approach, the classical Debye formula has been used. Peter Debye (1884 – 1966) won the Nobel Prize in chemistry in 1936 for his contributions to the study of molecular structure, specifically dipole moments. In 1915 he published "Zerstreuung von Röntgenstrahlen" or Scattering from X-rays. In this work Debye presented a method to compute scattering of electrons from nuclear position. This work was refined by adding precomputed atomic form factors by Crommer and Mann. While accurate for a given scattering angle range, the Debye approach is computationally expensive and the time-cost increases quadratically with the number of atoms in the protein. Shown below is the Debye Formula with the atomic form factors:

$$I(q) = \sum_{i=1}^{M} \sum_{j=1}^{M} F_i(q) F_j(q) \frac{\sin(q \cdot r_{ij})}{q \cdot r_{ij}} \quad (1)$$

As shown, the given intensity (I) is a function of scattering angle (q) and 3D position (r). F_i and F_j are the computed form factor for the given atom at the specified scattering angle q. Because of the iterative structure of this approach, the contribution of each atom can be quantified and adjusted.

As an alternative to the Debye formula, CRYSOL was developed in 1995. This program uses Gaussian sphere approximation to compute the scattering from the solvent in the excluded

volume and spherical harmonics to calculate both the envelope around the protein and the hydration layer that surrounds it:

$$\mathrm{I}_{\mathrm{pred}}(\mathrm{q}) = \langle \left| \mathbf{F}_{\mathrm{mol}} - \mathbf{F}_{\mathrm{disp}} + \delta\rho\mathbf{F}_{\mathrm{surf}} \right|^2 \rangle_{\Omega} \quad (2)$$

where F_{mol} is the complex scattering amplitude of the molecule, F_{disp} is the complex scattering amplitude of the displaced solvent, and $\delta\rho Fs_{urf}$ is the complex scattering amplitude of the increased density of the surface water layer. The average of these amplitudes is taken for a given q producing intensity (I). As opposed to the Debye formula, the computational cost grows linearly with the number of atoms in the protein. This method has been the gold standard for nearly 20 years in computing SAXS scattering profiles from atomic coordinates.

In 2010 AXES, was created by a team at the NIH to rival the accuracy of CRYSOL [48]. In this approach experimental SAXS data variability was explicitly addressed by adding tuning parameters to the Experimental SAXS profile:

$$\mathrm{I}_{\mathrm{expt}}(\mathrm{q}) = \mathrm{I}_{\mathrm{sample}}(\mathrm{q}) - \alpha \mathrm{I}_{\mathrm{buffer}}(\mathrm{q}) + \mathrm{c} \quad (3)$$

where I_{expt} is the experimental SAXS profile. In this case, the scattering from the buffer is subtracted from the scattering with the sample and buffer with a scaling factor α and an offset value c added. The scaling factor (α) accounts for instrument and sample concentration uncertainly during data collection. The offset value (c) accounts for variability in X-ray fluorescence. The predicted SAXS profile is computed the same way as CRYSOL with additional averages taken:

$$\mathrm{I}_{\mathrm{pred}}(\mathrm{q}) = \langle \langle \langle \left| \mathbf{F}_{\mathrm{mol}} - \mathbf{F}_{\mathrm{disp}} + \delta\rho\mathbf{F}_{\mathrm{surf}} \right|^2 \rangle_{\Omega} \rangle_{\mathrm{solv}} \rangle_{\mathrm{ens}} \quad (4)$$

where Ω is averaged over a set of molecular frame orientations relative to the incident beam, solv is averaged over the displaced and surface water sets, and ens is averaged over the

ensemble of macromolecular structures. Because of these different averaging tasks, AXES is more than an order of magnitude slower than CRYSOL. Several approaches to improve the speed of AXES are currently under development.

Although fast (CRYSOL) and accurate (AXES), spherical harmonics are not as easily modified iteratively as the Debye formula is. Because of the relative ease of implementation of the Debye formula, other groups have attempted to reduce the computational cost by making approximations to the Debye formula.

For example, the FoXs algorithm [44] approximates the atoms in a protein model as beads of different scattering masses but equal shape. This approach reduces the computation time in the Debye formula by about two orders of magnitude but the scattering profiles obtained deviates slightly from the scattering profiles obtained from CRYSOL [42]. The loss in accuracy due to this approximation is shown however to be within the margin of experimental error. For our purposes with loop and side chain approximations, we did not want to make an approximation of an approximation. This had the potential to introduce a compounding effect that could negatively impact our results.

Using a different approach to simplify the Debye formula, Stovgaard et. al [45] used dummy atoms with statistically derived form factors. This approach led to an order of magnitude increase in speed. The scattering profiles obtained by using this method matched the scattering profiles obtained through CRYSOL, but also relies on the accuracy of the form factor estimates. In the form factor estimates obtained by Stovgaard, the form factor contribution was very limited in the case of Isoleucine, Leucine and Valine. These are hydrophobic residues and were likely buried in the interior of the protein core during training of the system. These amino acids have different levels of hydrophobicity depending on the pH of

the surrounding solution. We were concerned that applying this approximation to proteins where either 1) the pH is not the same and 2) the structure of the protein does not have buried hydrophobic residues, would introduce error into our modeling.

Our solution to the limitations presented about these other methods was to avoid any shape and/or form factor approximations and use GPU acceleration to compute the scattering profile directly from the Debye formula. We would then compare our results with the Debye formula with the results obtained by spherical harmonics. To our knowledge this is the first approach to combine the Debye formula for atomic scatterers with parallel GPU threading. By utilizing this approach, we were able to avoid making further approximations to the Debye formula while addressing the computational bottleneck.

CHAPTER III

ALGORITHM DEVELOPMENT

Generate SAXS Profiles from Rigid Protein Bodies

To accurately determine the SAXS profile from the atomic coordinates of full atom protein models we utilized several key equations – the Debye formula for atomic scatterers and three equations to calculate the form factors[44, 45, 47, 49-51]. The form factors are continuous functions of the scattering momentum q. Using the Euclidean atomic coordinates from structures stored in the protein data bank (PDB), scattering profiles for rigid bodies are computed. The following equations, starting with the Debye formula (for completeness), depict the method:

$$I(q) = \sum_{i=1}^{M} \sum_{j=1}^{M} F_i(q) F_j(q) \frac{\sin(q \cdot r_{ij})}{q \cdot r_{ij}} \tag{5}$$

where the intensity, I(q) is a function of the momentum transfer q. The momentum transfer is given by $q = (4\pi \sin\theta)/\lambda$, where the scattering angle θ is given by 2θ, and λ is the wavelength of the incident beam. $F_i(q)$ and $F_j(q)$ are the atomic form factors and r_{ij} is the pairwise Euclidean distance between atom i and atom j. M is the number of atoms in the protein and the summations run over all atoms. To calculate the form factors, we subtracted the displaced solvent contribution from the form factor in vacuo:

$$\mathrm{F_i(q)} = \mathrm{fv(q)} - \mathrm{fs(q)} \tag{6}$$

where $f_v(q)$ is the atomic form factor in vacuo, and $f_s(q)$ is the form factor of the hypothetical atom that represents the displaced solvent. The atomic form factor in vacuo approximation is

based on the combination of relativistic Dirac-Slater wave functions and numerical Hartree-Fock wave functions [52]. These Hartree-Fock scattering factors were previously computed from q = 0 to q = 1.5 at intervals of 0.01Å^{-1}. For convenience, these scattering factors were previously fit to the 5-Gaussian (Cromer-Mann) analytic function:

$$\mathrm{fv_i}(q) = \sum_{i=1}^{4} \mathrm{ai} \cdot \mathrm{e}^{-\mathrm{bi}(\frac{q}{4\pi})^2} + \mathrm{c} \tag{7}$$

where a, b, and c are the constants for each atom, and q is the scattering angle in angstroms. This approximation is only valid with a q range from 0 to 2.0Å (see Table 1) [53].

Table 1: The Cromer-Mann coefficients for scattering factors between 0 to 2.0Å

Atom	A1	A2	A3	A4	B1	B2	B3	B4	C
H	0.493002	0.322912	0.140191	0.040810	10.510900	26.125700	3.142360	57.799700	0.003038
C	2.310000	1.020000	1.588600	0.865000	20.843900	10.207500	0.568700	51.651200	0.215600
N	12.212600	3.132200	2.012500	1.166300	0.005700	9.893300	28.997500	0.582600	11.52900
O	3.048500	2.286800	1.546300	0.867000	13.277100	5.701100	0.323900	32.908900	0.250800
S	6.905300	5.203400	1.437900	1.586300	1.467900	22.215100	0.253600	56.172000	0.866900

In SAXS scattering experiments, the valid scattering angle range is from 0 to ≈ 0.33Å. For larger scattering angles, a 6-Gaussian approximation must be used which is valid from 0 to ≈ 6.0 Å [54]. The displaced solvent scattering fs(q) was approximated by V_i, the excluded solvent volume V displaced by atom i (See Table 2):

$$\mathrm{fs_i}(q) = q_s V_i \mathrm{e}^{-\frac{q^2 V_i^{2/3}}{4\pi}} \tag{8}$$

where qs is the solvent density of 0.334e Å^{-3}. The combination of these equations yields a SAXS scattering profile from rigid body data stored in a pdb file.

Table 2: Excluded volume, radius, and bound hydrogen count by atom type

Type	Volume ($Å^3$)	Radius (Å)	Bound Hydrogen	Type	Volume ($Å^3$)	Radius (Å)	Bound Hydrogen
H	5.15	1.07	0	NH2	12.79	1.45	2
C	16.44	1.58	0	NH3	17.94	1.62	3
CH	21.59	1.73	1	O	9.13	1.30	0
CH2	26.74	1.85	2	OH	14.28	1.50	1
CH3	31.89	1.97	3	S	19.86	1.68	0
N	2.49	0.84	0	SH	25.10	1.81	1
NH	7.64	1.22	1				

Initial Validation of BCL::SAXS

I programmed BCL::SAXS using the equations described above and computed SAXS profiles for protein 1ENH using all atoms.

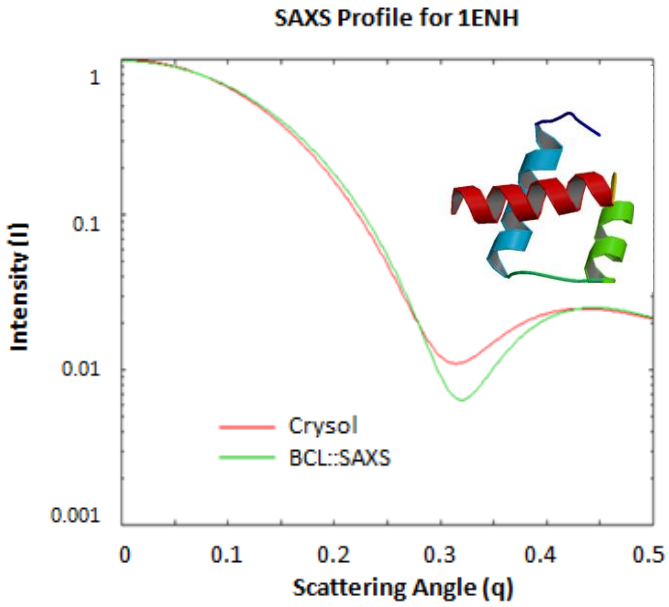


Figure 4: Initial SAXS profiles generated through BCL::SAXS.

As shown in figure 4, BCL::SAXS deviates from Crysol between 0.28 Å and 0.40 Å. The red curve represents the scattering profile computed by Crysol, while the green curve represents the

profile generated through BCL::SAXS. The large deviation from Crysol for all atom computations was unacceptable and had to be resolved. Using Taylor series expansions on sinx/x:

$$\lim_{x\to 0}\frac{sinx}{x}=\frac{x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\cdots}{x}=1-0+0-0+\cdots=1 \tag{9}$$

We realized that the Debye formula did not vanish at q = 0, but reduced to:

$$I(q)=\sum_{i=1}^{M}\sum_{j=1}^{M}F_i(q)F_j(q) \tag{10}$$

Adding this adjustment to the Debye formula solved the problem.

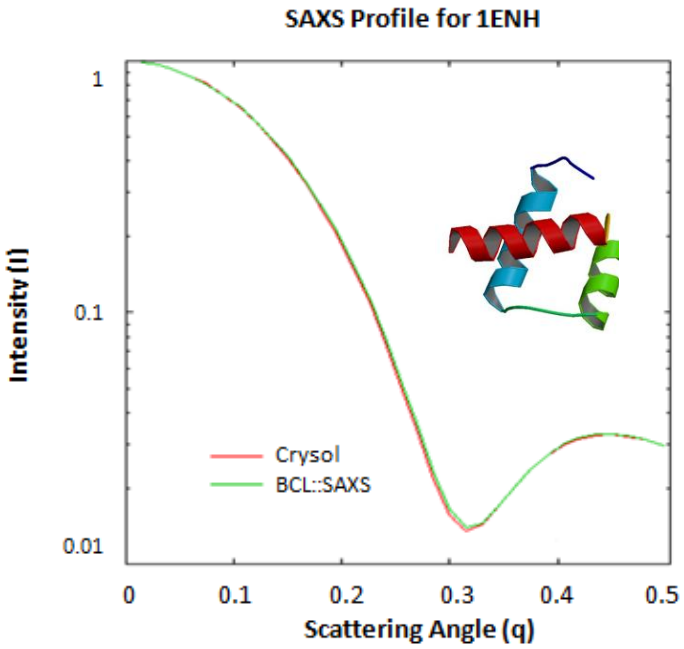


Figure 5: Computed SAXS profile with correction to Debye formula

GPU Parallel Processing to Accelerate Algorithm

The pairwise nature of the Debye formula has a computational cost of $O(N^2)$ for each value of the scattering angle (q) evaluated. N is the number of Cartesian coordinates (atoms) contained in the protein. This high computational cost and time requirement has precluded the

use of the direct calculation of SAXS profiles using the Debye formula during folding simulations. To overcome this computational limitation, alternative approaches for this calculation including multipole expansion methods for spherical harmonics [42] and approximation of the individual form factors have been developed [45]. To directly compute the SAXS profile using the Debye formula we leveraged the parallel architecture of graphical processing unit (GPU) threads using OpenCL and computed SAXS profiles directly.

The Debye formula can be visualized as an NxN square matrix of N-atom rows by N-atom columns where N is the number of atoms in the protein. The pairwise Euclidean distances were calculated for each entry in the matrix with the diagonal represented by zeros. Pairwise distance calculations in a Matrix form are an ideal calculation type for GPU acceleration because each GPU thread can calculate a single Euclidean distance. The algorithm was restructured to have each thread calculate a Debye partial sum for a current atom i:

$$I_{partial} = \sum_{j=1}^{M} F_i(q) F_j(q) \frac{\sin(q \cdot r_{ij})}{q \cdot r_{ij}} \quad (11)$$

This technique enables the application of this accelerated algorithm to very large multimeric systems in excess of 90,000 atoms while leveraging device shared memory in a tiling technique. The result of this partial sum is a matrix of q rows by N-atom columns where q is the scattering angle and N is the total number of atoms. These partial sums are then summed across each column to completion for each q using a GPU reduction sum kernel to arrive at the desired q number of sums (see figure 6). Table 3 shows the timing results of GPU vs. CPU benchmarks with BCL::SAXS.

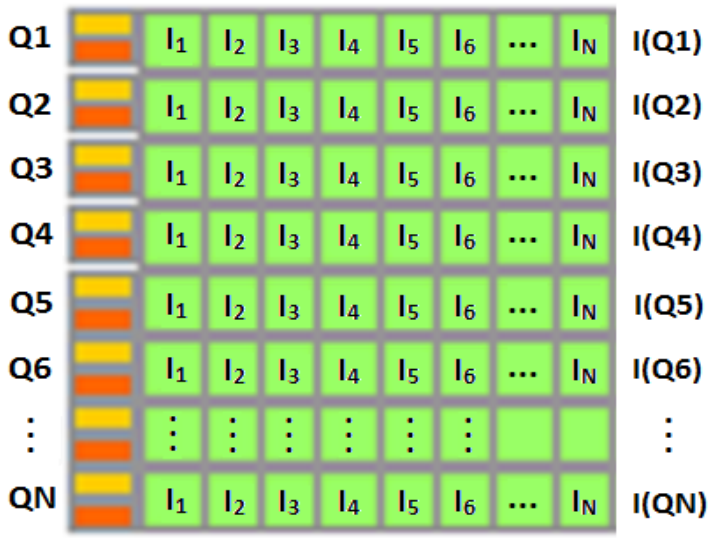


Figure 6: Schematic of threading used for GPU acceleration. Each partial intensity sum (I_N) is computed for a given scattering angle Q. The partial intensity values are then summed to give a final Intensity I(QN) for the given scattering angle.

GPU Acceleration Yields Orders of Magnitude Speed Improvements

The GPU accelerated Debye calculation was benchmarked on several protein systems from the PDB with sizes ranging from 1800 atoms to 92,000 atoms. The benchmark was performed on several devices ranging from low-end workstation class GPUs (Quadro 600) to high-end consumer grade GPUs (C1060). See Table 3 below.

Table 3: Timing results of GPU vs. CPU benchmarks. All timings are reported in seconds. Q600 indicates Quandro 600.

PDB	Atoms	CPU $1300	GPU Accelerated Chips						Maximum Speedup
			Q600 $150	GTX470 $200	GTX480 $250	GTX580 $450	GTX680 $600	C1060 $1200	
1O26	1832	3.6	0.1	0.07	0.07	0.07	0.07	0.09	5x
1WA5	7543	65	1	0.31	0.28	0.27	0.20	0.37	325x
1NR1	23217	624.3	9.3	2	1.9	1.8	1.2	2.7	520x
1ZUM	43243	2300	30	4.9	4.1	3.9	2.4	6.5	958x
1VSZ	91846	15365	132	19.8	16.9	15.8	9.0	26.3	1707x

CHAPTER IV

ALGORITHM APPROXIMATIONS

Approximate SAXS Profiles for Protein Models with Missing Side Chains and Loops

To approximate the side chain regions of a given amino acid, the form factors for the atoms with missing side chain coordinates were added to the C_β position of the respective amino acid. This approach is analogous to how the form factors for hydrogen are summed together with their bound heavy atom in CRYSOL [42].

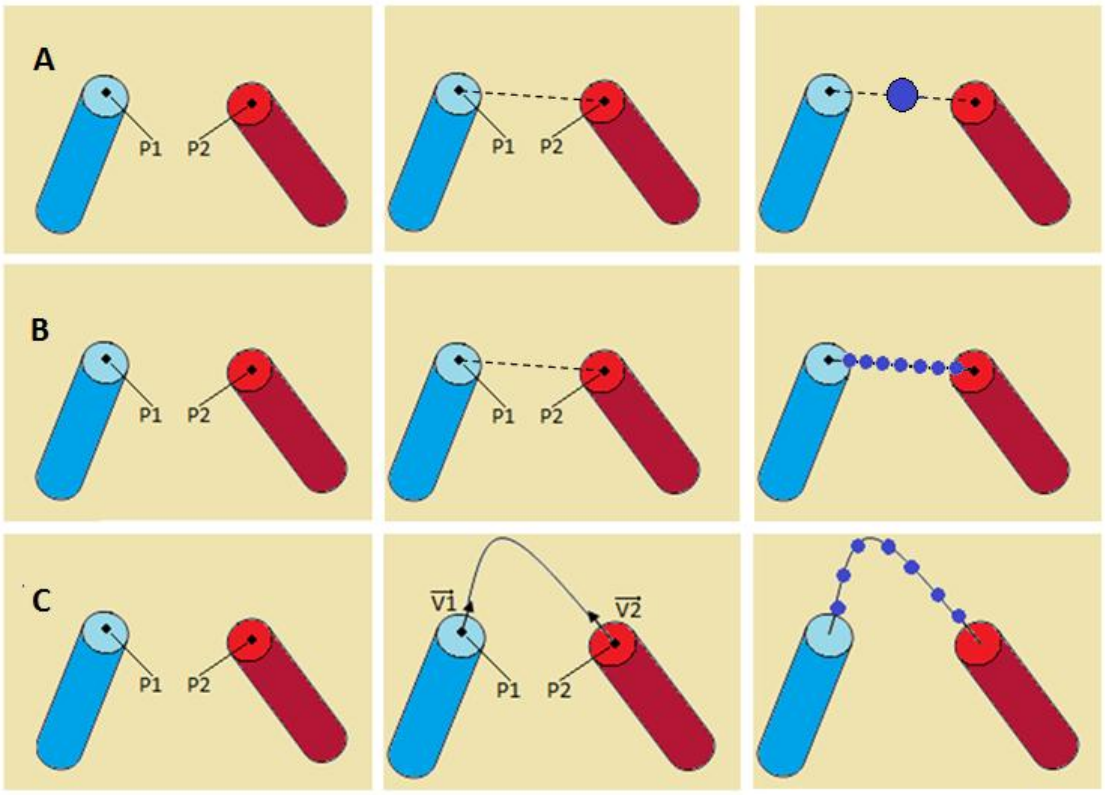


Figure 7: Protein model with two α-helical structures, p1 and p2 (left), approximated path with unit vectors v1 and v2 pointing in the direction of p1 and p2 (middle) and residues placed equidistant along the path (right). Panel A depicts the loop region residues summed at the midpoint between the SSEs. Panel B depicts the linear path with the loop region residues evenly spaced along a linear line between the SSEs. Panel C depicts the orientation dependent curvilinear path between SSE_1 and SSE_2.

The loop regions were first approximated by removing atomic coordinate data between secondary structure elements (SSEs) and computing the midpoint between the c-terminus of the first SSE to the n-terminus of the second SSE. All of the form factor contributions from all of the residues were summed at this location. This approach although fast, dramatically altered the SAXS profile. (See Appendix III) We were able to correctly identify five protein models (1UBIA, 1J27A, 1JL1A, 1NFNA, and 3B50A) out of a set of nine using this method.

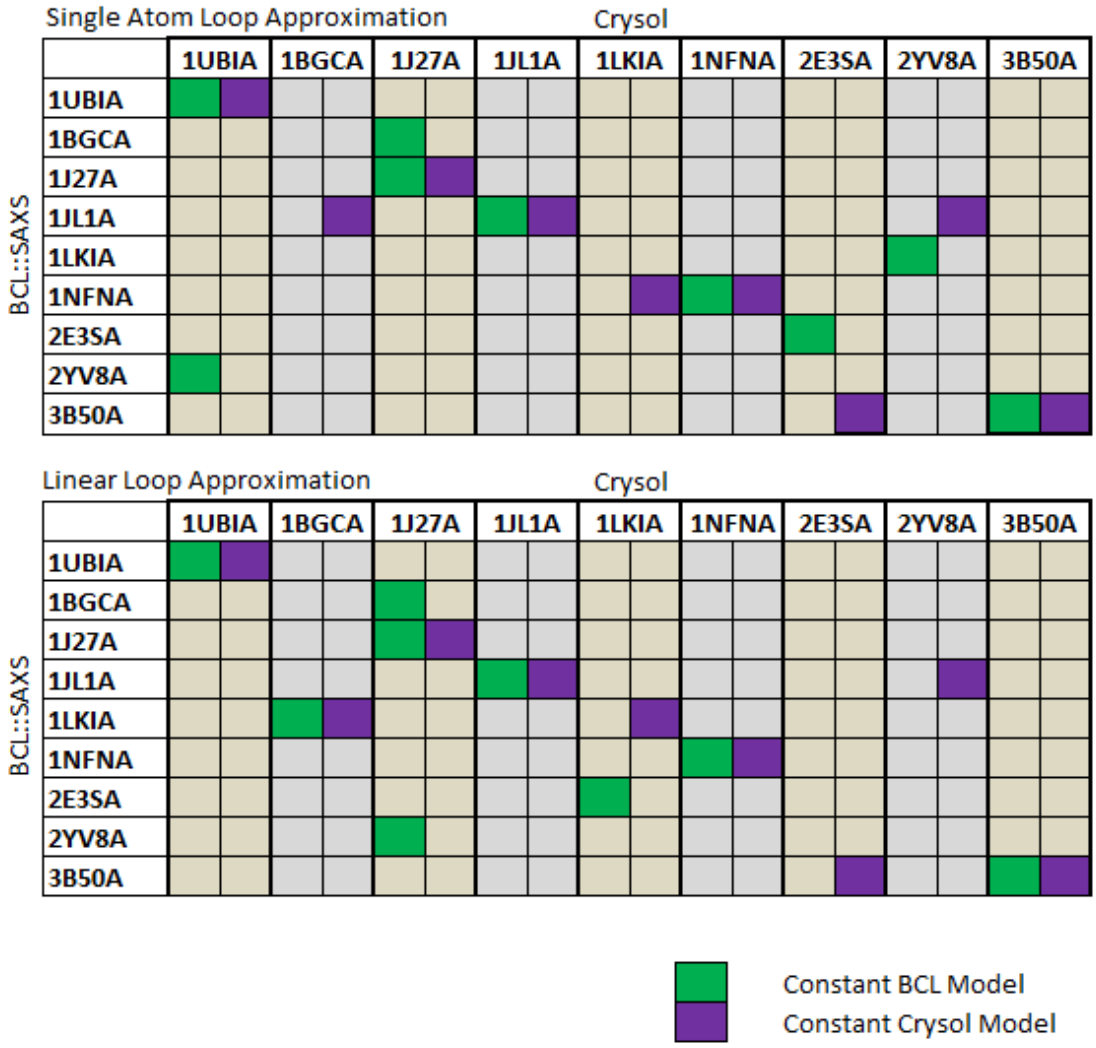


Figure 8: Scoring Matrix of Single Atom and Linear Loop Approximation. The vertical column represents saxs profiles generated through BCL::SAXS with approximated loop regions by the designated method. The horizontal column represents saxs profiles generated through CRYSOL. The Matrix is not symmetric. The green squares indicate the minimum saxs score when keeping the profile generated through BCL::SAXS constant and comparing across all other CRYSOL profiles. The purple square indicate the minimum saxs score when keeping the profile generated through CRYSOL constant and comparing across all other BCL::SAXS profiles.

Figure 8 depicts in color square our initial results with the single atom and linear loop approximation. In both cases we generated SAXS profiles from nine protein models with either the single atom or linear approximation for the loop regions. We then used CRYSOL to generate scattering profiles from the same protein set and measured the difference between the curves with our scoring function. The matrix is not symmetric which means that we obtain different results when we compare a CRYSOL model across a set of BCL::SAXS models vs. comparing a BCL::SAXS model across a set of CRYSOL models. To account for this, figure 8 contains two color columns per model. For a model to be correctly identified by our SAXS score, both evaluation types must be a minimum (both boxes green and purple must be present in the diagonal position).

The second approach was to compute a linear path between the two SSEs. Residues were then placed equidistant along this path to simulate loops. We were able to correctly identify the same five protein models (1UBIA, 1J27A, 1JL1A, 1NFNA, and 3B50A) out of a set of nine using a linear path approximation between the SSEs. This approximation was not realistic for systems that required more residues and had less space between SSEs. In the case, residues would be placed in overlapping regions along the linear line resulting in severe steric clashes.

To further optimize the method, we accounted for residue steric clashes by creating a curvilinear path between SSE_1 and SSE_2 and using their respective orientation. This path begins in the direction of SSE_1 and is gradually modified until it ends in the direction of SSE_2. This approach has the ability to generate parabolic, sigmoidal or linear pathways depending on SSE orientation. While crude, this approach is much more rapid than actual loop construction.

Vector Calculations to Approximate the Path Directions between Two SSEs

P_1 represents the C_β position of the last residue in the N-terminal SSE, while P_2 represents the C_β position of the first residue in the C-terminal SSE.

$$\mathrm{P_1} = \{\mathrm{x1_n}, \mathrm{y1_n}, \mathrm{z1_n}\} \tag{12}$$

$$\mathrm{P_2} = \{\mathrm{x1_c}, \mathrm{y1_c}, \mathrm{z1_c}\} \tag{13}$$

CP_1 represents the center position of the last residue in the N-terminal SSE, while CP_2 represents the center position of the first residue on the C-terminal SSE.

$$\mathrm{CP_1} = \{\mathrm{x2_n}, \mathrm{y2_n}, \mathrm{z2_n}\} \tag{14}$$

$$\mathrm{CP_2} = \{\mathrm{x2_c}, \mathrm{y2_c}, \mathrm{z2_c}\} \tag{15}$$

We computed a vector pointing in the same orientation of the SSE by subtracting the C_β position of the center of the SSE from P_1 and P_2.

$$V_n = \mathrm{P_n} - \mathrm{CP_n} \tag{16}$$

where n is the index of the point. The direction of the vectors V_1 and V_2 were computed by dividing them by their magnitude.

$$D_n = \frac{V_n}{\sqrt{V_{nx}^2 + V_{ny}^2 + V_{nz}^2}} \tag{17}$$

The distance (D_{sse}) between two SSEs was computed by subtracting P_2 from P_1 and then taking the norm of the resulting vector. The distance (D_x) between points along the path (S) was computed by dividing one by one more than the number of amino acids in the loop region.

$$D_x = \frac{1}{N_{aa} + 1} \tag{18}$$

The predicted loop length (P) was computed by multiplying the number of amino acids by the C_α - C_α spacing of 3.8 Å. The 3.8 Å term is the average distance between amino acids in a protein. It was computed by averaging the C_α distance between residues in the engrailed homeodomain (pdb id: 1ENH) [55].

$$P = N_{aa} \times 3.8 \tag{19}$$

Pathway Calculations for Loop Approximation

The path length (S) between two SSEs was approximated as a curve starting in the direction of SSE_1 and ending in the direction of SSE_2. The curve calculation consists of a linear, parabolic, and a directional component. The linear component is given by:

$$l(L) = (1 - \mathrm{L})\mathrm{P}_1 + \mathrm{LP}_2 \tag{20}$$

where L is between [0, 1]. When L=0, the equation reduces to the Euclidean coordinates of point 1. When L=1, the equation reduces to the Euclidean coordinates of point 2. The parabolic component is given by:

$$p(L) = \mathrm{N} \times \mathrm{L}(1 - \mathrm{L}) \tag{21}$$

where N is a normalization factor to size the height of the parabola and control parabolic path length. The directional component is given by:

$$d(L) = [(1 - \mathrm{L})\overrightarrow{\mathrm{d}_1} + \mathrm{L}\overrightarrow{\mathrm{d}_2}\,] \tag{22}$$

where d1 and d2 are unit directional vectors pointing in the direction of SSE_1 and SSE_2 respectively. The complete parabolic approximation function is:

$$\mathrm{P(L)} = (1 - \mathrm{L})\mathrm{P}_1 + \mathrm{LP}_2 + \mathrm{NL}(1 - \mathrm{L}) \times [(1 - \mathrm{L})\overrightarrow{\mathrm{d}_1} + \mathrm{L}\overrightarrow{\mathrm{d}_2}\,] \tag{23}$$

Triangle Approximation for Normalization Factor Calculation

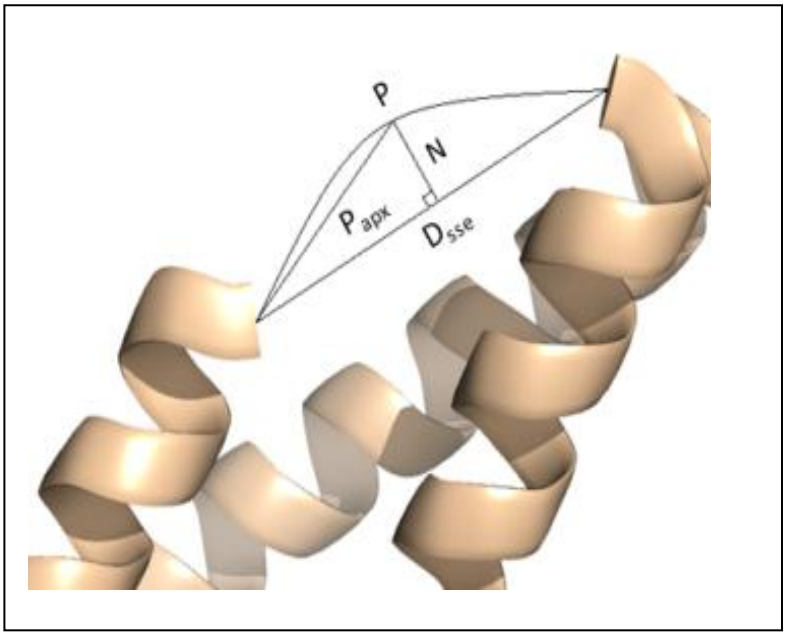


Figure 9: Depiction of parabolic height approximation. D_{sse} is the distance between SSE's, P_{apx} is the estimated distance of the parabolic loop. N is the normalization factor and controls the height of the parabola.

The normalization factor (N) controls the height of the curve and corresponding path length. To calculate N for a given loop region we divided the curve in half and approximated the arc to be the hypotenuse of a right triangle. The base of the triangle was the Euclidean distance between the SSEs divided by two (See figure 2). With these approximations, the normalization factor (N) is given by the Pythagorean Theorem:

$$N = \frac{1}{2}\sqrt{P^2 - D_{sse}^2} \tag{24}$$

Where N is the normalization factor, P is the predicted loop length, and D_{sse} is the Euclidean distance between P_1 and P_2.

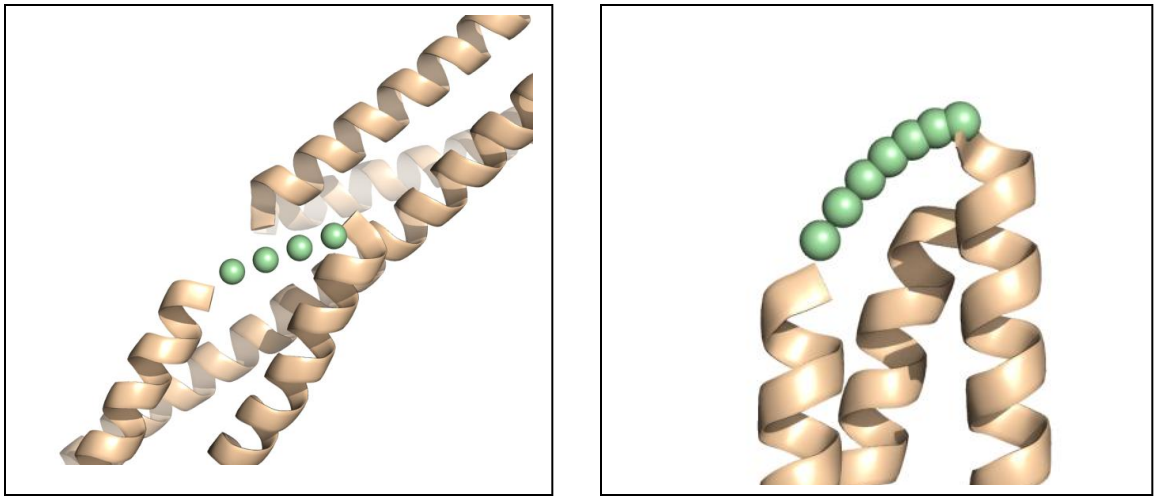


Figure 10: The triangulation method produces linear (left) and curved pathways (right) between SSEs depending on the value of the normalization factor.

Regula Falsi Approximation for Normalization Factor Calculation

Prior to implementing triangle approximation for loop regions, I computed the parabolic arc length based on the desired distance between the SSEs and a given normalization parameter to adjust the height of the arc. This was framed as a non linear optimization problem using the regula falsi procedure. In this instance, the objective function was the function to compute arc length, the parameter to optimize was the normalization factor n, and the goal was to minimize the difference between the desired arc length and the computed arc length.

Arc Length Calculations

The following is the derivation shown by Robert Donley Ph.D of the arc length computation along the path $f(x) = x^2$ from x=0 to x=1. The formula to compute arc length is:

$$L = \int_a^b \sqrt{1 + (f')^2}\,dx \tag{25}$$

Setting $f(x) = x^2$, gives $f'(x) = 2x$ and $(f'(x))^2 = 4x^2$. The formula becomes:

$$L = \int_0^1 \sqrt{1 + 4x^2}\,dx \tag{26}$$

This integral is not trivial and will require some work to evaluate. The first step is to introduce the hyperbolic trigonometric functions cosh(t) and sinh(t).

$$\cosh(t) = \frac{e^t + e^{-t}}{2} \tag{27}$$

$$\sinh(t) = \frac{e^t - e^{-t}}{2} \tag{28}$$

The cosh(t) function looks like $1+x^2$, but with much steeper growth at ∞. The sinh(t) is a function with an inverse function. Here are the derivative properties of the cosh(t) and sinh(t) functions:

$$\frac{d}{dx}\cosh(x) = \sinh(x) \tag{29}$$

$$\frac{\mathrm{d}}{\mathrm{dx}}\sinh(\mathrm{x}) = \cosh(\mathrm{x}) \tag{30}$$

The next step is to parameterize the cosh and sinh functions.

$$(\mathrm{x}, \mathrm{y}) = (\cosh(\mathrm{t}), \sinh(\mathrm{t})) \tag{31}$$

$$\cosh(\mathrm{t})^2 - \sinh(\mathrm{t})^2 = 1 \tag{32}$$

The hyperbolic trigonometric functions will be used to evaluate the integral in equation 22. To simplify this derivation I will refer to an arbitrary expression as a box. (■) Note: inverse functions undo the action of the function. For example, the inverse of e^x is the ln x. The inverse of x^3 is $x^{1/3}$.

$$\sinh^{-1}(■) = \mathrm{t} \leftrightarrow \sinh(\mathrm{t}) = ■ \tag{33}$$

Using the relation from equation 32:

$$\cosh^2(\mathrm{t}) = 1 + \sinh^2(\mathrm{t}) \tag{34}$$

Substituting the relation from 33 into 34:

$$\cosh^2(\mathrm{t}) = 1 + ■^2 \tag{35}$$

Take the square root of both sides:

$$\cosh(t) = \sqrt{1 + \blacksquare^2} \tag{36}$$

Using the relation in 33 and substituting that for the variable t in equation 36:

$$\cosh\left(\sinh^{-1}(\blacksquare)\right) = \sqrt{1 + \blacksquare^2} \tag{37}$$

The necessary forms of the double angle formula for both the cosh and sinh functions are:

$$\sinh(2\blacksquare) = 2\cosh(\blacksquare)\sinh(\blacksquare) \tag{38}$$

$$\cosh(2\blacksquare) = 2\cosh^2(\blacksquare) - 1 \tag{39}$$

For reference, equation 26 is shown again below:

$$L = \int_0^1 \sqrt{1 + 4x^2}\,dx \tag{26}$$

Substitute $4x^2$ with a function to remove the radical:

$$2x = \sinh(t) \tag{40}$$

$$L = \int_0^1 \sqrt{1 + \sinh^2(t)}\,dx \tag{41}$$

Using the relationship in equation 32 and apply to equation 41:

$$L = \int_0^1 \sqrt{\cosh^2(t)}\,dx \tag{42}$$

Removing the radical:

$$L = \int_0^1 \cosh(t)\,dx \tag{43}$$

To solve for dx in equation 43, take the derivative of 40 and solve for dx:

$$dx = \frac{\cosh(t)\,dt}{2} \tag{44}$$

Now substitute equation 44 into equation 43:

$$L = \int \cosh(t) \cdot \frac{\cosh(t)}{2}\,dt \tag{45}$$

Simplify and pull the constant in front of the integral:

$$L = \frac{1}{2}\int \cosh^2(t)\,dt \tag{46}$$

Use double angle formula from equation 39 to simplify $\cosh^2(t)$:

$$\cosh(2t) = 2\cosh^2(t) - 1 \tag{47}$$

$$\cosh(2t) + 1 = 2\cosh^2(t) \tag{48}$$

$$\frac{\cosh(2t)}{2} + \frac{1}{2} = \cosh^2(t) \tag{49}$$

Substitute equation 49 into equation 46:

$$L = \frac{1}{2}\int \left(\frac{1}{2} + \frac{\cosh(2t)}{2}\right) dt \tag{50}$$

To evaluate this integral use u substitution:

$$u = 2t$$

$$du = 2dt \tag{51}$$

$$dt = \frac{du}{2}$$

Substitute relationship from 51 into equation 50:

$$L = \frac{1}{2}\int \left(\frac{1}{2} + \frac{\cosh(u)}{2}\right)\frac{du}{2} \tag{52}$$

Simplify:

$$L = \frac{1}{2}\int \left(\frac{1}{4} + \frac{\cosh(u)}{4}\right) du \tag{53}$$

Integrate:

$$L = \frac{1}{2}\left[\frac{1}{4}u + \frac{\sinh(u)}{4}\right] + c \tag{54}$$

Simplify:

$$L = \frac{1}{8}u + \frac{\sinh(u)}{8} + c \tag{55}$$

Substitute u = 2t into equation 55.

$$L = \frac{1}{8}2t + \frac{\sinh(2t)}{8} + c \tag{56}$$

Simplify:

$$L = \frac{1}{4}t + \frac{\sinh(2t)}{8} + c \tag{57}$$

From equation 40 and the function relationship in 33:

$$t = \sinh^{-1}(2x) \tag{58}$$

Substitute 58 into 57:

$$L = \frac{1}{4}\sinh^{-1}(2x) + \frac{\sinh(2\sinh^{-1}(2x))}{8} + c \tag{59}$$

Apply the double angle formula for sinh from equation 38 into equation 59:

$$L = \frac{1}{4}\sinh^{-1}(2x) + \frac{2\sinh(\sinh^{-1}(2x)\cdot\cosh(\sinh^{-1}(2x)))}{8} + c \tag{60}$$

The inverse functions will cancel. Using the relation from equation 37 yields:

$$L = \frac{1}{4}\sinh^{-1}(2x) + \frac{1}{4}(2x)\cdot\sqrt{1+4x^2} + c \tag{61}$$

The arc length between 0 and 1 is given as:

$$L = F(1) - F(0) \tag{62}$$

$$F(0) = \frac{1}{4}\sinh^{-1}(2*0) + \frac{1}{4}(2*0)\cdot\sqrt{1+4*0^2} = 0 \tag{63}$$

$$F(1) = \frac{1}{4}\sinh^{-1}(2) + \frac{\sqrt{5}}{2} \approx 1.48 \tag{64}$$

Apply Arc Length Calculations Generally

Now that we can compute the arc length for a simple function such as x^2, we need to expand the method to include general functions while using all of the work previously shown. To do this we need to convert a general problem into the form from the previous section. For example, suppose we want to find the arc length along the parabola:

$$y = \frac{1}{10}x^2 - \frac{1}{10}x + 20 \tag{65}$$

Between x = 0 and x=20. The first step is to shift the parabola by moving the vertex to the origin. This is done by completing the square:

$$y = \frac{1}{10}\left[\left(x^2 - x + \frac{1}{4}\right) - \frac{1}{4}\right] + 20 \tag{66}$$

$$y = \frac{1}{10}\left[\left(x - \frac{1}{2}\right)^2 - \frac{1}{4}\right] + 20 \tag{67}$$

To center the parabola at the origin, remove the ¼ term and the vertical offset term of 20. This leaves:

$$y = \frac{1}{10}\left(x - \frac{1}{2}\right)^2 \tag{68}$$

Perform u substitution:

$$u = x - \frac{1}{2} \tag{69}$$

$$y = \frac{1}{10}u^2 \tag{70}$$

$$y' = \frac{1}{5}u \tag{71}$$

The limits of integration change. When x = 0, u = -0.5; and when x=20, u = 19.5. Now apply the arc length formula previously derived in equation 25 with equation 71:

$$L = \int_a^b \sqrt{1 + (f')^2}\,dx \tag{25}$$

$$L = \int_{-0.5}^{19.5} \sqrt{1 + \left(\frac{1}{5}u\right)^2}\,du \tag{72}$$

$$L = \int_{-0.5}^{19.5} \sqrt{1 + \frac{1}{25}u^2}\,du \tag{73}$$

In this form we can see the similarity from the formula given in part 1:

$$\int \sqrt{1+4x^2}dx = \frac{1}{4}\sinh^{-1}(2x) + \frac{1}{4}(2x)\cdot\sqrt{1+4x^2} + c \quad (61)$$

Perform another u substitution with:

$$\frac{1}{25}u^2 = 4x^2 \quad (74)$$

$$\frac{1}{5}u = 2x \quad (75)$$

$$u = 10x \quad (76)$$

$$du = 10dx \quad (77)$$

With this substitution the limits of integration change again. When u = -0.5, x = -0.05; when u = 19.5, x = 1.95. Equation 73 becomes:

$$L = \int_{-0.05}^{1.95} \sqrt{1+4x^2}\cdot 10dx \quad (78)$$

$$L = 10\cdot[F(1.95) - F(-0.05)] \approx 44.93 \quad (79)$$

Where F(x) is:

$$\frac{1}{4}\sinh^{-1}(2x) + \frac{1}{4}(2x)\cdot\sqrt{1+4x^2} \quad (80)$$

Numerically Implement Arc Length Calculations

The derivation above requires the inverse hyperbolic sin function. This function is expensive computationally. Alternatively, to compute the arc length along a path given by:

$$y = Ax^2 + Bx + C \quad (81)$$

Between x=L_1 and x = L_2, we use the formula:

$$L = \frac{1}{4A}\left[\ln\left(\frac{D_2 + S_2}{D_1 + S_1}\right) + D_2S_2 - D_1S_1\right] \quad (82)$$

Where D_1 and D_2 are the derivatives evaluated at the end points:

$$D_1 = 2AL_1 + B \quad (83)$$

$$D_2 = 2AL_2 + B \quad (84)$$

$$S_1 = \sqrt{1 + D_1^2} \tag{85}$$

$$S_2 = \sqrt{1 + D_2^2} \tag{86}$$

One Dimensional Optimization

There are multiple methods of one-dimensional unconstrained optimization[56, 57]. To find a local minimizer, the ideal case would be to find the x value where the derivative of the function is zero. There are a few stumbling blocks to this approach: 1) the derivative may not exist, 2) it may be difficult to calculate and 3) the location on the function where the derivative is zero may not be explicitly solvable for x. There are multiple approaches to solving this problem that address these concerns. One approach is to set the derivative of the function equal to zero and find where this occurs (Newton, Bisection, Regula Falsi, and Secant optimization methods). Another approach is to find the minimizer of f(x) without using the derivative function (Golden Section and Quadratic interpolation methods). Another approach is to use information from both the function and the derivative of the function (Cubic Interpolation method). Each method has strengths and weaknesses depending on how the system behaves. For example, the Newton method is very fast, but unstable. It may diverge, or converge to the opposite root. In either case (divergence or false convergence), the answer is incorrect. When the Newton method does converge, the convergence is quadratic (fast). The secant method may diverge as well. I chose regula falsi, because although a bit slower than the Newton or Secant method, it will not diverge and is faster than the Bisection Method.

Regula Falsi Method

This method is the method of false position. The idea is to bracket x* between a_k and b_k with $g(a_k)$ and $g(b_k)$ of opposite sign. G(x) is approximated by a secant line through $(a_k, g(a_k))$ and $(b_k, g(b_k))$:

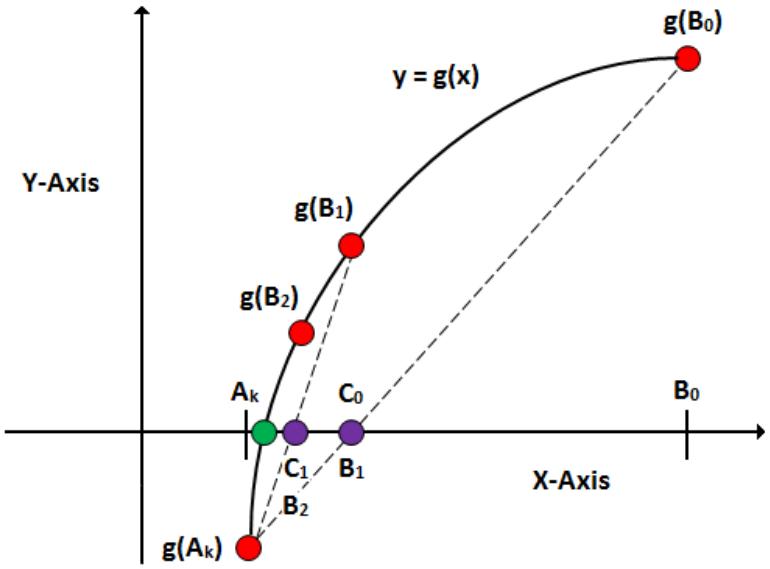


Figure 11: Regula Falsi Optimization. The green dot represents x* or the location of x when g(x) is zero. The secant line is depicted as a dashed line.

The equation for the secant line is:

$$y = g(a_k) + \frac{g(b_k) - g(a_k)}{b_k - a_k} \cdot (x - a_k) \tag{87}$$

Let C_k solve for where equation 87 (the secant equation) equals zero:

$$0 = g(a_k) + \frac{g(b_k) - g(a_k)}{b_k - a_k} \cdot (x - a_k) \tag{88}$$

$$c_k = \frac{a_k g(b_k) - b_k g(a_k)}{g(b_k) - g(a_k)} \tag{89}$$

Once c_k is found, the process is iterated until the termination criteria have been met. The update rules are as follows:

```
If g(ck) is the same sign as g(ak)
      ak+1 = ck, bk+1 = bk
If g(ck) is the opposite sign to g(ak)
      ak+1 = ak, bk + 1 = bk
If g(ck) = 0
      Stop
```

The algorithm is repeated until an acceptable margin of error or a predefined maximum number of iterations have been reached.

The method was implemented in BCL::SAXS and the normalization factor was computed with the regula falsi method optimization method. With a precise normalization factor I was able to generate loops with precise parabolic arc length distances between SSEs. During testing, the regula falsi method was slower and less accurate than the triangular approximation. Hence, the triangular approximation was chosen over the regula falsi approximation to compute the normalization factor. See the section titled: Selecting the Optimal Parabolic Height Approximation Algorithm.

CHAPTER V

EVALUATION OF ALGORITHM AND APPROXIMATIONS

Generation of SAXS Profiles from Atomic Coordinates with CRYSOL

Since SAXS is an emerging technique, there are few instances where experimental SAXS data exist for proteins that have been structurally determined to high resolution. Therefore, experimental scattering curves were approximated from high resolution protein structures in the PDB using the program CRYSOL [42]. This program computes the scattering profile using multipole expansion for fast calculation of the spherically averaged scattering profile. CRYSOL calculates the vacuum and excluded volume scattering components as well as a hydration layer contribution. To generate experimental scattering curves CRYSOL was run with the command line "crysol /dro 0.0 /sm 0.33 inputfile.pdb." The hydration shell does not need to be included in the Debye model for calculating SAXS profiles because it has a smaller impact on χ^2 than the errors in an experimentally measured SAXS profile[31]. The upper q-limit of 0.33 Å-1 was used in order to be within the expected valid experimental range.

SAXS Profile Analysis

For direct comparison, the calculated curve was multiplied by a scaling weight (c)

$$c = \frac{\sum_{k=1}^{Q} I_{cal}(q_k) * I_{exp}(q_k)}{\sum_{k=1}^{Q} I_{cal}^2(q_k)} \quad (90)$$

where I_{cal} is the intensity of the calculated curve, I_{exp} is the intensity of the experimental curve and q is the scattering angle. The scaling between I_{cal} and I_{exp} cannot be determined because concentration cannot be measured with enough accuracy for simulation. The scaling factor

minimizes the χ^2 measure. After the scaling weight was computed, the intensities of the calculated curve were multiplied by the scaling weight (see top graph on figure 9).

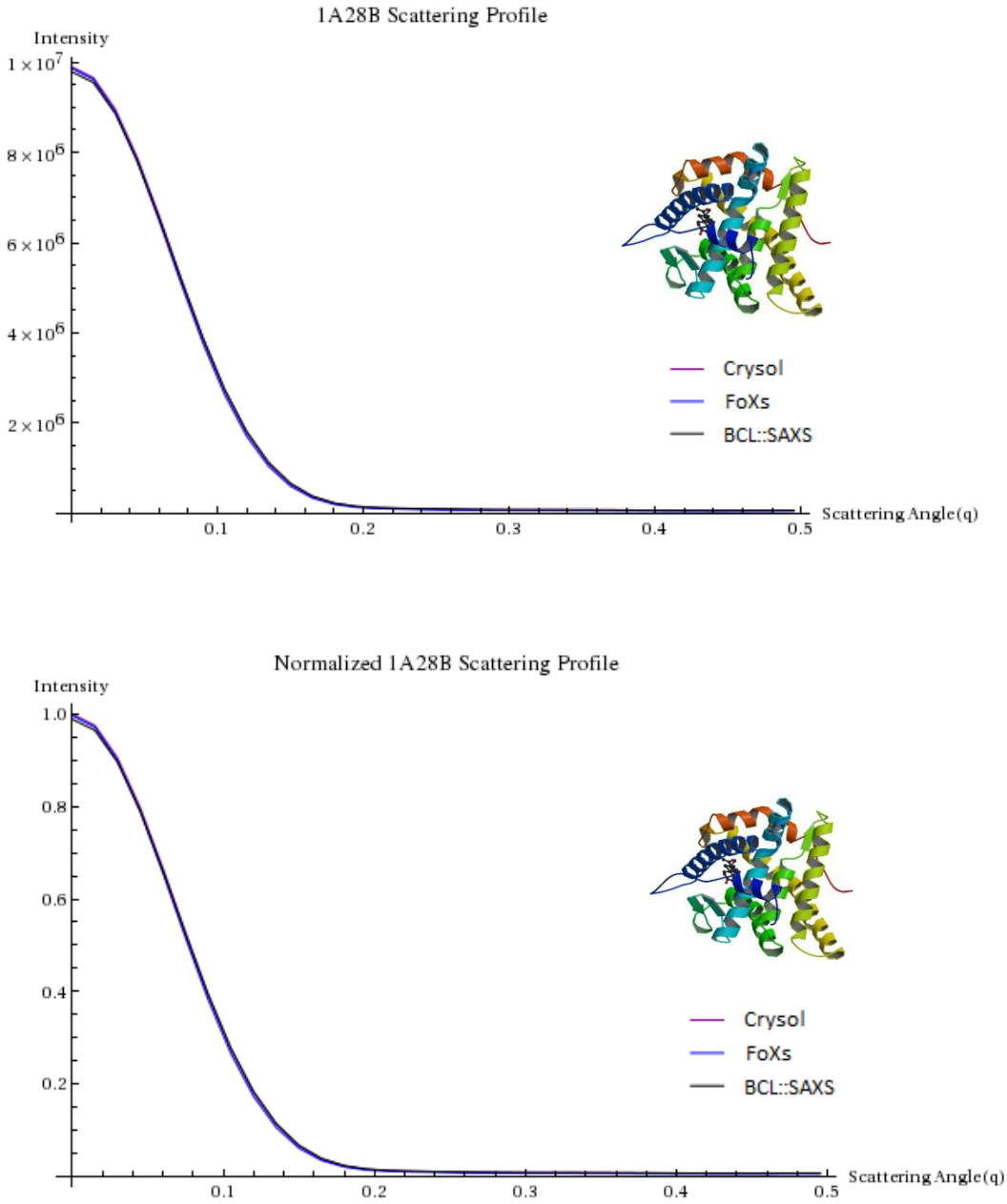


Figure 12: Original SAXS profile (Top) and Normalized SAXS profile of 1A28B (Bottom) for three different algorithms, Crysol, FoXs, and BCL::SAXS

To compare the scattering profiles, we first normalized the experimental and calculated scattering intensities to be between (0, 1] by multiplying all the intensities in both sets by a constant scaling factor (β) :

$$\beta = \frac{1}{q(0)_{exp}} \tag{91}$$

As shown in Figure 9, the normalization process does not change the morphology of the curves. The only difference is the scale on the y-axis. All three algorithms appear to predict the same scattering profile from atomic coordinates. To magnify the effects of small distances, (higher q values), the scattering intensities (I) for both data sets were converted to a $\log_{10}$ scale.

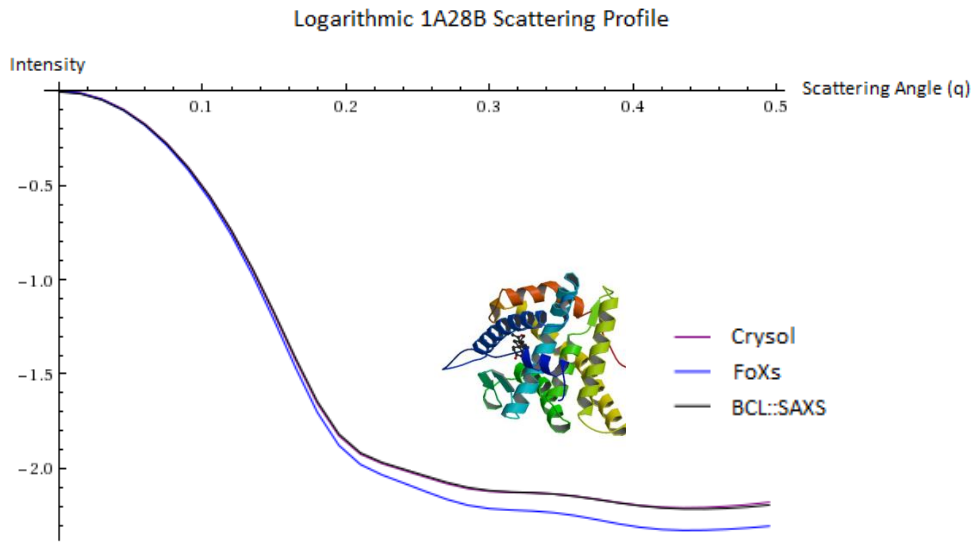


Figure 13: Logarithmic SAXS profile for 1A28B

The scattering profiles depicted on the logarithmic scale show the deviation in scattering profiles between the FoXs algorithm and the Crysol algorithm. The scattering profile generated through BCL::SAXS is directly superimposed on the scattering profile generated through Crysol. Using cubic splines, the derivative of the intensities for both data sets were computed. Similar to other approaches to modeling proteins from a SAXS scattering profile [28, 58, 59], we score a model based on the difference between the profile obtained by CRYSOL [42] and the profile computed by our algorithm BCL::SAXS. The measure was used to quantify the difference between the derivatives of the two scattering curves.

$$s=\sqrt{\frac{\sum_{k=1}^{Q}(I_{exp}(q_k)-I_{cal}(q_k))^2}{Q}} \quad (92)$$

where Q is the number of entries in the data set.

Model Similarity Assessed by dRMSD SAXS Score

To measure the similarity between two SAXS profiles we computed the derivative of the profiles generated through CRYSOL and BCL::SAXS to compute the similarity score (dRMSD). In Table 4 we show the scattering profile scores for 1J27A, 2HUJA, 3FRRA, and 1PBVA without the derivative. The difference between the curves resulting from BCL::SAXS and CRYSOL is in units of "experimental" standard deviations between scatting profiles (s). In Table 5 we show the scattering profile score for the same proteins with the derivative included in the score. The difference between the curves resulting from BCL::SAXS and CRYSOL is in units of "experimental" standard deviations between derivatives of the scatting profiles. (ds). In each case the scattering curve obtained from CRYSOL is compared to the BCL::SAXS curve with approximated side chain and loop coordinates. In both of these tables, the diagonal represents protein self matching. We observe that by using the dRMSD score, we can recover protein pairs that would otherwise be mislabeled by using the RMSD score alone.

Table 4: SAXS profile scores

RMSD		BCL::SAXS model with loop approximation in units of s			
	PDB ID	**1J27A**	**2HUJA**	**3FRRA**	**1PBVA**
CRYSOL	**1J27A**	**0.069**	0.179	0.305	0.382
	2HUJA	0.140	**0.067**	0.134	0.186
	3FRRA	0.221	0.127	**0.078**	**0.097**
	1PBVA	0.246	0.167	0.126	0.117

Table 5: dRMSD SAXS profile scores

dRMSD		BCL::SAXS model with loop approximation in units of ds			
	PDB ID	**1J27A**	**2HUJA**	**3FRRA**	**1PBVA**
CRYSOL	**1J27A**	**0.67**	1.80	3.02	3.73
	2HUJA	1.25	**0.94**	1.77	2.33
	3FRRA	2.36	1.94	**1.12**	1.20
	1PBVA	2.38	2.20	1.87	**0.98**

Table 6: Stovgaard SAXS profile scores

Stovgaard Score		BCL::SAXS model with loop approximation in units of S			
	PDB ID	**1J27A**	**2HUJA**	**3FRRA**	**1PBVA**
CRYSOL	**1J27A**	4.81	12.19	21.32	29.81
	2HUJA	**3.71**	4.76	8.98	14.38
	3FRRA	5.09	**4.04**	4.53	8.27
	1PBVA	5.42	**4.21**	4.22	6.86

Table 7: Cumulative Integral SAXS profile scores

Cumulative Integral		BCL::SAXS model with loop approximation in units of c			
	PDB ID	**1J27A**	**2HUJA**	**3FRRA**	**1PBVA**
CRYSOL	**1J27A**	0.462	0.445	0.469	0.623
	2HUJA	**0.403**	0.381	0.401	0.555
	3FRRA	0.405	**0.380**	**0.392**	0.545
	1PBVA	0.422	**0.400**	0.416	0.564

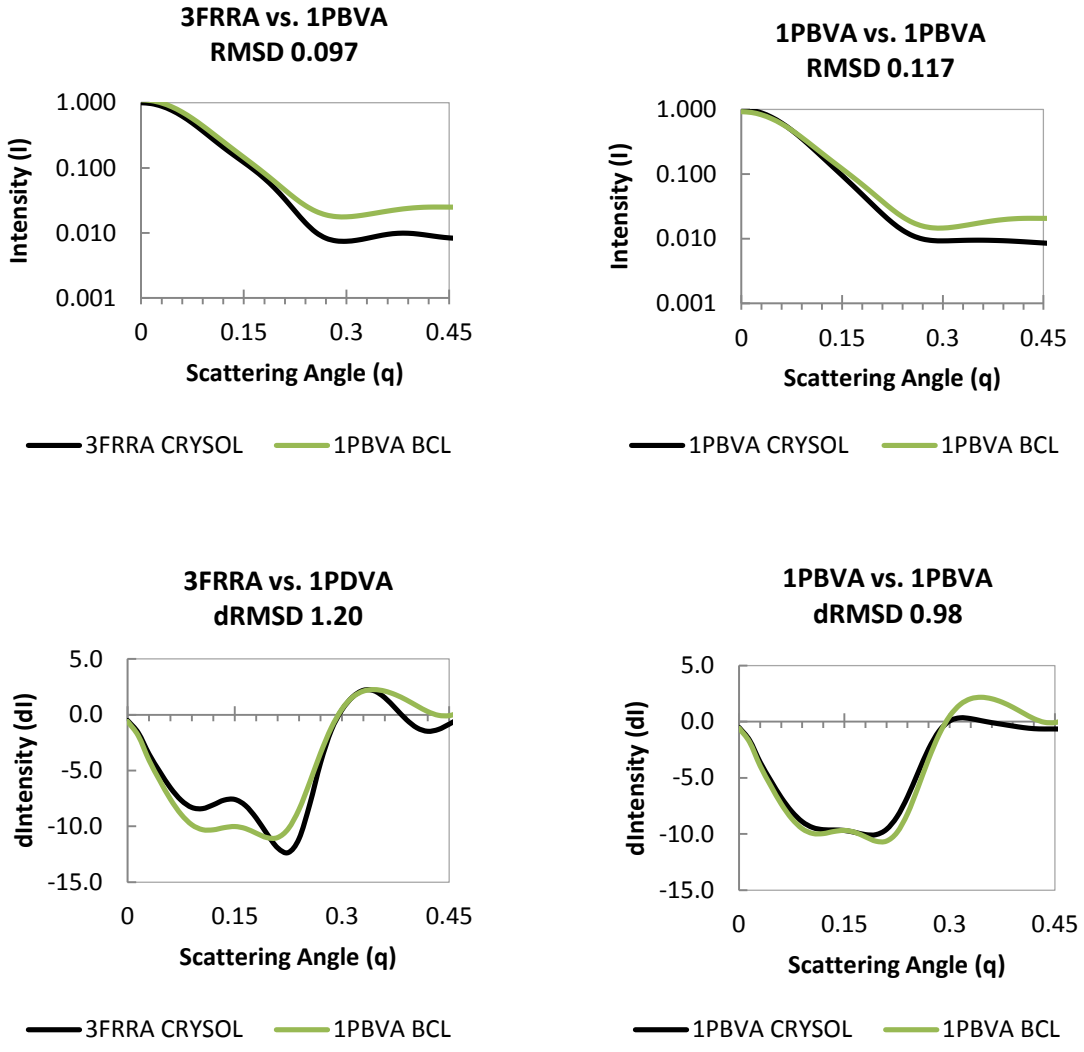


Figure 14: SAXS profiles of 3FRR and 1PBVA. The black curve is the scattering profile generated from the protein with the indicated PDB ID. The green curve is the scattering profile generated through BCL::SAXS for the protein with the indicated PDB ID with approximated side chains and loop regions.

The derivative score accentuates small directional changes in the SAXS profile. By doing this differences between the overall shapes of the two SAXS profiles under analysis are considered. (See Appendix IV) This properly corrects proteins that would otherwise be incorrectly identified.

Alternative Methods to Compare SAXS profiles

Stovgaard calculated the difference between two scattering curves with a similar approach [45]. The only difference is that rather than take the derivative of the functions, he divided by an estimated experimental error σ:

$$s=\sqrt{\frac{\sum_{k=1}^{Q}\left[\frac{\left(I_{exp}(q_k)-I_{cal}(q_k)\right)}{\sigma(q_k)}\right]^2}{Q}} \quad (93)$$

where σ is given by:

$$\sigma_q=I_q(q+\alpha)\beta \quad (94)$$

Previously the expression:

$$\sigma_q=I_q\beta \quad (95)$$

has been used in the literature as a realistic estimate of experimental error with β = 0.3.

To increase the precision in the portion of the curve between q = 0.1 and 1 =0.5 $Å^{-1}$, Stovgaard introduced a scaling factor (q+α) which is more strict at this scattering angle range with α = 0.15. Using this method, we measured the difference between BCL::SAXS and Crysol and compared it with the results reported in the Stovgaard paper.

Table 8: Replication of Stovgaard Score. Rg is radius of gyration, S is the Stovgaard Score, and Replicated S is the BCL::SAXS method scored using this method.

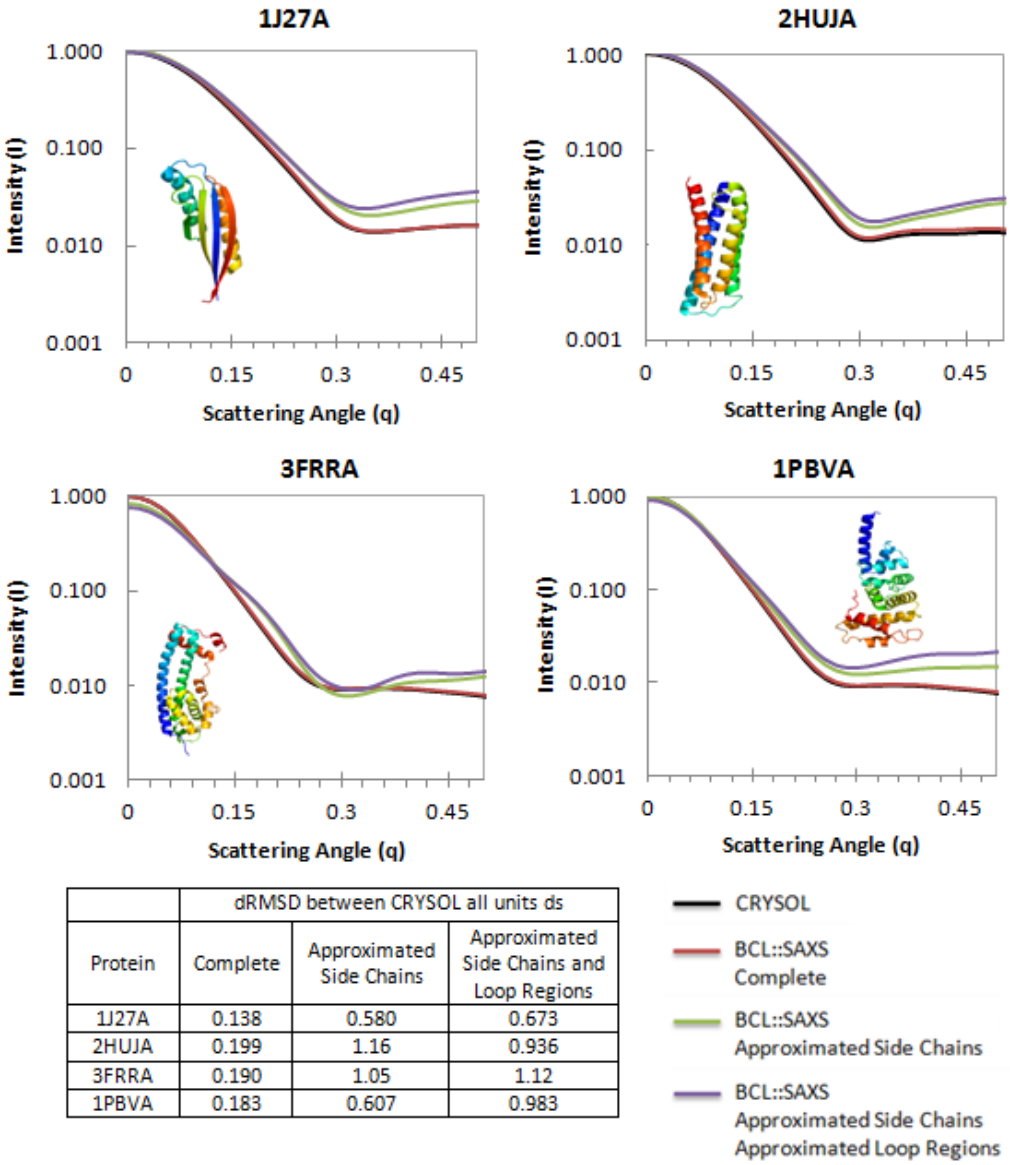
PDB ID	Chain	Length	Rg	S	Replicated S
1A28	B	249	10.39	0.300	0.173
1A3A	D	144	8.15	0.230	0.211
1AQU	A	281	10.73	0.177	0.165
1AQZ	A	142	8.48	0.208	0.183
1ATL	A	200	9.30	0.267	0.194
1ATZ	A	75	7.24	0.214	0.249
1AUO	A	218	9.34	0.137	0.204
1BBH	A	131	9.08	0.161	0.242

These results are significant for two reasons: 1) We successfully recreated the S score, and 2) In some cases BCL::SAXS outperformed the Stovgaard method using their scoring metric. (1A28, 1A3A, 1AQU, 1AQZ, and 1ATL).

When we applied the Stovgaard score in our loop approximation analysis, the score was unable to identify the native protein. (See Table 6) We tested a cumulative integral score that sums the difference between the sums of increasingly large portions of the range. Again, this method does not consider the shape of the curve in the analysis and failed to identify the native in three of four cases. (See Table 7) As a general trend we found that curve morphology was a critical feature for profile comparison with approximated loop regions and side chains. The derivative score was the only method that incorporated both morphology and curve deviation to correctly identify proteins from a small test set. This is the method we used for the rest of the analysis.

Accuracy of Calculated SAXS Profiles

To validate our SAXS method we compared our results with CRYSOL [42], SAXS profiles were generated using BCL::SAXS and CRYSOL for proteins in our benchmark dataset of 455 proteins. We then scored the difference between a given protein with itself and all other proteins using our scoring function described in the methods section. In figure 15, the SAXS comparison scores for complete models ranged from 0.138 ds (1J27A) to 0.199 ds (2HUJA). In our benchmark protein set the SAXS scores for complete models ranged from 0.0357 ds (1FD3A) to 1.424 ds (2ZWAA). Figure 15 depicts the SAXS scattering profiles generated from ridged bodies for 4 proteins: 1J27A, 2HUJA, 3FRRA, and 1PBVA. It also depicts the effect side chains approximation and loop region approximations have on the overall shape of the SAXS profile.



	dRMSD between CRYSOL all units ds		
Protein	Complete	Approximated Side Chains	Approximated Side Chains and Loop Regions
1J27A	0.138	0.580	0.673
2HUJA	0.199	1.16	0.936
3FRRA	0.190	1.05	1.12
1PBVA	0.183	0.607	0.983

Figure 15: BCL::SAXS comparison with CRYSOL for different levels of protein model approximation. Complete is for complete protein models, Approximated side chains represent the first approximation with the side chain summed and placed at the C-beta position of the residue. Approximated side chains and loop regions represent the first and second approximation with simulated loop regions. The dRMSD score is the RMSD of the derivative between the CRYSOL and the designated curve.

Non-redundant Dataset for Protein Discrimination Benchmark

To determine how well the SAXS score can distinguish different proteins from each other, we evaluated a random subset of 455 proteins with 20% identify cutoff, 1.6 Å resolution cutoff, and 0.25 R-factor cutoff from the PICES databank[60, 61]. These proteins can be formed

into a 455 x 455 matrix (207,025 pairings) where the diagonal represents a protein paired with itself (a true positive) and the off diagonal elements represents a protein paired with a different protein. Using scattering profiles generated through CRYSOL, we computed the difference between the native protein and the test protein for each pairing. If the minimum SAXS score for a given protein was on the diagonal for the ith row and jth column, then we correctly identified the protein from all other candidate proteins and classified that as a true positive. If the minimum SAXS score was not on the diagonal, we classified it as a false positive. Using receiver operating characteristic (ROC) curves, we plotted the false positive rate on the x axis and the true positive rate on the y-axis.

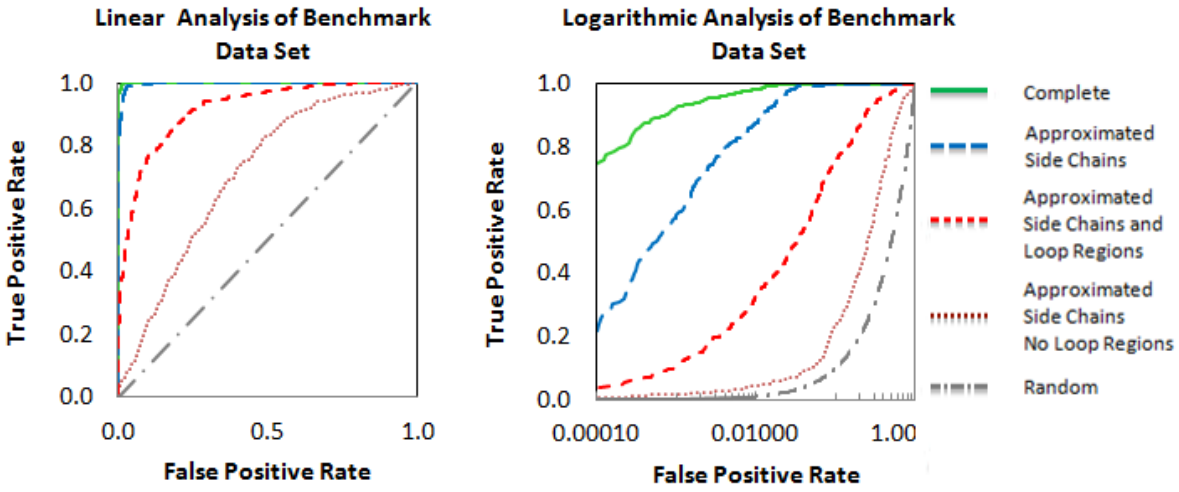


Figure 16: ROC analysis of 455 proteins from the benchmark dataset. The linear plot is on the left, while the logarithmic plot is on the right.

The area under the curve (AUC) for complete protein models is 99.95%. When side chains are removed, the AUC is 99.62%. The AUC for proteins without side chains and loop regions is 70.85%. When loop regions are approximated, the AUC is 92.64%. There were 207,025 total pairing evaluated in this experiment. In all but three cases the lowest SAXS score was the native protein when using complete protein models for analysis. For proteins 1YOZA and 3I31A the native was ranked second, while for protein 3L42A the native was ranked third.

SAXS dRMSD Score Analysis

Figure 17 depicts the range of SAXS scores from our benchmark set of 455 proteins for complete protein models. All of the self pairing scores had a dRMSD score below 1.5 ds, while non self pairing profiles had a dRMSD score ranging from 0.3 to over 13 ds. There is a window of overlap between the scores for matches and non-matches below 1.5 ds. As a loose guideline, based on this benchmark set, A SAXS dRMSD score above 1.5 ds indicate that the compared proteins are different. A SAXS dRMSD score below 1.5 ds indicate that the scattering profiles are similar, with the similarity increasing as the score decreases.

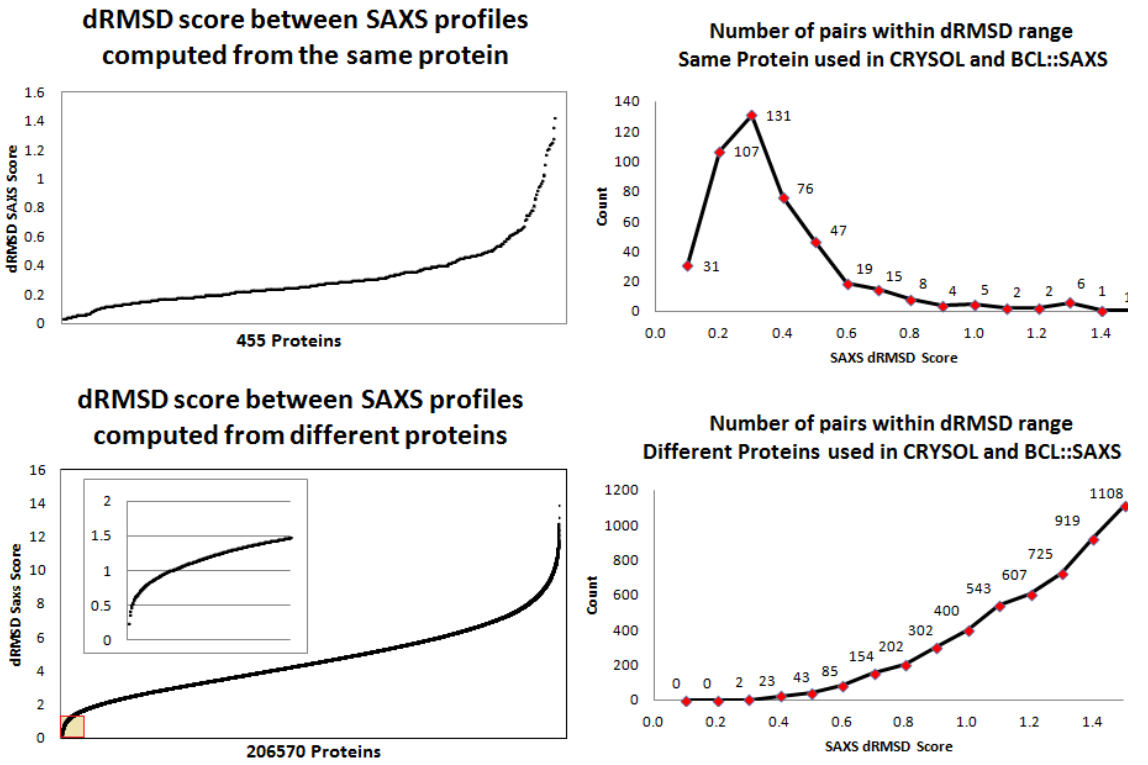


Figure 17: Range of SAXS dRMSD scores from benchmark set. Shown on the top left are the dRMSD scores computed between CRYSOL and BCL::SAXS profiles of the same protein. The top right plot depicts the distribution of these scores. Shown on the bottom left are the scores between CRYSOL and BCL::SAXS profiles where the protein used to compute the CRYSOL curve is different from the protein used to compute the BCL::SAXS curve. The bottom right depicts the distribution of these scores below a dRMSD cutoff of 1.5ds

In the benchmark dataset of 455 proteins there were 207,025 different combinations. Of these combinations, only 5568 have dRMSD scores below 1.5 ds. This is a reduction of 97% of the

models. The distribution of the scores in the range below 1.5 ds is skewed to the left in the case of identical protein pairs and skewed to the right in the case of different proteins. This explains why the AUC in the ROC analysis was 99.95. Most of the erroneous protein models were above 1.5 ds. For the models that were inside this range, the model with the minimum ds score was the native in all but three cases. This analysis shows that although multiple proteins can have the same SAXS profile, structurally distinct proteins can be distinguished by a SAXS profile below a cutoff of 1.5 ds.

Selecting the Optimal Parabolic Height Approximation Algorithm

Shown below are the results of the triangle and regula falsi loop approximation methods. The regula falsi method has a more pronounced arc in the path between two SSEs, while the triangle approximation has a smaller arc in the path. In comparison with the Native, we observe that the loop regions in 2HUJA are more flat that arched.

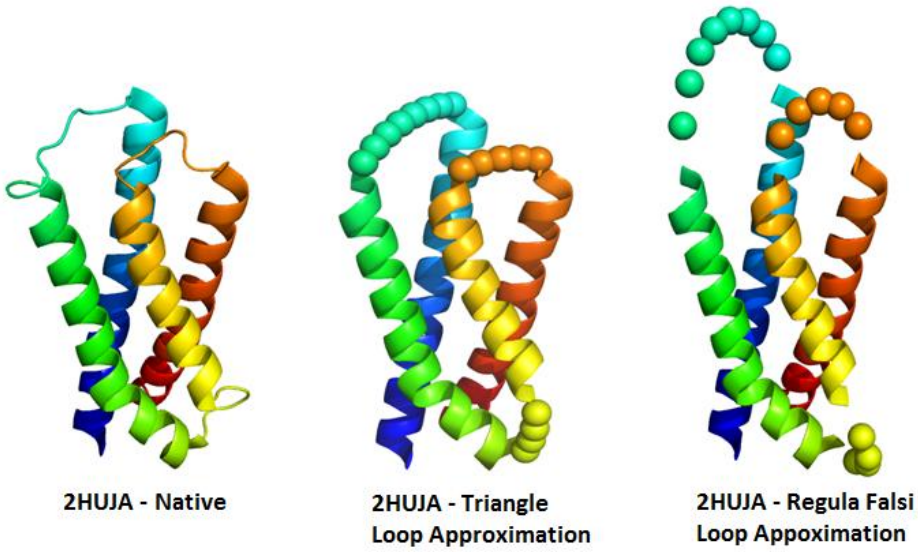


Figure 18: Loop Approximation results for triangle and regula falsi methods

The AUC for the triangle approximation was 92.64% and was used to on our benchmark protein set. The AUC for the regula falsi approximation was 81.74%. Although this was an

improvement over protein models with missing loop regions, the high arches in the pathway between SSEs negatively impacted the morphology of the SAXS profile. We decided to use the triangle approximation for rapid loop building.

Structural Similarity of Proteins with Similar SAXS Scores

To determine if protein models with similar SAXS scores were similar in protein structure, MAMMOTH [62] was used to rank structural similarity between two proteins. The 455 x 455 matrix (207,025 pairings) was used again to score the structural similarity of a pair of proteins. The diagonal represents self protein pairing. The higher the Z-score, the more similar the two structures are. The lower the SAXS dRMSD score, the more similar the two saxs profiles are. In this analysis, a high Z-Score and a low SAXS score indicate that proteins identified by SAXS as similar are structurally similar. Figure 19 shows that structurally similar proteins (high Mammoth Z-score) have a low SAXS score (bottom left corner). However, while structurally dissimilar proteins (low Mammoth Z-score) tend to have increased SAXS scores, the observed range of SAXS scores widens. As expected, structurally different proteins appear similar in a SAXS experiment if their overall shape is similar.

The MAMMOTH analysis shows that proteins with very similar z-scores (structurally similar proteins) also have a low SAXS dRMSD score. Importantly, the analysis shows that very similar structures do not have high SAXS scores. In the middle range of the analysis, we observe that SAXS scores are degenerate. Different structures can have similar SAXS scores. This degeneracy is inherently due to the spherical averaging of atoms in the SAXS data collection process. Because of this degeneracy SAXS cannot be used exclusively to predict protein structure.

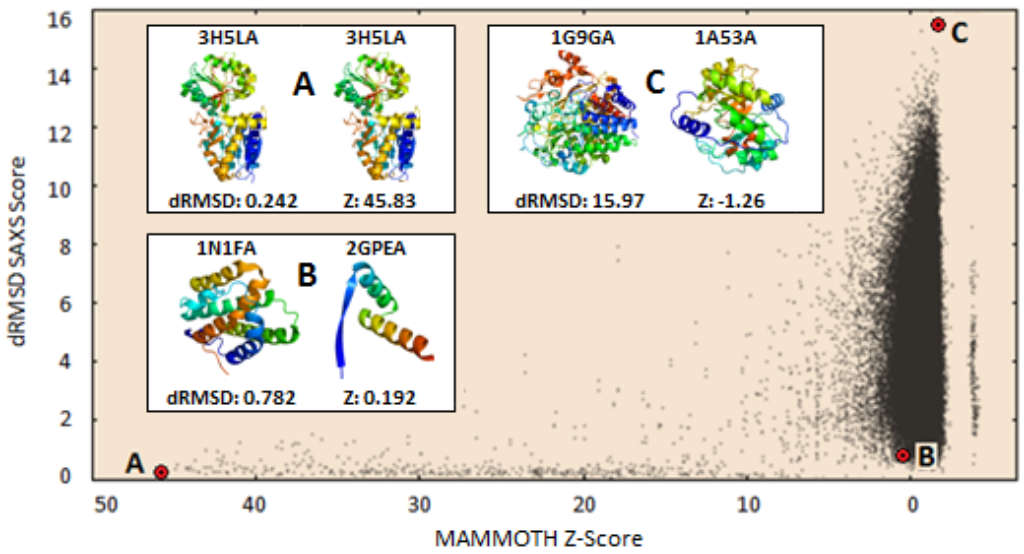
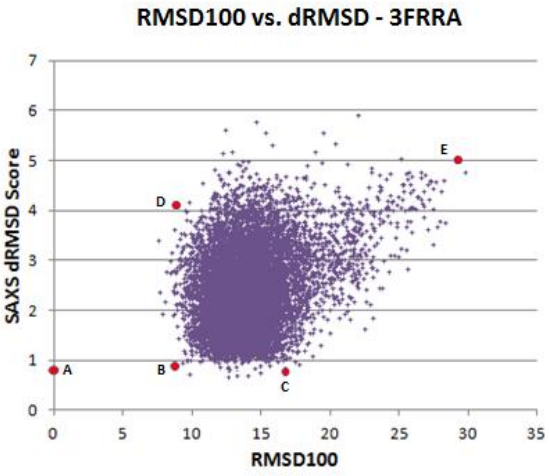
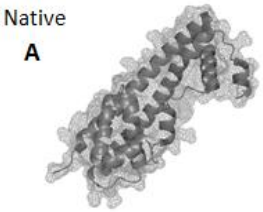


Figure 19: Structural MAMMOTH Z-score vs. SAXS profile similarity score. The x-axis represents the MAMMOTH z-score and with large values indicating structural similarity of two proteins. The y-axis represents the SAXS similarity score (dRMSD) with small values indicating similarity of two scattering profiles generated from proteins. Panels A, B, and C correlate with their respective red dot. Panel A depicts 3H5LA matched with itself. Panel B depicts 1N1FA matched with 2GPEA. Panel C depicts 1G9GA matched with 1A53A. In each panel, the SAXS score is dRMSD and the MAMMOTH Z-Score is Z.





Label	Protein Model	RMSD100	dRMSD
A	Native	0	0.80
B	1007_0048	8.75	0.88
C	1048_0054	16.77	0.77
D	1071_0034	8.81	4.05
E	1095_0009	29.19	5.02

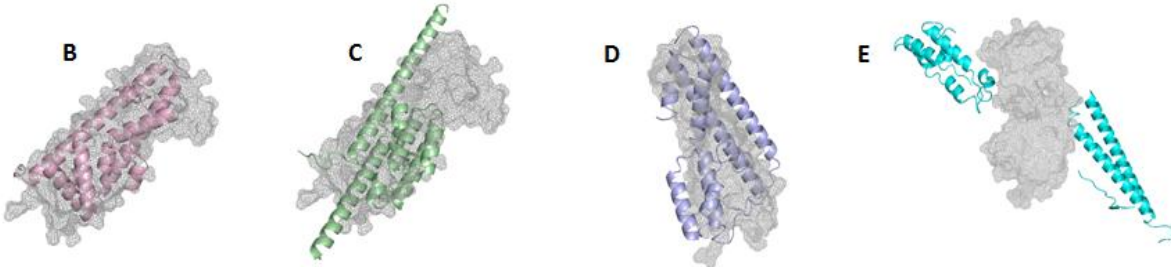


Figure 20: RMSD100 vs. dRMSD SAXS score for single protein. BCL::SAXS was used to score 10,000 protein confirmations of 3FRRA from the BCL::Fold benchmark set. In each case the surface mesh of the native confirmation of 3FRRA is in shown in gray. Each model represents a different state and is identified by a red dot in the RMSD100 vs. SAXS dRMSD plot.

In the MAMMOTH analysis (figure 19), we compared two different proteins. Figure 20 depicts how the SAXS dRMSD score is used to compare different topologies of the same protein. RMSD100 is a normalized root-mean-square distance for comparing protein structures[63].

SAXS Restraints Incorporated into BCL::Fold during Protein Folding

The BCL::SAXS score was added to the minimization process in BCL::Fold for 1CI6A, 3PDYA, 1PBVA, and 1J27A. During the assembly process of secondary structure elements, the SAXS score was used to penalize configurations that deviated from the experimental restraint. For each folding simulation, 1000 models were generated in BCL::Fold The results of four soluble protein examples are shown in Table 9 and visualized in figure 21.

Table 9: Mean RMSD100 score and standard deviation for top 5% models folded in BCL::Fold with and without SAXS restraints. The PVALUE is the probability that the mean shift is due to chance. The top 10% enrichment is given by the intersection of the set of top 10% models ranked by RMSD100 and the set of top 10% models ranked by SAXS dRMSD score.

PDB	BCL::Fold Type	Mean	STDEV	PVALUE	Top 10% Enrichment
1CI6A	No SAXS Restraint	9.89	2.60	$2.79\ e^{-24}$	3.6
	SAXS Restraint	2.22	1.27		0.7
1PBVA	No SAXS Restraint	11.39	0.48	$2.13\ e^{-9}$	1.4
	SAXS Restraint	10.70	0.48		1.4
3PDYA	No SAXS Restraint	14.56	1.20	$3.10\ e^{-3}$	4.8
	SAXS Restraint	13.71	0.90		1.4
1J27A	No SAXS Restraint	9.11	0.69	$4.06\ e^{-1}$	0.9
	SAXS Restraint	9.08	0.96		0.9

Figure 21 shows the RMSD100 distribution plots for our example proteins during protein folding with BCL::Fold. For each protein, 1000 models were generated through BCL::Fold with and without SAXS restraints.

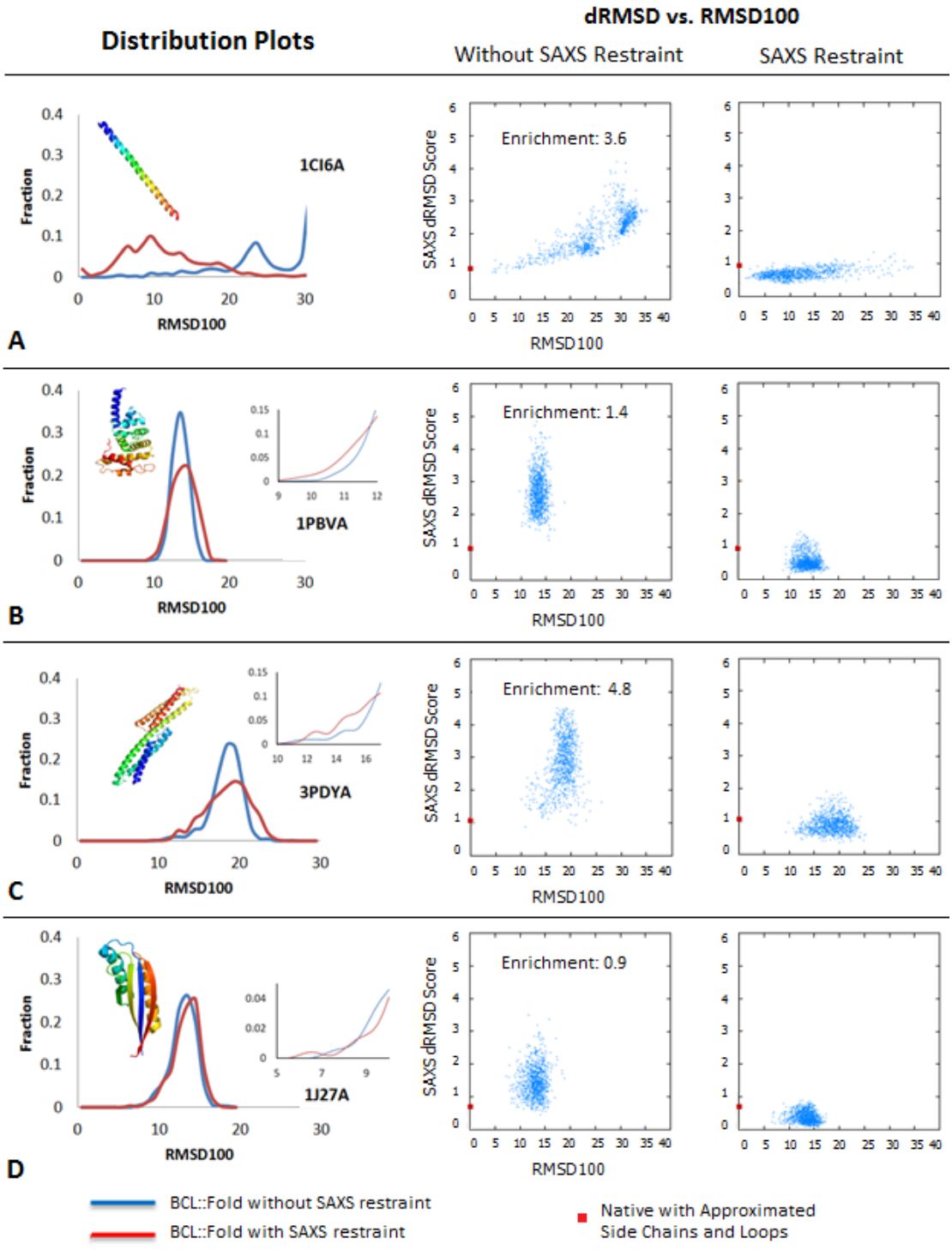


Figure 21: Folding results for 1000 models generated with and without using SAXS as a weighted term in energy function of BCL::Fold. The proteins 1CL6A, 1PBVA, 3PDYA and 1J17A are shown. (Panels A, B, C, and D respectively) On the left the distribution plots depict the difference between folding with and without SAXS restraints. On the right the SAXS score vs. RMSD100 plots are shown. In these plots, the native protein (red dot) was obtained from the PDB. The side chain and loop region coordinates were removed and then

The top 10% enrichments were computed by counting the number of models given by the intersection of set of top 100 models ranked by RMSD100 and the set of top 100 models ranked by SAXS dRMSD score and dividing by ten. A perfect enrichment score would be ten. An enrichment value of one means the dRMSD score did not help nor hinder the ranking process. A score below one indicates that the dRMSD score negatively affected the ranking, while a score above one indicates that the dRMSD score positively affected the protein ranking. The dRMSD vs. RMSD100 plots are shown on the right side of figure 21.

CHAPTER VI

DISCUSSION

We have implemented an innovative technique to compute SAXS profiles from atomic coordinates. In our approach we did not make approximations to the Debye formula, rather we used GPU acceleration to handle the double summation of all atoms. To our knowledge this is the first time GPU acceleration has been used in the Debye formula to compute SAXS profiles. We were able to consistently replicate the scattering profiles generated by CRYSOL. By using the Debye formula we obtained direct control of the scattering profile calculation. This provided the opportunity to rapidly approximate the side-chain and loop region positions of a given protein model and compute a scattering profile. The deviation between this scattering profile and the scattering profile generated by CRYSOL was used as a restraint in BCL::Fold.

Because of the low resolution of the SAXS method, it cannot be used exclusively to identify the native protein configuration from a set of similar protein configurations. We have shown however, that although the information content in the SAXS profile is limited, it can be used to filter erroneous protein models early in the prediction process thus focusing computation time on models that fit the experimental data.

For this project to be successful, there were some key challenges that had to be solved. First, we had to find a method to compute SAXS profiles from atomic coordinates. Second, we had to have a scoring function to compare the similarity of two SAXS profiles. Third, we had to develop a method to approximate models with missing side chains and missing loop regions. Forth, we had to benchmark our results. Once we could generate SAXS profiles from complete

protein models, it was clear how important the shape of the SAXS profile is when comparing two profiles. When we summed the side chain coordinates to the C-β position of its respective residue and compared the SAXS profile generated with this approximation to the profile generated through CRYSOL, the profile generated from the approximation was vertically offset from the profile generated from CRYSOL. The overall shape of the curve remained the same. To account for this behavior, we computed the first derivative of the profiles and then computed the similarity score, dRMSD, between the derivatives of the SAXS profiles. By using the derivative score, we reduced the amount of false positives obtained during our analysis with our benchmark protein set.

Using this scoring metric, BCL::SAXS was 99.95% accurate in picking the native protein from a set of other proteins. With the side chains approximated, BCL::SAXS was 99.62% accurate in picking the native protein from a set of other proteins. With the loop regions removed, the accuracy dropped from 99.62% to 70.85%. This result shows that loop regions play an important role in protein topology. Using our loop approximation algorithm, the accuracy increased to 92.64%. This result shows that having an approximate estimate of a protein location can have significant impact on the accuracy of SAXS scattering profiles generated from atomic coordinates.

The derivation of the loop approximation method was a learning process. We first attempted the midpoint approximation, followed by the linear approximation, and then used the curvilinear approximation. Using the curvilinear approximation we had to derive the normalization factor N. Our first approach to calculate N was the regula falsi optimization protocol with parabolic arc length computations. This was computationally expensive and mathematically complex. After successfully implementing the method we noticed that loop

regions between SSEs are not large parabolic arcs between SSEs (see figure 18), but rather they curve and weave around the SSEs. The precision gained by the optimization protocol was unnecessary. Substituting the entire protocol with one line of code (the triangular approximation) increased the speed and accuracy of the calculation. This experience reminded me of the words of Dr. Richard Hamming; "The purpose of computing is insight, not numbers."

There are two classification levels of protein structure that we considered in this work. The first level is how the SAXS score can be used to distinguish different proteins from one another. The second level is how the SAXS score can be used to distinguish different conformations of the same protein. The MAMMOTH analysis (figure 19) depicts different proteins. This analysis highlights the point that structurally similar proteins have a low saxs scores (sample A), structurally different proteins can have low saxs scores (sample B) and structurally different proteins can have high saxs scores (sample C). The trend is that two proteins that are structurally similar do not have a high saxs score. If two proteins are structurally similar then they will have a low saxs score. The opposite is not true. If two proteins have a low saxs score, that does not mean they are structurally similar.

Figure 20 depicts different confirmations of the same protein. Sample B and C have similar SAXS scores with the native, but have RMSD value of 8.75 and 16.77 respectively. This enforces the observation that if two proteins have a low SAXS score, that does not mean they are structurally similar. In fact, we observe many different topologies with low SAXS scores but large RMSD100 differences.

With these observations in mind, we used SAXS restraints during protein folding. Figure 21 shows the results two different folding runs of four proteins with 1000 models generated for each protein in BCL::Fold. In the first folding run, BCL::SAXS was used to rank the

1000 models generated by BCL::Fold. We observe enrichment all cases except for 1J27A. In this case the enrichment value was 0.9. In the second case, we used SAXS as a term in the knowledge based scoring function of BCL::Fold. By doing this we clearly observed that low SAXS scores do not imply structural similarity. There was a statistically significant improvement of the mean RMSD100 values in the top 5% of the models, but no models were below 8Å cutoff. These results show that SAXS can be very useful to score models after they have been generated but may not be the best choice for generating initial protein models. This may in fact restrict our ability to effectively sample the energy landscape of a given protein by prematurely forcing models to adhere to SAXS restraints.

Computation of SAXS profiles can be used to validate high-resolution models in solution and to identify biologically active protein conformations. SAXS can also be used to characterize complexes whose components have known atomic structures. These components act as building blocks that can be arranged to form complexes where the scattering from the complex fits the experimental data. Investigators interested in large complexes can use BCL::SAXS to generate computed SAXS profiles for permutations of a given complex and identify the experimental configuration.

Investigators interested in protein docking studies can use BCL::SAXS to generate computed SAXS profiles of receptor-ligand complexes to identify likely receptor-ligand configurations and compare their proposed models with experimental data to identify the correct configuration of the system.

Furthermore, SAXS is another experimental technique that can now be used by BCL::Fold to aid in protein structure prediction. Although, SAXS cannot unambiguously identify the correct protein topology from a group of structures of similar shape, it can be used to filter

away erroneous models, thus focusing further computation on more feasible backbone topologies. Small globular proteins are not amenable to this approach in protein structure prediction. Interestingly, the SAXS experimental technique seems to be suited best for large, highly variable protein topologies. - Opposite that of X-ray crystallography and NMR. SAXS provides a means of studying assembly and large-scale conformational changes. Further work must be done to benchmark SAXS with large variable proteins.

CHAPTER VI

CONCLUSION

Summary

The SAXS profile simulated from protein models in silico can be used to distinguish different proteins from each other, but cannot be used exclusively to distinguish different permutations of the same protein topology. Despite this limitation, SAXS can be used to filter protein models that are very different from the native from further analysis during protein structure prediction saving time, money, and computational resources. BCL::SAXS has potential use for the scientific community to investigate protein-protein confirmations, and quaternary protein structure orientation. These SAXS profiles can provide additional structural information and can be very useful when combined with NMR or EPR.

We showed that side chains do not dramatically change shape of the overall scattering profile obtained from a given protein and that they can be approximated as a point at the C_β position of the peptide backbone. We showed that the loop regions between secondary structure elements are critical to the overall morphology of the SAXS profile and that they can be coarsely approximated to improve protein identification. We were able to identify the correct fold for 1CI6A, but as the complexity of the protein increased, the number of degenerate confirmations increased.

Study Limitations

Because of the lack of experimental SAXS data, CRYSOL was used to simulate experimental SAXS curves. This work must be benchmarked with experimental SAXS data.

Future Work

Further work should be done to extend this analysis to larger multimeric protein complexes and membrane proteins. The literature shows that SAXS has been successfully combined with NMR to produce more native-like protein structures as opposed to using either method alone. BCL::SAXS should be combined with other experimental methods such as NMR and EPR to incorporate as many sparse experimental restraints as possible for protein structure prediction in BCL::Fold.

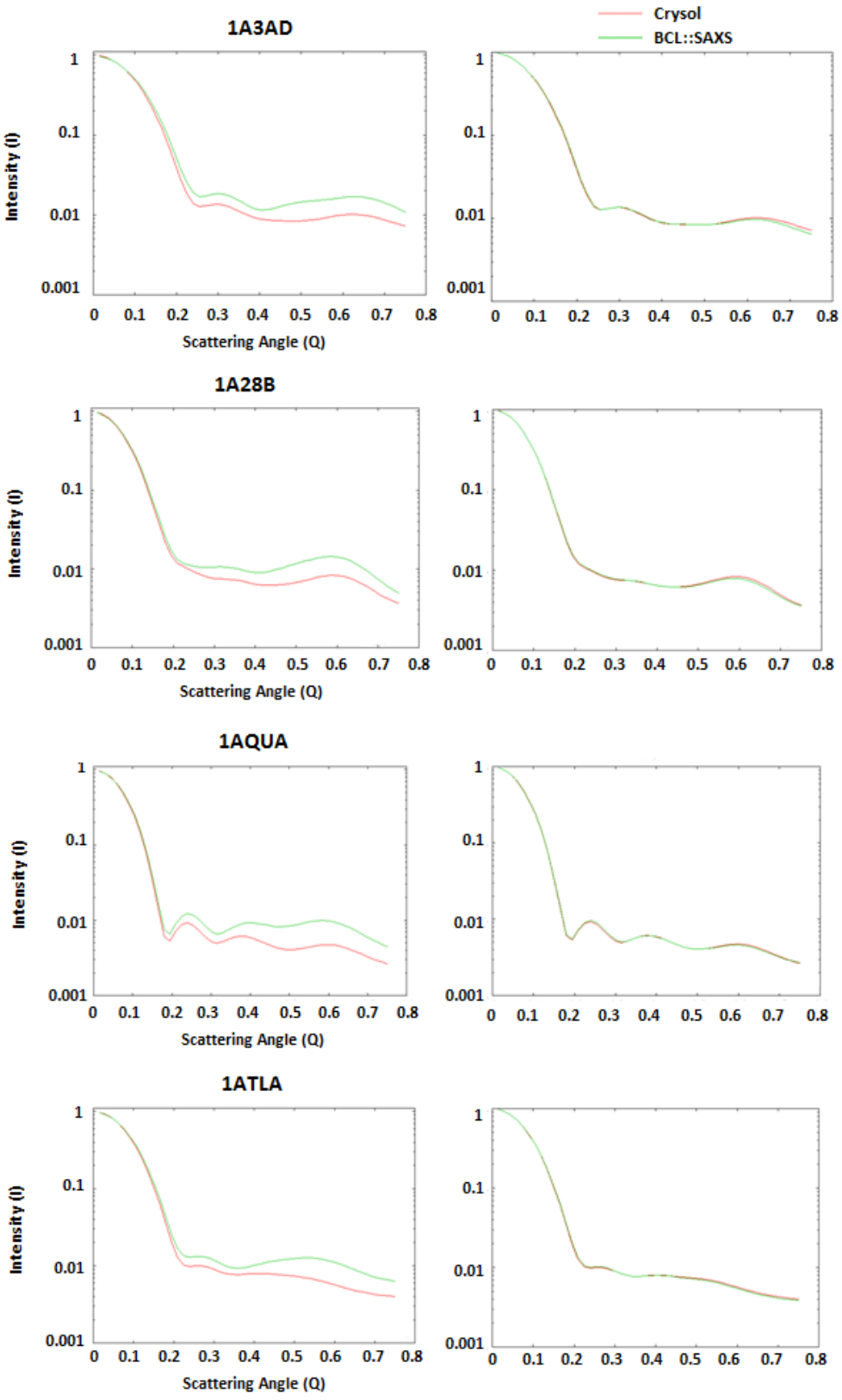
CHAPTER VII

APPENDIX I – 1ENH C_α- C_α spacing: 3.8Å Average

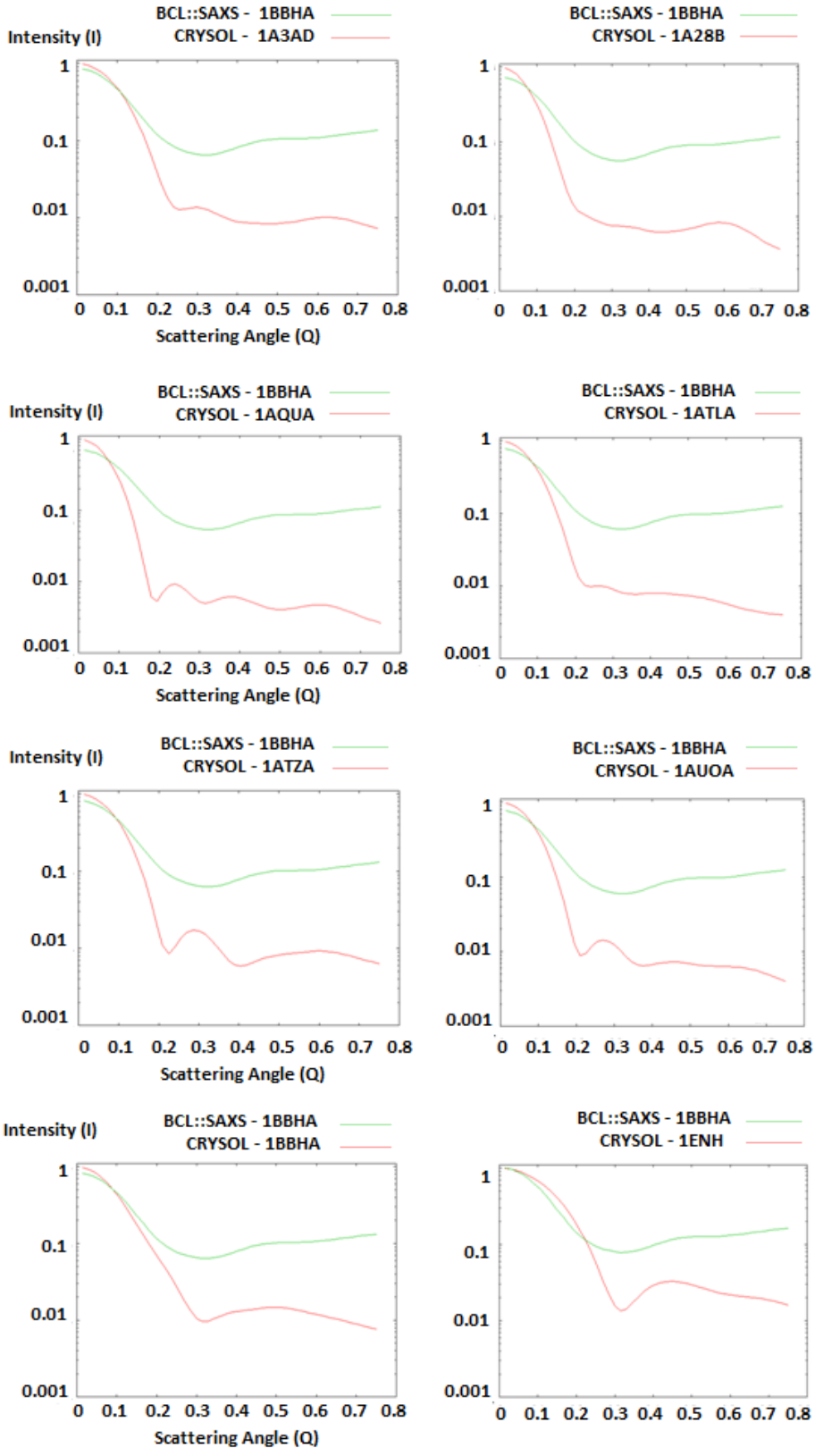
Atom Num	Atom	Residue	Residue Num	X	Y	Z	delta X	delta Y	delta Z	Distance (Spacing)
2	CA	ARG	3	3.22	44.97	51.87	0.00	0.00	0.00	0.00
13	CA	PRO	4	0.60	47.33	50.35	2.62	-2.36	1.52	3.84
20	CA	ARG	5	-0.88	46.16	47.07	1.48	1.17	3.28	3.78
31	CA	THR	6	0.73	47.49	43.90	-1.61	-1.33	3.17	3.79
38	CA	ALA	7	-1.24	48.84	40.93	1.97	-1.36	2.98	3.82
43	CA	PHE	8	-1.94	46.20	38.32	0.70	2.65	2.61	3.78
54	CA	SER	9	-2.13	46.94	34.66	0.19	-0.74	3.66	3.74
60	CA	SER	10	-5.24	46.24	32.55	3.11	0.71	2.11	3.82
66	CA	GLU	11	-3.39	43.46	30.86	-1.85	2.78	1.69	3.74
75	CA	GLN	12	-2.14	42.02	34.15	-1.25	1.44	-3.29	3.80
84	CA	LEU	13	-5.65	42.06	35.60	3.51	-0.03	-1.44	3.80
92	CA	ALA	14	-7.06	40.39	32.48	1.41	1.67	3.12	3.80
97	CA	ARG	15	-4.55	37.61	32.78	-2.51	2.78	-0.30	3.76
108	CA	LEU	16	-5.17	37.25	36.53	0.61	0.37	-3.75	3.82
116	CA	LYS	17	-8.92	36.99	36.11	3.75	0.25	0.42	3.79
125	CA	ARG	18	-8.54	34.40	33.41	-0.39	2.59	2.70	3.76
136	CA	GLU	19	-6.18	32.33	35.60	-2.36	2.07	-2.20	3.83
145	CA	PHE	20	-8.55	32.67	38.56	2.37	-0.34	-2.96	3.81
156	CA	ASN	21	-11.34	31.34	36.36	2.80	1.33	2.20	3.80
164	CA	GLU	22	-9.25	28.29	35.62	-2.10	3.05	0.74	3.77
173	CA	ASN	23	-8.22	27.49	39.18	-1.03	0.80	-3.56	3.79
181	CA	ARG	24	-8.78	29.76	42.10	0.56	-2.27	-2.92	3.74
192	CA	TYR	25	-6.13	28.22	44.33	-2.65	1.55	-2.23	3.79
204	CA	LEU	26	-2.54	28.89	43.45	-3.58	-0.67	0.88	3.75
212	CA	THR	27	0.23	26.33	43.84	-2.78	2.56	-0.39	3.80
219	CA	GLU	28	3.67	27.79	44.68	-3.44	-1.46	-0.85	3.83
228	CA	ARG	29	4.98	26.73	41.31	-1.31	1.06	3.37	3.76
239	CA	ARG	30	2.15	28.37	39.34	2.83	-1.64	1.98	3.82
250	CA	ARG	31	2.53	31.53	41.41	-0.38	-3.16	-2.08	3.80
261	CA	GLN	32	6.21	31.54	40.48	-3.68	-0.01	0.94	3.80
270	CA	GLN	33	5.47	31.02	36.80	0.74	0.52	3.68	3.79
279	CA	LEU	34	2.82	33.74	36.92	2.65	-2.72	-0.12	3.80
287	CA	SER	35	5.27	36.08	38.61	-2.45	-2.35	-1.69	3.79

293	CA	SER	36	7.79	35.51	35.74	-2.52	0.57	2.87	3.86
299	CA	GLU	37	5.23	35.94	33.03	2.56	-0.43	2.71	3.75
308	CA	LEU	38	3.46	38.99	34.43	1.77	-3.05	-1.40	3.79
316	CA	GLY	39	6.34	40.90	35.99	-2.88	-1.91	-1.56	3.79
320	CA	LEU	40	4.66	41.00	39.41	1.68	-0.10	-3.42	3.81
328	CA	ASN	41	6.08	39.98	42.80	-1.42	1.02	-3.38	3.81
336	CA	GLU	42	4.80	36.52	43.85	1.27	3.47	-1.06	3.84
345	CA	ALA	43	3.51	38.18	46.98	1.29	-1.66	-3.13	3.77
350	CA	GLN	44	1.29	40.51	45.00	2.22	-2.33	1.99	3.78
359	CA	ILE	45	-0.29	37.60	43.14	1.58	2.91	1.86	3.80
367	CA	LYS	46	-0.93	35.67	46.35	0.64	1.93	-3.21	3.80
376	CA	ILE	47	-2.74	38.63	47.93	1.81	-2.97	-1.59	3.82
384	CA	TRP	48	-4.69	39.28	44.74	1.95	-0.64	3.19	3.80
398	CA	PHE	49	-5.83	35.61	44.76	1.13	3.67	-0.02	3.84
409	CA	GLN	50	-6.77	35.65	48.49	0.94	-0.04	-3.73	3.85
418	CA	ASN	51	-8.79	38.85	48.15	2.02	-3.21	0.34	3.80
426	CA	LYS	52	-10.47	37.65	45.03	1.69	1.21	3.13	3.75
435	CA	ARG	53	-11.66	34.48	46.81	1.19	3.17	-1.79	3.83
446	CA	ALA	54	-12.90	36.57	49.72	1.24	-2.09	-2.91	3.79
451	CA	LYS	55	-14.94	38.86	47.44	2.03	-2.29	2.28	3.82
460	CA	ILE	56	-16.25	35.74	45.71	1.32	3.11	1.73	3.80

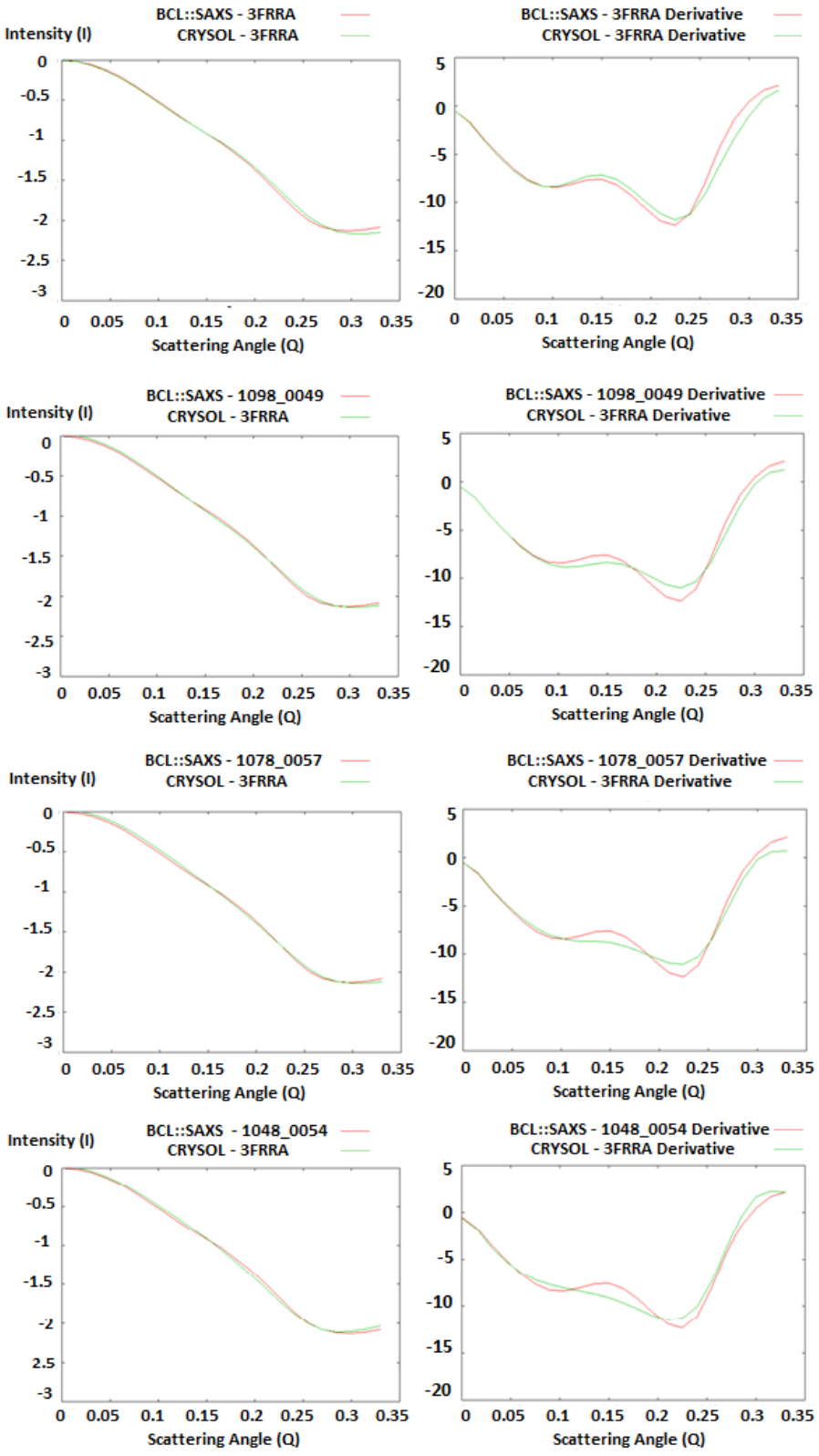
APPENDIX II – SAXS profiles before and after the Debye formula correction



APPENDIX III – 1BBHA BCL::SAXS profile with Midpoint Loop Approximation



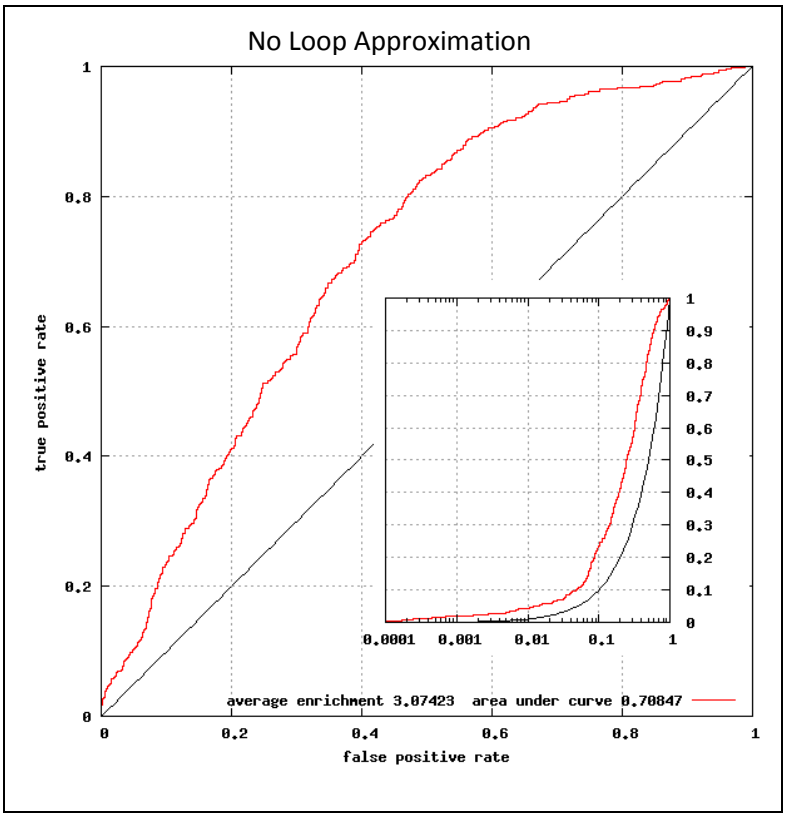
APPENDIX IV- Logarithmic and Derivative SAXS Profiles of 3FRRA Topologies

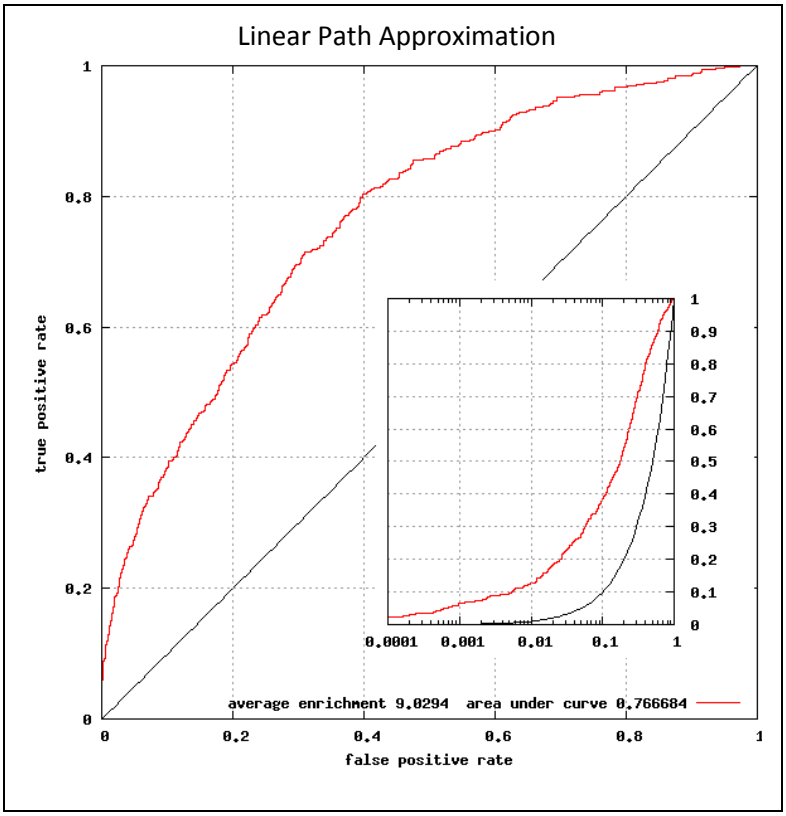


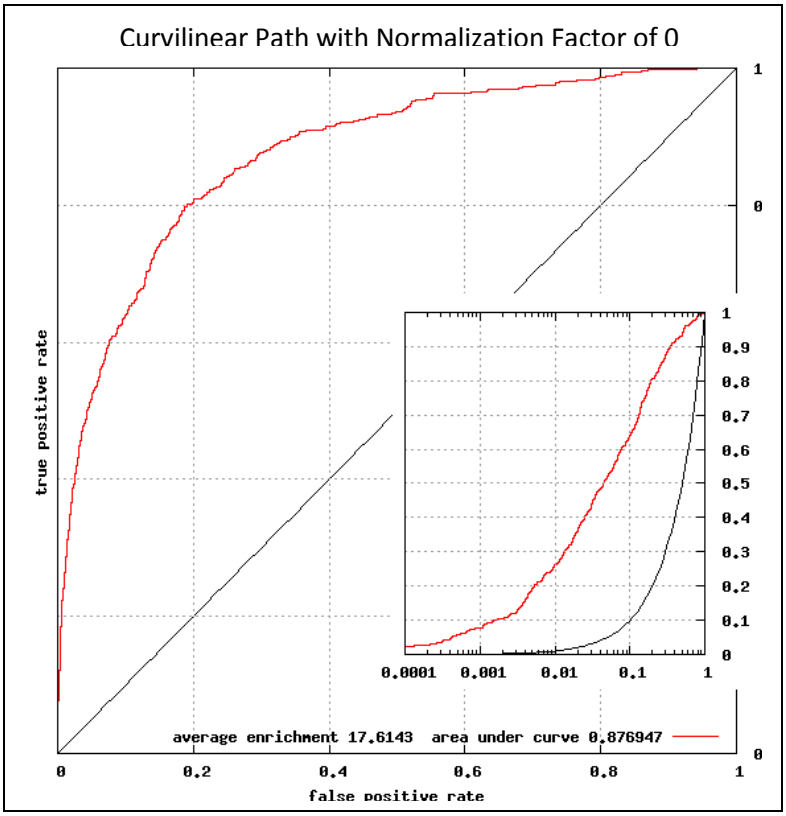
APPENDIX V –BCL::SAXS Benchmark set of 455 proteins

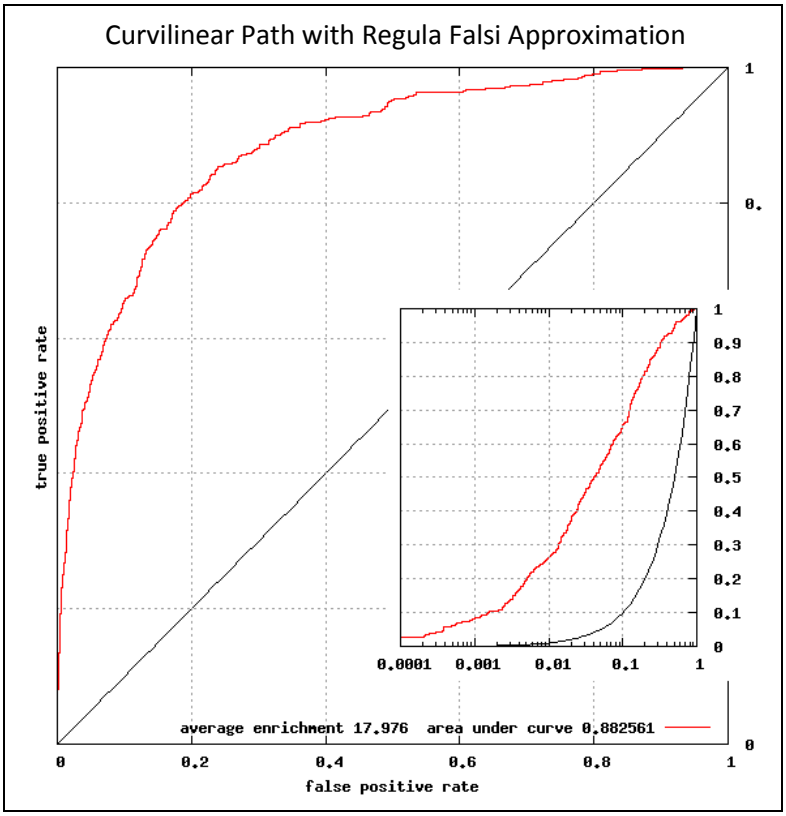
PDB ID								
1A53A	1GWYA	1MG7A	1RYIA	1WC2A	2BJFA	2GAGB	2OPCA	2VQPA
1AH7A	1H03P	1MK4A	1RYPL	1WHIA	2BKMA	2GDMA	2OV0A	2VTWA
1AOLA	1H32A	1MTPA	1S2WA	1WLZA	2BMOA	2GIYA	2OYAA	2VXTI
1B0NB	1H97A	1MUWA	1SA3A	1WOJA	2BOUA	2GPEA	2P14A	2W1JA
1B8ZA	1HE1A	1MXRA	1SFPA	1WRAA	2BURA	2GUDA	2P51A	2W3QA
1BKRA	1HQ0A	1N1FA	1SKZA	1WUIS	2C0HA	2GYQA	2P9XA	2W5NA
1BX7A	1HXNA	1N5UA	1SR8A	1WWRA	2C61A	2H2ZA	2PBIB	2W83C
1C1YB	1I0RA	1N7SD	1SVBA	1WZUA	2C8MA	2H8EA	2PGNA	2WB0X
1C7SA	1I2KA	1NFPA	1SZHA	1X6OA	2CBZA	2HBAA	2PMUA	2WDSA
1CEOA	1I71A	1NKDA	1T0TV	1XAKA	2CE2X	2HEUA	2PQ8A	2WFWA
1COZA	1IG0A	1NLSA	1T3YA	1XE7A	2CI1A	2HIMA	2PTHA	2WI8A
1CV8A	1IM5A	1NPYA	1T6SA	1XGKA	2CKKA	2HLYA	2PVBA	2WKJA
1D0CA	1ISUA	1NTHA	1T9IA	1XKRA	2CNQA	2HQSC	2Q0SA	2WNHA
1D2VA	1J0HA	1NWZA	1TFEA	1XOVA	2CVEA	2HU9A	2Q5CA	2WQIA
1DC1A	1J2RA	1O26A	1TIGA	1XU1R	2D0OA	2HYKA	2Q9FA	2WUXA
1DG6A	1JB7B	1O97C	1TP6A	1XZZA	2D5BA	2I53A	2QDXA	2WY4A
1DJ8A	1JEKA	1OD3A	1TT8A	1Y66A	2DC4A	2I8TA	2QFEA	2WZOA
1DQPA	1JH6A	1OGDA	1TVXA	1Y96B	2DEJA	2IDLA	2QISA	2X5XA
1DY5A	1JIXA	1OIHA	1U07A	1YD9A	2DOKA	2II2A	2QKPA	2YSKA
1E4FT	1JMVA	1OOEA	1U5KA	1YGTA	2DQWA	2IMFA	2QQ9A	2YXOA
1EAYC	1JQ5A	1OSYA	1U7PA	1YLKA	2DTJA	2IPIA	2QTWA	2Z0TA
1EF8A	1JX6A	1OZ2A	1UCRA	1YOZA	2DYJA	2IW1A	2QWOB	2Z3QA
1ELKA	1JYOE	1P5VA	1UG6A	1YT3A	2E2DC	2IY2A	2R25A	2Z6OA
1EUVA	1K3YA	1PBYC	1UJPA	1YZ1A	2E56A	2J1VA	2R6JA	2Z98A
1EXTA	1K8KC	1PJXA	1UPQA	1Z21A	2E85A	2J6LA	2RB8A	2ZD7A
1F00I	1KA1A	1PSRA	1USGA	1Z6OA	2EBNA	2J9OA	2REEA	2ZFYA
1F3UA	1KMJA	1Q0PA	1UUYA	1ZC3B	2EHZA	2JDID	2RINA	2ZK9X
1F86A	1KPTA	1Q6OA	1UWKA	1ZHXA	2ENDA	2JE6I	2RKLA	2ZPTX
1FD3A	1KU3A	1QAZA	1V2BA	1ZL0A	2ET1A	2JGPA	2SPCA	2ZSIB
1FM0D	1KYFA	1QHDA	1V5VA	1ZUUA	2F01A	2JLQA	2UUYB	2ZWAA
1FS7A	1L3KA	1QNRA	1V84A	1ZZKA	2F5GA	2NNUA	2UXQA	2ZYZB
1FYEA	1L7LA	1QQP4	1VBWA	2A2KA	2FAOA	2NRRA	2UZ1A	3A1FA
1G4YB	1LFWA	1QW9A	1VE2A	2A7BA	2FCWA	2NVHA	2V33A	3A6FA
1G6XA	1LM5A	1R17A	1VHWA	2AEBA	2FFUA	2NX4A	2V7FA	3ABDX
1G9GA	1LQTA	1R6JA	1VLSA	2AKZA	2FIPA	2O0QA	2V9KA	3B47A
1GK9B	1LTZA	1R8SA	1VSRA	2ARCA	2FMAA	2O6FA	2VCHA	3B6HA
1GO3F	1LZLA	1RFYA	1W07A	2AXWA	2FPHX	2O90A	2VFRA	3BBBA
1GS5A	1M1NB	1RK6A	1W4XA	2B3GA	2FTXB	2ODFA	2VHKA	3BFOA
1GU2A	1M4LA	1RSSA	1W6SB	2B9DA	2FYGA	2OH5A	2VLQA	3BHWA
1GVDA	1M9ZA	1RW7A	1WA5C	2BCMA	2G40A	2OKMA	2VOVA	3BL9A
3BNEA	3CJKB	3DR9A	3EW8A	3FVHA	3H0UA	3I31A	3K8UA	3LKEA
3BONA	3CL6A	3DVOA	3F2EA	3FXQA	3H5LA	3I84A	3KF6B	3LLUA
3BQAA	3CP3A	3E1RA	3F6GA	3G2BA	3H79A	3IDBB	3KJDA	3LQSA
3BSOA	3CT5A	3E4WA	3F8MA	3G4EA	3H8TA	3IISM	3KQ5A	3LW3A
3BWUD	3CYPB	3E8TA	3FB9A	3G9MA	3HFWA	3IM3A	3KUPA	3M1IB
3C1RA	3D2QA	3ECBB	3FF5A	3GBGA	3HKWA	3IP4C	3KZDA	3M7VA
3C5NA	3D3MA	3EFYA	3FHDA	3GG7A	3HO6A	3ITVA	3L42A	3MEAA
3C7TA	3D85C	3EKIA	3FL2A	3GJ8B	3HR6A	3JTZA	3L60A	3MMSA
3C9HA	3DANA	3EMFA	3FO8D	3GMXA	3HUPA	3JXOA	3LATA	3MWCA
3CBZA	3DGPA	3EOJA	3FQMA	3GP4A	3HY0A	3K2MC	3LHQA	3VUBA
3CHJA	3DKSA	3ETJA	3FSIA	3GWAA				

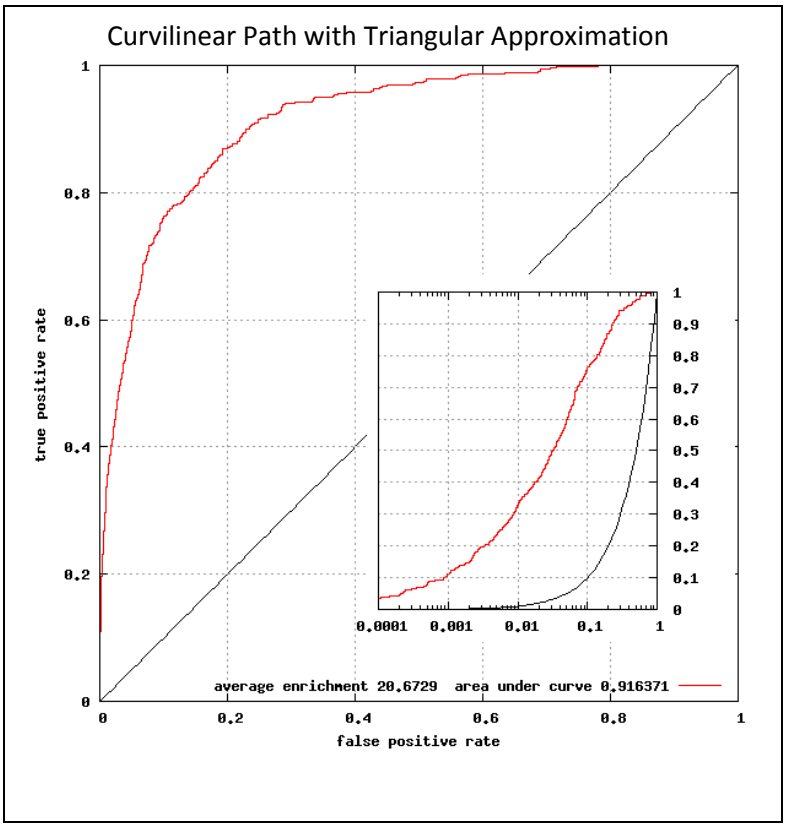
APPENDIX VI – Loop and Normalization Factor Optimization with Benchmark Set






Curvilinear Path with Normalization Factor of 0
true positive rate
false positive rate
average enrichment 17.6143 area under curve 0.876947
0
0.2
0.4
0.6
0.8
1
0.9
0.7
0.5
0.3
0.1
0.0001
0.001
0.01


Curvilinear Path with Regula Falsi Approximation
true positive rate
false positive rate
average enrichment 17.976 area under curve 0.882561
0
0.2
0.4
0.6
0.8
1
0.9
0.7
0.5
0.3
0.1
0.0001
0.001
0.01


Curvilinear Path with Triangular Approximation
true positive rate
false positive rate
average enrichment 20.6729 area under curve 0.916371
0
0.2
0.4
0.6
0.8
1
0.0001
0.001
0.01
0.1
0.9
0.7
0.5
0.3

APPENDIX VI –BCL::SAXS Commandlines

Comparing SAXS profiles

The BCL application, "SimulateSaxsData" is used to create SAXS profiles from given pdb file and compare the profile generated with the experimental SAXS profile. There are three levels of approximation. The first level is complete protein models without any missing regions. A sample command line is:

```
./bcl.exe SimulateSaxsData 'SaxsDebye(consider loops=0, analytic=0)' -pdb
1ENH.pdb -saxs_input_format crysol -exp_data 1ENH00.int -output_file 1ENH.data
-aaclass AAComplete -rmsd
```

The command line for approximating side chains is:

```
./bcl.exe SimulateSaxsData 'SaxsDebye(consider loops=0, analytic=0)' -pdb
1ENH.pdb -saxs_input_format crysol -exp_data 1ENH00.int -output_file 1ENH.data
-aaclass AABackBone -rmsd
```

The command line for approximating both side chains and loop regions is:

```
./bcl.exe SimulateSaxsData 'SaxsDebye(consider loops=1, analytic=0)' -pdb
1ENH.pdb -saxs_input_format crysol -exp_data 1ENH00.int -output_file 1ENH.data
-aaclass AABackBone -rmsd -min_sse_size 5 3 999
```

These will create a SAXS profile for protein 1ENH and compare the protein with the SAXS profile generated through CRYSOL for the desired approximation level. The "input" folder must contain 1ENH.pdb and 1ENH00.int.

BCL::Fold Availability

All components of BCL::Fold, including scoring, sampling, and clustering methods are implemented as part of the BioChemical Library (BCL) that is currently being developed in the Meiler laboratory (www.meilerlab.org). BCL::Fold is freely available for academic use along with several other components of the BCL library.

BIBLIOGRAPHY

1. Watson, J.D. and F.H. Crick, *Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid.* Nature, 1953. **171**(4356): p. 737-8.

2. Brenner, S., F. Jacob, and M. Meselson, *An unstable intermediate carrying information from genes to ribosomes for protein synthesis.* Nature, 1961. **190**: p. 576-581.

3. Anfinsen, C.B., *Principles that Govern the Folding of Protein Chains.* Science, 1973. **181**(4096): p. 223-230.

4. Fang, Y., A.G. Frutos, and J. Lahiri, *Membrane protein microarrays.* J Am Chem Soc, 2002. **124**(11): p. 2394-5.

5. Bernstein, F.C., et al., *The protein data bank: A computer-based archival file for macromolecular structures.* Archives of Biochemistry and Biophysics, 1978. **185**(2): p. 584-591.

6. Honig, B., *Protein Folding: From the Levinthal Pardox to Structure Prediction.* J. Mol. Biol, 1999. **293**: p. 283-293.

7. J.T. Ngo, J.M., M. Karplus, *Computational complexity, protein structure prediction, and the Levinthal paradox*, in *The Protein Folding Problem and Tertiary Structure Prediction*, K.J.L.G. Merz, S., Editor. 1994: Birkhauser, Boston, MA. p. 435-508.

8. Karplus, M., *The Levinthal paradox: yesterday and today.* Fold Des, 1997. **2**(4): p. S69-75.

9. Levinthal, C., *Are there pathways for protein folding?* J. Chim. Phys, 1968. **65**(1): p. 44-45.

10. Skrisovska, L., M. Schubert, and F.H. Allain, *Recent advances in segmental isotope labeling of proteins: NMR applications to large proteins and glycoproteins.* J Biomol NMR, 2010. **46**(1): p. 51-65.

11. Bill, R.M., et al., *Overcoming barriers to membrane protein structure determination.* Nat Biotechnol, 2011. **29**(4): p. 335-40.

12. Baker, D. and A. Sali, *Protein structure prediction and structural genomics.* Science, 2001. **294**(5540): p. 93-6.

13. White, S.H. and W.C. Wimley, *Membrane protein folding and stability: physical principles.* Annu Rev Biophys Biomol Struct, 1999. **28**: p. 319-65.

14. Crippen, G.M., *Global optimization and polypeptide conformation.* Journal of Computational Physics, 1975. **18**(2): p. 224-231.

15. Reeke, G.N., *Protein Folding: Computational Approaches to an Exponential-Time Problem.* Annual Review of Computer Science, 1988. **3**(1): p. 59-84.

16. Dibrov, A., Y. Myal, and E. Leygue, *Computational modelling of protein interactions: energy minimization for the refinement and scoring of association decoys.* Acta Biotheor, 2009. **57**(4): p. 419-28.

17. Karakas, M., et al., *BCL::Fold - De Novo Prediction of Complex and Large Protein Topologies by Assembly of Secondary Structure Elements.* PLoS One, 2012. **7**(11): p. e49240.

18. Li, Z. and H.A. Scheraga, *Monte Carlo-minimization approach to the multiple-minima problem in protein folding.* Proc Natl Acad Sci U S A, 1987. **84**(19): p. 6611-5.

19. Woetzel, N., et al., *BCL::Score-Knowledge Based Energy Potentials for Ranking Protein Models Represented by Idealized Secondary Structure Elements.* PLoS One, 2012. **7**(11): p. e49242.

20. Simons, K.T., et al., *Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions.* J Mol Biol, 1997. **268**(1): p. 209-25.

21. Bradley, P., et al., *Free modeling with Rosetta in CASP6.* Proteins, 2005. **61 Suppl 7**: p. 128-34.

22. Alber, F., et al., *Integrating diverse data for structure determination of macromolecular assemblies.* Annu Rev Biochem, 2008. **77**: p. 443-77.

23. Lindert, S., et al., *EM-fold: De novo folding of alpha-helical proteins guided by intermediate-resolution electron microscopy density maps.* Structure, 2009. **17**(7): p. 990-1003.

24. Robinson, C.V., A. Sali, and W. Baumeister, *The molecular sociology of the cell.* Nature, 2007. **450**(7172): p. 973-82.

25. Koch, M.H.J., P. Vachette, and D.I. Svergun, *Small-angle scattering: a view on the properties, structures and structural changes of biological macromolecules in solution.* Q Rev Biophys, 2003. **36**(2): p. 147-227.

26. Putnam, C.D., et al., *X-ray solution scattering (SAXS) combined with crystallography and computation: defining accurate macromolecular structures, conformations and assemblies in solution.* Q Rev Biophys, 2007. **40**(3): p. 191-285.

27. Svergun, D.I. and M.H.J. Koch, *Small-angle scattering studies of biological macromolecules in solution.* Reports on Progress in Physics, 2003. **66**(10): p. 1735-1782.

28. Svergun, D.I., M.V. Petoukhov, and M.H. Koch, *Determination of domain structure of proteins from X-ray solution scattering.* Biophys J, 2001. **80**(6): p. 2946-53.

29. Tsuruta, H. and T.C. Irving, *Experimental approaches for solution X-ray scattering and fiber diffraction.* Curr Opin Struct Biol, 2008. **18**(5): p. 601-8.

30. Mertens, H.D. and D.I. Svergun, *Structural characterization of proteins and complexes using small-angle X-ray solution scattering.* J Struct Biol, 2010. **172**(1): p. 128-41.

31. Forster, F., et al., *Integration of small-angle X-ray scattering data into structural modeling of proteins and their assemblies.* J Mol Biol, 2008. **382**(4): p. 1089-106.

32. Bernado, P., et al., *A structural model for unfolded proteins from residual dipolar couplings and small-angle x-ray scattering.* Proc Natl Acad Sci U S A, 2005. **102**(47): p. 17002-7.

33. Gabel, F., et al., *A structure refinement protocol combining NMR residual dipolar couplings and small angle scattering restraints.* J Biomol NMR, 2008. **41**(4): p. 199-208.

34. Chen, B., et al., *Multiple conformations of SAM-II riboswitch detected with SAXS and NMR spectroscopy.* Nucleic Acids Res, 2011.

35. Sondermann, H., et al., *Computational docking and solution x-ray scattering predict a membrane-interacting role for the histone domain of the Ras activator son of sevenless.* Proc Natl Acad Sci U S A, 2005. **102**(46): p. 16632-7.

36. Zheng, W. and S. Doniach, *Fold recognition aided by constraints from small angle X-ray scattering data.* Protein Eng Des Sel, 2005. **18**(5): p. 209-19.

37. Petoukhov, M.V. and D.I. Svergun, *Global rigid body modeling of macromolecular complexes against small-angle scattering data.* Biophys J, 2005. **89**(2): p. 1237-50.

38. Boura, E., et al., *Solution structure of the ESCRT-I complex by small-angle X-ray scattering, EPR, and FRET spectroscopy.* Proc Natl Acad Sci U S A, 2011. **108**(23): p. 9437-42.

39. Mishraki, T., et al., *Structural effects of insulin-loading into HII mesophases monitored by electron paramagnetic resonance (EPR), small angle X-ray spectroscopy (SAXS), and attenuated total reflection Fourier transform spectroscopy (ATR-FTIR).* J Phys Chem B, 2011. **115**(25): p. 8054-62.

40. Wang, J., et al., *Determination of multicomponent protein structures in solution using global orientation and shape restraints.* J Am Chem Soc, 2009. **131**(30): p. 10507-15.

41. Grishaev, A., et al., *Refined solution structure of the 82-kDa enzyme malate synthase G from joint NMR and synchrotron SAXS restraints.* J Biomol NMR, 2008. **40**(2): p. 95-106.

42. Svergun, D., C. Barberato, and M.H.J. Koch, *CRYSOL - A program to evaluate x-ray solution scattering of biological macromolecules from atomic coordinates.* Journal of Applied Crystallography, 1995. **28**: p. 768-773.

43. Svergun, D.I. and H.B. Stuhrmann, *New developments in direct shape determination from small-angle scattering. 1. Theory and model calculations.* Acta Crystallographica Section A Foundations of Crystallography, 1991. **47**(6): p. 736-744.

44. Schneidman-Duhovny, D., M. Hammel, and A. Sali, *FoXS: a web server for rapid computation and fitting of SAXS profiles.* Nucleic Acids Res, 2010. **38**(Web Server issue): p. W540-4.

45. Stovgaard, K., et al., *Calculation of accurate small angle X-ray scattering curves from coarse-grained protein models.* BMC Bioinformatics, 2010. **11**: p. 429.

46. Tjioe, E. and W.T. Heller, *ORNL_SAS: software for calculation of small-angle scattering intensities of proteins and protein complexes.* Journal of Applied Crystallography, 2007. **40**: p. 782-785.

47. Debye, P., *Zerstreuung von Röntgenstrahlen.* Annalen der Physik, 1915. **351**: p. 809-823.

48. Grishaev, A., et al., *Improved fitting of solution X-ray scattering data to macromolecular structures and structural ensembles by explicit water modeling.* J Am Chem Soc, 2010. **132**(44): p. 15484-6.

49. Cromer, D.T. and J.B. Mann, *X-ray scattering factors computed from numerical Hartree-Fock Wave Functions*, in *Los Alamos Scientific Laboratory Report*1967, University of California: Los Alamos.

50. Cromer, D.T. and J.T. Waber, *Scattering Factors Computed from Relativistic Dirac-Slater Wave Functions.* Acta Crystallographica, 1965. **18**: p. 104-&.

51. Fraser, R.D.B., T.P. MacRae, and E. Suzuki, *An Improved Method for Calculating the Contribution of Solvent to the X-ray Diffraction Pattern of Biological Molecules.* Journal of Applied Crystallography, 1978. **11**: p. 693-694.

52. Doyle, P.A. and P.S. Turner, *Relativistic Hartree–Fock X-ray and electron scattering factors.* Acta Crystallographica Section A, 1968. **24**(3): p. 390-397.

53. Brown, P.J.R., A.G.;Maslen, E.N.;O'Keefe, M.A.;Willis, B.T.M, *Intensity of diffracted intensities*, in *International Tables for Crystallography*, E. Prince, Editor. 2006, John Wiley and Sons. p. 554-595.

54. Fox, A.G., M.A. Okeefe, and M.A. Tabbernor, *Relativistic Hartree-Fock X-Ray and Electron Atomic Scattering Factors at High Angles.* Acta Crystallographica Section A, 1989. **45**: p. 786-793.

55. Clarke, N.D., et al., *Structural studies of the engrailed homeodomain.* Protein Sci, 1994. **3**(10): p. 1779-87.

56. Daniel, J.W., *The Conjugate Gradient Method for Linear and Nonlinear Operator Equations.* SIAM Journal on Numerical Analysis, 1967. **4**(1).

57. Di Fiore, C., S. Fanelli, and P. Zellini, *Low complexity secant quasi-Newton minimization algorithms for nonconvex functions.* Journal of Computational and Applied Mathematics, 2007. **210**(1-2): p. 167-174.

58. Grishaev, A., et al., *Refinement of multidomain protein structures by combination of solution small-angle X-ray scattering and NMR data.* J Am Chem Soc, 2005. **127**(47): p. 16621-8.

59. Walther, D., F.E. Cohen, and S. Doniach, *Reconstruction of low-resolution three-dimensional density maps from one-dimensional small-angle X-ray solution scattering data for biomolecules.* Journal of Applied Crystallography, 2000. **33**(2): p. 350-363.

60. Wang, G. and R.L. Dunbrack, Jr., *PISCES: recent improvements to a PDB sequence culling server.* Nucleic Acids Res, 2005. **33**(Web Server issue): p. W94-8.

61. Wang, G. and R.L. Dunbrack, *PISCES: a protein sequence culling server.* Bioinformatics, 2003. **19**(12): p. 1589-1591.

62. Ortiz, A.R., C.E. Strauss, and O. Olmea, *MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison.* Protein Sci, 2002. **11**(11): p. 2606-21.

63. Carugo, O. and S. Pongor, *A normalized root-mean-square distance for comparing protein three-dimensional structures.* Protein Sci, 2001. **10**(7): p. 1470-3.